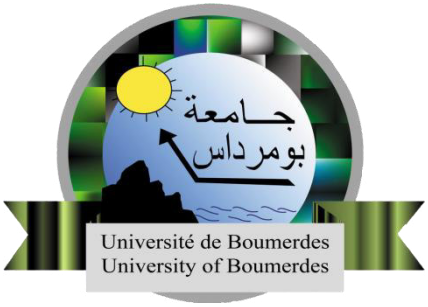
République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE M'HAMED BOUGARA-BOUMERDES



Faculté des **S**ciences de l'**I**ngénieur

Mémoire de Master

Présenté par :

Mr Boulares Ahmed et Mlle Mntambo Marjorie

En vue de l'obtention du diplôme de **Master** en

Génie Electrique

Option : Automatique

Thème :

Etude de la commande et simulation des circuits d'un pendule inversé.

Président	Hamdaoui	MAA	UMBB
Rapporteurs	H.Akroum	MCB	UMBB
Examinateurs	K.Khalfi	MAA	UMBB
	M.S.Boumediene	MCB	UMBB

- Promotion Juin 2017 -

Dédicaces

Je tiens à dédier ce mémoire :

A ma chère mère Gertrude Thandiwe Mntambo qui n'a jamais arrêté de m'aimer et de m'encourager durant toutes mes années d'études. Malgré la distance entre nous, ses sacrifices illimités et ses réconforts moraux m'ont poussé de l'avant.

En mémoire de mon père Joshua Mntambo qui a été plus qu'un père pour moi mais aussi un ami.

A ceux qui sont la source de mon inspiration et mon courage, à qui je dois de l'amour et de la reconnaissance :

Mes sœurs et frères de Boumerdes Daisy Chepkorir, Danielle N'guetta, Richard Lilanda, Innocent Nyirenda,Joseliah Mboumba, Julie qui m'ont soutenu de temps en temps et m'ont aidé à continuer avec le but de vivre la vie en abondance.

A tous mes proches des familles notamment mes sœurs qui sont au Zimbabwe.

Sans oublié mon ami préféré Munyaradzi Mukasa qui m'inspire chaque fois avec son sagesse et sa réussite.
A tous mes amis.

Marjorie Mntambo.

I dedicate my work to:

My parents, in memory of my uncle Younes and his son Youcef.

Boulares Ahmed

Remerciements

Nous rendons nos profondes gratitudes à Dieu tout puissant qui nous a aidés à réaliser ce modeste travail.

Nous exprimons nos profondes gratitudes à nos parents pour leurs encouragements, leurs soutiens et pour les sacrifices qu'ils ont enduré.

Nous remercions. Notre promoteur Mr Akroum Hamza pour les efforts qu'il a déployé, pour nous aider, conseiller, encourager et corriger.
Nous tenons à remercier les membres de jury d'avoir accepté d'examiner notre travail.

Nous remercions aussi tout le corps enseignant dans le département de Génie Electrique qui a contribué à notre formation universitaire.
Sans oublier tous nos amis

Que tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, trouvent ici notre sincère reconnaissance.

ملخص

في هذا العمل قدمنا دراسة في التحكم والمحاكاة للدارةالمستعملة للتحكم في النواس العكسي الذي يعتبر نظام غير خطي و غير مستقر. نموذج رياضي قد فصل متبوعا بالمحاكاة لنظام النواس المعكوس باستعمال الماتلاب أين كان تطبيق المتحكم البي أي دي من أجل استقرار النواس في الموضع المراد. باستعمال برنامج البروتوس، دارة التحكم صممت و بنيت في بيئة افتراضية. و قد تم برمجتها باستعمال برنامج الأردوينو و نقله باستعمال لوحة الأردوينو أونو.

الكلمات المفتاحية: النواس العكسي، غير خطي، ماتلاب، متحكم بي أي دي، بيئة افتراضية، بيئة التطوير

RESUME

Dans ce travail nous présentons l'étude de la commande et la simulation d'un pendule inverse qui est un système instable et non-linéaire. Un modèle mathématique du système a été détaillé suivi par une simulation sous Matlab où le contrôle PID a été appliqué pour stabiliser la position du pendule. En utilisant l'outil Proteus, le système a été fait dans un environnement virtuel. Ensuite la conception et programmation des circuits ont été faites en utilisant la carte Arduino Uno qui est programmé dans le logiciel IDE Arduino.

Mots clés

Le pendule inversé, non-linéaire, Matlab, contrôle PID, environnement virtuel, IDE Arduino

ABSTRACT

In this work we present a study of the control and simulation of the circuit that make up an inverted pendulum which is a non-linear and unstable system. A mathematical model of the system was detailed followed by the simulation of the system in Matlab where PID control was applied to stabilise the position of the pendulum. Using the Proteus tool, the system was designed and made in a virtual environment. The circuits were then designed and programmed using the Arduino Uno circuit board which is programmed in the Arduino Software.

Key words: Inverted pendulum, non-linear, Matlab, PID control, virtual environment, IDE Arduino.

Listes des figures

Liste des tableaux

Liste des abréviations

SIMO	Single Input Single Output
PD	Proportionnelle Dérivative
PID	Proportionnelle Intégral Dérivative
GA	Genetic Algorithme
LQR	Linear Quadratic Regulator
SIRM	Single Input Rule Module
EOPD	Electro-Optic Proximity Detector
SISO	Single Input Single Output
Vcc	Common Collector Voltage
DRL	Diode de Roue Libre
TTL	Transistor-transistor Logic
PWM	Pulse Width Modulation
ROM	Read Only Memory
RAM	Random Access Memory
GND	Ground
USB	Universal Serial Bus
LED	Light Emitting Diode
FTDI	Future Technology Devices International
SPI	Interface Série Périphérique
TWI	Two-Wire interface
UART	Universal Asynchronous Receiver/Transmitter
RX	Receiver
TX	Transmitter
IDE	Integrated Development Environment
VSPE	Virtual Serial Port Emulator
COMPIM	COM Physical Interface Model

Table des matières

Introduction Générale

Dans de nombreuses cultures, en particulier en Afrique, l'équilibre des objets comme les grands conteneurs d'eau est une compétence que les femmes apprennent dès leur plus jeune âge. Instinctivement, lorsque l'objet a tendance à tomber dans n'importe quelle direction, la tête le suit rapidement dans le but de le maintenir équilibré sur la tête. Avec le temps, ces personnes deviennent si excellentes à ce point que l'on s'aperçoit qu'elles ne font aucun effort!

Ce que la plupart de ces personnes ne savent pas est le fait que leurs corps agissent comme des systèmes à pendules inversés quand ils font cela. Le cerveau permet de calculer rapidement une chute dans n'importe quelle direction et d'envoyer des instructions à la tête pour qu'elle change de position et continue de maintenir l'objet droit vers le haut.

C'est exactement le même défi que propose le système automatisé du pendule inversé. Alors que cet exercice semble assez simple et instinctif pour l'homme, il sera nécessaire de définir des stratégies précises pour assurer le maintien automatisé du pendule inversé. Bien évidemment, les performances obtenues grâce à un système automatisé sont de loin supérieures à celles qui seraient obtenues par l'homme.

En Automatique, les ingénieurs se trouvent toujours en train de rechercher a recopier, éliminer ou réduire les efforts de l'homme en créant les systèmes et les machines automatisées qui peuvent faire les tâches que dans le passé ne pouvait être faites que par des humains.

Le pendule inversé est un outil didactique puissant utilisé en automatique depuis maintenant plus de 50 ans. Nous pouvons d'ailleurs retrouver une vaste littérature à son sujet, puisque de nombreux services universitaires d'automatique ou de mécatronique créent un jour leur propre pendule inversé [20].

Ce type de dispositif était à l'origine utilisé afin d'illustrer et de comparer différentes techniques de régulation linéaire comme la stabilisation de systèmes instables et de tester l'efficacité des nouvelles techniques. Au moment présent il existe plusieurs dispositifs qu'on peut utiliser pour stabiliser ce système.

Ce projet est une excellente occasion pour nous d'étudier et de concevoir notre propre système du pendule inversé. Ce mémoire est organisé en quatre chapitres :

Dans le premier chapitre, nous développons des informations générales concernant les systèmes des pendules inversés. Nous allons voir ici les exemples de ce système dans le monde réel.

Le deuxième chapitre est consacré à la modélisation mathématique de ce système où les lois de Newton sont appliquées à l'ensemble de chariot et pendule afin de déduire les équations décrivant le système. Ensuite, nous avons le programme de simulation utilisant Matlab et le contrôle PID pour stabiliser le pendule

En chapitre trois nous détaillons les matériaux nécessaires et leurs importance dans notre système. Inclus dans ce chapitre est une description détaillé des caractéristiques et fonctionnement de la carte Arduino.

Le dernier chapitre se compose de l'information sur le logiciel de programmation IDE Arduino, notre simulation dans l'environnement virtuel fourni par le logiciel Proteus et les circuits conçus de notre système. Nous terminons avec une Conclusion Générale.

Chapitre 1 : Généralités sur les pendules inversés

I.1 Introduction

Avant d'entrer spécifiquement au travail que nous avons effectué dans la conception de notre système qui est le but de ce mémoire, nous allons dans ce premier chapitre, détailler les informations importantes sur les pendules inversés. La question de « qu'est-ce qui est un pendule inversé » et « pourquoi nous nous sommes intéressés à ce système » sera répondu clairement dans les lignes qui suivent.

Dans ce chapitre, nous décrivons les caractéristiques générales du système et nous expliquons ses applications variées. Il a été important de bien expliquer le principe de fonctionnement d'un pendule inversé, afin d'aider les lecteurs à l'imaginer.

I.2 Qu'est-ce qu'un pendule ?

Grâce à la force de rappel due à la gravité, lorsqu'un pendule est déplacé latéralement de sa position d'équilibre, Il peut être ramené encore à cette position d'équilibre. Lorsqu'il est relâché, cette force de rappel combinée avec la masse de pendule le fait osciller d'avant en arrière autour de la position d'équilibre. [27]

Définition : Un pendule, selon *le Concise Oxford English Dictionary* est un poids suspendu à un pivot pour pouvoir osciller librement.

Le plus basique des pendules est une masse accrochée sur un fil léger. Pour ce système, le point d'équilibre est là où se trouve la masse lorsque l'amplitude du pendule est égale à 0° (c'est-à-dire que la masse est suspendue verticalement à partir du pivot). Selon la figure 1.1, B est le point d'équilibre du pendule.

Les pendules sont utilisés dans les instruments scientifiques comme les accéléromètres, les sismomètres et les pendules horloges [ ]. Dans les conditions idéales (où les forces perturbantes comme le frottement, l'amortissement et la résistance de l'aire sont éliminées), la période d'un pendule reste constante voilà son utilité à capter le temps précisément dans une horloge. [27]

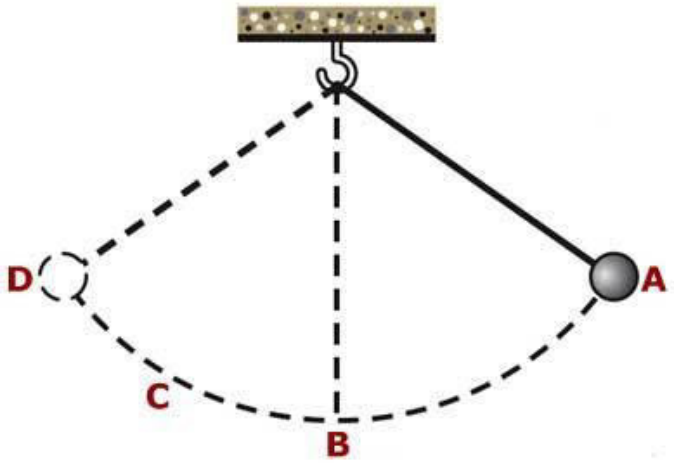


Figure I.1 un pendule oscillant

I.3 Le pendule inversé

Un pendule inversé est un type de pendule dont sa position d'équilibre est instable parce qu'il est maintenu verticalement par rapport au point d'équilibre d'un pendule simple et faisant un angle de 180° avec ce dernier. Il s'agit d'un système SIMO (Single Input Multiple Output) ce qui signifie Une Entrée à Plusieurs Sorties. Ce système est instable et il est décrit par un modèle non linéaire.

Au lieu d'osciller autour du point B (figure 1.1), le pendule inversé fait son mouvement en haut c'est-à-dire, au dessus de l'oscillation supposée du pendule simple. Vu que nous avons déjà cité précédemment que la force due à la gravité travaille toujours pour amener le pendule vers la position B (voir figure 1.1) à cause de cela, la position d'équilibre d'un pendule inversé est instable et nous avons besoin d'un système de contrôle pour le maintenir à sa position d'équilibre.

Généralement, le pendule inversé est constitué d'un chariot sur un rail et d'un pendule suspendu sur le chariot. Ce système mécanique simple est souvent utilisé pour modéliser les problèmes de commande rencontrés dans le vol des fusées et des missiles dans les premières étapes de lancement, lorsque la vitesse de l'aire est trop petite pour la stabilité aérodynamique. Une fusée spatiale au décollage se comporte comme un pendule inversé et toute comme un robot marcheur [19]

I.3.1 Historique d'un pendule inversé

La non-linéarité du pendule inversé et la simplicité de structure nous permettent d'utiliser ce système comme une référence pour vérifier les performances et la robustesse des nouvelles méthodes de commande [1] [2].

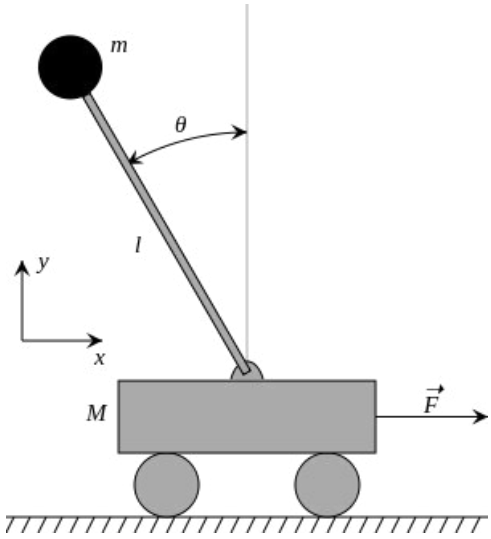


Figure I.2 : Mécanisme simple d'un pendule inversé.

En 1960 Roberge James à traité avec succès un pendule inversé dans sa thèse, *The Mechanical Seal*[26]. Il a conçu un servomécanisme pour équilibrer un balai en utilisant les critères de Routh de Bode et ceux de Nyquist. Son système prendrait environ dix minutes pour se stabiliser et une fois stabilisé, il pourrait rester pendant environ trois heures dans cette position stable. Après cela, beaucoup des systèmes plus rapides et plus robustes, mais basés sur le même principe ont été mis en place partout dans le monde.

Juste pour mentionner quelques œuvres faites sur les pendules inversés, en [21] les auteurs ont équilibré le pendule en utilisant un potentiomètre comme capteur et un régulateur « Proportionnelle Dérivative » (PD) comme l'algorithme de conception. Le régulateur PD à été utilisé pour générer un signal pour commander la vitesse et la direction du moteur. Le modèle cependant possède quelque limitations comme : les vibrations, glissement des roues, le courant insuffisant et l'échauffement du transistor. Voila pourquoi les auteurs ont proposé pour les prochains travaux, l'utilisation de plusieurs capteurs au lieu d'un et d'autres régulateurs comme les PID et GA.

Prasad [22] à réalisé le même système dans une manière différente. Les états de ce système non-linéaire ont été gavés au LQR qui est conçu en utilisant le modèle linéaire d'espace d'état. Ici les méthodes de PID et LQR ont été utilisées ensembles pour commander la position de chariot et stabiliser le pendule inversé dans la position verticale. On à conclue que ce système est optimal donc préférable à celui qui est commandé uniquement par un PID.

Il existe beaucoup des recherches déjà faites sur la stabilisation des pendules inversés en utilisant la logique floue. Par exemple selon Yi et Yubazaki [1], un régulateur basé sur le modèle d'inférence flou connecté dynamiquement (le SIRM) pour la commande de stabilisation des systèmes de pendule inversé présenté. Le régulateur a quatre entrées, chacun d'eux a un SIRM et un degré d'importance dynamique et il fait la commande de l'angle de pendule et de position du chariot en parallèle.

Berenji[3] à construire le réseau des neurones de l'évaluation d'action et le réseau flou de la sélection d'action et les deux étaient réglé en utilisant l'apprentissage par renforcement, pour commander un pendule inversé et lui positionné en haut-droit. Lin a proposée pour ajuster les fonctions d'appartenance d'une base de règle floue basée sur l'adaptation d'un mode glissement et l'a appliqué au commande angulaire de pendule inversé.

Lu [7] à utilisé l'algorithme génétique pour générer automatiquement les règles floue et les facteurs de mise à l'échelle pour le contrôle du pendule inversé. .Margaliot [8] montrait une nouvelle approche pour déterminer la structure de commande floue pour le pendule inversé par floue synthèse de Lyapunov alors que Mikukcik [10] a extrait les règles floues pour le contrôle du pendule inversé par une méthode de regroupement flou.

Saez a utilisé le contrôleur prédictif généralisé pour déterminer les paramètres du modèle flou Takagi-Sugeno pour contrôler un pendule inversé. Wong a adopté l'algorithme génétique pour accorder toutes les fonctions d'appartenance d'un système flou alors que Yamakawa [4] à conçu un système matériel de régulateur floue en haute vitesse et il a utilisé que sept règles floue pour commander l'angle d'un pendule inversé.

Bien que le contrôle de stabilisation d'un système de pendule inversé inclue également le contrôle de position du chariot et le contrôle angulaire du pendule, en raison de la longueur limitée du rail, les approches indiquées ci-dessus ont seulement pris en compte le contrôle angulaire du pendule.

Pour commander l'angle de pendule et la position de chariot, Kandadai [18] à modifié le structure de Berinji [3] au commande hiérarchique et l'as permit de générer la base de connaissance floue automatiquement. Il à pris plus de 12.0 secondes, cependant pour que le système approche la stabilisation avec un certain décalage en plus de sa complexité de structure.

Sur la base de la théorie des systèmes de structure variable et de la trajectoire de l'équation dynamique linéarisée d'un pendule inversé sur le plan de phase, Kawaji[13] a construit un contrôleur flou composé de deux modules de règles simples. Un module était pour l'amplitude de la variable manipulée et l'autre pour le signe de la variable manipulée. Étant donné que le contrôle de la position du chariot peut être considéré comme une perturbation de l'angle du pendule, l'information de la position du chariot a d'abord été transformée en un angle de cible virtuel, et cet angle a ensuite été intégré dans la commande de l'angle du pendule. Bien que cette méthode soit plutôt simple, il était difficile de stabiliser complètement le système dans un court intervalle de temps.

Kyung [4] a présenté un contrôleur flou, dont la base de règles a été dérivée de trois réseaux de neurones. Bien que le contrôleur flou puisse stabiliser un système pendule inversé en environ 8,0 s, il lui fallait 396 règles même après une procédure de lissage et une procédure de réduction logique.

Matsuura [9] et Yasunobu [19] ont utilisé les informations du chariot pour créer un ensemble de 49 règles floues pour la réalisation de l'angle de cible virtuel, puis utilisé l'angle de cible virtuel et l'information du pendule pour construire un autre ensemble de 49 règles floues pour la stabilisation totale. Sakai [16] a appliqué une méthode d'optimisation non linéaire pour former un contrôleur flou pour la stabilisation, mais le contrôleur a passé plus de 200,0 s pour stabiliser le système.

I.3.2 Intérêt de l'étude d'un pendule inversé

L'étude de pendule inversé est idéale pour les étudiants comme nous en Automatique et en Robotique qui font les laboratoires de recherches parce qu'ils font intervenir beaucoup de notions intéressantes pour eux : la programmation, l'automatisation, la mécanique et l'électronique.

Le bébé après son premier pas de marche joue comme un pendule inversé dont les deux axes de rotation fondamentaux sont les hanches et les chevilles. Lorsqu'il est en position de debout, ses articulations de corps travaillent sans s'arrêter pour le maintenir en équilibre. [2]

Mentionné dans cette section sont quelques applications de cette concept au monde réel dans les domaines variées.

I.3.2.1 Le domaine de Robotique

La robotique utilise ce type de concept dans leur principe de commande. Le robot auto-balancé est constitué d'une plateforme solidaire de deux roues. L'axe des roues est perpendiculaire au l'axe de déplacement du robot et le principe du contrôle de position est basé sur le pendule inversé. [20]

La figure 1.3 à gauche est une image d'un robot de voiture dansante qui s'agit d'un jouet pour les enfants qui utilise le principe du pendule inversé pour se tenir debout, exposant ainsi un spectacle intéressant. A droite nous avons un robot auto balancé et assez stable pour servir les boissons comme un serveur humain.

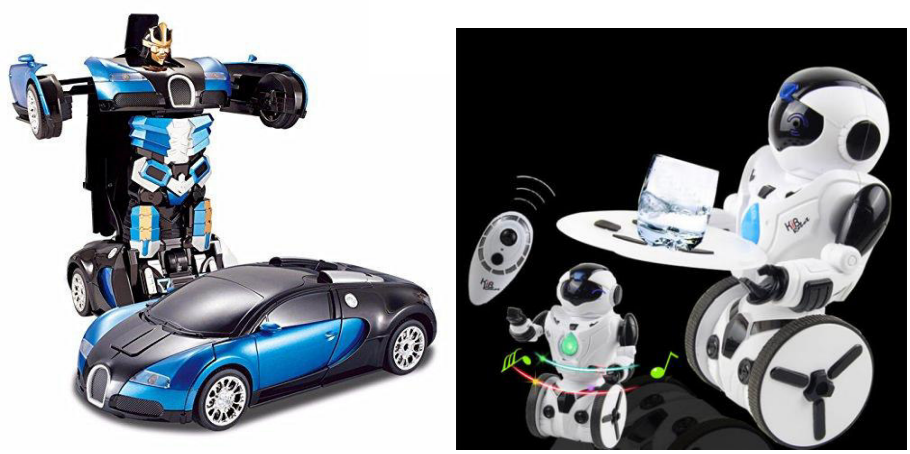


Figure I.3 Exemples des robots inversés.

Nbot(figure I.4 à gauche) est un robot d'équilibrage à deux roues construit par David .P Anderson, ce robot utilise un capteur inertiel disponible dans le commerce et positionne l'information à partir du codeur du moteur pour équilibrer le système. Steven Hassenplug a construit avec succès un robot d'équilibrage appelé Legway (figure I.4 à droite) en utilisant le kit de robotisation LEGO Mindstorms. Deux capteurs de détecteur de proximité électro-optique (EOPD) ont été utilisés pour fournir l'angle d'inclinaison du robot au contrôleur qui est programmé dans BrickOS, un langage de programmation similaire à C / C ++.[31]

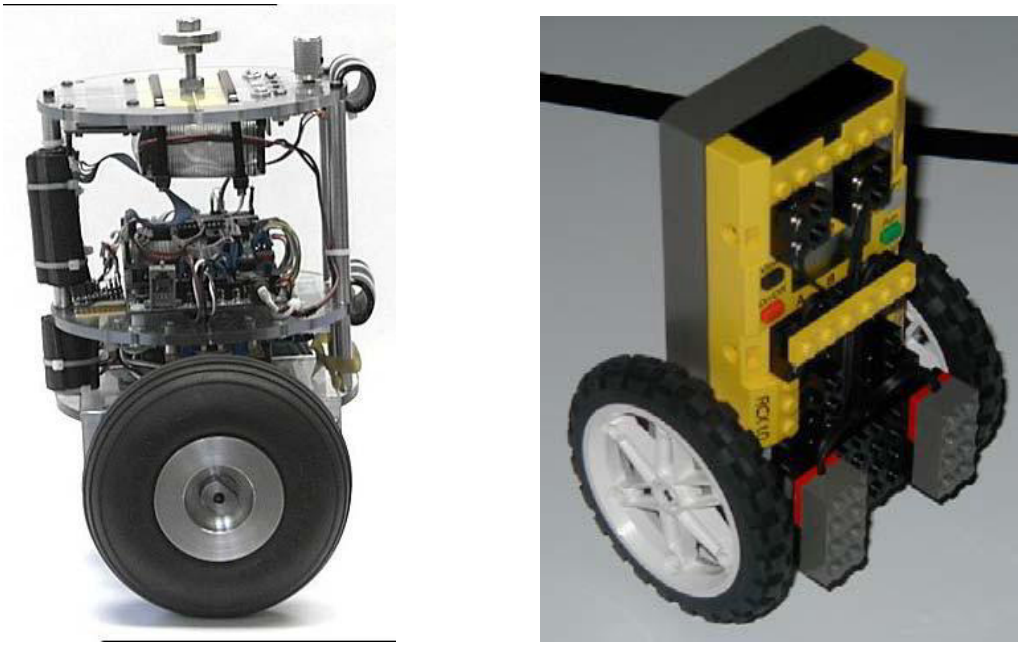


Figure I.4 Nbot(à gauche) et Legway(à droite)*[31]*

I.3.2.2 Le domaine de la médicine

Les spécialistes qui travaillent dans le domaine de réalisation de prothèses pour les hanches (remplacement chirurgical d'un organe, la pièce ou l'appareil de remplacement : prothèse dentaire)sont amenés à utilisé un pendule double inversé pour calculer l'ensemble des contraintes qui sont soumises à la prothèse comme le montre la figure suivante

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 penchant en avant et on même que le pendule inversé [27].

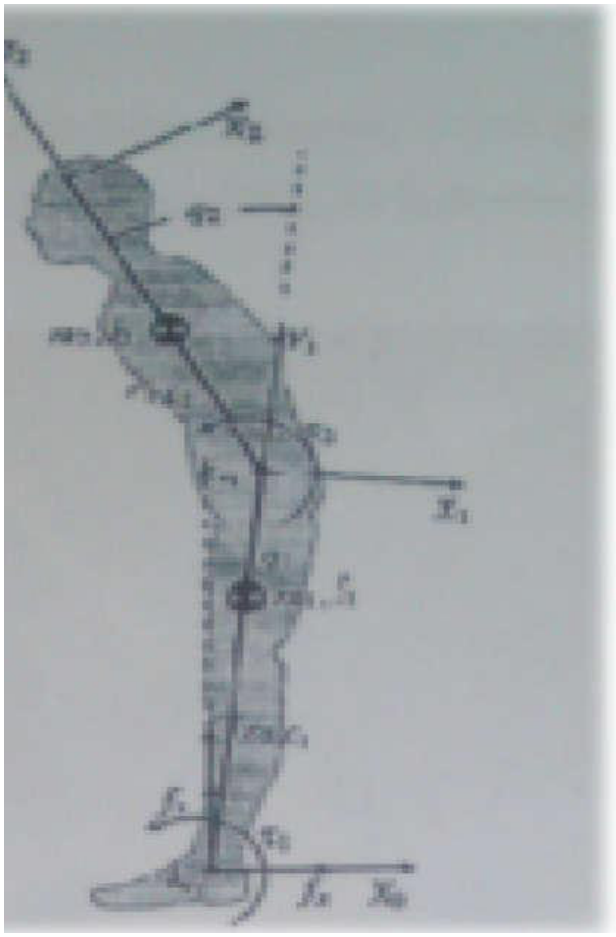


Figure I.5 Le corps de l'être humain vu comme double pendule *[27]*

I.3.2.3 L'appareil de transport personnel *Segway*

Montré en (Figure I.6) ci-dessous est un exemple pratique d'un pendule inversé, où la motion des roues est commandée par un ordinateur avec le but de maintenir le corps du véhicule dans une position verticale. Quand le cycliste incline en avant, il pousse le contrôleur de tourné les roues avec l'accélération nécessaire pour reste en balance [6].

Dee Kamen qui détient plus de 150 brevets américains et étrangers liés aux dispositifs médicaux, aux systèmes de contrôle climatique et à la conception d'hélicoptères, a inventé le «SEGWAY HT». Cette innovation utilise cinq gyroscopes et une collection d'autres capteurs d'inclinaison pour se maintenir debout. Seuls trois gyroscopes sont nécessaires pour l'ensemble du système, les capteurs supplémentaires sont inclus comme mesure de sécurité. [31]

Figure I.6 Le Segway HT *[31].*

I.3.2.4 Le domaine de l'aérospatiale :

Dans ce domaine l'étude des pendules inversés a une grande importance, par exemple pour commander et stabiliser l'attitude du satellite, le lancement des fusés...etc.[28]

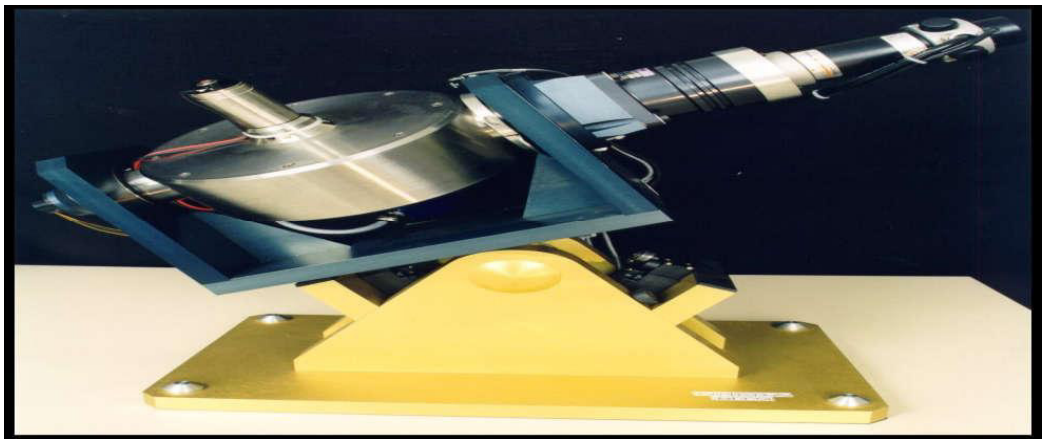


Figure I.7 : Pendule gyroscopique inversé *[28]*

I.3.2.4 Le domaine de loisir

Dans les parcs d'attractions on trouve les tours qui utilisent le principe du pendule inversé. La machine en figure 1.8 bascule vers le haut autour d'un axe de sorte qu'il agit comme un pendule inversé mais dans son première phase de mouvement il agit comme un pendule normal.

Figure1.8 Tour d'amusement (Ciseaux)

I.3.3 Propriétés de pendule inversé

Malgré les différentes tailles et différents matériaux utilisés pour fabriquer des pendules, ils ont des caractéristiques communes mentionnées ci-dessous :

1.3.3.1 La non-linéarité :

Les systèmes linéaires sont des systèmes qui sont régis par les équations différentiels linéaires à coefficients constants et d'ordre fini d'où le principe de la superposition peut être appliqué. Néanmoins, la non-linéarité est la particularité, en mathématiques, de systèmes dont le comportement n'est pas linéaire, c'est-à-dire soit ne satisfaisant pas le principe de superposition, soit dont la sortie n'est pas proportionnelle à l'entrée. Ci-dessous sont les équations du système du pendule inversé qui montrent que ce système est non-linéaire.

$$(M+m)\,\ddot{x} + b\dot{x} + \mathrm{ml}\ddot{\theta}\cos\theta - \mathrm{ml}\dot{\theta}^2\sin\theta = \mathrm{F} \quad \text{(II.3)}$$

$$(I+ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta \quad \text{(II.6)}$$

I.3.3.2 Système holonome :

Un système mécanique S est holonome si la position de ses différentes parties peut être caractérisée par *n* variables indépendantes q_1,.... q_n, appelées coordonnées généralisées du système. On dit alors que S est un système holonome à n degrés de liberté.

Le pendule inversé est également un système holonome avec pour coordonnées généralisées q1=X et q2=θ [11] [12].

I.3.3.3 L'instabilité en boucle ouverte

La position d'équilibre du pendule inversé est instable voila pourquoi le système nécessite les capteurs pour mesurer la distance déplacée par le chariot et l'angle du pendule. Apres avoir mesuré ces variables il faut les renvoyer au système de contrôle, créant ainsi un système en boucle fermée.

I.3.3.4 Système sous-actionné

Les systèmes mécaniques sous-actionnés sont définis comme étant des systèmes dont le nombre d'actionneurs est inférieur au nombre de degrés de liberté. Le pendule inversé stabilisé par volant d'inertie est, du fait qu'il a moins d'actionneurs de degrés de liberté, un système mécanique sous-actionnés.

I.3.4 Principe de fonctionnement [18] :

C'est virtuellement impossible de balancer un pendule dans la position inverse sans applique quelque force extérieure au système. Le système de chariot-pendule-moteur permet au mouvement de rotation du moteur d'être convertis en mouvement de translation du chariot au travers d'une courroie attachée au chariot et passant par le moteur.

Le principe de fonctionnement du système est très simple en théorie : quand le pendule penche vers la droite, le chariot doit le rattraper en effectuant un mouvement vers la droite et inversement au cas où le pendule penche vers la gauche. La difficulté est de régler l'intensité et la forme de la réaction du chariot en fonction de l'angle que

le pendule fait avec la verticale.

I.4 Conclusion

Apres avoir défini et détaillé les caractéristiques d'un pendule inversé, nous pouvons conclure que c'est un dispositif important dans le domaine d'Automatique pour tester les algorithmes de contrôle. Dans notre travail nous allons détailler notre système en expliquant chaque étape utilisé pour le concevoir. Le chapitre suivant est consacré à la modélisation mathématique et simulation en Matlab.

Chapitre 2 : Modélisation mathématique et simulation sous Matlab

II.1. Introduction

Le problème de commande exige la recherche du modèle mathématique du système à commander, donc une phase de modélisation est nécessaire pour permettre l'étude en simulation. Dans ce chapitre nous allons présenter la modélisation mathématique du système avant de faire l'étude en simulation sous MATLAB. Le contrôle en PID a été utilisé pour stabiliser la position du pendule.

II.2. Modélisation mathématique du pendule inversé

Le système du pendule inversé est un exemple couramment utilisé dans les manuels de systèmes de contrôle et la littérature de recherche. Sa popularité découle en partie du fait qu'elle est instable sans contrôle, c'est-à-dire que le pendule tombera simplement si le chariot n'est pas déplacé pour l'équilibrer. En outre, la dynamique du système est non-linéaire. L'objectif de la commande de système est d'équilibrer le pendule inversé en appliquant une force au chariot sur lequel le pendule est fixé.

Dans ce cas, nous considérons un problème bidimensionnel où le pendule est forcé de se déplacer dans le plan vertical représenté sur la figure ci-dessous. Pour ce système, l'entrée de commande est la force F qui déplace le chariot horizontalement et les sorties sont la position angulaire du pendule θ et la position horizontale du chariot x.

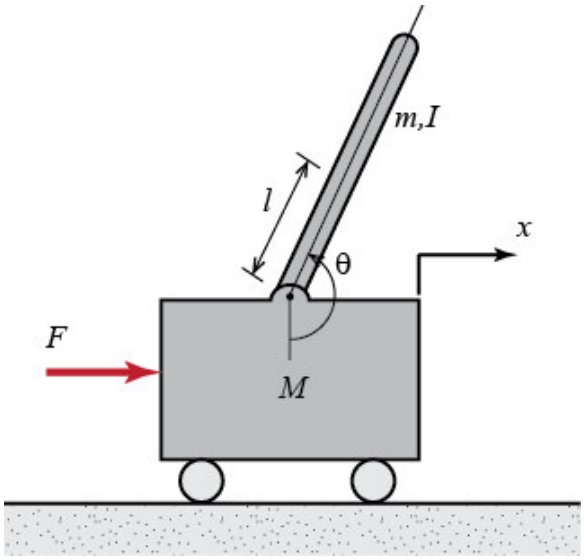


Figure II.1. Schéma de l'ensemble chariot et pendule inversé

Les paramètres de notre système sont les suivants :

Paramètre	Description	Valeur	Unité
M	Masse du chariot	0.175	kg
m	Masse du pendule	0.055	kg
b	Coefficient de frottement du chariot	0.1	N /m/sec
l	demi-longueur du pendule	0.165	mètre
I	Moment d'inertie du pendule	0.003025	Kg.m²
F	Force appliqué au chariot		Newton
x	Position du chariot		mètre
θ	Angle de rotation du pendule		radian

Tableau II.1 Récapitulatif des variables utilisées dans la modélisation

Pour la section du contrôle PID de ce problème, nous ne serons intéressés que dans le contrôle de la position du pendule. C'est parce que cette technique est la mieux adaptée pour les systèmes à une entrée et une sortie unique (SISO Single Input Single Output en Anglais). Nous allons cependant étudier l'effet du contrôleur sur la position du chariot après que le contrôleur a été conçu. Pour cette section, nous concevrons un contrôleur pour restaurer le pendule vers une position verticale vers le haut après avoir subi une poussée impulsive sur le chariot.

En utilisant des techniques d'espace d'état, il est plus facile a traiter un système avec plusieurs sortis. Dans notre cas, le système du pendule inversé à une entrée et multi-sorties (SIMO). Par conséquent, pour la section de l'espace d'état de l'exemple du pendule inversé, nous contrôlerons à la fois l'angle du pendule et la position du chariot.

II.2.1.Analyse de forces et équations de système

Voici le diagramme de corps libres des deux éléments du système de pendule inversé.

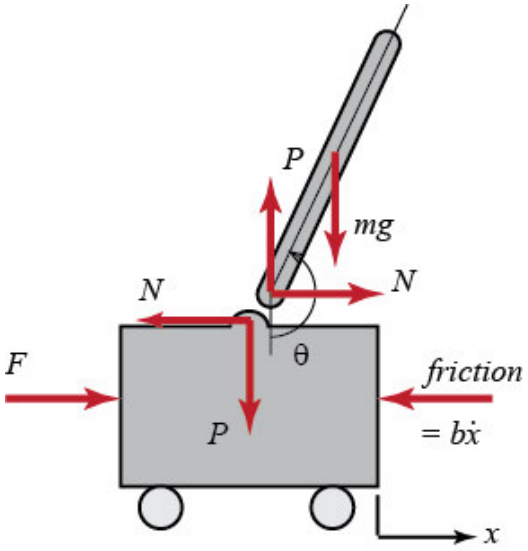


Figure II.2. Diagramme du corps libre du chariot et du pendule.

En additionnant les forces dans le diagramme de corps libre du chariot dans le sens horizontal, on obtient l'équation de mouvement suivante :

$$M\ddot{x} + b\dot{x} + N = F \qquad \text{(II.1)}$$

Nous pourrions résumer les forces dans la direction verticale également, mais aucune information utile ne serait obtenue à partir de cela.

En additionnant les forces dans le diagramme du corps libre du pendule dans le sens horizontal, on obtient l'expression suivante pour la force de réaction N

$$N = m\ddot{x} + ml\ddot{\theta}\cos\theta \ - ml\dot{\theta}^2\sin\theta \qquad \text{(II.2)}$$

Si nous substituons cette équation à la première équation, nous obtenons l'une des deux équations principales pour ce système.

$$(M+m)\,\ddot{x} + b\dot{x} + \mathrm{ml}\ddot{\theta}\cos\theta - \mathrm{ml}\dot{\theta}^2\sin\theta = \mathrm{F} \qquad \text{(II.3)}$$

Pour obtenir la deuxième équation du mouvement pour ce système, résume les forces perpendiculaires au pendule. La résolution du système selon cet axe simplifie considérablement les mathématiques. Nous obtenons l'équation suivante.

$$P\sin\theta + N\cos\theta - mg\sin\theta = ml\ddot{\theta} + m\ddot{x}\cos\theta \qquad \text{(II.4)}$$

Pour éliminer les termes P et N dans l'équation ci-dessus, résume les moments autour du centre du pendule pour obtenir l'équation suivante.

$$-Pl\sin\theta - Nl\cos\theta = I\ddot{\theta} \quad \text{(II.5)}$$

En combinant ces deux dernières expressions, nous obtenons la deuxième équation principale

$$(I + ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta \quad \text{(II.6)}$$

Étant donné que la technique de contrôle que nous utiliserons dans cet exemple s'applique uniquement aux systèmes linéaires, cet ensemble d'équations doit être linéarisé. Plus précisément, nous allons linéariser les équations autour de la position de l'équilibre vers le haut vertical, θ = π, et supposerons que le système reste dans un petit quartier de cet équilibre.

On considère que ϕ représente la petite déviation de la position d'équilibre, c'est-à-dire

$\theta = \pi + \phi$

En présumant une petite déviation ϕ par rapport à l'équilibre, nous pouvons utiliser les approximations de petit angle suivantes :

$$\cos\theta = \cos(\pi + \phi) \approx -1 \quad \text{(II.7)}$$

$$\sin\theta = \sin(\pi + \phi) \approx -\phi \quad \text{(II.8)}$$

$$\dot{\theta}^2 = \dot{\phi}^2 \approx 0 \quad \text{(II.9)}$$

Après avoir remplacé les approximations ci-dessus en nos équations non linéaires, principales, nous arrivons aux deux équations de mouvement linéarisées. F a été remplacé par u

$$(I + ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x} \quad \text{(II.10)}$$

$$(M + m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u \quad \text{(II.11)}$$

II.2.2. La fonction de transfert

Pour obtenir les fonctions de transfert des équations du système linéarisé, nous devons d'abord prendre la transformée de Laplace des équations du système en supposant des conditions initiales nulles. Les transformées Laplace qui en résultent sont présentées ci-dessous.

$$(I + ml^2)\phi(s)s^2 - mgl\phi(s) = mlX(s)s^2 \quad \text{(II.12)}$$

$$(M+m)X(s)s^2 + bX(s)s - ml\phi(s)s^2 = U(s) \quad \text{(II.13)}$$

Une fonction de transfert représente la relation entre une seule entrée et une seule sortie à la fois. Pour trouver notre première fonction de transfert pour la sortie ϕ(s) et une entrée de U(s), nous devons éliminer X(s) des équations ci-dessus. Résolvez la première équation pour X (s).

$$X(s) = \left[\frac{I+ml^2}{ml} - \frac{g}{s^2}\right]\phi(s) \quad \text{(II.14)}$$

Ensuite, substituez ce qui précède à l'équation II.12

$$(M+m)\left[\frac{I+ml^2}{ml} - \frac{g}{s^2}\right]\phi(s)s^2 + b\left[\frac{I+ml^2}{ml} - \frac{g}{s^2}\right]\phi(s)s - ml\phi(s)s^2 = U(s) \quad \text{(II.15)}$$

La fonction de transfert est alors la suivante :

$$\frac{\phi(s)}{U(s)} = \frac{\frac{ml}{q}s^2}{s^4 + \frac{b(I+ml^2)}{q}s^3 - \frac{(M+m)mgl}{q}s^2 - \frac{bmgl}{q}s} \quad \text{(II.16)}$$

Où :

$$q = [(M+m)(I+ml^2) - (ml)^2] \quad \text{(II.17)}$$

À partir de la fonction de transfert ci-dessus, on constate qu'il y a à la fois un pôle et un zéro à l'origine. Ceux-ci peuvent être annulés et la fonction de transfert devient la suivante.

$$P_{pend}(s) = \frac{\phi(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(I+ml^2)}{q}s^2 - \frac{(M+m)mgl}{q}s - \frac{bmgl}{q}} \quad \left[\frac{rad}{N}\right] \quad \text{(II.18)}$$

Deuxièmement, la fonction de transfert avec la position du chariot X (s) comme la sortie peut être dérivée de manière similaire pour arriver à la suivante :

$$P_{char}(s) = \frac{X(s)}{U(s)} = \frac{\frac{(I+ml^2)s^2 - mgl}{q}}{s^4 + \frac{b(I+ml^2)}{q}s^3 - \frac{(M+m)mgl}{q}s^2 - \frac{bmgl}{q}s} \quad \left[\frac{m}{N}\right] \quad \text{(II.19)}$$

II.2.3. L'espace d'état

Les équations linéarisées de mouvement d'en haut peuvent également être représentées sous forme d'espace d'état si elles sont réarrangées en une série d'équations différentielles de premier ordre. Comme les équations sont linéaires, elles peuvent ensuite être placées dans la matrice standard illustrée ci-dessous.

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\\ \dot{\phi}\\ \ddot{\phi}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Ml^2} & \frac{m^2gl^2}{I(M+m)+Ml^2} & 0\\ 0 & 0 & 0 & 1\\ 0 & \frac{-mlb}{I(M+m)+Ml^2} & \frac{mgl(M+m)}{I(M+m)+Ml^2} & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \phi\\ \dot{\phi}\end{bmatrix} + \begin{bmatrix}0\\ \frac{I+ml^2}{I(M+m)+Ml^2}\\ 0\\ \frac{ml}{I(M+m)+Ml^2}\end{bmatrix}u \quad \text{(II.20)}$$

$$y = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \phi\\ \dot{\phi}\end{bmatrix} + \begin{bmatrix}0\\ 0\end{bmatrix}u \quad \text{(II.21)}$$

La matrice C a deux lignes car la position du chariot et la position du pendule font partie de la sortie. Plus précisément, la position du chariot est le premier élément de la sortie y et l'écart du pendule par rapport à sa position d'équilibre est le deuxième élément de y.

II.3 Simulation en Matlab.

Matlab est l'outil de référence pour la simulation numérique, notamment en ce qui concerne l'Automatique. Il offre des possibilités avancées que ce soit en matière d'identification ou de commande. Il permet, de manière plus générale, de résoudre une grande diversité de problèmes de simulation, dans des domaines aussi varies que le traitement du signal, les statistiques ou la vision, pour ne citer que quelques exemples

II.3.1Fonction de transfert

Nous pouvons représenter les fonctions de transfert dérivées ci-dessus pour le système de pendule inversé dans MATLAB en utilisant les commandes suivantes :

```
clear all
clc
M = 0.175;
m = 0.055;
b = 0.1;
I = 0.003025;
g = 9.8;
l = 0.165;
q = (M+m)*(I+m*l^2)-(m*l)^2;
s = tf('s');
P_char = (((I+m*l^2)/q)*s^2 - (m*g*l/q))/(s^4 + (b*(I + m*l^2))*s^3/q - ((M + m)*m*g*l)*s^2/q - b
P_pend = (m*l*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q - b*m*g*l/q);
sys_tf = [P_char ; P_pend];
inputs = {'u'};
outputs = {'x'; 'phi'};
set(sys_tf,'InputName',inputs)
set(sys_tf,'OutputName',outputs)
sys_tf
```

On trouve la fonction du transfert suivante:

From input "u" to output

$$x = \frac{4.149.10^{-9}s^2 - 8.159.10^{-8}}{8.786.10^{-10}s^4 + 4.149.10^{-10}s^3 - 1.876.10^{-8}s^2 - 8.159.10^{-9}s}$$

$$\phi = \frac{8.325.10^{-9}s}{8.786.10^{-10}s^3 + 4.149.10^{-10}s^2 - 1.876.10^{-8}s - 8.159.10^{-9}}$$

II.3.2 L'espace d'état

Les commandes MATLAB supplémentaires suivantes créent un modèle d'espace d'état du pendule inversé et produisent la sortie ci-dessous lorsqu'elle est exécutée dans la fenêtre de commande

```
m = 0.055;
b = 0.1;
I = 0.003025;
g = 9.8;
l = 0.165;

p = I*(M+m)+M*m*l^2; %denominator for the A and B matrices

A = [0      1              0           0;
     0 -(I+m*l^2)*b/p   (m^2*g*l^2)/p   0;
     0      0              0           1;
     0 -(m*l*b)/p        m*g*l*(M+m)/p  0];
B = [     0;
     (I+m*l^2)/p;
          0;
        m*l/p];
C = [1 0 0 0;
     0 0 1 0];
D = [0;
     0];

states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'u'};
outputs = {'x'; 'phi'};

sys_ss = ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs)
sys_tf = tf(sys_ss)
```

sys_ss =

$$a = \begin{matrix} 0 & 1 & 0 & 0 \\ 0 & -0.4722 & 0.8427 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.9475 & 21.36 & 0 \end{matrix} \qquad b = \begin{matrix} 0 \\ 4.722 \\ 0 \\ 9.475 \end{matrix}$$

$$c = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{matrix} \qquad d = \begin{matrix} 0 \\ 0 \end{matrix}$$

Continuous-time state-space model.

sys_tf =From input "u" to output...

$$x = \frac{4.722s^2 + 4.194.10^{-15}s - 92.85}{s^4 + 0.4722s^3 - 21.36s^2 - 9.285s}$$

$$\phi = \frac{9.475s - 5.26.10^{-16}}{s^4 + 0.4722s^3 - 21.36s^2 - 9.285s}$$

Continuous-time transfer function.

II.3.3 L'analyse du système

Rappelons que les deux fonctions de transfert (II.18 et II.19) ne sont valables que pour de petites valeurs de l'angle ϕ où ϕ est l'écart du pendule de la position verticale vers le haut. En outre, l'angle de pendule absolu θ est égal à π+ϕ.

II.3.3.1 La réponse impulsionelle en boucle ouverte

Nous pouvons maintenant examiner la réponse impulsionnelle en boucle ouverte du système. Plus précisément, nous examinerons comment le système répond à une force impulsive appliquée au chariot utilisant l'impulsion de commande MATLAB.

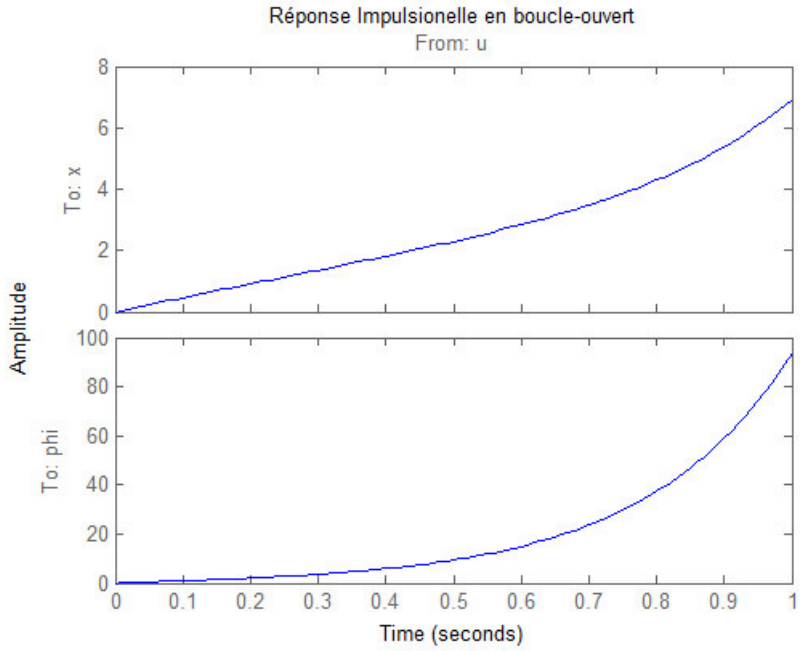


Figure II.3 Réponse impulsionelle en boucle ouverte

Les résultats montrent que la réponse du système est totalement insatisfaisante. En fait, il n'est pas stable en boucle ouverte parce que la position du chariot se déplace infiniment loin vers la droite et la position du pendule augmente de plus de 100 radians (15 révolutions), mais le modèle n'est valable que pour les petites ϕ.

Les zéros et pôles de système (où la position du chariot est la sortie) peuvent être trouvés par les commandes suivantes :

```
[zeros poles] = zpkdata(P_pend,'v')
```

zeros =0

$$\text{poles} = \begin{matrix} 4.6043 \\ -4.6420 \\ 0.4344 \end{matrix}$$

De la même manier on trouve les zeros du système losque l'angle de pendule est la sortie :

```
[zeros poles] = zpkdata(P_cart,'v')
```

$$\text{zeros} = \begin{matrix} 4.4346 \\ -4.4346 \end{matrix}$$

$$\text{poles} = \begin{matrix} 0 \\ 4.6043 \\ -4.6420 \\ -0.4344 \end{matrix}$$

Les pôles pour les deux fonctions de transfert sont identiques. Le pôle à 4.6043 indique que le système est instable car le pôle a une partie réelle positive. En d'autres termes, le pôle est dans la moitié droite du plan s complexe. Cela correspond à ce que nous avons observé ci-dessus.

II.3.3.2. Réponse Indicielle en Boucle Ouverte

Étant donné que le système a un pôle avec une partie réelle positive, sa réponse à une entrée échelon augmentera également sans bornes. Nous vérifierons cela en utilisant la commande « *lsim* » qui peut être utilisée pour simuler la réponse des modèles des systèmes à la fois linéaire et invariant à des entrées arbitraires. Dans ce cas, une entrée échelon de 1Newton sera utilisée.

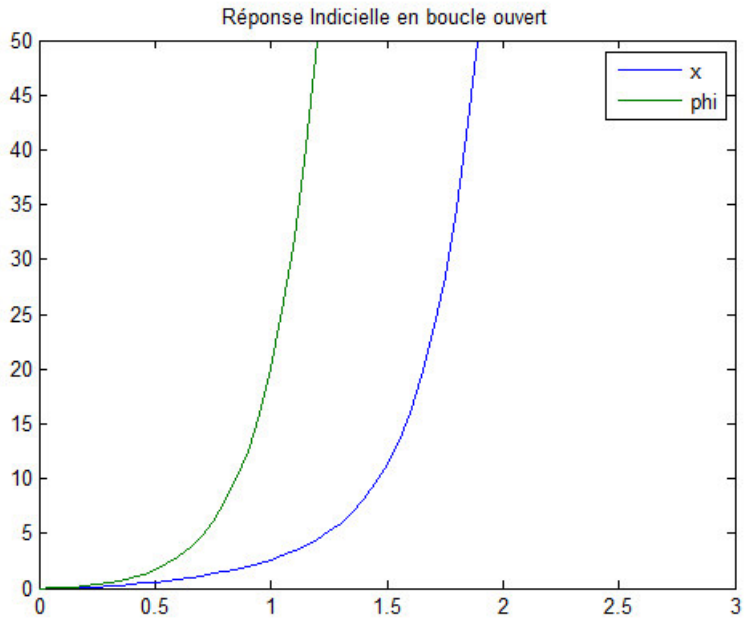


Figure II.4. Réponse Indicielle en boucle-ouvert

Les résultats de figure II.4 confirment notre attente que la réponse du système à un échelon unitaire est instable. Il ressort de l'analyse ci-dessus qu'une sorte de contrôle devra être conçu

pour améliorer la réponse du système. Nous allons appliquer le control PID sur la position de pendule.

II.3.3.3 Les pôles et zéros du système en boucle ouverte

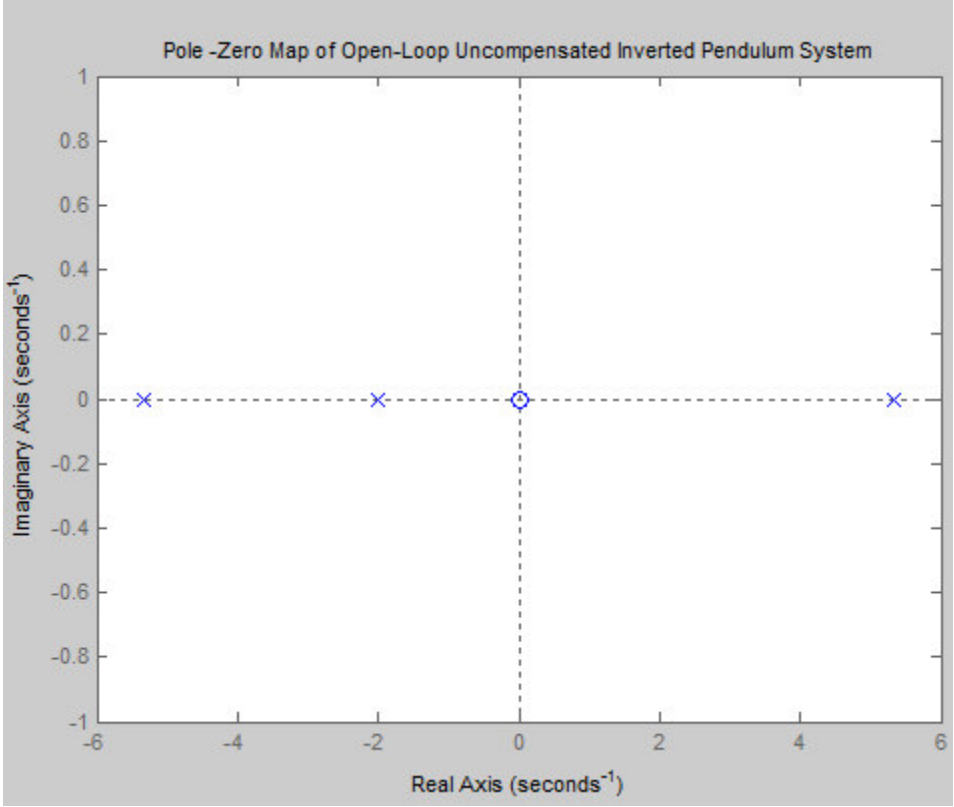


Figure II.5 Pôles et zéros du système.

La position des pôles du pendule inversé (en boucle ouverte) montre que le système est instable, car l'un des pôles de la fonction de transfert se trouve sur le demi-côté droit du plan S.

II.3.3.4 L'analyse de lieu des racines

En concevant une compensation pour n'importe quelle installation, il est important d'observer la réaction de rétroaction unitaire en boucle fermée pour vérifier la stabilité. De nombreux systèmes sont instables en boucle ouverte mais stables en boucle fermée. L'inverse est également possible que le système soit stable en boucle ouverte mais instable en boucle fermée, bien que ce cas soit rare. Le système non compensé en boucle fermée peut être étudié en visualisant la trajectoire d'analyse de lieu des racines du système. La figure II.6 montre le parcours du système local.

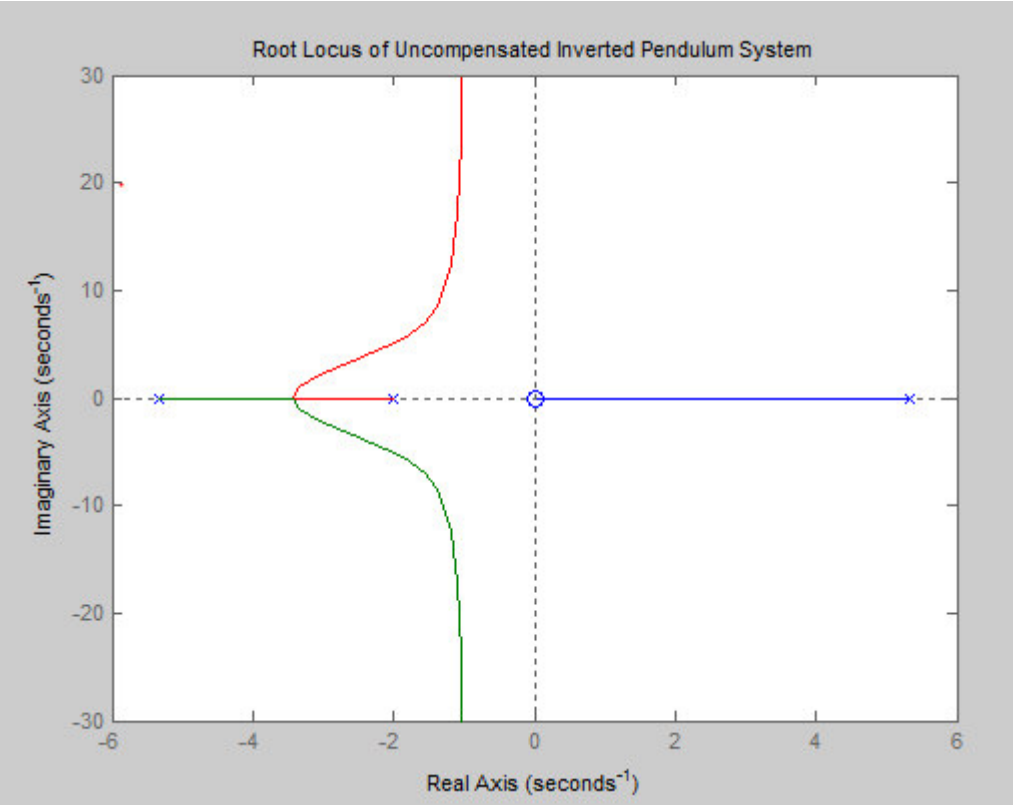


Figure II.6 Lieu des racines du système en boucle ouverte

Le lieu des racines a une branche sur le côté droit de l'axe imaginaire, ce qui indique que le système est instable en boucle fermée pour toutes les valeurs de K.

II.3.4. Contrôle PID

Dans cette section, nous concevrons un contrôleur PID pour le système du pendule inversé. Dans le processus de conception, nous supposerons une installation à une seule entrée à une seule sortie, telle que décrite par la fonction de transfert de la position du pendule. Autrement dit, nous voulons contrôler l'angle du pendule sans tenir compte de la position du chariot.

Le "PID" dans "Contrôle PID" signifie "Proportionnel, Intégral, Dérivé". Ces trois termes décrivent les éléments de base d'un contrôleur PID. Chacun de ces éléments effectue une tâche différente et a un effet différent sur le fonctionnement d'un système. Ces effets sont résumés dans le tableau IV.1 :

Coefficient	Temps de montée	Temps de stabilisation	Dépassement	Erreur Statique
Kp	Diminue	Augmente	Augmente	Diminue
Ki	Diminue	Augmente	Augmente	Annule
Kd	–	Diminue	Diminue	-

Tableau II.2 Tableau récapitulant l'influence d'un PID série sur le système qu'il corrige si l'on augmente séparément l'action proportionnelle (P), intégrale (I) ou dérivée (D)[*20]*

Dans notre cas le contrôleur tentera de maintenir le pendule verticalement vers le haut lorsque le chariot est soumis à une impulsion de 1 Nsec. Dans ces conditions, les critères de conception sont:

Temps de réponse de moins de 2 secondes

Le pendule ne doit pas déplacer plus de 0,04 radians loin de la verticale

II.3.4.1. La structure du système

Puisque nous essayons de contrôler la position du pendule, qui devrait revenir à la verticale après la perturbation initiale, le signal de référence que nous suivons devrait être nul. La force externe appliquée au chariot peut être considérée comme une perturbation impulsive.

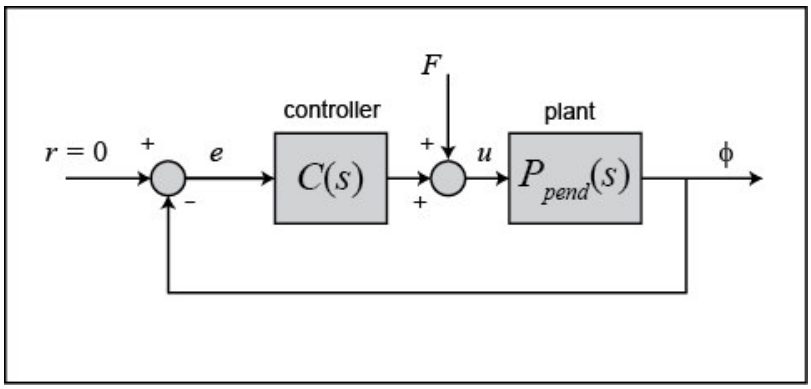


Figure II.7 Schématique du système de pendule et régulateur.

En réarrangent ce schéma, on obtient

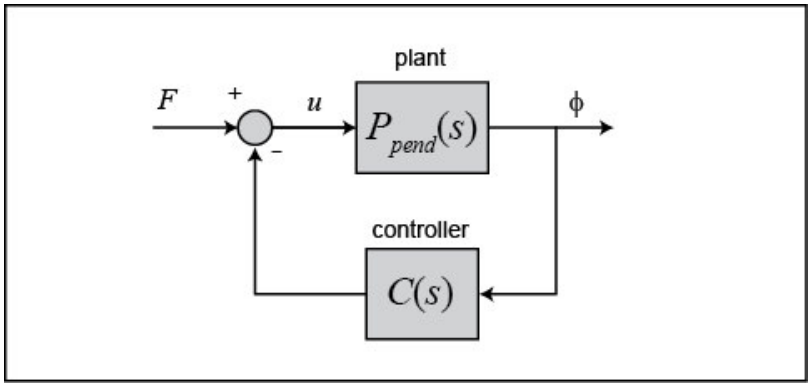


Figure II.8 Schématique simplifié du système

La fonction de transfert résultante T (s) pour le système en boucle fermée d'une entrée de force F à une sortie de l'angle de pendule ϕ est alors déterminée comme suit :

$$T(s) = \frac{\phi(s)}{F(s)} = \frac{P_{pend}}{1+C(s)} \qquad \text{(II.22)}$$

Ensuite, voici comment nous concevons un contrôleur PID dans MATLAB :

Plus précisément, nous définissons notre contrôleur à l'aide de l'objet PID dans MATLAB. Nous utilisons ensuite la commande de feedback pour générer la fonction de transfert en boucle fermée T (s) tel que représenté dans le programme ci-dessus où la force de perturbation F est l'entrée et l'écart de l'angle du pendule par rapport à la verticale ϕ est la sortie.

```
Kp = 1;
Ki = 1;
Kd = 1;
C = pid(Kp,Ki,Kd);
T = feedback(P_pend,C);
```

Maintenant, nous pouvons commencer à régler notre contrôleur. Tout d'abord, examinons la réponse du système en boucle fermée à une perturbation impulsionnelle pour cet ensemble initial de gains de contrôle.

```
t=0:0.01:10;
impulse(T,t)
title('Response of Pendulum Position to an Impulse Disturbance under PID Control: Kp = 1, Ki = 1, Kd = 1');
```

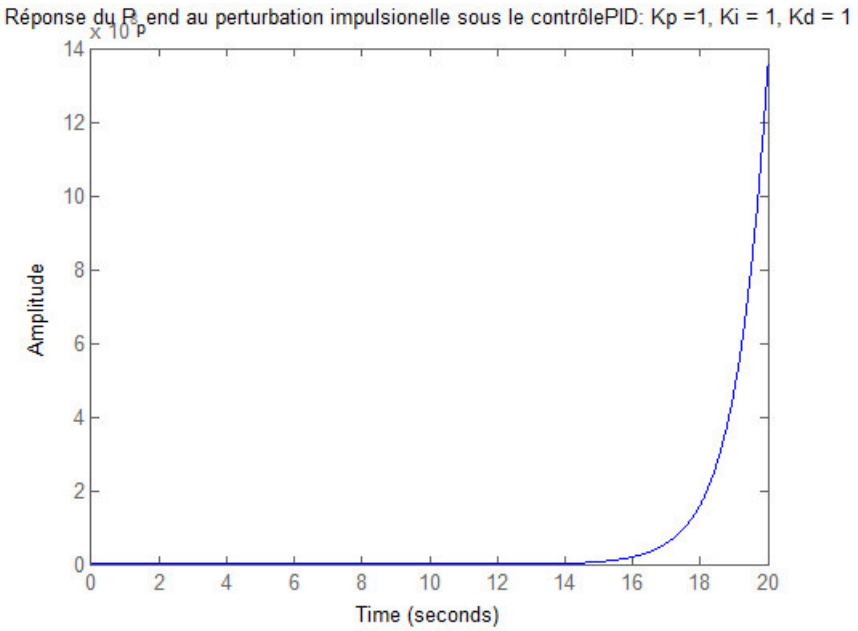


Figure II.9 Contrôle PID avec Kp =1, Ki = 1, Kd = 1

Cette réponse n'est toujours pas stable. Commençons à modifier la réponse en augmentant le gain proportionnel. Augmenter la variable Kp pour voir quel effet il a sur la réponse

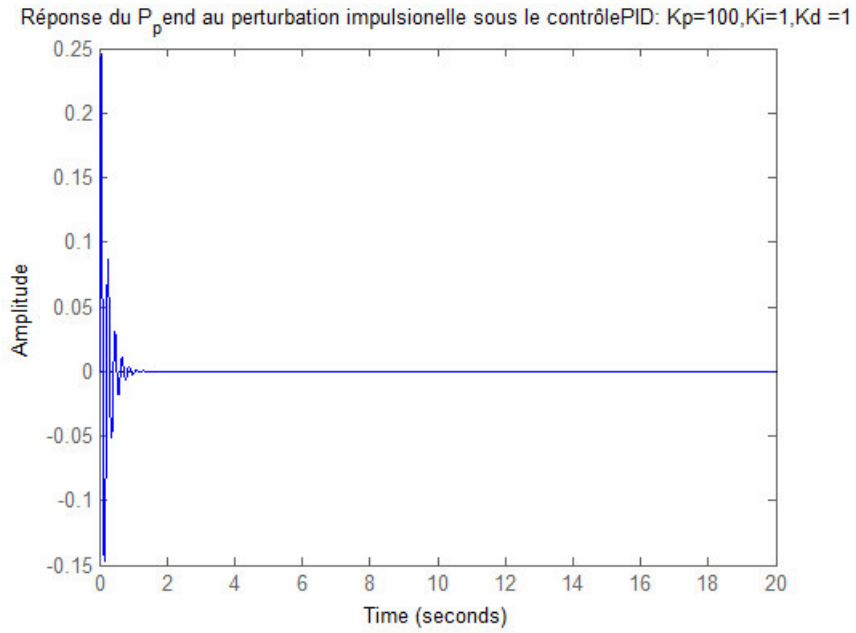


Figure II.10 Contrôle PID avec Kp =100, Ki = 1, Kd = 1

Le temps de réponse du système est 1.55secondes et le dépassement maximal est de 0.242 Radian. Vu que la réponse tend vers 0 dans une manière suffisamment rapide, nous constatons que l'action intégrale en plus n'est pas nécessaire. Cependant, le dépassement maximale peuvent être diminue en augmentant le contrôle dérivative. Apres quelques essais on constate qu'un gain dérivé de Kd = 3 fournit une réponse satisfaisante.

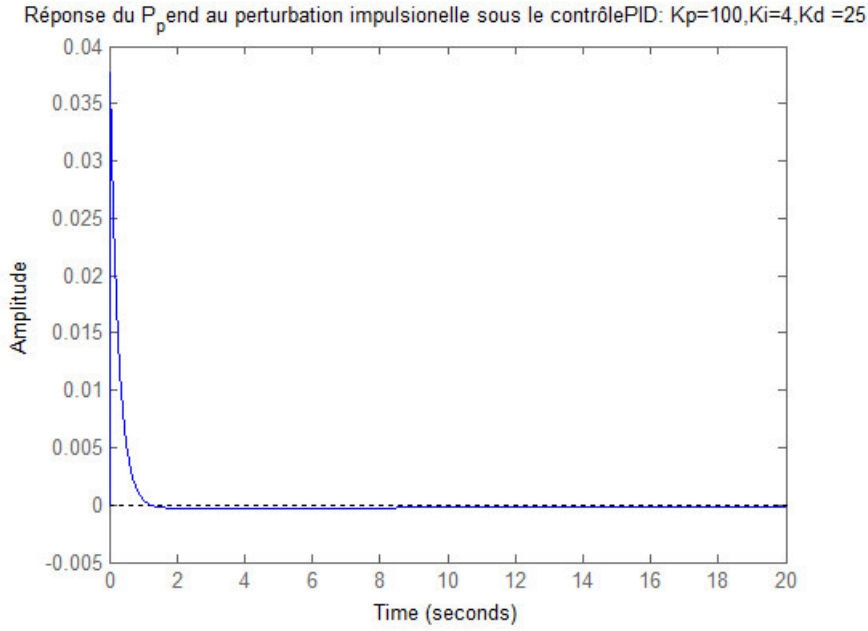


Figure II.11 Contrôle PID avec Kp =100, Ki = 4, Kd = 25

Le dépassement a été réduit de sorte que le pendule ne déplace pas plus de 0,04 radians loin de la verticale. Comme toutes les exigences de conception ont été remplies, aucune autre itération n'est nécessaire.

II.4. Conclusion

Vu que le contrôle en PID est idéal pour les systèmes SISO nous avons dans ce chapitre l'appliqué pour la commande de position du pendule. Nous avons arrivé à stabilisé le système vu que la qualité essentielle pour un système régulé qui est exigé à tout prix est la stabilité. Avec plusieurs essaies en changeant les gains K_p K_i et K_d nous avons trouvé le compromis entre la précision et la rapidité du système. Assez rapidement et avec moins de déviations par rapport à la position d'équilibre, le pendule est stabilisé par le contrôleur PID que nous avons conçu.

Chapitre 3 : Les matériaux utilisés

III .1. Introduction

Après avoir étudié la modélisation mathématique et simulation de notre projet, nous arrivons à le réaliser pratiquement. Dans ce chapitre nous allons décrire les matériels utilisés et leurs fonctions dans le système. Nous devions considérer beaucoup de facteurs pour faire les meilleurs choix des matériaux. Grâce à cette partie de notre projet nous avons augmenté nos connaissances dans les différents aspects de l'Automatique.

Nous avons vu l'utilité des modules que nous avons étudié lors de notre étude de Génie Electrique et puis l'Automatique. Juste pour mentionner quelques-uns, le module de Machines Electriques nous a familiarisés avec le fonctionnement des moteurs alors que les Capteurs et Mesures étaient importants pour notre choix éclairé des capteurs utilisés et leurs fonctions. La connaissance de Régulation Automatique et la Commande des systèmes ont joué un grand rôle pour rendre notre système automatique comme nous le voulions.

III.2 La structure du système :

Notre système du pendule inversé est divisé en deux parties principales :

1. La partie mécanique
2. La partie électronique

III.2.1 La partie mécanique

Cette partie du système est constituée de l'ensemble des rails, chariot et pendule. En utilisant du métal et du bois, nous avons réussi à mettre en place les rails et le chariot. Il était essentiel que les deux rails soient complètement parallèles l'un à l'autre pour assurer le bon mouvement du chariot sur les rails. Le chariot doit glisser facilement sur les rails pour qu'il puisse garder le pendule dans la position haute désirée. Il doit faire des mouvements de va-et-vient le long des rails sans dépasser les extrémités.

Le but de ces deux(les rails et le chariot) consiste à maintenir le pendule en place. En outre, nous avons appliqué de l'huile sur les rails pour réduire le frottement entre les rails et la face inférieure du chariot qui est également en métal. Le support en bois est nécessaire pour la stabilité des rails et aussi pour le placement de moteur et la deuxième poulie aux extrémités des rails.

Nous avons attaché un pendule de 33cm qui pèse 55 grammes sur le chariot en le permettant d'avoir qu'un axe de rotation. Le pendule est attaché de telle façon qu'il a un axe de rotation juste en haut de chariot. Le montage de ces trois matériaux est en figure 4.1.

Figure III.1 : Les rails, le chariot et le pendule

III.2.2 La partie électronique

Nous avons dans cette partie les composants qu'utilise l'énergie électrique pour commander le bon fonctionnement du système mécanique.

III.2.2.1 Le moteur à Courant Continu

Nous avons un seul actionneur dans ce système, le moteur à courant continue. Comme le dit son nom, ce moteur marche en courant continu et il convertit l'énergie électrique en énergie mécanique. Pour notre système nous avons utilisé ce moteur (Figure III.2) pour commander le mouvement linéaire de chariot. Ceci a était possible grâce à la courroie qu'on a attaché sur le chariot qui parcourt le moteur. Le moteur lui-même est contrôlé par le program en Arduino et il agit également comme une poulie.

Caractéristiques du moteur

Micro moteurs à courant continu 12V

Diamètre 38.5mm

Longueur 57mm

Arbre 3.17mm.

Figure III.2 Moteur à courant continue

III.2.2.1.1 Généralités sur le moteur à courant continu

En comparaison aux autres technologies, **le moteur à courant continu** est une machine qui transforme l'énergie électrique en énergie mécanique, il est constitué de : l'inducteur (stator), l'induit (rotor), un aiment, un axe, le balai et le collecteur.

L'inducteur (parfois appelé champ) produit le flux magnétique dans la machine. Il est constitué d'un électro-aimant qui engendre la force magnétomotrice nécessaire à la production du flux. Dans les machines bipolaires (à deux pôles), deux bobines excitatrices sont portées par deux pièces polaires montées à l'intérieur d'une culasse.

Les bobines excitatrices sont alimentées en courant continu, et le courant qui les traverse porte le nom de courant d'excitation. Elles sont composées de plusieurs centaines de spires et portent un courant relativement faible. Les bobines sont bien isolées des pièces polaires afin de réduire les risques de court-circuit à la terre. Dans certains moteurs à courant continu, les bobines et pièces polaires sont remplacées par des aimants permanents.

L'induit est composé d'un ensemble de bobines identiques réparties uniformément autour d'un noyau cylindrique. Il est monté sur un arbre et tourne entre les pôles de l'inducteur. L'induit constitue donc un ensemble des conducteurs que coupe le flux magnétique. Les bobines sont disposées de telle façon que leurs deux côtes coupent respectivement le flux provenant d'un pôle nord et d'un pôle sud de l'inducteur.

Le collecteur est un ensemble cylindrique de lames de cuivre isolées les unes des autres par des feuilles de mica. Le collecteur est monté sur l'arbre de la machine, mais isolé. Les deux fils sortant de chaque bobine de l'induit sont successivement et symétriquement soudés aux lames du collecteur [20].

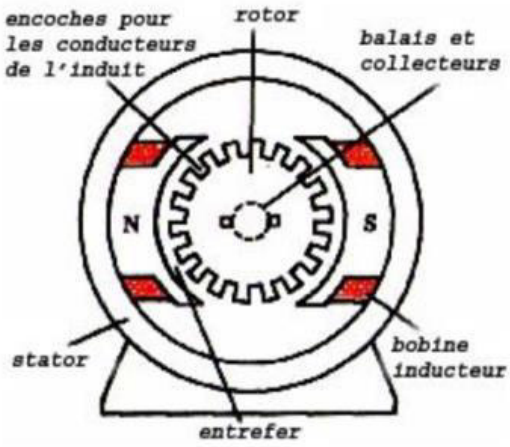


Figure III.3 Constitution du moteur à courant continu *[20]*

Les balais sont fait en carbone, car ce matériau possède une bonne conductivité électrique et est assez doux pour ne pas user indûment le collecteur. Pour améliorer leur conductivité, on ajoute parfois au carbone une petite quantité de cuivre. La pression des balais sur le collecteur peut être réglée à une valeur appropriée grâce à des ressorts ajustables

III.2.2.2 Le Pont en H

La tension maximale que la carte Arduino peut supporter est de 5Volts. Cependant, le moteur que nous utilisons fonction avec 12Volts. Pour cette raison, nous avons besoin d'un pont en H ce qui nous permettre d'alimenter le moteur avec 12V et en même temps commander son mouvement utilisant la carte Arduino. Une attention particulière a été prise pour ne pas confondre les alimentations car cela pourrait griller notre carte Arduino.

III.2.2.2.1 Le Principe de fonctionnement du pont en H [20]

Le pont en H peut être symbolisé par une structure à quatre interrupteurs ouverts au repos (Figure IV.4). Dans ce cas, la tension aux bornes du moteur est nulle. Pour mettre en rotation

le moteur il suffit de fermer un couple d'interrupteurs et de laisser les deux autres au repos. Le moteur est donc aliment´e avec la tension UM=Vcc. Pour modifier le sens de rotation, il suffit de permuter le couple d'interrupteurs fermés et ouverts. On aura donc UM= - Vcc.

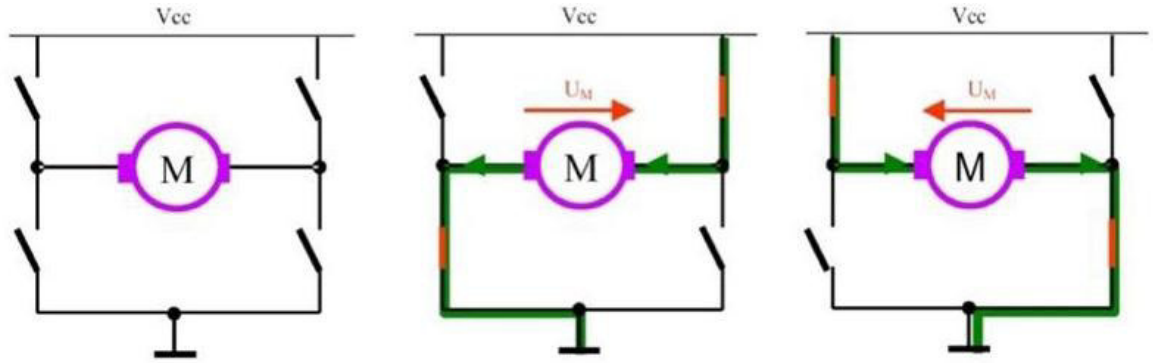


Figure III.4. Schéma de principe de fonctionnement du pont en H *[20]*

III.2.2.2.2 La structure autour de transistors

Les transistors vont assurer la fonction des interrupteurs et fonctionneront en saturer/bloquer. Ainsi en bloquant un couple de transistor et en saturant l'autre, on établit une tension UM aux bornes du moteur, dont on change le signe en modifiant les couples saturés/bloqués. Les diodes placées entre émetteurs et collecteurs des transistors jouent le rôle de diode de roue libre (DRL). Elles permettent de protéger les transistors en devenant passantes lorsque ces derniers sont tous bloqués et qu'il faut évacuer le courant IM de l'inductance du moteur.

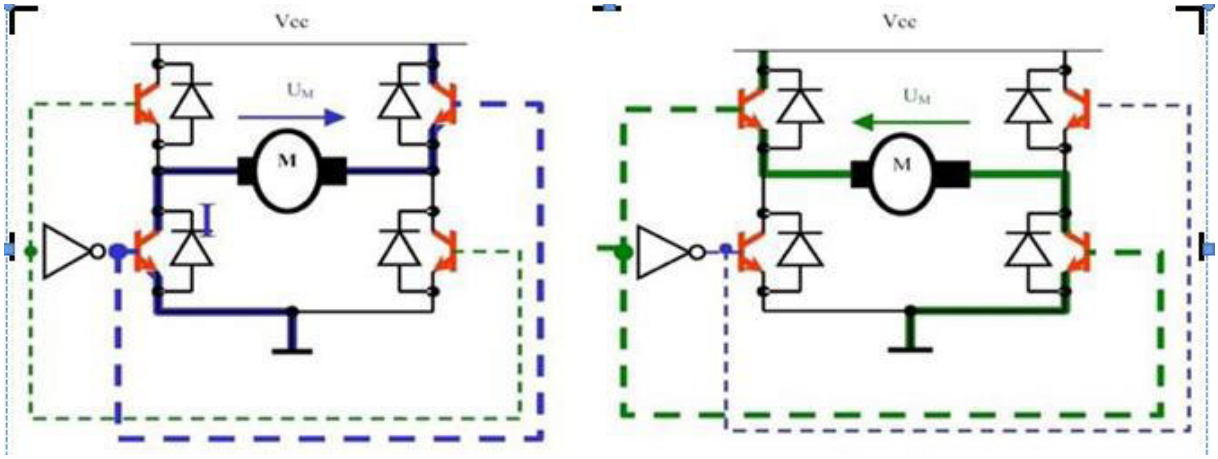


Figure III.5 Pont en H de transistors *[20]*

Il est possible d'activer la rotation d'un moteur à l'aide d'un relais ou d'un transistor. L'inconvénient de l'option transistor (ou relais) est qu'il n'est possible de facilement contrôler le sens de rotation du moteur. La solution réside dans l'utilisation d'un pont H. composant

constitué de plusieurs transistors mais vendu pré assemblé sous forme de circuit intégré. Pour cela on a utilisé dans notre carte de commande le circuit intégré **L298N**

III.2.2.2.3 Pont-H L298N

Ce *breakout board* est un Double Pont-H destiné au contrôle de moteur continu (H-Bridge Motor Driver). Il est basé sur le composant L298N qui est un double Pont-H conçu spécifiquement pour ce cas d'utilisation.

C'est un module extrêmement utile pour le contrôler de robots et ensembles mécanisés. Il peut contrôler deux moteurs à courant continu ou un moteur pas-à-pas 4 fils 2 phases. il est conçu pour supporter des tensions plus élevées, des courants importants tout en proposant une commande logique TTL (basse tension, courant faibles, idéal donc pour un microcontrôleur).

Il peut piloter des charges inductives comme des relais, solénoïdes, moteurs continus et moteurs pas-à-pas. Les deux types de moteurs peuvent être contrôlés aussi bien en vitesse (PWM) qu'en direction. Toutes les sorties en puissance sont déjà protégées par des diodes anti-retour.

Il s'agit d'un module prêt à l'emploi.

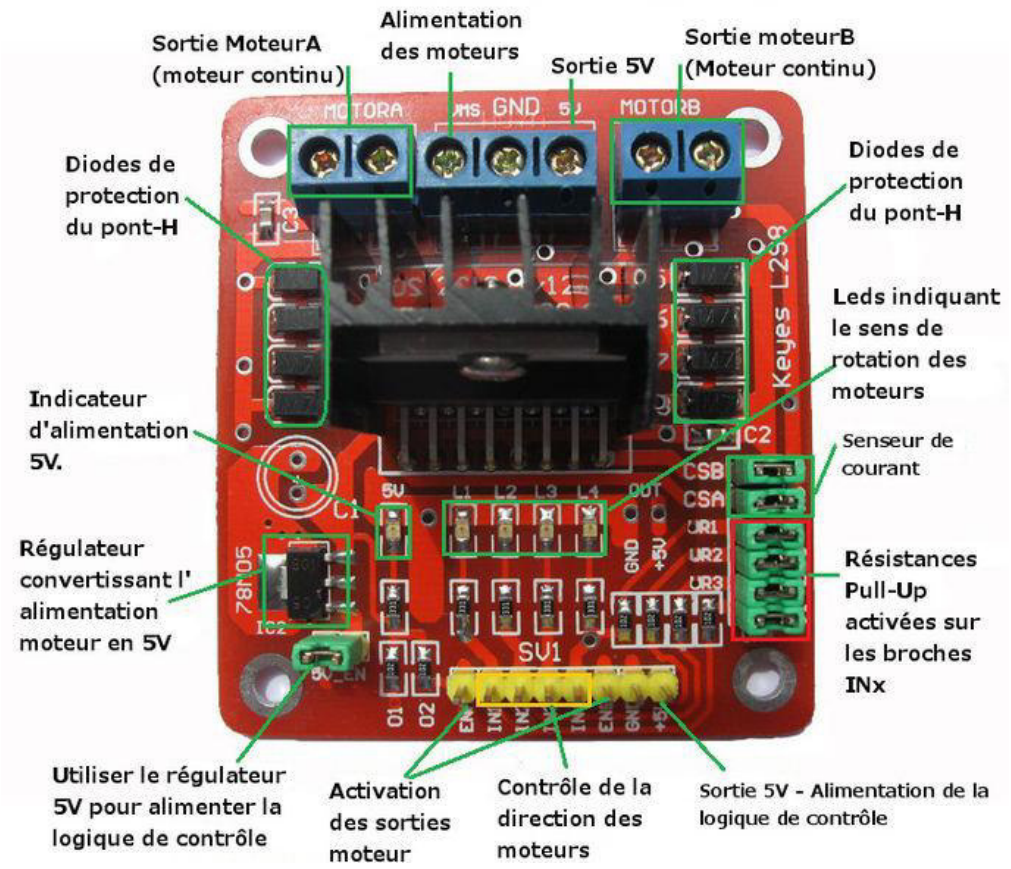


Figure III.6 Les détails techniques du L298N

III.2.2.3 Les microcontrôleurs

Il est très important que nous expliquions le fonctionnement des microcontrôleurs ici en considérant qu'Arduino est un panneau d'interface microcontrôleur. Les microcontrôleurs sont beaucoup utilisés pour les applications intégrées et dans les appareils automatiques, les outils électriques, les jouets, les dispositifs médicaux implantables, les machines de bureau, les systèmes de contrôle des moteurs, les appareils, les télécommandes et autres types de systèmes embarqués.

Sa nature principale est l'autosuffisance et le faible coût. Il ne pas destiné à être utilisé comme un dispositif de calcul dans le sens classique, c'est-à-dire qu'un microcontrôleur n'est pas conçu pour être une machine de traitement de données mais plutôt un noyau intelligent pour un système dédié spécialisé.

Un microcontrôleur comprend habituellement un processeur central, des ports d'entrée et de sortie, une mémoire pour le stockage de programme et de données, une horloge interne et un ou plusieurs périphériques tel que des minuteries, des compteurs, des convertisseurs

analogique-numériques, des équipements de communication série et des circuits de surveillance.[9]

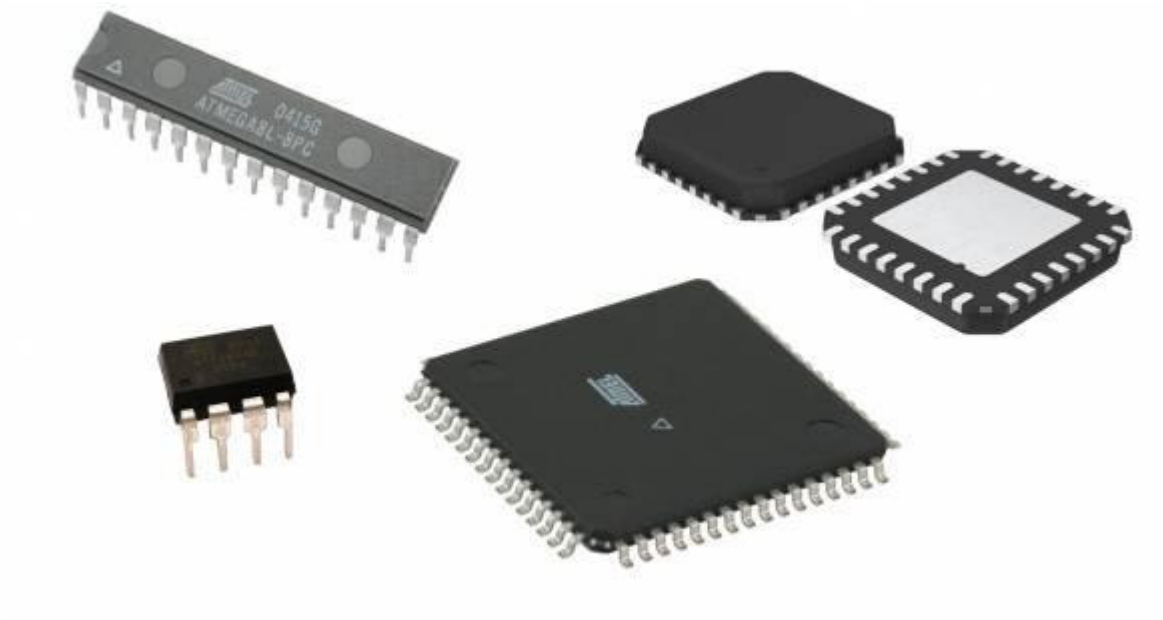


Figure III.7. Les différents microcontrôleurs

III.2.2.3.1 Critères de choix des microcontrôleurs

Il existe un très grand nombre de modèles de microcontrôleurs, et de nombreux fabricants proposent chacun plusieurs familles de microcontrôleurs, comptant chacune parfois des centaines de modèles. On trouve des microcontrôleurs allant du petit circuit à 6 pattes coûtant une fraction d'€ jusqu'à des circuits comptant des centaines de pattes et coûtant des dizaines d'€.

Les critères à considérer pour le choix de microcontrôleurs sont :

- Le nombre de pattes d'entrées-sorties.
- Taille de la mémoire de programme (ROM)
- Taille de la mémoire vive (RAM)
- Consommation électrique (tension de fonctionnement, courant consommer)
- « La Puissance » du processeur
- La fréquence de l'horloge
- Le prix

- L'environnement de développement (matériel et logiciel, Le coût et la disponibilité)
- L'expérience

III.2.2.4 L'Arduino[32]

Nous l'avons déjà mentionné plusieurs fois alors nous allons clarifier ici ce que c'est l'Arduino est quelle est son fonctionnement dans le système. En bref, nous pouvons dire que les cerveaux de notre système s'y trouvent. Pour que le moteur change son direction de rotation par exemple, il faut que l'Arduino 'pense' à cela et puis 'dit' au moteur du le faire et aussi les calculs nécessaires pour le bon fonctionnement des capteurs sont faites dedans.

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses.

Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles. Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartez 16 MHz. Pour programmer cette carte, on utilise l'logiciel IDE Arduino. [30]

C'est une plateforme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur. Arduino peut être utilisé pour développer des objets interactifs, pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteurs, et pouvant contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielles.

Les projets Arduino peuvent être autonomes, ou bien ils peuvent communiquer avec des logiciels tournant sur votre ordinateur (tels que Flash, Processing ou MaxMSP). Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées pré-assemblées; le logiciel de développement open-source peut être téléchargé gratuitement. Le langage de

programmation Arduino est une implémentation de Wiring, une plateforme de développement similaire, qui est basée sur l'environnement multimédia de programmation Processing.

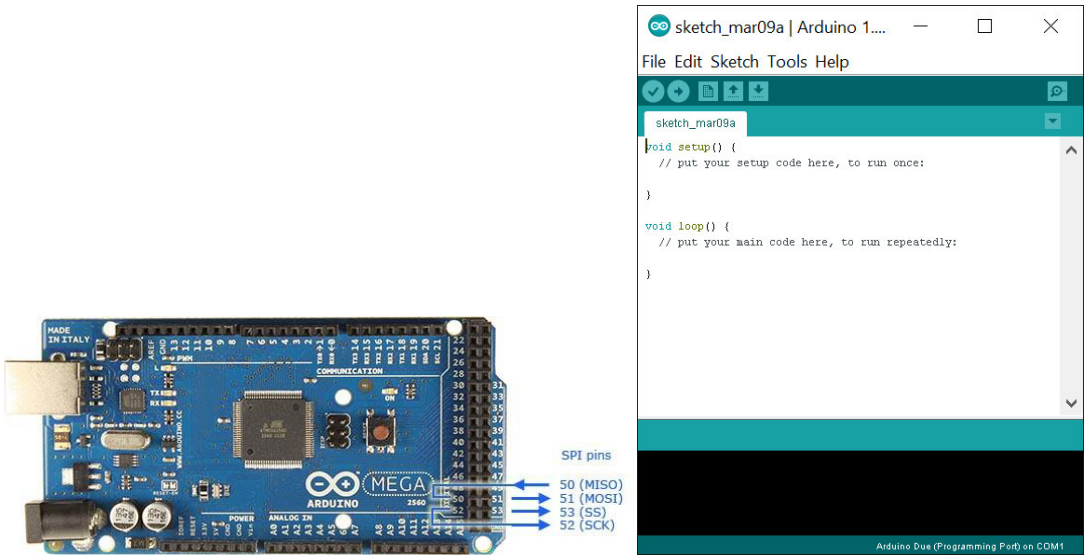


Figure III.8 La carte Arduino Mega et le logiciel affiché sur l'ordinateur

III.2.2.4.1 Pourquoi Arduino ?

Il y a de nombreux microcontrôleurs et de nombreuses plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, et beaucoup d'autres qui offrent des fonctionnalités comparables. Tous ces outils prennent en charge les détails compliqués de la programmation des microcontrôleurs et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs, tout en offrant plusieurs avantages pour les enseignants, les étudiants et les amateurs intéressés par les autres systèmes :

- **Pas cher** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plateformes. La moins chère des versions du module Arduino peut être assemblée à la main!
- **Multi-plateforme** : Le logiciel Arduino, écrit en Java, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- **Un environnement de programmation clair et simple**: L'environnement de programmation Arduino c'est-à-dire le logiciel Arduino est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puisse en tirer profit également. Pour les enseignants, il est basé sur l'environnement de programmation Processing : les étudiants qui apprennent à programmer dans cet environnement seront déjà familiarisés avec l'aspect du logiciel Arduino.

- **Logiciel Open Source et extensible** : Le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmateurs expérimentés. Le langage peut être aussi étendu à l'aide de librairies C++, et les personnes qui veulent comprendre les détails techniques peuvent reconstruire le passage du langage Arduino au langage C pour microcontrôleur AVR sur lequel il est basé. De la même façon, vous pouvez ajouter du code du langage AVR-C directement dans vos programmes Arduino si vous voulez.
- **Matériel Open source et extensible** : Les cartes Arduino sont basé sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, etc. Les schémas des modules sont publiés sous une licence Creative Commons, et les concepteurs de circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dans le but de comprendre comment elle fonctionne et pour économiser de l'argent.[8]

III.2.2.4.2 Les différentes cartes Arduino

Il existe plusieurs différentes cartes Arduino qui ont les caractéristiques et spécifications différentes. Citer ci-dessous est quelque uns :

- Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un ATmega8.
- L'extrémité d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- L'Arduino Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- Le NG d'Arduino plus, avec une interface d'USB pour programmer et usage d'un ATmega168.
- L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- L'Arduino Diecimila, avec une interface d'USB et utilise un microcontrôleur ATmega168.
- L'Arduino Duemilanove ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- L'Arduino Mega, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilisations microcontrôleur ATmega328.

- L'Arduino Mega2560, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- L'Arduino Leonardo, avec un morceau ATmega3U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.

Figure III.9. Les différentes cartes Arduino selon leurs tailles

III.2.2.4.3 Arduino UNO

Dans notre système nous utilisons la carte Arduino Uno. Le schéma qui suit montre les différents composants de cette carte:

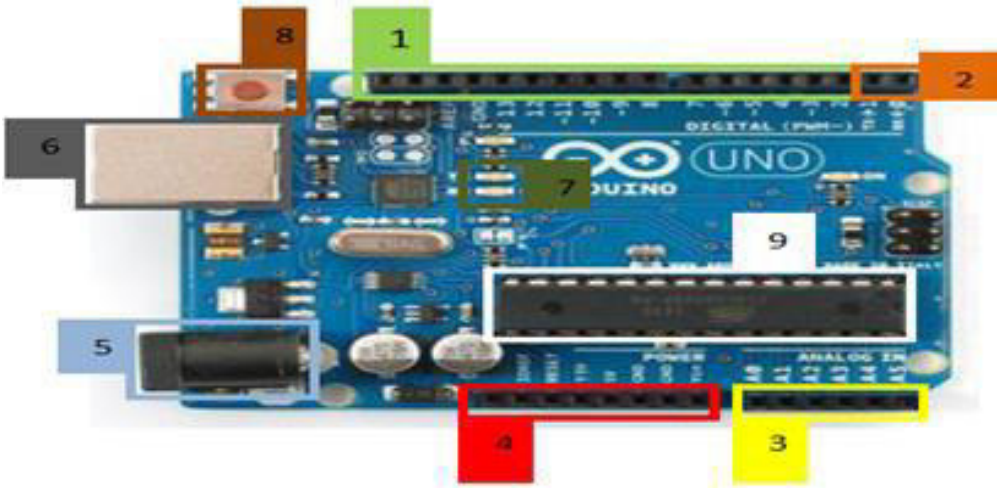


Figure III.10. Les composants d'Arduino Uno *[30]*

1 : Entrées/Sorties digitales (14 pin)

2 : Pin de communication

3 : Entrées/Sorties Analogique (peuvent aussi servir comme E/S digitales, 6 pin)

4 : Broches d'alimentation pour le montage (5V, 3,3V, GND…)

5 : Entrée d'alimentation externe pour la carte

6 : Port USB pour la communication entre le PC et la carte

7 : LED indiquant la communication avec l'ordinateur (Tx, Rx)

8 : Botton RESET

9 : Microcontrôleur ATmega328

III.2.2.4.3.1 La constitution de la carte Arduino UNO

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un boot loader de façon à ce qu'un programmateur dédié ne soit pas nécessaire.

III.2.2.4.3.2 Le Microcontrôleur ATMega328

Un microcontrôleur ATMega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino UNO.[30]

Figure III.11. Microcontrôleur ATMega328 *[29]*

Le microcontrôleur ATMega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- ***La mémoire Flash***: C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont boot loader de 0.5 ko).
- ***RAM*** : c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- ***EEPROM*** : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. [10]

III.2.2.4.3.4 Les sources de l'alimentation de la carte

On peut distinguer deux genres de sources d'alimentation (Entrée Sortie) et cela comme suit :

- ***Vin*** : La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- ***5V*** : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V

régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation Vin via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

- ***3,3 V*** : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA. [9]

III.2.2.4.3.5 Les entrées & sorties

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions pinMode( ), digitalWrite( ) et digitalRead( ) du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digital Write (broche, HIGH).

En plus, certaines broches ont des fonctions spécialisées

• Interruptions Externes: Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. -Impulsion PWM (largeur d'impulsion modulée): Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analog Write ( ).

• SPI (Interface Série Périphérique): Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.

• I2C: Broches 4 (SDA) et 5 (SCL), supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire interface - interface "2 fils").

• LED: Broche 13, il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

La carte UNO dispose de 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (c'est-à-dire sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analogRead( ) du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023).

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

III.2.2.4.3.6 Les ports de communications

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX).

On utilise (RX) pour recevoir et (TX) pour transmettre les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB-vers-série de la carte, pour assure l'interface entre les niveaux TTL et le port USB de l'ordinateur

Comme il y a un port de communication virtuel pour le logiciel sur l'ordinateur, la connexion série de l'Arduino est très pratique pour communiquer avec un PC

III.2.2.5 Le capteur (gyroscope/accéléromètre)

Pour mesurer électroniquement les angles du pendule, nous avons utilisé le capteur MPU6050 Gyroscope/accéléromètre. Cet appareil intégré contient un gyroscope du 3-axes et un accéléromètre du 3-axes. L'accéléromètre mesure la force appliqué à l'appareil dans chaque axe alors que le gyroscope mesure la vitesse de rotation (en degré/ seconde). En utilisant le code en Arduino nous arrivons à transformé la valeur de gyroscope vers l'angle(en degrés).

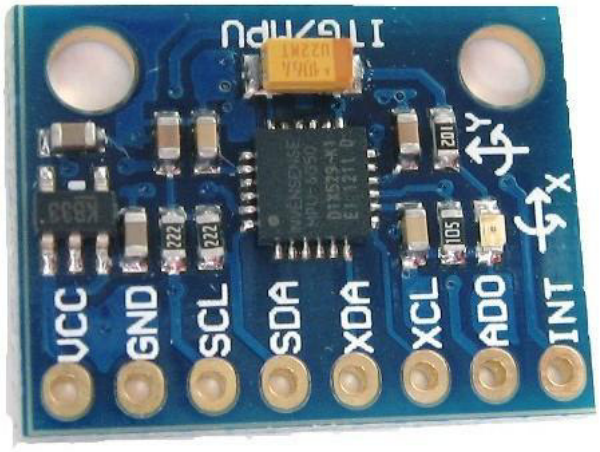


Figure III.12 Gyroscope/Accéléromètre

III.2.2.6 Les interrupteurs de la fin de course

Deux interrupteurs de fin des courses ont été placés aux deux extrémités des rails pour limité le mouvement du chariot au-delà des rails.

Figure III.13 Deux interrupteurs de la fin de course

III.2.2.7 L'alimentation DC

Nous avons utilisé l'alimentation du courant continu montré en figure III.14 pour alimenter le moteur. La carte Arduino est alimentée par l'ordinateur (les deux sont connectés pour faciliter le chargement des programmes de l'ordinateur vers Arduino)

Figure III.14 L'alimentation DC

III.3 Conclusion

Dans ce chapitre nous avons donné quelques concepts théoriques et nous avons expliqué chaque composant principal du notre montage. Le bon fonctionnement de tous les matériaux cités ici est essentiel pour le fonctionnement global de notre système.

Pour l'Arduino, nous avons expliqué un des ses deux parties essentielles (la partie matérielle) en citant ces caractéristiques, les différents types des cartes Arduino et la composition particulièrement d'Arduino Uno. La partie de programmation sera expliquée dans le prochain chapitre.

Chapitre 4 : Conception et programmation du système

IV.1 Introduction

Le système que nous réalisons consiste à deux parties essentielles ; la partie mécanique et la partie électronique. Dans le chapitre précédent nous avons décrit tous les composants constituant la partie mécanique mais concernant la partie électronique, nous n'avons pas traité la programmation impliquée lors de la réalisation du système.

Dans ce dernier chapitre nous introduisons la partie logicielle de notre carte Arduino qu'on va utiliser pour programmé le système virtuel en Proteus et puis le système réel que nous réalisons.

IV.2 Le logiciel Arduino

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information du ce programme appelé IDE Arduino.

IV.2.1 Structure générale du programme (IDE Arduino)

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C.

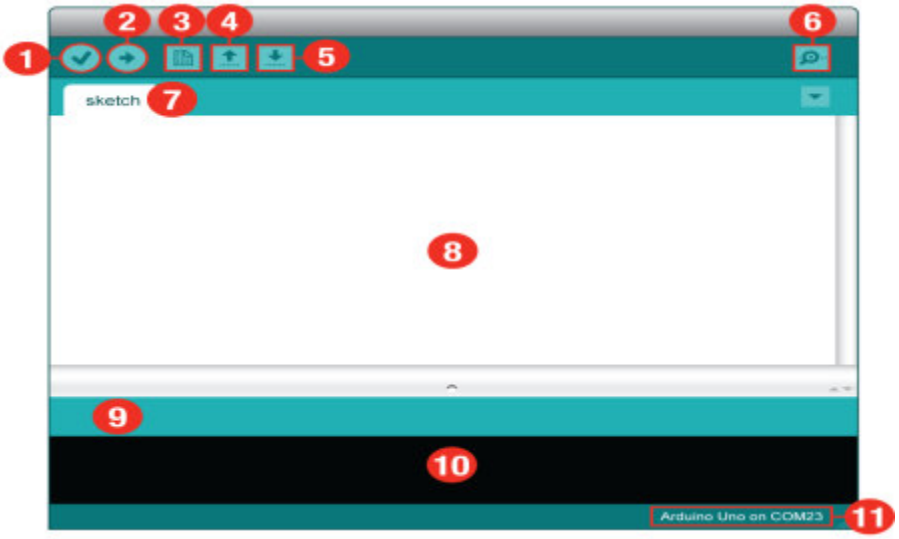


Figure IV.1 Description de l'interface d'Arduino IDE

1 : Compiler et vérifier le code.

2 : Compiler et téléverser/télécharger le code vers la carte Arduino.

3 : Nouvel onglet de la fenêtre de code.

4 : Ouvrir un projet.

5 : Sauvegarder.

6 : Moniteur série (pour voir les messages qui a été programmes dans le programme).

7 : L'onglet en cours avec le nom de programme.

8 : Editeur de code.

9 : Zone de message.

10 : Le Console des erreurs (zone de notification).

11 : Type de la carte Arduino + port série sur lequel la carte est branchée.

IV.2.2 Injection du programme

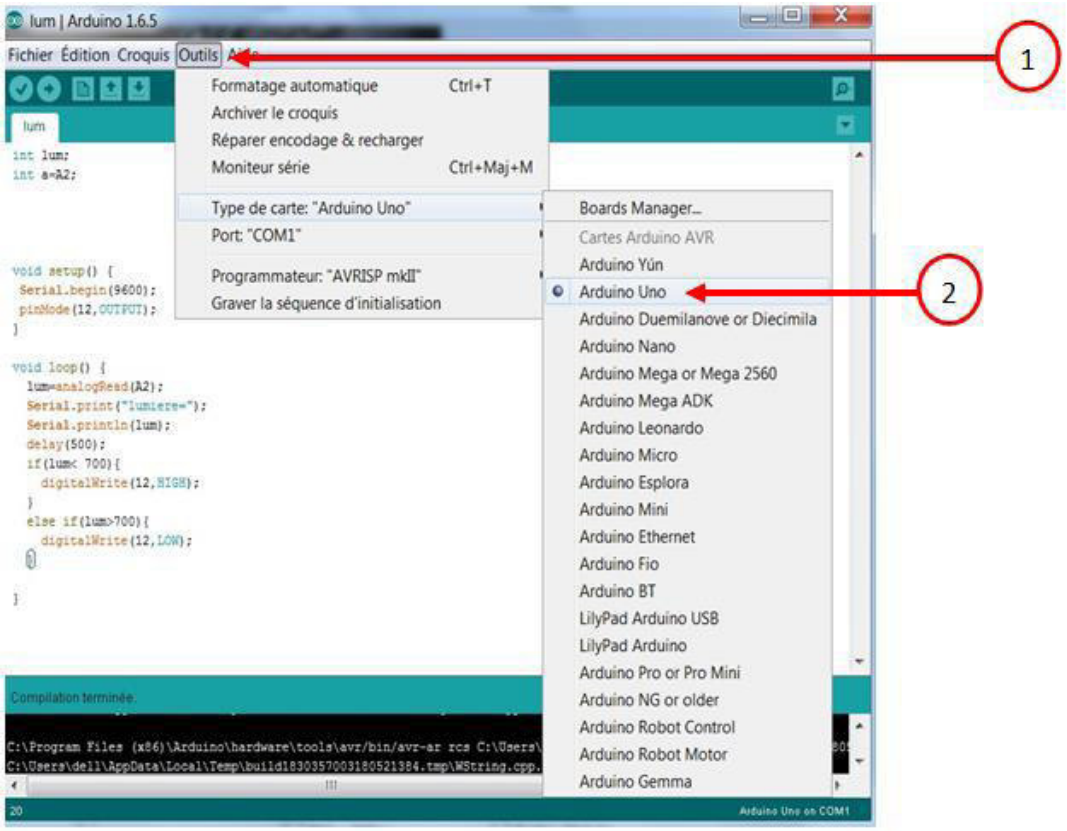


Figure IV.2 Paramétrage de la carte étape1

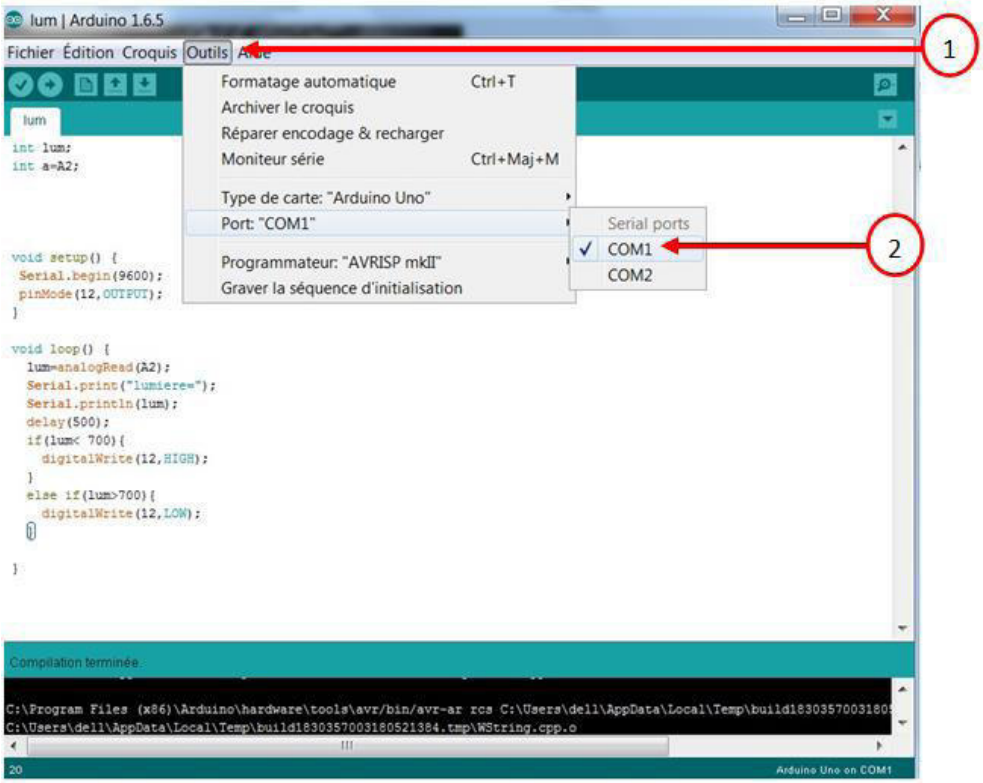


Figure IV.3 Paramétrage de la carte étape2

IV.2.3 Description du programme

Un programme Arduino est une suite d'instructions élémentaires sous forme textuelle (ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de codes.

- ***Commentaires***

Les **commentaires** sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme.

- ***Définition des variables***

Une variable est un nombre. Ce nombre est stocké dans un espace de la mémoire vive (RAM) du microcontrôleur et il a la particularité de changer de valeur. Le nom de variable accepte quasiment tous les caractères sauf le point, la virgule et les accents. Voilà les types de variables les plus répandus

Type	Nombre stocké	Valeurs maximales du nombre stocké	Nombre sur X bits	Nombre d'octet
int	entier	-32 768 à +32 767	16 bits	2 octet
long	entier	-2 147 483 648 à +2 147 483 647	32 bits	4 octets
char	entier	-128 à +127	8 bits	1 octet
float	décimale	$-3.4* 10^{38}$ à $+3.4*10^{38}$	32 bits	4 octet
double	décimale	$-3.4* 10^{38}$ à $+3.4*10^{38}$	32 bits	4 octet

Tableau IV.1 Les différentes types des variables

- ***Configuration des entrées et des sorties void setup ( )***

Les broches numériques de l'Arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques; pin mode (nom, état) est une des quatre fonctions relatives aux entrées/sorties numériques.

- ***Programmation des interactions void loop ( )***

Dans cette boucle, on définit les opérations à effectuer dans l'ordre **digital write** (nom, état) est une autre des quatre fonctions relatives aux entrées – sorties numériques.

IV.2.4 Les étapes de téléchargement du programme

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la carte Arduino via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.

2. On vérifie ce programme avec le logiciel Arduino (compilation).

3. Si des erreurs sont signalées, on modifie le programme.

4. On charge le programme sur la carte.

5. On câble le montage électronique.

6. L'exécution du programme est automatique après quelques secondes.

7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (pile 9 volts par exemple).

8. On vérifie que notre montage fonctionne.

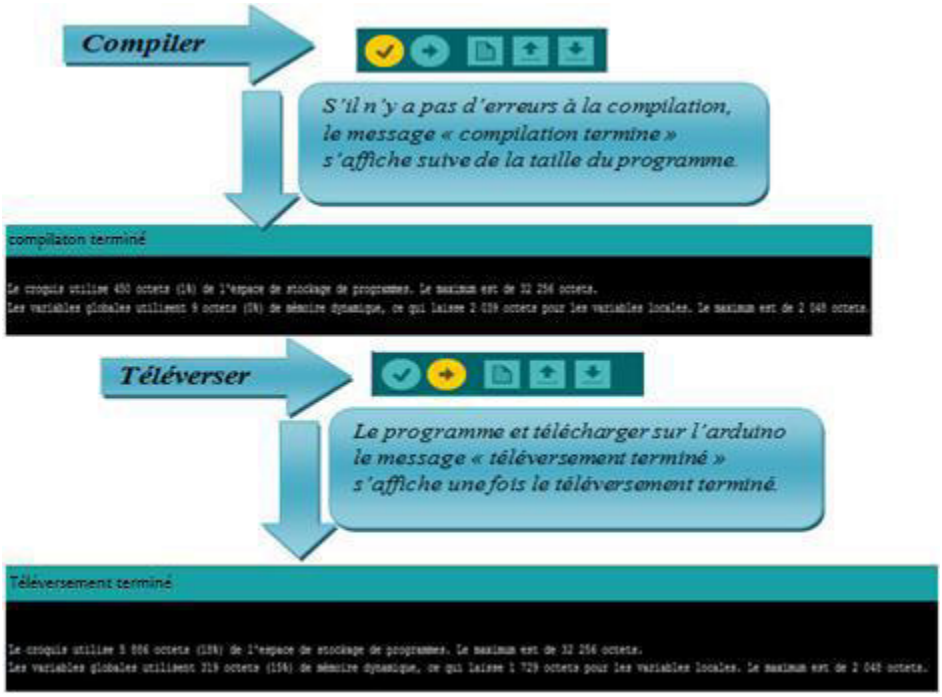


Figure IV.4 Les étapes de téléchargement du code

IV.3 Simulation en Proteus

Avant de passer à la réalisation pratique de notre système, nous avons eu recours à la simulation des différentes parties du système, pour cela on utilise le logiciel Proteus qui est un très bon logiciel de simulation en électronique.

La simulation permet d'ajuster et de modifier le circuit comme si on manipulait un montage réel. Ceci permet d'accélérer le prototypage et de réduire son coût. Il faut toujours prendre en

considération que les résultats obtenus de la simulation sont un peu différents de celles du monde réel.

Figure IV.5 Le programme de Proteus en ouverture

ISIS est un éditeur de schémas qui intègre un simulateur analogique, logique ou mixte. Toutes les opérations se fassent dans cet environnement, aussi bien la configuration des différentes sources que le placement des sondes et le tracé des courbes.

IV.3.1 La fenêtre du logiciel

Dans cette section nous allons commencer par la présentation de la fenêtre du logiciel ISIS

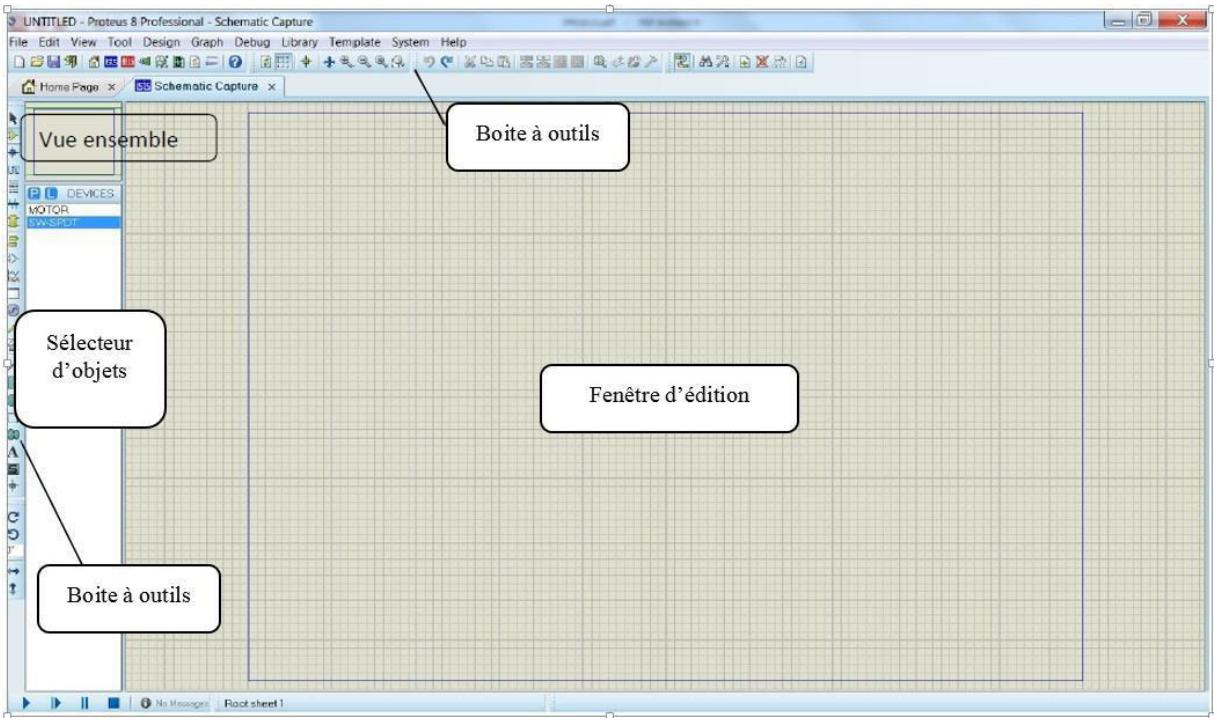


Figure IV.6 Fenêtre principale d'ISIS.

a. Fenêtre d'ensemble (Vue d'ensemble)

Le cadre en bleu délimite l'espace de travail tel qu'il a été défini par la commande *'Définir taille des feuilles'* du menu *'système'*.

Le cadre en vert délimite La zone de travail, c'est à dire la partie du schéma visible dans la fenêtre principale.

- Nous pouvons déplacer cette zone de travail en pointant la souris sur la zone désirée de la fenêtre d'ensemble et en effectuant un clic gauche.
- Nous pouvons redéfinir la zone de travail dans la fenêtre d'ensemble en appuyant sur la touche majuscule 'shift 'du clavier, associée au déplacement de la souris en maintenant appuyé le bouton gauche.

b. Fenêtre d'édition

La surface la plus grande de l'écran s'appelle "Fenêtre d'édition" et se comporte comme une fenêtre de dessin. C'est là que nous plaçons et câblez les composants.

c. La boite à outils

Elle est composée d'un ensemble d'icônes dont les fonctions seront détaillées ultérieurement et d'un sélecteur d'objet utilisé pour choisir les boîtiers, le style des pastilles, des traces, des traversées, etc.

IV.3.2 Placement et câblage des composants

Le circuit que nous allons tracer comme un exemple est représenté en Fig (III.30).

Nous commencerons par un exemple de circuit simple qui se compose d'un moteur on pont-H par interrupteur double plus l'alimentation (Vcc = 5Volt et la masse GND). Commençons par un clic gauche sur l'icône d'un composant, Nous avons ainsi accès aux bibliothèques de composants. Un clic sur "P", du "sélecteur d'objets" ouvre une nouvelle fenêtre .

Nous pouvons maintenant choisir un composant dans les différentes bibliothèques. Un double clic place le composant sélectionné dans le sélecteur d'objets et le rend disponible pour l'édition.

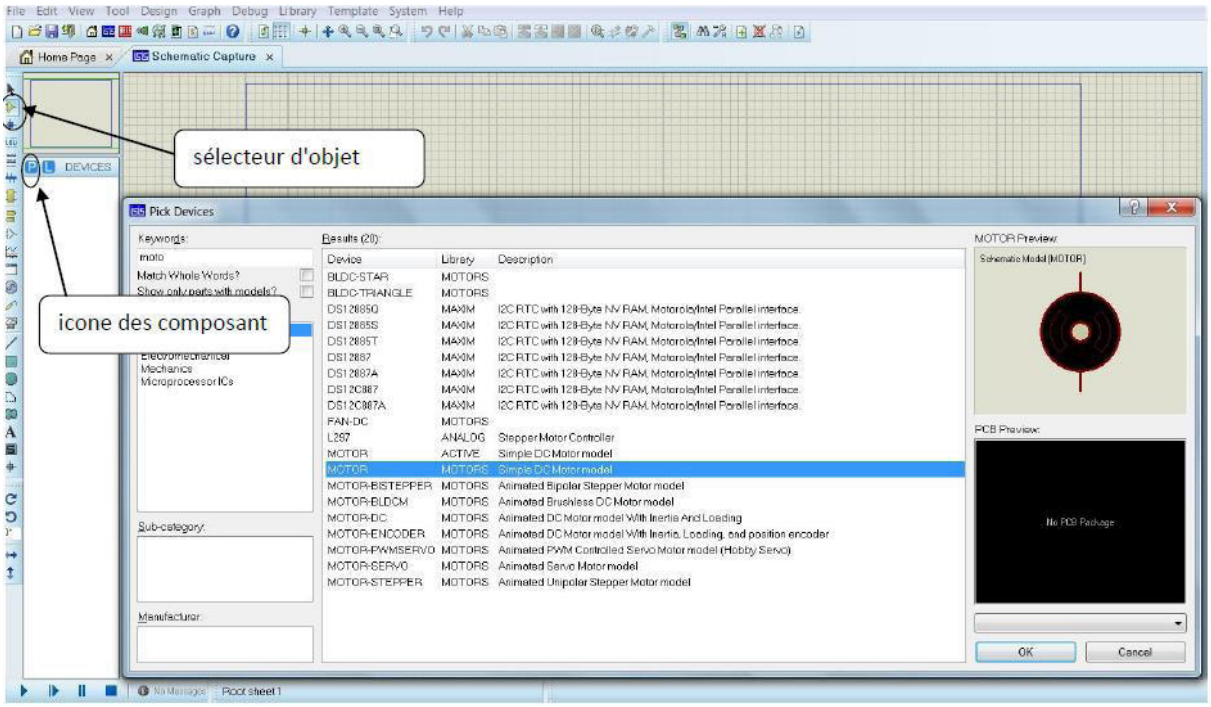


Figure IV.7 Choix des composants.

Ce logiciel a la possibilité d'emporter même des codes hexadécimaux pour les réalisations qui contiennent des composants programmables ou des cartes programmables « Arduino » comme dans notre réalisation.

IV.4 Le programme en Proteus de notre système

Dans cette partie nous avons connecté notre circuit électronique dans le programme de PROTEUS et nous avons utilise la bibliothèque d'Arduino qui n'est pas contenu dans la bibliothèque électronique de PROTEUS. Avec le programme VSPE, nous avons crée un port sérial virtuel pour faciliter la communication entre le serial moniteur Arduino et Proteus. Le logiciel VSPE a été crée pour aidé les ingénieurs à créer, tester et débogueur les applications qu'utilise les ports en série.

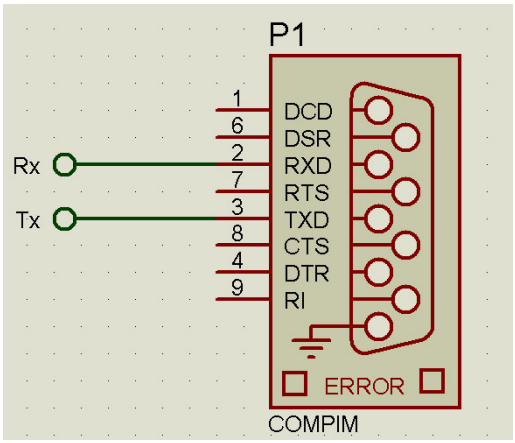


Figure IV.8 : Le port serial virtuel « COMPIM »

La figure montre le circuit complet électronique de notre système que nous avons simule avec Proteus.Le circuit content un appareil de "COMPIM" , cette appareil représente le port sérial de PROTEUS, avec la configuration entre lui, VSPE et le serial moniteur de l'IDE d'Arduino (figure III.8)

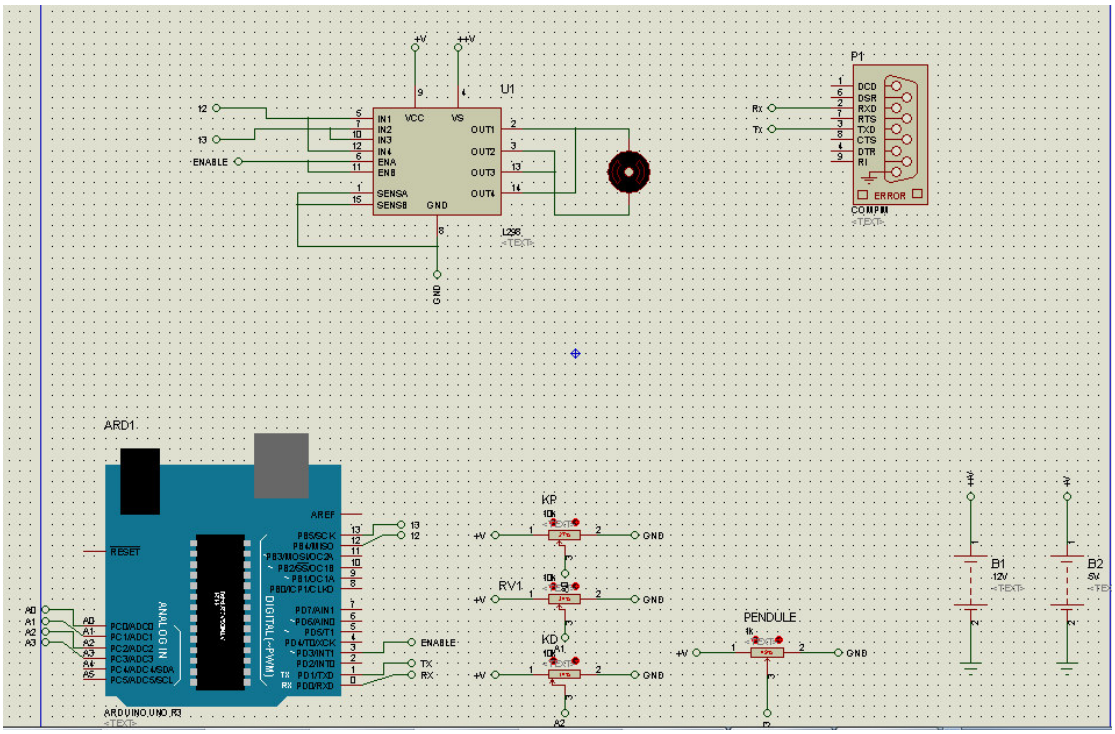


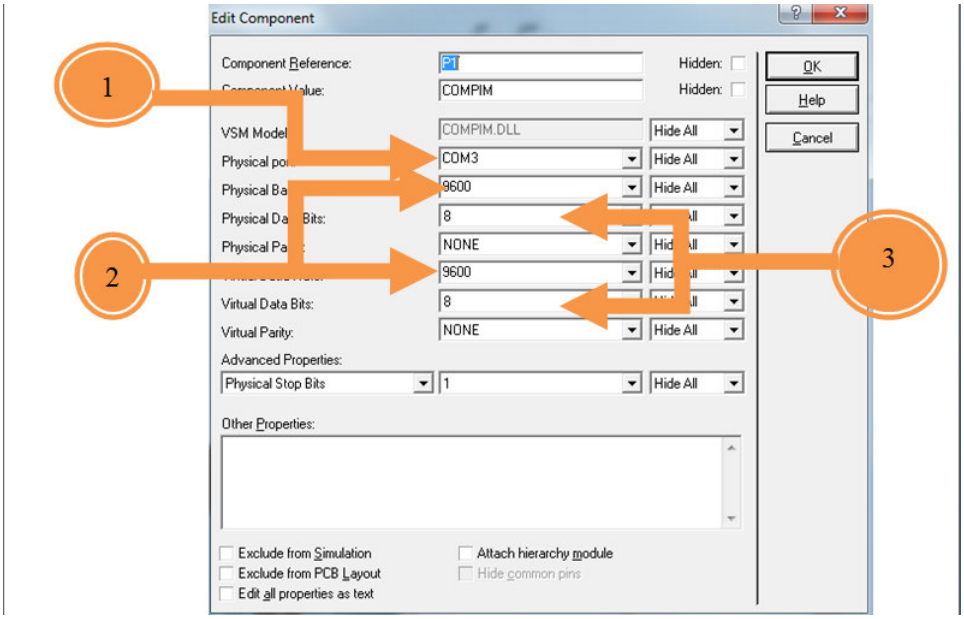
Figure IV.9 : Le circuit électronique de notre système.

Dans notre circuit de figure IV.9 on a un très important équipement appelé circuit intégré qui commande le démarrage, l'arrêt et la vitesse de DC moteur. Ce circuit intégrait est le **L298N** et il est capable de commande deux moteurs à courant continu de 2Ampère chacun.

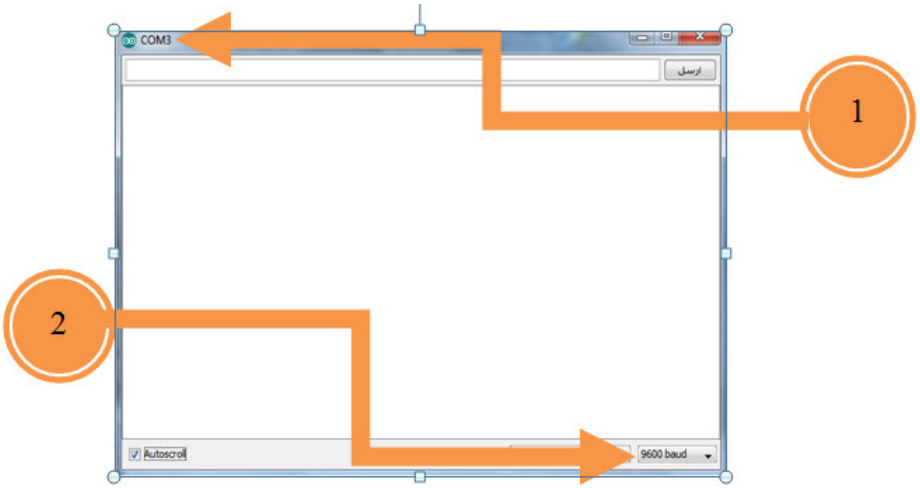
Pour voir les résultants de la simulation voici quelques étapes qu'on a suivies en Proteus, en Arduino et en VSPE:

- la première étape c'est de choisi le nombre de porte uni entre les trois programme figure IV.10 (Numéro 1),
- après ça on choisit la même vitesse de transfert des données (Numéro 2), ici on a choisi 9600 bauds,
- aussi on choisi les bits de donnée (Numéro 3).

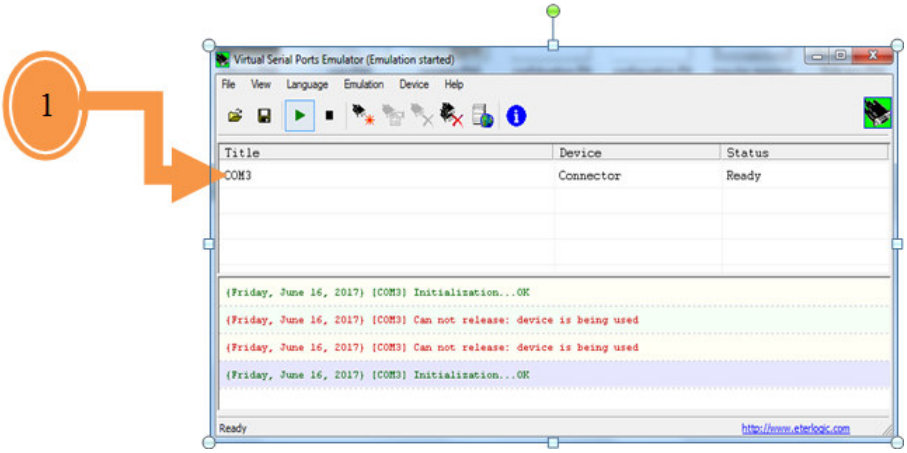
,



(a)



(b)



(c)

Figure IV.10 : (a) configuration de COMPIM, (b) configuration de serial moniteur, (c) configuration de VSPE.

IV.5 Les circuits conçus

Les circuits qui constituent notre système sont les suivants :

IV.5.1 Le pont-H et le moteur

Le L298N pont H composant peut être utilisé avec deux moteurs à courant continue. Le branchement des broches d'Arduino avec le L298N est résumé dans le tableau suivant (IV.2). Nous utilisons qu'un seul moteur donc nous avons besoins seulement des broche EN1, IN1 et IN2 du pont-H. Lorsque le EN1 (Enable pin 1) est branché vers le GND le moteur s'éteindre mais lorsqu'il est branché vers 5V d'Arduino il s'allume. Pour varier la vitesse du moteur il

faut branche ce broche vers une broche d'Arduino qui est capable de **MLI** (Modulation de Largeur d'Impulsion) ce qu'on appelle en anglais **PWM** (Pulse Width Modulation). Pour cette raison, dans notre cas nous avons utilisé la broche numérique 9 d'Arduino.

Arduino Uno broche	L298N
9	EN1
8	IN1
7	IN2
5	IN3
4	IN4
3	EN2

Tableau IV.2 Branchement de pont-H vers la carte Arduino

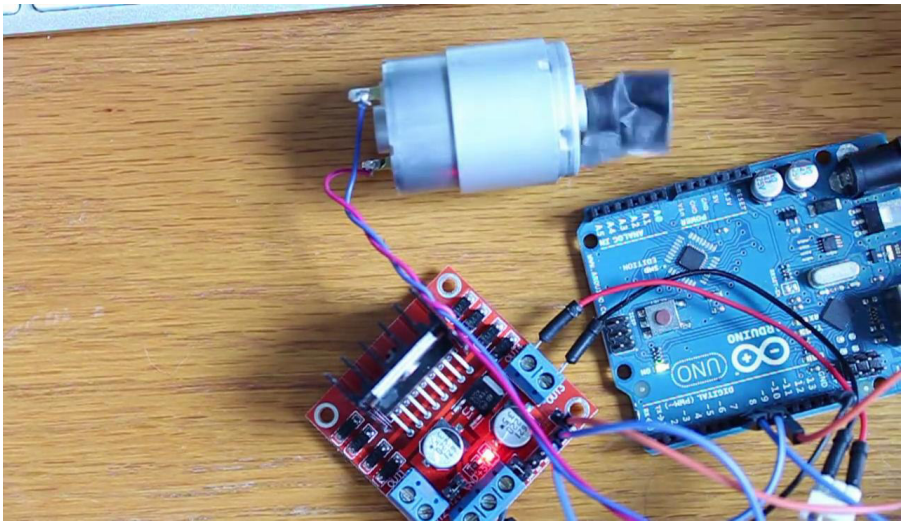


Figure IV.11 Branchement du moteur avec l'Arduino au travers le pont-H

En utilisant ce montage, le moteur est programme pour les mouvements vers la droite et puis la gauche avec les vitesses variées.

IV.5.2 Le capteur gyroscope/accéléromètre

Ce capteur est coller sur le pendule pour mesurer l'angle de rotation selon l'axe x parce que le pendule fait ses mouvement seulement selon cette axe. Le gyroscope/accéléromètre est branché avec l'Arduino comme le montre la figure IV.12

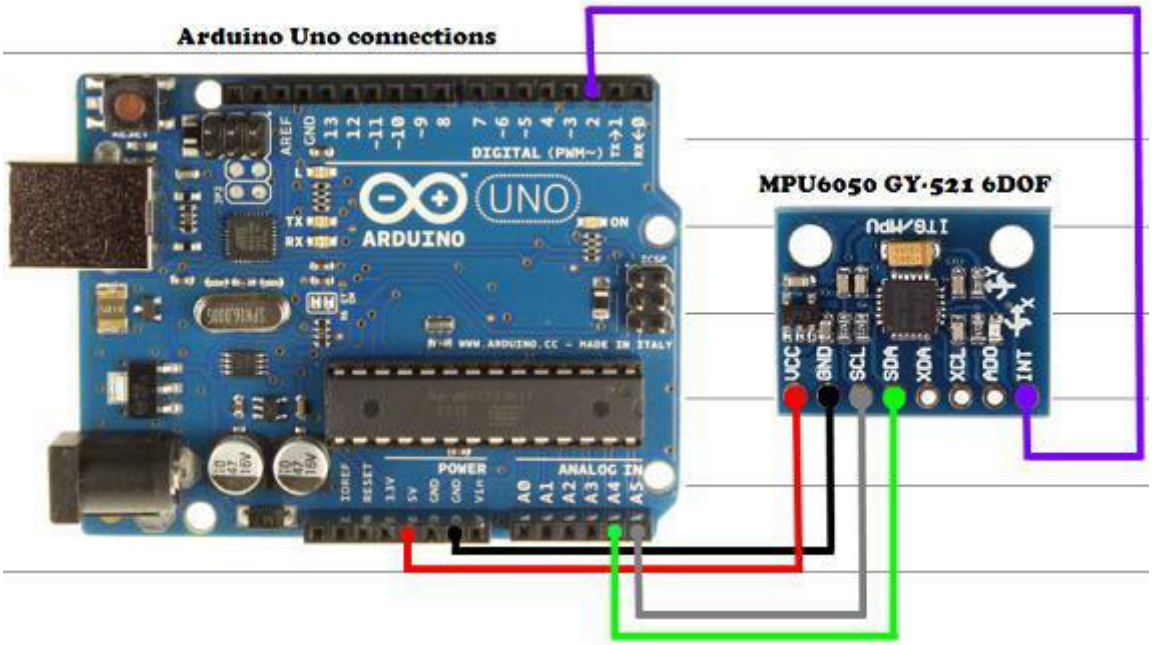


Figure IV.12 Branchement d'Arduino et Gyroscope/Accéléromètre

Les sorties de notre capteur sont six, trois sorties du gyroscope et trois d'accéléromètre. Selon notre programme nous avons intégré ces sorties et puis se focalisé sur l'angle de rotation selon l'axe des x.

IV.5.3 Les interrupteurs de la fin de course

Nous les avons nommé lim1 et lim2 et puis nous avons les branché avec les broches numériques de la carte Arduino. Leur fonctionnement est simple : des que le chariot touche un des deux, l'interrupteur se ferme et puis selon le programme la direction du moteur change. Le chariot aussi change le direction avant d'arriver a la fin des rails.

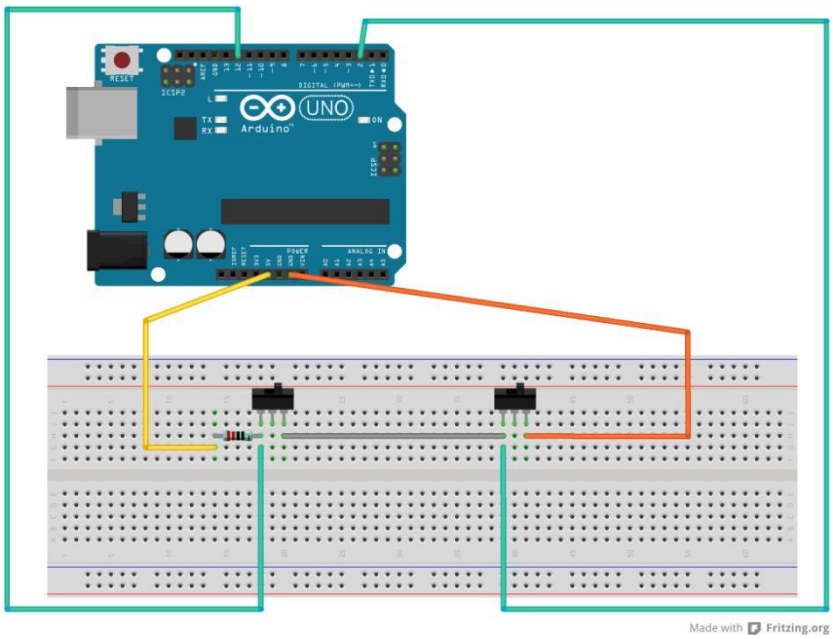


Figure IV.13 Branchement des fins de courses

IV.6 Conclusion

Dans ce chapitre nous avons utilisé le logiciel Arduino pour programmer les différentes circuits qui sont nécessaire pour le système du pendule inversé dans l'environnement virtuel Proteus.

Conclusion Générale

Ce travail d'étude et de recherche concernant les pendules inversés a été une excellente occasion pour nous d'augmenter nos connaissances dans le domaine de l'Automatique. Nous avons vu l'importance du ce système comme une plate-forme pour tester les méthodes de contrôle et aussi dans ses diverses applications dans le monde réel comme mentionné dans le premier chapitre.

Le chapitre deux nous a permis d'appliquer notre expérience de simulation Matlab à un système réel et de comprendre encore le contrôle PID en l'appliquant pour rendre stable le système du pendule inversé.

Du côté pratique de ce travail, nous avons beaucoup appris sur les composants électroniques et leur programmation. Nous avons surtout appris l'importance des microcontrôleurs pour la commande des systèmes et apprécié leur disponibilité ainsi que leur faible coût. Les cartes de circuit imprimée comme Arduino et qui sont faciles à utiliser rendent le travail de la construction de circuits plus facile et beaucoup plus agréable.

Ayant réussi à concevoir les circuits qui composent un pendule inversé, dans le proche avenir, nous prévoyons de les mettre ensemble pour réaliser l'ensemble du système.

Bibliographie

[1]J.Yi, N. Yubazaki, “ Stabilization fuzzy control of inverted pendulum” Technology Research Center, Japan, 2000

[2]Ferhat Lahouazi, “ Mise en œuvre d’une stratégie de commande neuro floue : Application à un pendule inversé” Mémoire de magistère, Université de Moloud Mammeri, Tizi-Ouzou ,2011.

[3] Berenji HR, “A reinforcement learning-based architecture for fuzzy logic control”, International Journal of Approximate Reasoning, 1992

[4] Yamakawa T, “Stabilization of inverted pendulum by a high-speed fuzzy logic controller hardware system Fuzzy set and systems,” 1989

[5] Lin S, Chen Y, “Design of adaptive fuzzy sliding mode for non-linear system control,” Proceedings of FUZZ-IEEE’94, vol. 1, 1994

[6] Kuphaldt T, “ Lessons in Industrial Instrumentation,” Version 2.11,2015

[7] Lu H,Tzeng S,Yeh M, “Fuzzy logic controller design using genetic algorithm,” ”, Proceedings of IIZUKA’96, vol. 2, 1996

[8] Margaliot M, Langholz G, “Adaptive fuzzy controller design via fuzzy lyapunov synthesis”, Proceedings of FUZZ-IEEE’98, 1998

[9] Brechet Thomas, Tribino Julien, “Le Pendule inversé et son application en robotique”, 2008

[10] Mikulcic A, Chen J, “Experiments on using fuzzy clustering for fuzzy control system design”, Proceedings of FUZZ-IEEE’96, vol. 3, 1996.

[11] Bourles.H, “ Systèmes linéaires de la modélisation à la commande”, Hermès Sciences Publication, Paris ,2006.

[12] Lamnabhi-Lagarrigue F, Rouchon P, “Commande non linéaire”,Hermès Sciences Publication, Paris, 2003.

[13] Kawaji S, Maeda T, “Fuzzy servo control system for an inverted pendulum”, Proceedings of IFES’91, vol. 2, 1991

[14] Kyung KH, Lee BH, “ Fuzzy rule base derivation using neural network-based fuzzy logic controller by self-learning”, Proceedings of IECON’93, vol. 1, 1993

[15]Matsuura K, “Fuzzy control of upswing and stabilization of inverted pole including control of cart position”, Journal of Japan Society for Fuzzy Theory and Systems ,1993.

[16] Yasunobu S, Mori M, “Swing up fuzzy controller for inverted pendulum based on a human control strategy”, Proceedings of FUZZIEEE’97, vol. 3, 1997

[17] Sakai S, Takahama T, “Learning fuzzy control rules for inverted pendulum by simplex method”, Proceedings of the13th Fuzzy System Symposium, 1997

[18]Khalil Sultan, “Inverted Pendulum analysis, Design and implementation.”,Institute of Industrial electronics engineering, Pakistan.

[19]Lundberg, Kent H, Taylor W. Barton, "History of Inverted-Pendulum Systems.", IFAC Proceedings Volumes 42.24 : 131-135,2010

[20] Khemissat A, “Plateforme de prototypage rapide pour la robotique mobile : Application à un robot Auto-balancé" mémoire de Master, UMBB, Boumerdes,2016.

[21]Hasan, Mahadi, "Balancing of an inverted pendulum using PD controller." Dhaka University Journal of Science 60.1 : pg115-120,2012

[22]Prasad, Lal Bahadur, Barjeev Tyagi, and Hari Om Gupta. "Optimal control of nonlinear inverted pendulum dynamical system with disturbance input using PID controller & LQR." Control System, Computing and Engineering (ICCSCE), 2011 IEEE International Conference on. IEEE, 2011.

[23] Gage, William H., et al. "Kinematic and kinetic validity of the inverted pendulum model in quiet standing." Gait & posture 19.2 (2004): 124-132.

[24] Yoshida, Kazunobu, "Swing-up control of an inverted pendulum by energy-based methods" American Control Conference, 1999, Vol. 6. IEEE, 1999.

[25] Magana, Mario E, Frank Holzapfel, "Fuzzy-logic control of an inverted pendulum with vision feedback", IEEE transactions on education 41.2 (1998): 165-170.

[26] Roberge K, "The Mechanical Seal", Massachusetts Institute of Technology, USA,1960

[27] Dr James.E.Parks ,“The simple pendulum”, PDF

[28] Massinissa T,Salia A, “Stabilisation d’un pendule inverse par des contrôleurs utilisant la structure à deux degrés de liberté ”,Mémoire du Master Academique, UMMTO,2016

[29] John Boxall, “Arduino Workshop A hands-on introduction with 65 project”, PDF

[30] Nafa N, Arabi B, “Conception et réalisation d’un système de sécurité commande à distance” mémoire de Master, UMBB,Boumerdes,2016.

[31]Rich Chi Ooi,“Balancing a Two-Wheeled Autonomous Robot” The University of Western Australia School of Mechanical Engineering, Final Year Thesis, 2003

[32]:http://www.generationrobots.com/fr/152-arduino