



ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET D'ANALYSE DES
SYSTÈMES - RABAT

Rapport de Stage :
Régulation et contrôle de vitesse d'un moteur DC
à base Arduino

Filière : ISEM

Réalisé par :

DKHEILA Anass

Encadré par :

Pr. Tamou NASSER

Jury :

Pr. Tamou NASSER
Pr. Asmaa HAMYANI

Année académique 2021/2022



Remerciements :

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon projet de stage clôturant la première année de formation d'ingénieur en Ingénierie des Systèmes Embarqués et Mobiles à l'ENSIAS.

Je tiens dans un premier temps à remercier mon encadrante du projet, Pr. TAMOU Nasser, pour sa transmission de savoir, sa disponibilité et ses conseils, qui ont contribué au bon déroulement du projet.

Je remercie également les membres de jury pour leur lecture attentive de ce rapport, ainsi que pour les remarques qu'ils m'adresseront lors de la soutenance afin d'améliorer mon travail.

Je remercie enfin tous les enseignants de notre École Nationale Supérieure d'Informatique et d'Analyse des Systèmes de nous avoir partagé leurs connaissances et expériences tout au long de cette année de formation.



Abstract

This project assignment aims to implement a multitude of technologies allowing us to control and regulate the speed of a DC motor.

To achieve our system, we have used for the hardware part : an Arduino Mega 2560 board, a motor Driver (L298N) and a DC motor with a built-in encoder.

For the software part :Matlab/Simulink (MATLAB Support Package for Arduino Hardware) to communicate with the Arduino board via a USB cable.

The application domain is so vast as DC motors are the most used motors in the industrial field, it is indispensable to have control over certain physical parameters (Speed,Position...).

Future improvements can be made to this work to make it more innovative by developping this application into a complete and complex project such as : Autonomous Car or Obstacle Robot.

Key Words : DC motor, control and regulation of speed, Arduino Mega 2560, Matlab/- Simulink, L298N, Encoder.

Résumé

Ce projet d'étude a pour but d'implémenter plusieurs technologies pour le contrôle et la régulation de la vitesse d'un MCC .

Pour réaliser notre système, nous avons recouru pour la partie matérielle à : une carte Arduino Mega 2560, un Motor Driver (L298N) et finalement un moteur DC avec encodeur intégré. Pour la partie logicielle : Matlab/Simulink (Support Package MATLAB pour Arduino matériel) pour communiquer de manière interactive avec la carte Arduino via un câble USB.

Le domaine applicatif est tellement vaste vue que les moteurs à courant continu sont les plus utilisée dans le milieu industriel, c'est indispensable d'avoir contrôle sur certains paramètres physiques (Vitesse,Position...). Des amélioration futures peuvent être apportés à ce travail afin de le rendre plus novateur en développant cette application en un projet plus Complexe et Complet tels qu'une voiture autonome or un robot éviteur d'obstacles.

Mots-Clés : MCC, Contrôle et la régulation de la vitesse, Arduino Mega 2560, Matlab/-Simulink, L298N, Encodeur.

Table des matières

Acronymes Glossaire :	1
1 Contexte général du projet :	3
1.1 Introduction :	3
1.2 Présentation de l'organisme d'accueil : ENSIAS	3
1.2.1 Présentation de l'ENSIAS :	3
1.2.2 Mission :	4
1.3 Présentation de la mission du stage :	4
1.3.1 Cahier de charge :	4
1.3.2 Problématique :	4
1.3.3 Objectif du projet :	5
1.3.4 Structure du rapport :	5
1.4 Conclusion :	5
2 Généralités et concepts sur les technologies :	6
2.1 Introduction :	6
2.2 L'état d'art d'un moteur à courant continu :	6
2.2.1 Définition d'une machine à courant continu :	7
2.2.2 Constitution d'un MCC :	7
2.2.2.1 Inducteur :	7
2.2.2.2 Induit :	8
2.2.2.3 Collecteur et les balais :	8
2.2.3 Types des MCC et principe de fonctionnement :	8
2.2.3.1 Types de MCC :	8
2.2.3.2 Principe de fonctionnement	9
2.3 Encodeur Rotatif :	9
2.3.1 Codeur rotatif incrémental :	10
2.3.2 Codeur rotatif absolu :	10
2.4 Pilotage du MCC :	10
2.4.1 Modulation de largeur d'impulsion (MLI) :	10
2.4.1.1 Définition :	10
2.4.1.2 Principe de la commande MLI :	11
2.4.1.3 Les avantages de la commande PWM :	11
2.4.2 Alimentation par convertisseur électronique :	11
2.4.2.1 Les Hacheurs :	11
2.5 Régulation et asservissement :	12
2.5.1 Précision et Stabilité des systèmes asservis :	13
2.5.1.1 Précision :	13
2.5.1.2 Stabilité :	13
2.5.2 Correcteur P :	14
2.5.3 Correcteur PI :	14
2.5.4 Correcteur PID :	14
2.6 Microcontrôleurs :	15
2.6.1 Définition :	15
2.6.2 Les caractéristiques principales d'un microcontrôleur :	15
2.6.3 Les avantages des microcontrôleurs :	16
2.7 Conclusion :	16

3 Conception et Modélisation du système :	17
3.1 Introduction :	17
3.2 Outils et Méthodes utilisées :	17
3.2.1 Matlab :	17
3.2.1.1 Simulink :	18
3.2.2 Modélisation du MCC :	19
3.2.2.1 Mise en équation en fonction du temps :	19
3.2.2.2 Schéma fonctionnel du moteur :	20
3.2.2.3 Simulation du moteur sous Matlab/Simulink :	21
3.2.3 Pilotage du MCC :	23
3.2.3.1 Commande MLI :	23
3.2.3.2 Pont en H	23
3.2.3.3 Principe de fonctionnement du pont en H :	23
3.2.3.4 Modélisation du pont en H avec MLI :	24
3.2.3.5 Simulation du pilotage du MCC :	25
3.2.4 Correcteur PID :	25
3.2.4.1 Simulation du régulation avec le correcteur PID :	26
3.2.4.2 Effets du correcteur PID :	26
3.2.5 Modélisation et simulation des circuits :	27
3.2.6 Système en boucle Ouverte :	27
3.2.6.1 Modélisation/Schéma Bloc :	27
3.2.6.2 Simulation et Résultats :	28
3.2.7 Système en boucle Fermée :	29
3.2.7.1 Modélisation/Schéma Bloc :	29
3.2.7.2 Simulation et Résultats :	30
3.3 Conclusion :	31
4 Implémentation du système de contrôle et régulation de vitesse :	32
4.1 Introduction :	32
4.2 Software :	32
4.2.1 Simulink support package for Arduino Hardware :	32
4.3 Hardware :	33
4.3.1 La carte Arduino Mega 2560 :	33
4.3.2 Le module L298N :	34
4.3.3 Motoréducteur (DC,Encodeur 740 ppr) :	35
4.3.3.1 Décodage des signaux de l'encodeur intégré :	36
4.3.4 Branchement du système :	37
4.4 Réalisation et Résultats :	38
4.4.1 Boucle Ouverte :	38
4.4.1.1 Schéma Bloc :	38
4.4.1.2 Résultats :	39
4.4.2 Boucle fermée :	39
4.4.2.1 Schéma Bloc :	39
4.4.2.2 Résultats :	40
4.5 Conclusion :	40
5 Conclusion et Perspectives :	41
5.1 Conclusion :	41
5.2 les prochaines étapes :	41

Table des figures

1.1 Espace ENSIAS	3
2.1 Moteur à courant continu :	6
2.2 Fonctionnement d'un MCC :	7
2.3 Inducteur ou Stator :	7
2.4 Induit ou Rotor :	8
2.5 Collecteur et Balais :	8
2.6 Illustration du principe de la Loi de Laplace :	9
2.7 Codeur Rotatif Moteur :	10
2.8 un signal MLI :	10
2.9 Schéma de commande de moteur en MLI :	11
2.10 Structure de commande :	12
2.11 Le pont en H :	12
2.12 Schéma bloc d'un système en boucle ouverte :	13
2.13 Schéma bloc d'un système en boucle fermée :	13
2.14 Correcteur P :	14
2.15 Correcteur PI :	14
2.16 Correcteur PID :	15
2.17 Différents types des microcontrôleurs :	15
3.1 Interface menu du logiciel Matlab :	17
3.2 Interface menu Simulink :	18
3.3 Menu des librairies Simulink :	18
3.4 Schéma électrique :	19
3.5 Schéma mécanique :	20
3.6 Schéma fonctionnel du MCC :	21
3.7 Allure de la vitesse Ω :	22
3.8 Allure du Courant :	22
3.9 Allure du Couple :	22
3.10 Visualisation du signal MLI :	23
3.11 Sens du courant dans les 2 sens de rotation :	24
3.12 Pont en H avec MLI :	24
3.13 Diagramme de tension par la commande plaine onde :	24
3.14 Pilotage du MCC par Pont en H avec MLI :	25
3.15 Allure de Vitesse :	25
3.16 Schéma bloc du correcteur PID en régulation de vitesse :	26
3.17 Allure de la vitesse Ω :	26
3.18 Schéma bloc du système en boucle ouverte :	27
3.19 Visualisation du signal MLI :	27
3.20 Allure de vitesse :	28
3.21 Allure de tension :	28
3.22 Allure de Courant :	29
3.23 Schéma bloc du système en boucle fermée :	29
3.24 Paramètres du correcteur :	30
3.25 Allure de vitesse :	30
3.26 Allure de tension :	30
3.27 Allure de Courant :	31
4.1 Simulink support package for Arduino Hardware :	32
4.2 Arduino Mega 2560 :	33

4.3	Caractéristiques d'une Carte Arduino Mega 2560	33
4.4	Le Module L298N :	34
4.5	Détails Techniques du module L298N :	35
4.6	MCC avec codeur intégré :	35
4.7	Visualisation des signaux aux bornes des phases A et B :	36
4.8	Bloc décodage des signaux du codeur :	37
4.9	Branchemet du système sous logiciel Fritzing :	37
4.10	Branchemet réel du système :	38
4.11	Schéma bloc boucle ouverte :	38
4.12	Allure de Vitesse :	39
4.13	Schéma bloc boucle fermée :	39
4.14	Résultats de la simulation boucle fermée :	40

Liste des abréviations

- **MCC** : Moteur à Courant Continue.
- **MLI** : Modulation de largeur d'impulsion.
- **PWM** : Pulse Width Modulation.
- **P** : Proportionnel.
- **PI** : Proportionnel Intégral.
- **PID** : Proportionnel Intégral Dérivé.
- **E/S** : Entrée/Sortie.
- **SRAM** : Static random-access memory.
- **EEPROM** : Electrically-erasable programmable read-only memory.
- **CMOS** : Complementary Metal-Oxide Semiconductor.
- **HCMOS** : High-speed CMOS.
- **MOS** : Metal–oxide–silicon transistor.
- **PPR** : Pulses Per Revolution.
- **RPM** : Revolutions per minute.

Introduction générale

La technologie moderne a permis le développement des sciences tout en imposant l'exploration de domaines théoriques de plus en plus complexes. Parmi ces sciences en pleine expansion et intégrant rapidement l'apport des technologies modernes, on compte l'automatique.

Les moteurs en générale représentent le pilier de l'industrie actuelle vu leurs puissances et leur rendement important tout dépend de la tâche à réaliser ; Y on a ceux qui utilisent les moteurs asynchrones en raison de leur fiabilité/productivité et son prix d'achat moins élevé. En outre, ces moteurs sont beaucoup plus robustes car ils ne nécessitent pratiquement pas d'entretien. Ainsi, d'autres introduisent les machines synchrones vu que la technologie de contact synchrone combine les avantages des moteurs à induction et des moteurs à aimants permanents.

Néanmoins, les moteurs à courant continu demeurent les plus utilisées dans le milieu industriel vu le simple découplage naturel dont il dispose, ce qui facilite sa tache de commande. Pour cela, ces MCCs sont très utilisés dans plusieurs domaines : automobile, les sous-marins et dans l'électroménager. En effet, leur vitesse de rotation intervient dans beaucoup d'application industrielle telle que la robotique, la manutention ainsi que l'entraînement des pompes

Dans la plupart des processus industriels, en particulier les moteurs électriques, il est indispensable de maîtriser certains paramètres physiques (vitesse, position, angle...etc.), il est donc très souvent nécessaire d'avoir recours à une commande.

Dans ce projet les objectifs visés sont :

- Le premier est de regrouper suffisamment d'informations sur les technologies utilisées.
- Le deuxième consiste à faire une réalisation pratique destinée à mettre en évidence ce système de régulation.

Chapitre 1

Contexte général du projet :

1.1 Introduction :

Dans ce chapitre, on présentera l'organisme d'accueil et le contexte du stage (Cahier de charge, Objectif, Problématique)

1.2 Présentation de l'organisme d'accueil : ENSIAS

1.2.1 Présentation de l'ENSIAS :

L'École Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS) est l'une des grandes écoles d'ingénieurs marocaines rattachée à l'université Mohammed V de Rabat. La qualité de ses ingénieurs, reconnue au Maroc et à l'étranger, a permis à l'ENSIAS de se positionner parmi les meilleures grandes écoles du Royaume durant les dix dernières années.

Fondée en 1992, l'École Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS) est la première école d'ingénierie spécialisée en informatique au Maroc. Dans sa mission pour former des ingénieurs d'état et la recherche en vue de développement technologique et économique du Maroc, elle met tout en oeuvre pour garantir à la fois des connaissances scientifiques et techniques et dote ses ingénieurs des aptitudes les plus qualifiantes dans leurs domaines en s'appuyant sur de nombreux projets afin d'aborder avec confiance le monde du travail et de prétendre rapidement à des postes de responsabilités dans les domaines les plus variés de la société de l'information.. L'originalité de son cursus alliant compétences scientifiques et techniques, et ouverture sur l'entreprise en font une des écoles les plus prisées par les étudiants et les recruteurs.



FIGURE 1.1 – Espace ENSIAS

1.2.2 Mission :

La mission principale de L'ENSIAS est la formation d'ingénieurs d'État en informatique en trois années. L'ENSIAS assure également le développement d'actions de coopération et de recherche au niveau national et international, la formation de formateurs en informatique, et l'organisation d'actions de formation continue. Régulièrement classée parmi les meilleures écoles d'ingénieurs marocaines, l'ENSIAS conforte depuis de nom-breuses années sa position de leader national pour les formations en technologies de l'information.

Neuf (9) filières sont offertes aux élèves ingénieurs depuis la deuxième année, la première étant un tronc commun entre toutes les filières. Elles émanent des 6 départements de l'ENSIAS. Les spécialisations sont :

- 1- Business Intelligence and Analytics (BI and A)
- 2- Génie de la Data (GD)
- 3- Ingénierie en Data Science and IOT (IDSIT)
- 4- Ingénierie Digitale pour la Finance (IF)
- 5- Ingénierie Intelligence Artificielle (2IA)
- 6- Sécurité des Systèmes d'Information (SSI)
- 7- Smart Supply Chain and Logistics (2SCL)
- 8- Génie Logiciel (GL)
- 9- Smart System Engineering (SSE)

L'ENSIAS délivre également cinq diplômes de master :

- 1- Master Recherche en Sciences des données et Big Data.
- 2- Master spécialisé en Internet des Objets : Logiciel et Analytique (MIOLA).
- 3- Master spécialisé en Biotechnologie médicale option Bioinformatique.
- 4- Master spécialisé en Systèmes embarqués pour l'automobile.
- 5- Master Recherche en Sécurité des Systèmes et Services (M3S).

1.3 Présentation de la mission du stage :

1.3.1 Cahier de charge :

- Développer un système de contrôle et de régulation de la vitesse d'un MCC à base d'Arduino.
- Faire une réalisation pratique destinée à mettre en évidence ce système.

1.3.2 Problématique :

Problématique tiré du cahier des charges :

- Recherche des différents technologies qui permettent la réalisation du projet.
- Modélisation du système de régulation
- Le choix de software convenable pour réaliser une implémentation pratique.
- Le choix de hardware convenable pour réaliser une implémentation pratique.

1.3.3 Objectif du projet :

L'objectif principal de ce projet et de développer un système de contrôle et la régulation de la vitesse d'un MCC. En effet le projet contient plusieurs parties qui nous ont aider a comprendre le fonctionnement et l'utilité de cette application. Ainsi se familiariser avec le hardware et le software utilisé afin de réaliser notre système.

1.3.4 Structure du rapport :

Ce rapport est organisé en trois chapitres décrivant l'ensemble du travail. L'objectif du premier chapitre est de présenter les généralités sur les technologies. Le deuxième déborde modélisation du système. Le Troisième chapitre explique le hardware et présente le software qu'on a utilisé pour réaliser le système et finalement une conclusion.

1.4 Conclusion :

Lors de ce chapitre, On a présenté le contexte de la mission de stage afin d'établir un plan de travail pour la suite du projet.

Chapitre 2

Généralités et concepts sur les technologies :

2.1 Introduction :

Lors de ce chapitre on présentera les différents technologies et les concepts qu'on peut utiliser pour réaliser le système de régulation de notre étude.

2.2 L'état d'art d'un moteur à courant continu :

L'évolution de l'industrie conduit à utiliser des machines qui nécessitent des vitesses de rotation variables et précises pour l'entraînement des systèmes.

Dans notre étude, on s'intéresse aux moteurs à courant continu ; car ils offrent une introduction faciles au fonctionnement de ses homologues, en donnant des repères clairs.

les moteurs et les générateurs furent les premières machines électrique utilisées au XIXème siècle pour produire de la puissance motrice. Le moteur électrique transforme de l'énergie électrique en mécanique par contre le générateur électrique converti l'énergie mécanique en électrique. Dans cette partie on va élaborer une présentation détaillé sur le moteur à courant continu l'objet de commande dans notre réalisation.



FIGURE 2.1 – Moteur à courant continue :

2.2.1 Définition d'une machine à courant continu :

La machine à courant continu (MCC) est un convertisseur électromécaniques d'énergie permettant la conversion bidirectionnel d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique. Cette MCC est une machine électrique tournante mettant en jeu des tensions et des courants continus. La machine à courant peut fonctionner en deux modes :

- Fonctionnement en mode moteur, l'énergie électrique est transformée en énergie mécanique.
- Fonctionnement en mode génératrice, l'énergie mécanique est transformée en énergie électrique

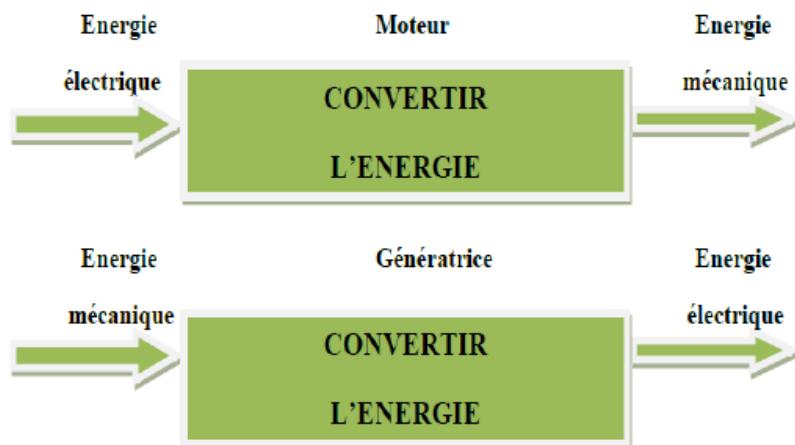


FIGURE 2.2 – Fonctionnement d'un MCC :

2.2.2 Constitution d'un MCC :

Le MCC est constitué principalement des éléments suivants :

2.2.2.1 Inducteur :

L'inducteur : Il appelé aussi stator, est la partie fixe du moteur. Il est constitué d'un aimant permanent ou d'un électroaimant alimenté par le courant. Il a pour fonction de créer un champ magnétique reçu par les bobines du rotor.



FIGURE 2.3 – Inducteur ou Stator :

2.2.2.2 Induit :

L'induit : Il appelé aussi rotor est la partie tournante du moteur. Il est constitué d'encoches dans lesquelles est enroulé un bobinage de conducteurs alimentés en courant continu.

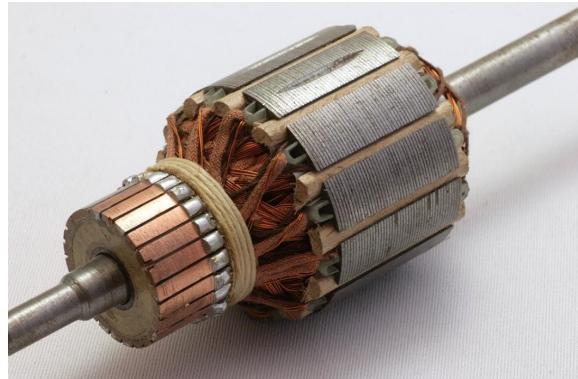


FIGURE 2.4 – Induit ou Rotor :

2.2.2.3 Collecteur et les balais :

Le collecteur : C'est un ensemble de lames de cuivre où sont reliées les extrémités du bobinage de l'induit

Les balais : Ils sont situés au stator frottent sur les lames du collecteur, et permettent d'établir un contact glissant entre la partie tournante qui tourne et la partie fixe de la machine.

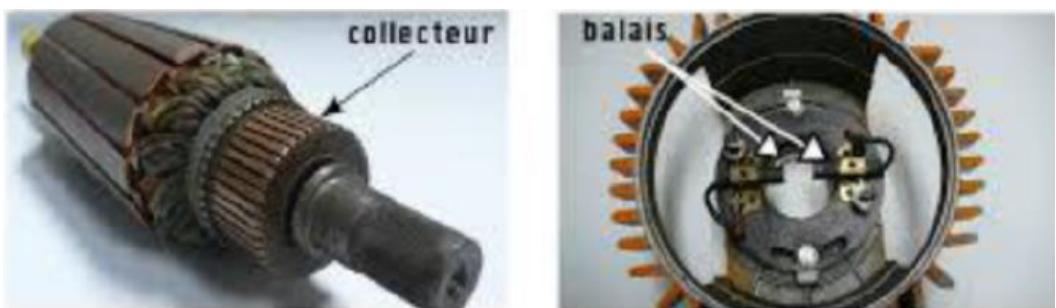


FIGURE 2.5 – Collecteur et Balais :

2.2.3 Types des MCC et principe de fonctionnement :

2.2.3.1 Types de MCC :

Il existe 5 types de MCC :

Excitation indépendante : Le bobinage inducteur des moteurs à excitation indépendante est raccordé à une alimentation à courant continu séparée. Grande souplesse de commande. Utilisé en milieu industriel, associé avec un variateur électronique de vitesse et surtout sous la forme moteur d'asservissement. Large gamme de vitesse.

Excitation par aimants permanents : Les moteurs à aimants permanents comportent des aimants permanents plutôt que des enroulements inducteurs pour produire le champ magnétique du stator.

Excitation série : Dans un moteur à excitation série ou moteur série, le circuit d'excitation est en série avec le circuit d'induit.

Excitation parallèle : Le circuit inducteur est branché en parallèle avec l'induit et, de ce fait, alimenté sous la même tension. Ce moteur a les mêmes caractéristiques que le moteur à excitation indépendante.

Excitation composée : La machine comporte une inductrice de dérivation placée en parallèle avec l'induit. Ce dernier étant en série avec l'inducteur série de la machine. On distingue deux types de branchements :

- — Courte dérivation : L'enroulement parallèle est aux bornes du moteur.

- Longue dérivation : L'enroulement parallèle est aux bornes de la source

2.2.3.2 Principe de fonctionnement

Lorsque le bobinage d'un inducteur du moteur est alimenté par un courant continu, sur le même principe qu'un moteur à aimant permanent, il crée un champ magnétique (flux d'excitation) de direction Nord-Sud. Une spire capable de tourner sur un axe de rotation est placée dans le champ magnétique. De plus, les deux conducteurs formant la spire sont chacun raccordés électriquement à un demi collecteur et alimentés en courant continu via deux balais frotteur.

L'induit (rotor) porte des conducteurs parcourus par un courant continu. Les spires sont le siège de forces (loi de Laplace) qui créent un couple ; l'induit se met à tourner.

La rotation du Rotor résulte en une variation du flux ce qui variation qui engendre une force électromotrice (f.e.m.) aux bornes de la spire.

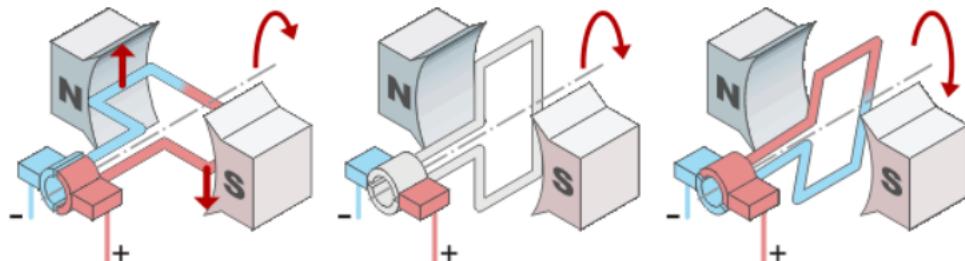


FIGURE 2.6 – Illustration du principe de la Loi de Laplace :

2.3 Encodeur Rotatif :

Un codeur rotatif ou capteur rotatif est un type de capteur permettant de fournir une information d'angle, en mesurant la rotation effectuée autour d'un axe.

L'information de vitesse peut alors être déduite de la variation de la position par rapport au temps. Plus le codeur rotatif tourne lentement, plus la déduction de vitesse perd en précision. Il existe 2 principaux types :



FIGURE 2.7 – Codeur Rotatif Moteur :

2.3.1 Codeur rotatif incrémental :

Il ajoute ou soustrait (selon le sens de rotation) une unité à un compteur à chaque rotation supérieure à la résolution du capteur. Le compteur est généralement remis à zéro lorsque l'appareil est allumé. C'est le cas de la souris d'ordinateur à boule.

2.3.2 Codeur rotatif absolu :

Il intègre son propre compteur. Ce genre de capteur est généralement calibré et initialisé une seule fois, et il conserve normalement sa valeur lors de l'arrêt de l'appareil. C'est le cas des compteurs kilométriques de voitures.

2.4 Pilotage du MCC :

2.4.1 Modulation de largeur d'impulsion (MLI) :

Pour faire varier la vitesse d'un moteur à courant continu, la variation de la tension aux bornes de l'induit est indispensable.

Avec le développement de ma commande MLI il est possible de faire varier la vitesse de rotation d'un MCC sans passer par la variation de tension à leur bornes.

2.4.1.1 Définition :

Une Modulation à Largeur d'Impulsion est un signal à période constante mais à rapport cyclique réglable. La MLI (ou PWM) est utilisée dans la famille des variateurs de fréquence : les cyclo-convertisseurs, les onduleurs, les redresseurs et les hacheurs.

Son application dans les variateurs de vitesse de moteur à courant continu. Le variateur de fréquence étant très utilisé dans l'industrie, l'étude des MLI devient très important.

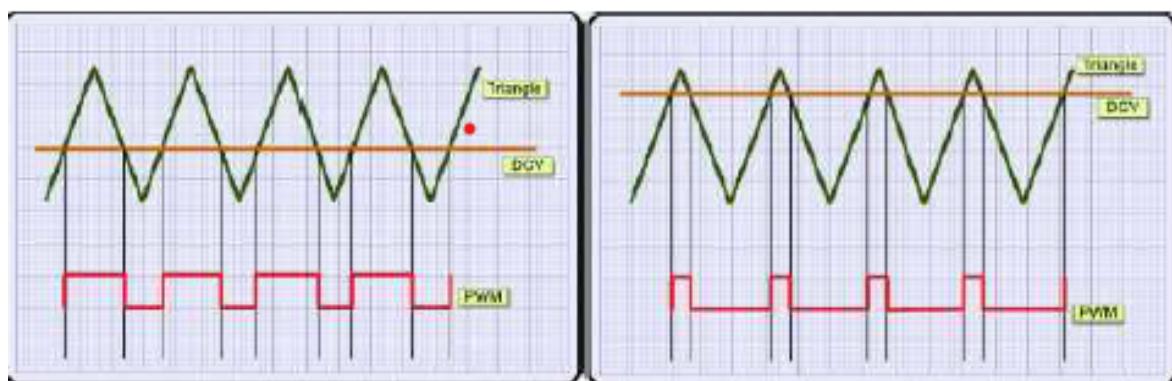


FIGURE 2.8 – un signal MLI :

2.4.1.2 Principe de la commande MLI :

Le Principe de base de la Modulation de la Largeur d'impulsion MLI est fondé sur le découpage d'une pleine onde rectangulaire. Ainsi, la tension de sortie est formée par une succession de crêteau d'amplitude égale à la tension continue d'alimentation et de largeur variable. La technique la plus répondue pour la production d'un signal MLI est de comparer entre deux signaux :

- Le premier, appelé signal de référence, est un signal continue.
- Le second, appelé signal de la porteuse, définit la cadence de la commutation des interrupteurs statiques du convertisseur. C'est un signal de haute fréquence par rapport au signal de référence (l'intersection de ces signaux donne les instants de commutation des interrupteurs)

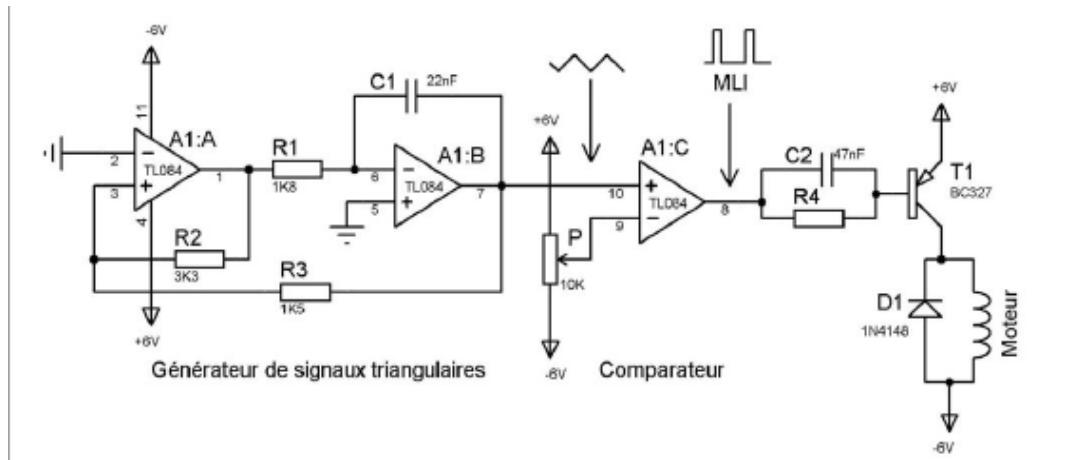


FIGURE 2.9 – Schéma de commande de moteur en MLI :

2.4.1.3 Les avantages de la commande PWM :

Il y a plusieurs avantages à piloter la vitesse d'un moteur en PWM :

- Le moteur peut être piloté par la sortie numérique d'un microcontrôleur.
- Amélioration considérable du rendement énergétique.
- Un variateur de vitesse très compacte.

2.4.2 Alimentation par convertisseur électrique :

2.4.2.1 Les Hacheurs :

Les hacheurs sont des convertisseurs statiques continu-continu permettant de fabriquer une source de tension continue variable à partir d'une source de tension continue constante.

Pour varier la tension, il suffit de varier la valeur moyenne de la tension de sortie en découpant la tension de source. Cela est possible en commandant l'ouverture et la fermeture des interrupteurs électriques.

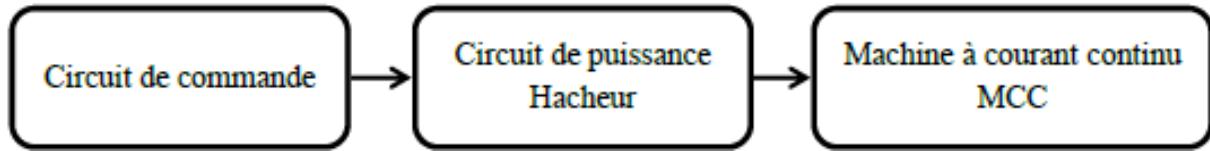


FIGURE 2.10 – Structure de commande :

Il existe plusieurs types hacheur qui assure la commande de vitesse d'MCC :

Hacheur Série : Le hacheur série permet de transformer une tension continue fixe en une tension d'alimentation plus faible réglable par l'utilisateur.

Hacheur Parallèle : Un convertisseur Boost, ou hacheur parallèle, est un convertisseur statique qui convertit une tension continue en une autre tension continue de plus forte valeur.

Pont en H : Le pont en H est une structure électronique servant à contrôler la polarité aux bornes d'un dipôle. Il est composé de quatre éléments de commutation généralement disposés schématiquement en une forme de H d'où le nom. Les commutateurs peuvent être des relais, des transistors, ou autres éléments de commutation en fonction de l'application visée.

Cette structure se retrouve dans plusieurs applications de l'électronique de puissance incluant le contrôle des moteurs, les convertisseurs et hacheurs, ainsi que les onduleurs.

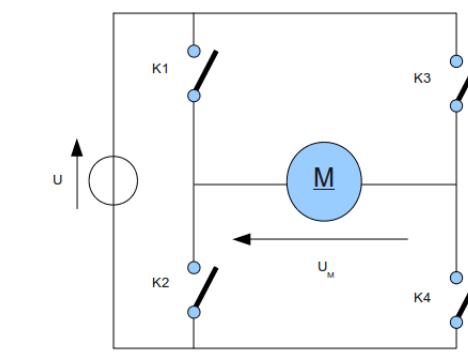


FIGURE 2.11 – Le pont en H :

2.5 Régulation et asservissement :

Dans la plupart des appareils dans des installations industrielles et domestiques, il est nécessaire de maintenir des grandeurs physiques à des valeurs déterminées, en dépit des variations externes ou internes influant sur ces grandeurs. Par exemple, le niveau d'eau dans un réservoir, la température d'une étuve, la vitesse et la position des moteurs, étant par nature variables, doivent donc être réglés par des actions convenables sur le processus considéré. Si les perturbations influant sur la grandeur à contrôler sont lentes ou négligeables, un simple réglage dit en boucle ouverte, permet d'obtenir et de maintenir la valeur demandée (par exemple : action sur un robinet d'eau).

Dans la majorité des cas, cependant, ce type de réglage n'est pas suffisant, parce que trop grossier ou instable. Il faut alors comparer, en permanence, la valeur mesurée de la grandeur réglée à celle que l'on souhaite obtenir et agir en conséquence sur la grandeur d'action, dite grandeur réglante. On a, dans ce cas, constitué une boucle de régulation et plus généralement une boucle d'asservissement. Cette boucle nécessite la mise en oeuvre d'un ensemble de moyens de mesure, de traitement de signal ou de calcul, d'amplification et de commande d'actionneur,

constituant une chaîne de régulation ou d'asservissement.

L'objectif d'une boucle de régulation de processus continu est de maintenir la mesure à une valeur – ou dans un intervalle de valeurs – désirée, malgré les perturbations, en agissant sur l'action (l'entrée du système physique).

Boucle Ouverte : Le maintien se fait manuellement.

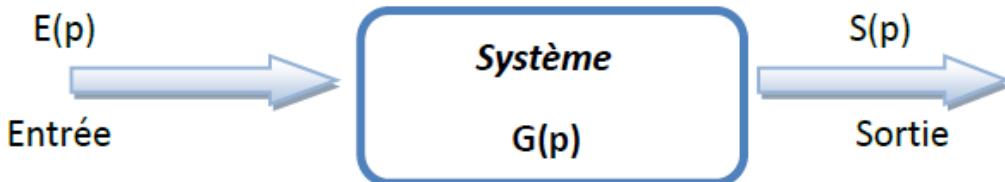


FIGURE 2.12 – Schéma bloc d'un système en boucle ouverte :

Boucle Fermée : Le moyen utilisé pour la commande automatique est la boucle de commande (boucle d'asservissement, boucle de régulation) : l'action à appliquer est calculée automatiquement par un système de traitement de l'information, le système de commande (correcteur, régulateur), en fonction d'informations sur la valeur désirée pour la mesure (consigne, référence) et sa valeur réelle (sortie, mesure)

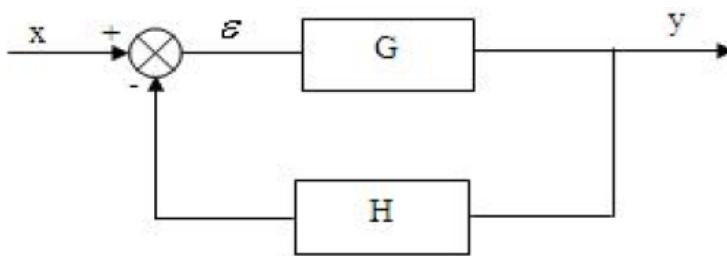


FIGURE 2.13 – Schéma bloc d'un système en boucle fermée :

2.5.1 Précision et Stabilité des systèmes asservis :

2.5.1.1 Précision :

Un système asservi en boucle fermée est dit précis, si sa sortie $s(t)$ est proche de la consigne $e(t)$. On peut représenter l'erreur entre la consigne et la sortie :

$$\epsilon(p) = E(p) - S(p)$$

2.5.1.2 Stabilité :

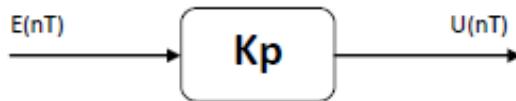
On définira la stabilité par une des propositions suivantes : Un système linéaire est stable :

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

il existe 3 types de correcteur, régulateur qu'on peut utiliser en système de commande.

2.5.2 Correcteur P :

Le correcteur proportionnel est le plus simple des correcteurs. Il ne permet pas toujours d'obtenir des performances très élevées mais il peut suffire dans certains cas si le cahier des charges n'est pas trop contraignant ou si le système a un comportement assez simple.



$$U(nT) = K_p E(nT) \Rightarrow U(z) = K_p E(z)$$

FIGURE 2.14 – Correcteur P :

2.5.3 Correcteur PI :

C'est la combinaison des correcteurs P et I, l'intérêt de la correction de type intégrale est de permettre une erreur statique nulle. Si l'entrée de l'intégrateur est reliée à l'erreur de régulation, l'effet du terme intégral sera bien d'annuler cette erreur. Notons que l'erreur est annulée même en présence d'une perturbation.

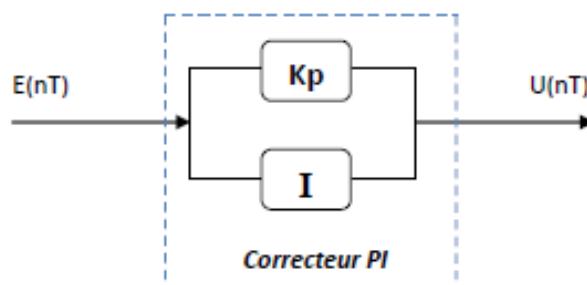


FIGURE 2.15 – Correcteur PI :

2.5.4 Correcteur PID :

C'est la combinaison des correcteurs P et I et d'un correcteur D, le correcteur PID est un système de contrôle permettant d'améliorer les performances d'un asservissement, c'est-à-dire un système ou procédé en boucle fermée. C'est le régulateur le plus utilisé dans l'industrie où ses qualités de correction s'appliquent à de multiples grandeurs physiques.

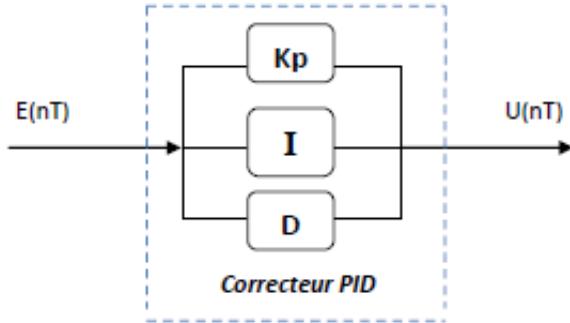


FIGURE 2.16 – Correcteur PID :

2.6 Microcontrôleurs :

2.6.1 Définition :

Un microcontrôleur est un circuit intégré et compact. Il comprend, conçu pour régir une opération spécifique et dans un système intégré. Il comprend :

- Un processeur.
- Des mémoire (morte et vive).
- Des périphériques E/S.
- Des interfaces E/S.

Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels. Depuis l'avènement de l'informatique embarquée, on le retrouve partout (automatisation, industrie, les appareils de bureau. . .)

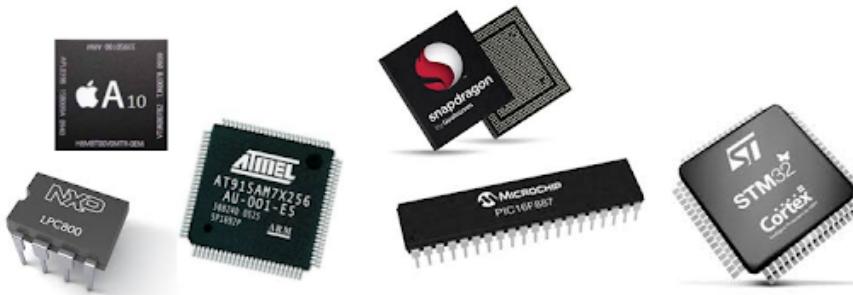


FIGURE 2.17 – Différents types des microcontrôleurs :

2.6.2 Les caractéristiques principales d'un microcontrôleur :

Les microcontrôleurs sont des composants qui permet la gestion des cartes, ils sont caractérisés par :

- De nombreux périphériques d'E/S.
- Une mémoire de programme.
- Une plus faible consommation électrique.

- Une mémoire vive (en général de type SRAM).
- Un coût réduit.
- Un processeur 8 bits ou 16 bits.
- une mémoire EEPROM destinée à la sauvegarde par programme de données à la coupure de l'alimentation.

2.6.3 Les avantages des microcontrôleurs :

Les points forts des microcontrôleurs sont nombreux et bien réels. Il suit, pour s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants de circuits intégrés en ce domaine depuis quelque année. Tout d'abord, un microcontrôleur intégré dans un seul et même boîtier ce qui avant nécessitait une dizaine d'éléments séparés. Cette intégration a aussi comme conséquence immédiate de simplifier la trace du circuit imprimé, puisqu'il n'est plus nécessaire de véhiculer des bus d'adresse et de données d'un composant à autre. Aussi le microcontrôleur permet :

- Augmentation de la fiabilité du système.
- Diminution de l'encombrement du matériel et du circuit imprimé.
- Simplification du tracé du circuit imprimé (plus besoin de tracer de bus!).
- Intégration en technologie MOS, CMOS, ou HCMOS (Diminution de la consommation).
- Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux : Moins cher que les composants qu'il remplace, diminution des coûts de main d'œuvre (conception et montage).
- Environnement de programmation et de simulation évolués.

2.7 Conclusion :

Lors de ce chapitre on a présenté les différentes technologies et concepts qu'on peut utiliser dans notre système afin de passer à l'étape de modélisation.

Chapitre 3

Conception et Modélisation du système :

3.1 Introduction :

Dans ce chapitre, On va élaborer la modélisation du système de régulation et les différentes technologies utilisées (MLI,Pont en H,Correcteur PID...)

3.2 Outils et Méthodes utilisées :

Pour modéliser et schématiser notre système on utilisera les Outils et les Méthodes convenables suivantes : Notre étude (Simulation, Réalisation) est faite principalement sous l'environnement MATLAB Simulink :

3.2.1 Matlab :

MATLAB est à la fois un langage de programmation et un environnement de développement fondé et commercialisé par la société américaine The Math Works. Il est utilisé dans les domaines de l'éducation, de la recherche et de l'industrie pour le calcul numérique mais aussi dans les phases de développement de projets. Ainsi, MATLAB est un environnement puissant, complet et facile à utiliser destiné aux calculs scientifiques. Il apporte aux ingénieurs, chercheurs et à tout scientifique un système interactif intégrant calcul numérique et visualisation. C'est un environnement performant, ouvert et programmable qui permet de remarquables gains de productivité et de créativité.

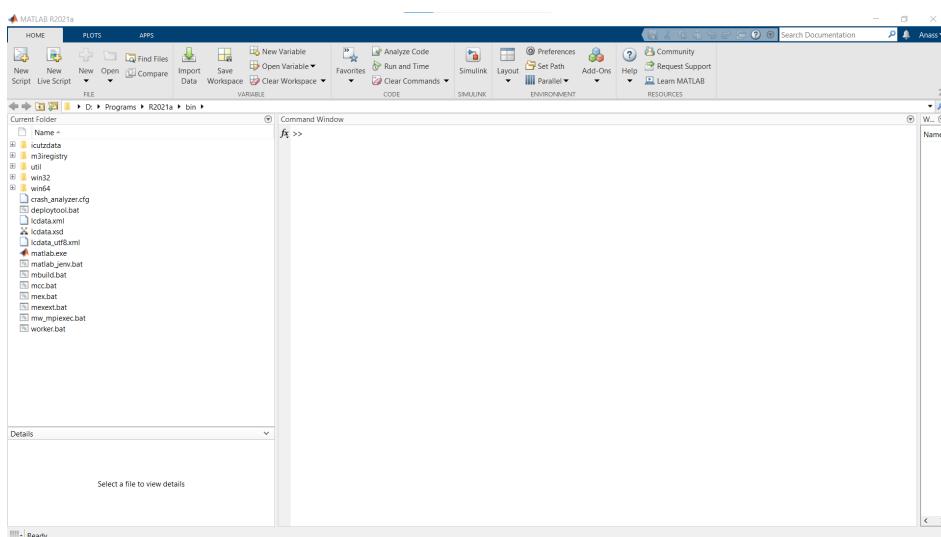


FIGURE 3.1 – Interface menu du logiciel Matlab :

MATLAB comprend aussi un ensemble d'outils spécifiques à des domaines, appelés Toolboxes (ou Boîtes à Outils). Indispensables à la plupart des utilisateurs, les Boîtes à Outils sont des collections de fonctions qui étendent l'environnement MATLAB pour résoudre des catégories spécifiques de problèmes.

3.2.1.1 Simulink :

Simulink, un produit complémentaire à MATLAB, est un logiciel de modélisation système multi-physique qui fournit un environnement graphique interactif pour la modélisation, la simulation et l'analyse de systèmes dynamiques. Il permet la construction rapide de prototypes virtuels pour explorer les concepts de conception à n'importe quel niveau de détail avec un minimum d'effort.

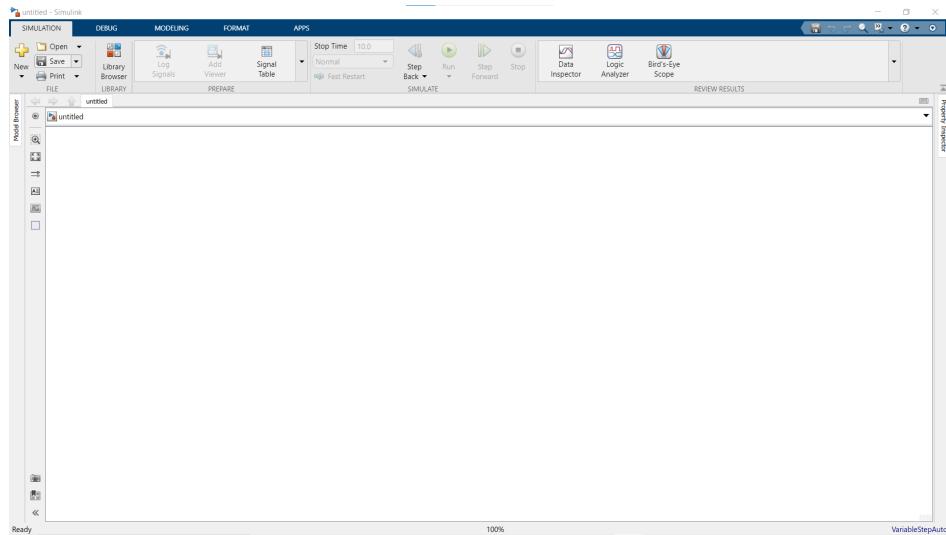


FIGURE 3.2 – Interface menu Simulink :

Pour la modélisation, Simulink fournit une interface utilisateur graphique (GUI) pour la construction de modèles sous forme de diagrammes fonctionnels. Il comprend une bibliothèque complète de blocs prédéfinis à utiliser pour construire des modèles graphiques de systèmes à l'aide d'opérations de souris par glisser-déposer. Ci dessous on trouve le menu des librairies contenant les blocs prédéfinis :

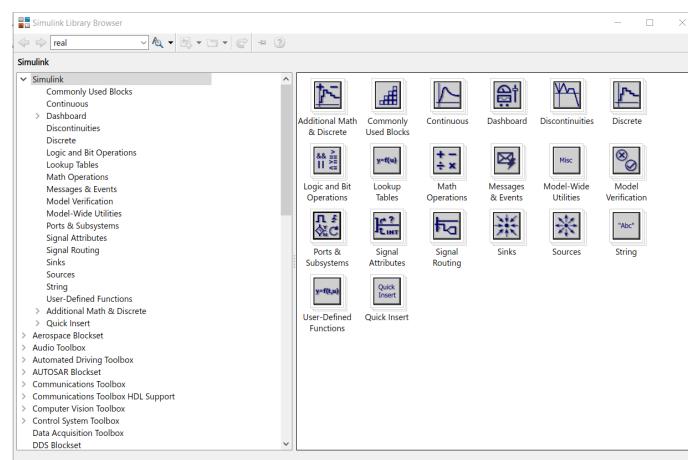


FIGURE 3.3 – Menu des librairies Simulink :

Pour utiliser simulink il suffit de cliquer sur l'icône de Simulink présente sur l'interface menu Matlab ou Exécuter la commande "**Simulink**" sur le Matlab prompt.

3.2.2 Modélisation du MCC :

Dans cette partie on entamera une étude du modèle mathématique et une modélisation du MCC à aimant permanent.

3.2.2.1 Mise en équation en fonction du temps :

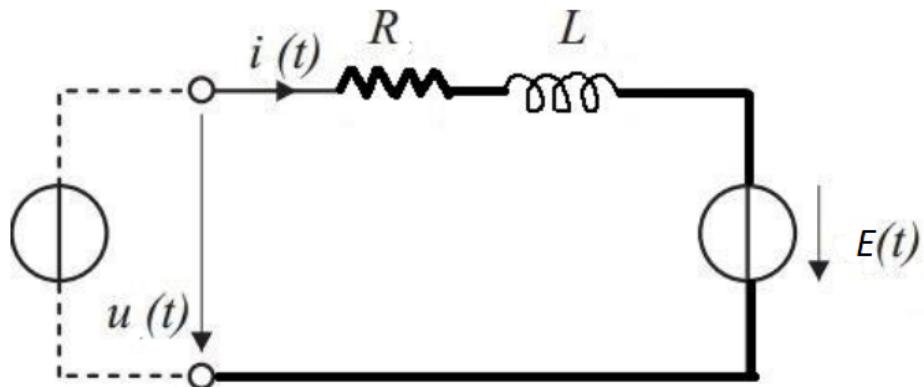


FIGURE 3.4 – Schéma électrique :

Équation électrique :

L'équation électrique du MCC est exprimée par l'équation ci-dessous :

$$U(t) = RI(t) + L \frac{dI(t)}{dt} + E(t)$$

$$\text{avec } E(t) = K_e \omega(t)$$

On a :

$$K_e \omega = K_e \omega(t) = cte$$

donc :

$$E(t) = K_e \omega(t)$$

Finalement, on a :

$$U(t) = RI(t) + L \frac{dI(t)}{dt} + K_e \omega(t)$$

Après la transformée de Laplace :

$$U(p) = RI(p) + LpI(p) + K_e \omega(p)$$

On aura donc :

$$I(p) / (U(p) - K_e \omega(p)) = 1 / (R + Lp) \quad (1)$$

- $U(t)$: Tension appliquée au moteur
- $I(t)$: Intensité traversant le moteur.
- ω : Vitesse de rotation du rotor.
- $E(t)$: Force contre électromotrice.
- K_e : Constante électrique.

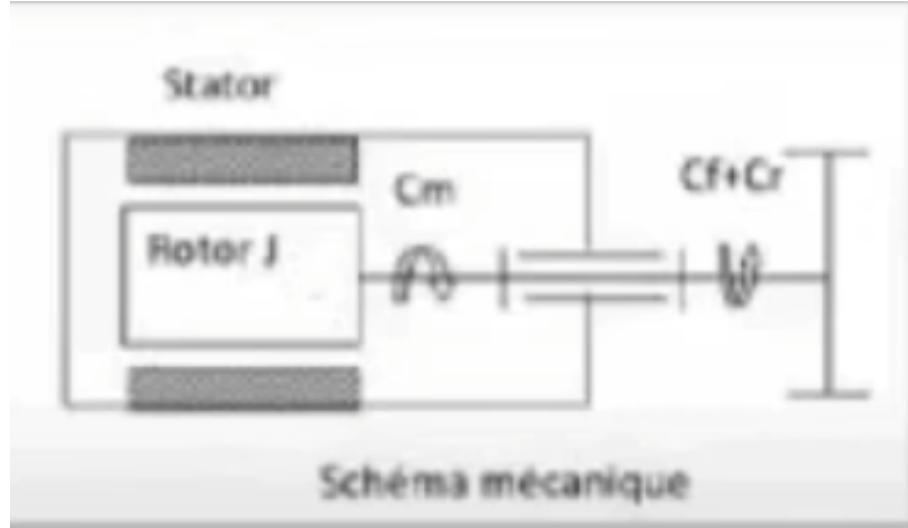


FIGURE 3.5 – Schéma mécanique :

Équation mécanique :

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

$$J(d\omega(t)/dt) = Ce + Cr + f\omega(t)$$

On a :

$$Ce(t) = K_m I(t)$$

Finalement, on aura :

$$J(d\omega(t)/dt) = K_m I(t) + Cr + f\omega(t)$$

Après la transformée de Laplace :

$$Jp\omega(p) = K_m I(p) - Cr - f\omega(p)$$

Finalement, on a :

$$\omega(p)/(K_m I(p) - Cr) = 1/(f + Jp) \quad (2)$$

- **J** : Moment d'inertie de l'axe du rotor
- **f** : Coefficient de frottement visqueux.
- **Cr** : Couple résistant.
- **E(t)** : Force contre électromotrice.
- **Ce** : Couple moteur généré.

3.2.2.2 Schéma fonctionnel du moteur :

D'après les deux équations (1) et (2) on peut établir le schéma fonctionnel du MCC.

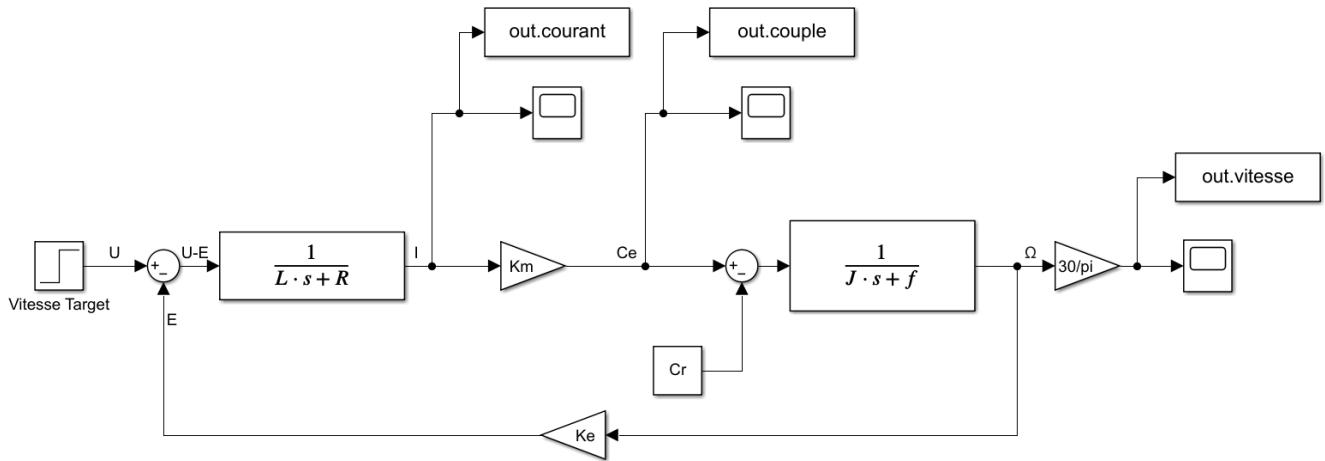


FIGURE 3.6 – Schéma fonctionnel du MCC :

Notre étude est faite à vide donc $C_r(p) = 0$, on peut obtenir la fonction de transfert suivante :

$$F(p) = \Omega(p)/U(p)$$

$$\text{avec } \Omega(p) = 1/(Ls + R) \cdot K_m \cdot (1/(Js + f))$$

$$\text{et } U(p) = 1 + (1/(Ls + R))s \cdot K_m \cdot (1/(Js + f)) \cdot K_e$$

On aura finalement :

$$F(p) = K / (1 + T_{em}p)(1 + T_{el}p)$$

Avec :

$$K = K_m / (fR + K_m K_e)$$

$$T_{em} = RJ / (fR + K_m K_e)$$

$$T_{el} = L/R$$

3.2.2.3 Simulation du moteur sous Matlab/Simulink :

Les valeurs des paramètres utilisées pour simuler notre système :

L	0.011 H	R	0.8 Ω
J	0.2 Kg.m ²	f	0.01 N.m.s
K _e	1.91 SI	K _m	1.91 SI

Diagrammes :

La figure ci dessous représente l'allure du vitesse Ω :

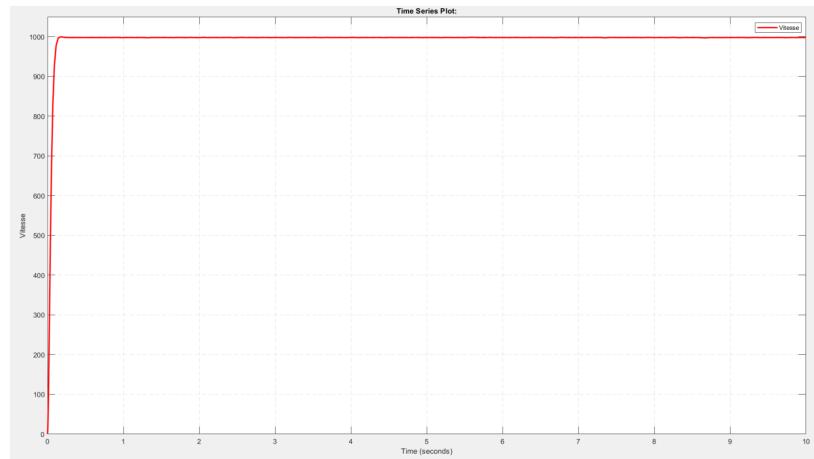


FIGURE 3.7 – Allure de la vitesse Ω :

La figure ci dessous représente l'allure du courant :

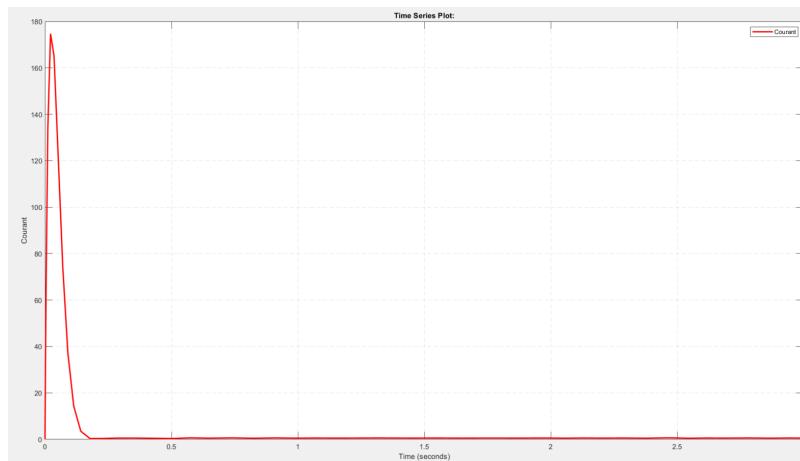


FIGURE 3.8 – Allure du Courant :

La figure ci dessous représente l'allure du couple :

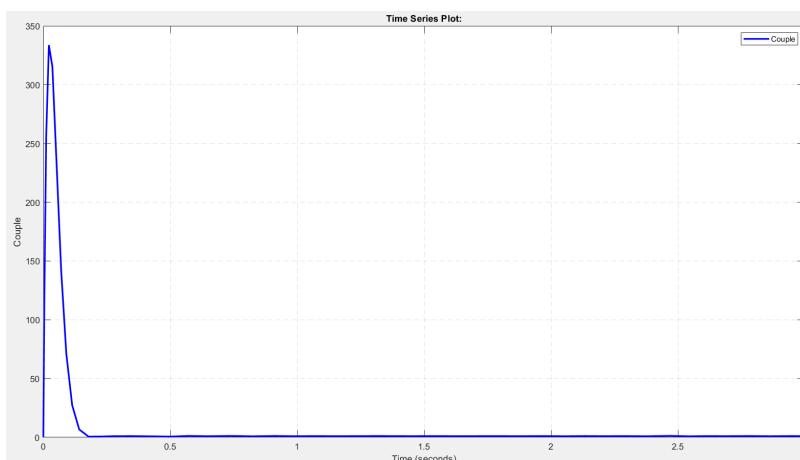


FIGURE 3.9 – Allure du Couple :

3.2.3 Pilotage du MCC :

3.2.3.1 Commande MLI :

Comme mentionné dans le Chapitre précédent on utilisera la commande MLI (PWM) en pilotage pour varier la vitesse de notre moteur.

Les figure ci dessous illustrent la visualisation du signal MLI sous le logiciel waveforms avec l'oscilloscope "Analog Discovery 2" :

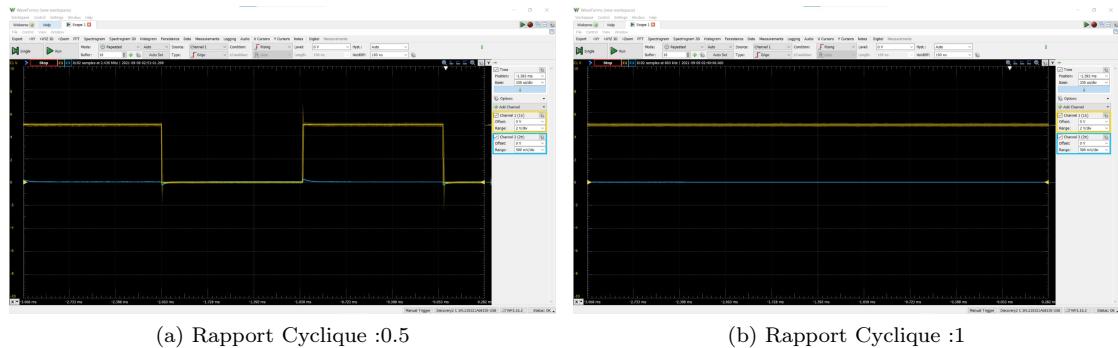


FIGURE 3.10 – Visualisation du signal MLI :

3.2.3.2 Pont en H

Le pont en H est une structure électronique servant à contrôler la polarité aux bornes d'un dipôle. IL est composé de quatre éléments de commutation généralement disposés schématiquement en une forme de H d'où le nom.

Le pont en H permet de faire passer un courant dans une charge dans les 2 sens. Il va nous permettre de :

- Faire tourner le MCC dans les 2 sens et de l'arrêter.

3.2.3.3 Principe de fonctionnement du pont en H :

Les interrupteurs fonctionnent deux par deux. Le A est associé au D et le B est associé au C. Dans le schéma ci-dessous, rien ne se passe car tous les interrupteurs sont ouverts donc le moteur est arrêté.

Sur le schéma de gauche, les interrupteurs A et D sont fermés, donc le courant entre par la patte gauche du moteur et sort par sa droite et par conséquent le moteur tourne. Sur le schéma de droite, les interrupteurs B et C sont fermés, donc le courant entre par la patte droite du moteur et sort par sa gauche ce qui fait tourner le moteur dans le sens inverse.

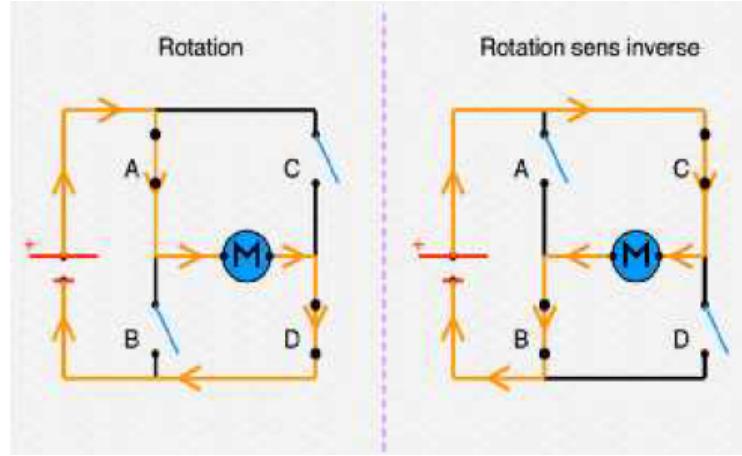


FIGURE 3.11 – Sens du courant dans les 2 sens de rotation :

3.2.3.4 Modélisation du pont en H avec MLI :

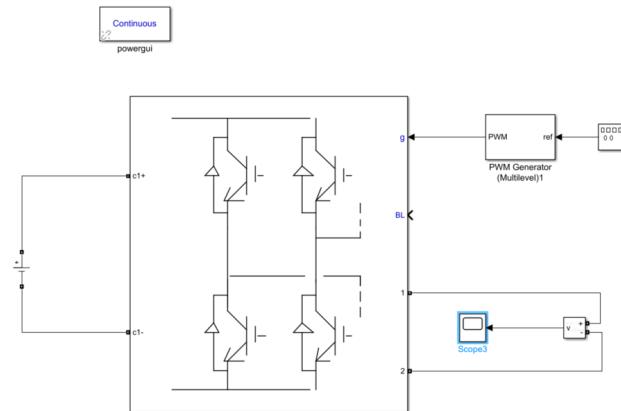


FIGURE 3.12 – Pont en H avec MLI :

Le résultat de la simulation est représenté sur la figure ci dessous :

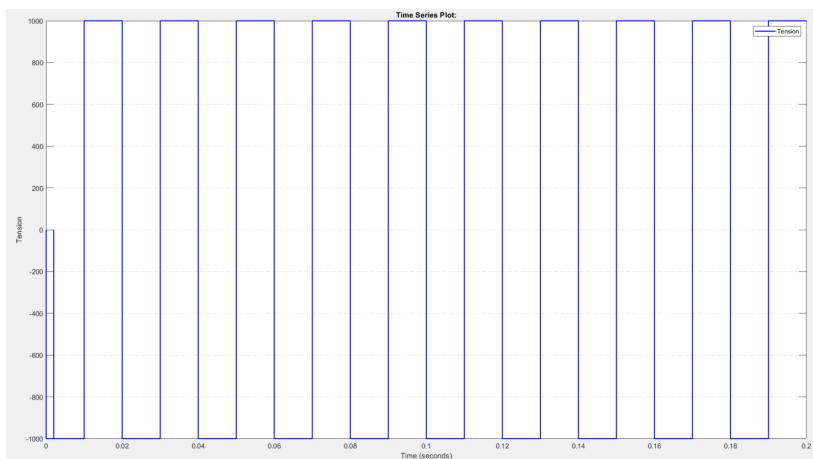


FIGURE 3.13 – Diagramme de tension par la commande plaine onde :

3.2.3.5 Simulation du pilotage du MCC :

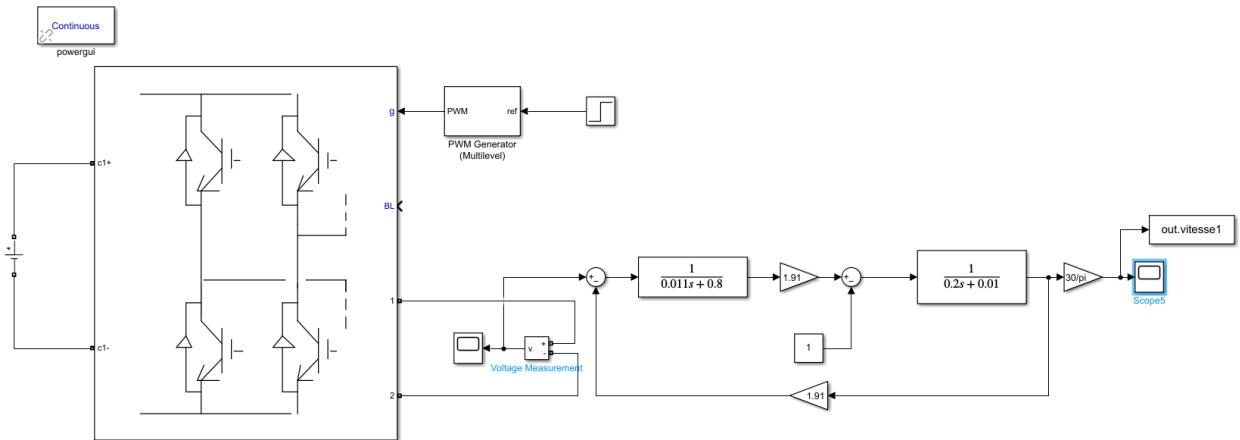


FIGURE 3.14 – Pilotage du MCC par Pont en H avec MLI :

Le résultat de la simulation est représenté sur la figure ci dessous :

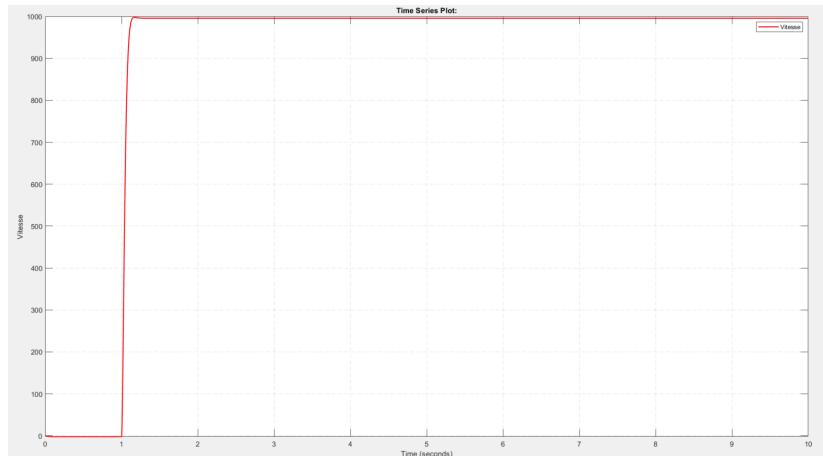


FIGURE 3.15 – Allure de Vitesse :

3.2.4 Correcteur PID :

La commande proportionnel-intégral-dérivé (PID) est insérée dans la chaîne directe de l’asservissement, en série avec le processus. Ce régulateur élabore à partir du signal d’erreur (t) une commande $U_r(t)$ en fonction de trois actions proportionnelle, intégrale, dérivée : On a :

$$U_r(t) = K_p \epsilon(t) + (1/T_i) \int_0^t \epsilon(t) dt + T_d \cdot (d\epsilon(t)/dt)$$

donc :

$$U_r(t) = K_p \epsilon(t) + K_i \int_0^t \epsilon(t) dt + K_d \cdot (d\epsilon(t)/dt)$$

Après la transformée de Laplace :

$$U_r(t) = K_p \epsilon(p) + K_i \cdot (d\epsilon(p)/dp) + K_d \cdot d\epsilon(p)p$$

- K_p : gain d'action proportionnelle
- $K_i = 1/T_i$: gain d'action intégrale.
- $K_d = T_d$: gain d'action dérivée.
- T_i : constante de temps, dite temps d'action intégrale.
- T_d : constante de temps, dite temps d'action dérivée.

Le régulateur PID est donc conçu dans le domaine temporel comme la somme des trois actions. On obtient alors un asservissement composée d'un terme proportionnel, un terme intégral et un terme dérivé, mises en parallèle, on parle d'asservissement PID :

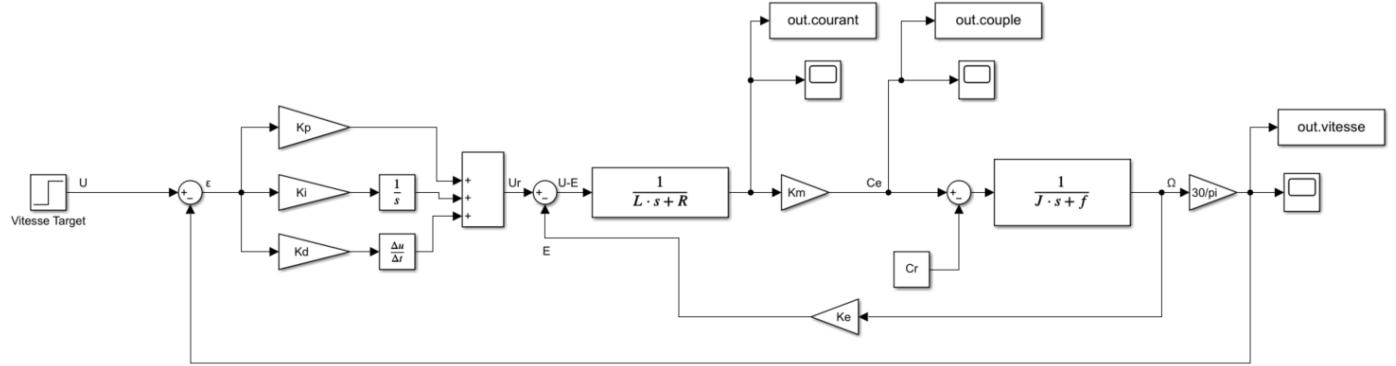


FIGURE 3.16 – Schéma bloc du correcteur PID en régulation de vitesse :

3.2.4.1 Simulation du régulation avec le correcteur PID :

On impose une vitesse consigne de 500 tr/min : Le résultat de la simulation est représenté sur la figure ci dessous :

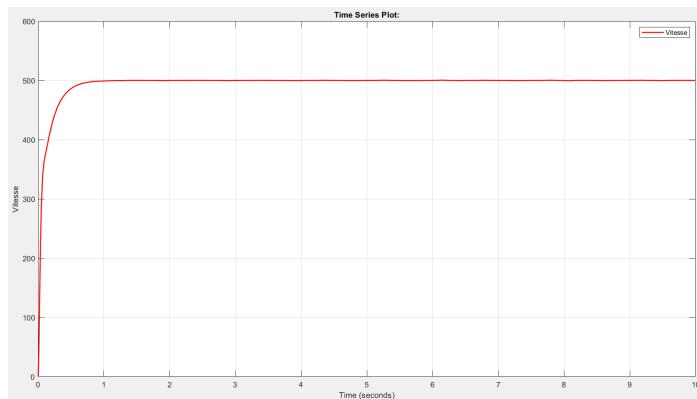


FIGURE 3.17 – Allure de la vitesse Ω :

3.2.4.2 Effets du correcteur PID :

- Diminution du temps de montée.
- Élimination de l'erreur statique.
- Diminution du temps de stabilisation.
- Diminution du dépassement.

3.2.5 Modélisation et simulation des circuits :

3.2.6 Système en boucle Ouverte :

3.2.6.1 Modélisation/Schéma Bloc :

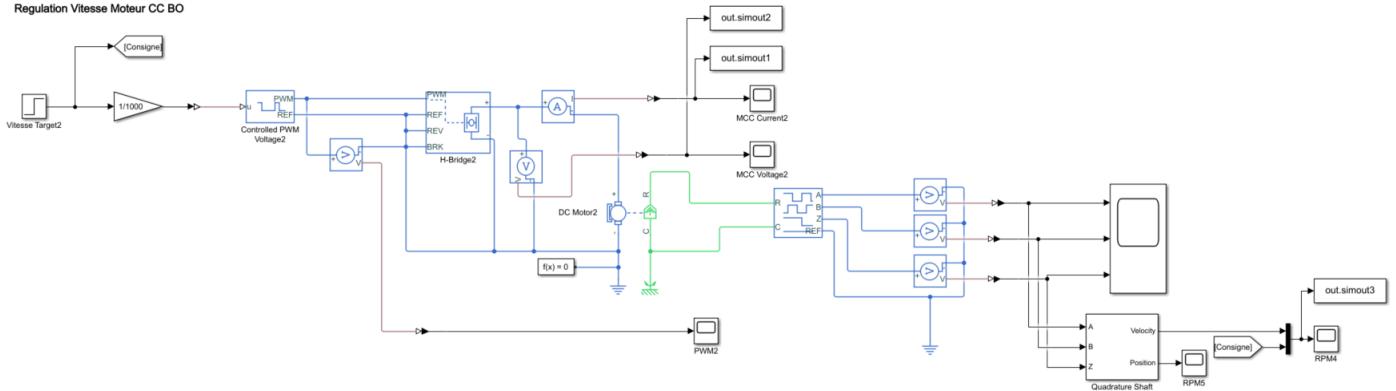


FIGURE 3.18 – Schéma bloc du système en boucle ouverte :

Dans ce bloc on fait l'étude de contrôle et régulation de vitesse du moteur modélisé auparavant.

Blocs utilisés :

— Bloc "DC motor" avec les propriétés du machine déjà modéliser auparavant :

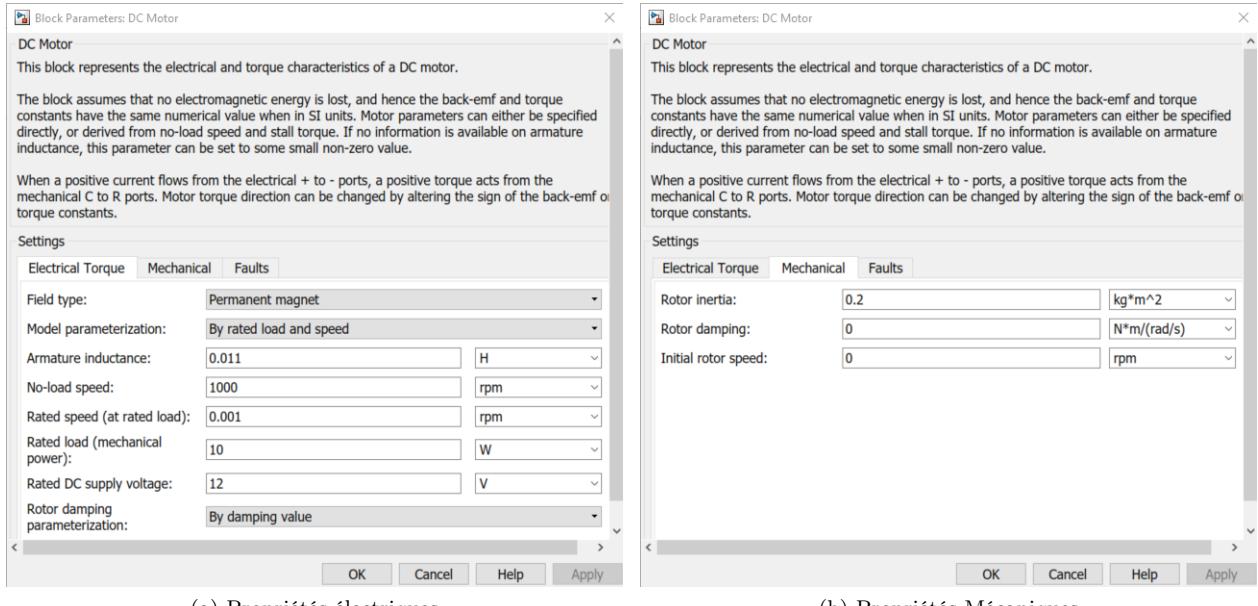


FIGURE 3.19 – Visualisation du signal MLI :

— Le bloc "Step (Vitesse Target)" génère au 2ème seconde un signal égale au consigne (Consigne est la valeur finale dans les paramètres du bloc), cette valeur passe par un gain 1/1000 pour être conforme au rapport cyclique du signal PWM (Valeur Input pour un rapport cyclique 0 Pourcent = 0, Valeur Input pour un rapport cyclique de 100 Pourcent= 1).

- Le bloc "Controlled PWM Voltage" reçoit la valeur du consigne après gain cette valeur est le rapport cyclique du PWM (Dans ce cas le rapport cyclique du signal PWM est 1) qui génère un signal suffisant pour tourner le moteur à la vitesse consigne.
- Le bloc "H-Bridge" reçoit le signal PWM comme entrée et entraîne / tourne le moteur.
- Le bloc "Incremental Shaft Encoder" représente un dispositif qui convertit les informations sur la position angulaire de l'arbre moteur en impulsions électriques.
- Le bloc "Quadrature Shaft Decoder" mesure la rotation de l'arbre à partir du bloc encodeur et affiche la vitesse et la position.

3.2.6.2 Simulation et Résultats :

On impose une vitesse consigne de 1000RPM.

Diagrammes :

La figure ci dessous représente l'allure du vitesse :

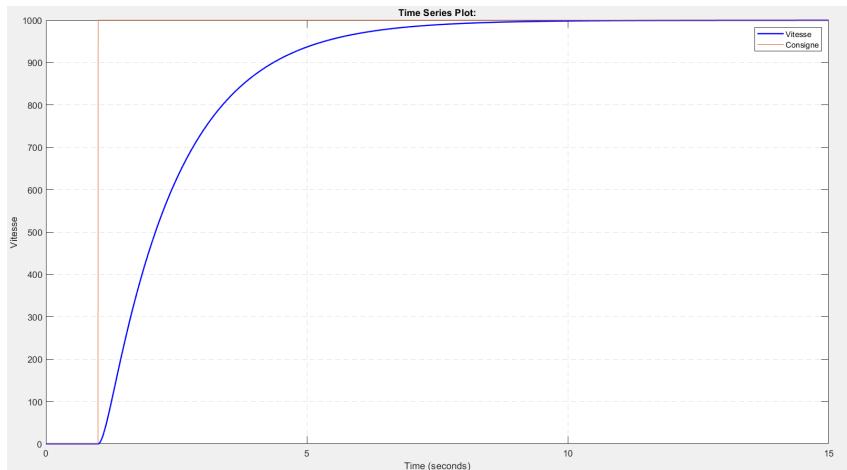


FIGURE 3.20 – Allure de vitesse

La figure ci dessous représente l'allure du tension :

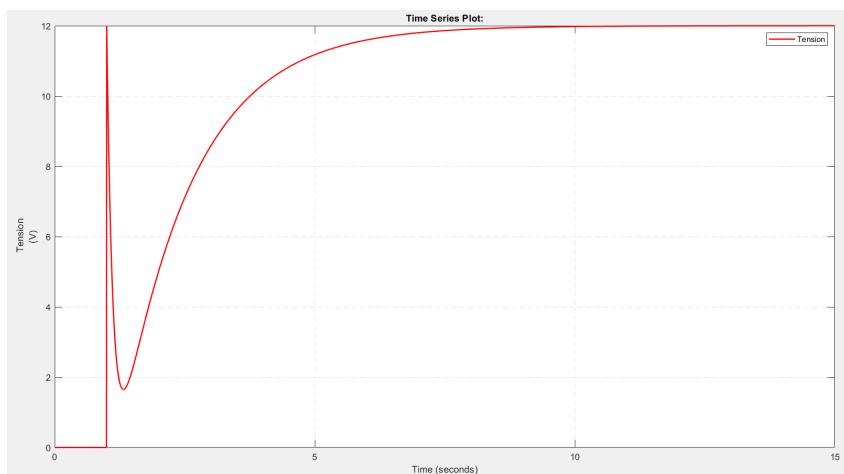


FIGURE 3.21 – Allure de tension

La figure ci dessous représente l'allure du courant :

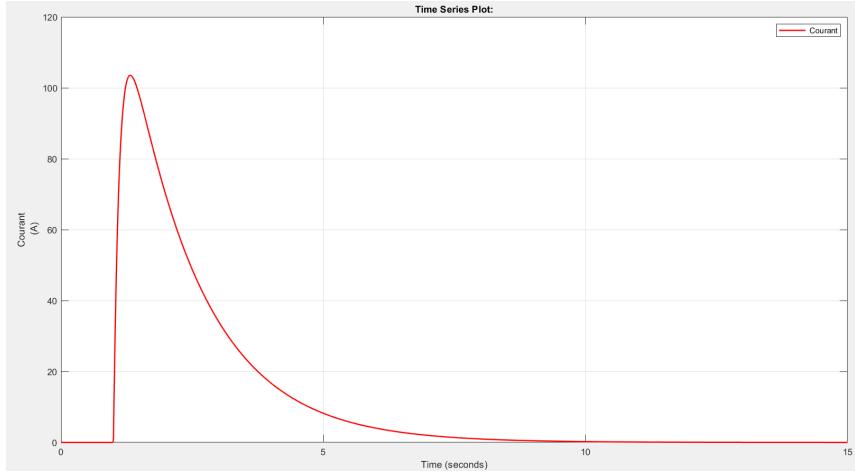


FIGURE 3.22 – Allure de Courant

Limitation de boucle ouverte :

- Avantages du système de contrôle à boucle ouverte :
 - Simple dans la construction et le design.
 - Facile à maintenir.
 - Généralement stable.
- Inconvénients du système de contrôle à boucle ouverte :
 - Ils sont inexacts.
 - Ils ne sont pas fiables.
 - Tout changement de sortie ne peut pas être corrigé automatiquement.

3.2.7 Système en boucle Fermée :

La boucle fermée est capable de stabiliser un système instable en boucle ouverte. Dans une régulation en boucle fermée, une bonne partie des facteurs perturbateurs externes sont automatiquement compensés par la contre-réaction à travers le procédé.

3.2.7.1 Modélisation/Schéma Bloc :

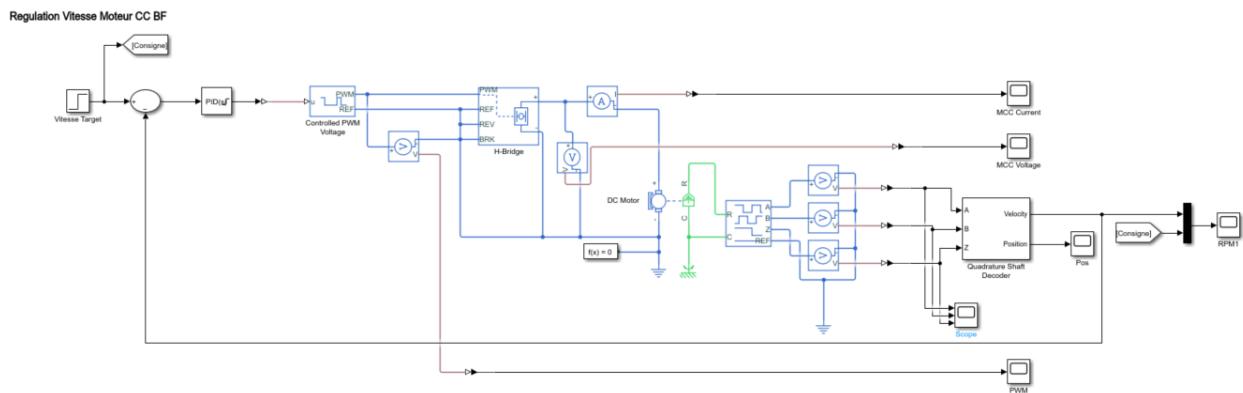


FIGURE 3.23 – Schéma bloc du système en boucle fermée :

Dans ce bloc on refait la même étude que celle de la boucle ouverte. Mais dans ce cas la correction se fait automatiquement par le bloc régulateur "PID". Pour assurer la bonne régulation il faut choisir les paramètres du correcteur K_p, K_i, K_d . On utilise les paramètres qu'on a trouvé dans la modélisation auparavant.

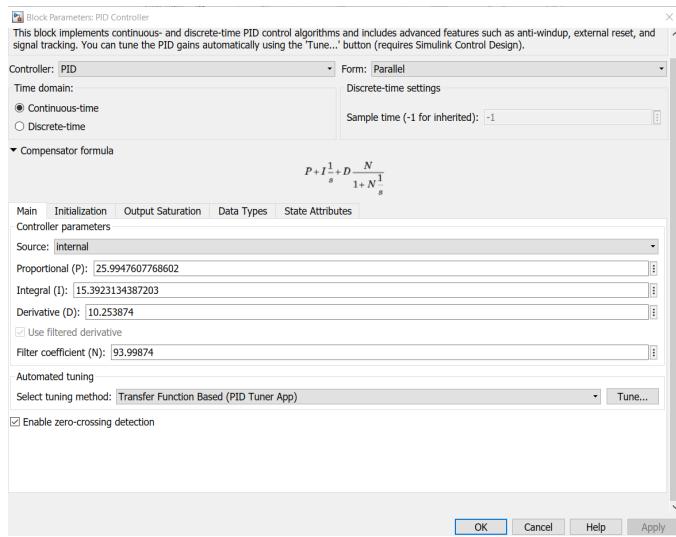


FIGURE 3.24 – Paramètres du correcteur :

3.2.7.2 Simulation et Résultats :

On impose une vitesse consigne de 500RPM. Dans ce cas le rapport cyclique est 0.5. La figure ci dessous représente l'allure du vitesse :

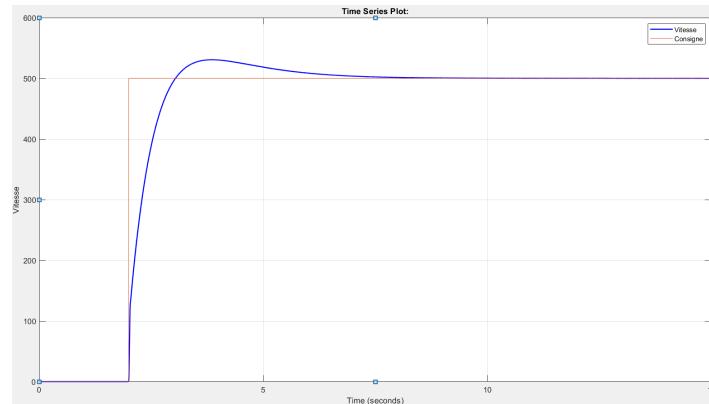


FIGURE 3.25 – Allure de vitesse

La figure ci dessous représente l'allure du tension :

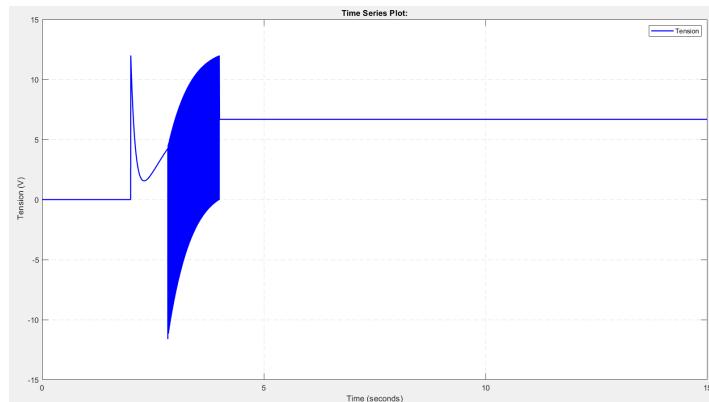


FIGURE 3.26 – Allure de tension

La figure ci dessous représente l'allure du courant :

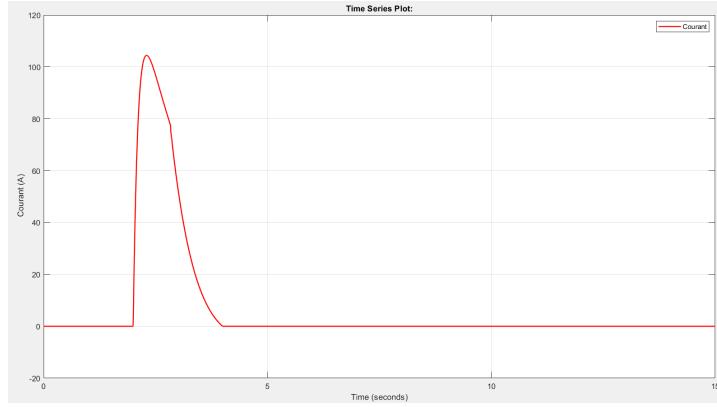


FIGURE 3.27 – Allure de Courant

3.3 Conclusion :

Dans ce chapitre, nous avons présenté la modélisation du système en boucle ouverte et fermée puis nous avons présenté les simulations de notre moteur en boucle ouverte/fermée et les résultats de ses simulations.

Chapitre 4

Implémentation du système de contrôle et régulation de vitesse :

4.1 Introduction :

Dans ce chapitre on va commencer la partie réalisation et implémentation de notre système de régulation et contrôle de vitesse d'un MCC. Le but de ce travail est de faire une réalisation pratique destinée à mettre en évidence une commande PID d'un MCC par la carte Arduino et le logiciel Matlab/Simulink.

4.2 Software :

Dans notre projet, on utilise le logiciel Matlab Simulink pour programmer l'Arduino parce que il fournit un environnement graphique interactif pour la modélisation, il permet aussi de la visualisation des signaux et résultats en temps réel, mais il faut télécharger le "Simulink Support Package for Arduino Hardware" pour pouvoir programmer l'Arduino via Matlab Simulink.

4.2.1 Simulink support package for Arduino Hardware :

Avec Support Package Simulink pour Arduino matériel, vous pouvez utiliser Simulink pour communiquer de manière interactive avec une carte Arduino via un câble USB. Le package permet de créer et d'exécuter des modèles Simulink sur des cartes Arduino. Le package de support comprend une bibliothèque de blocs Simulink pour configurer et accéder aux capteurs, actionneurs et interfaces de communication Arduino. Il vous permet également de surveiller et de régler de manière interactive les algorithmes développés dans Simulink lorsqu'ils s'exécutent sur Arduino.

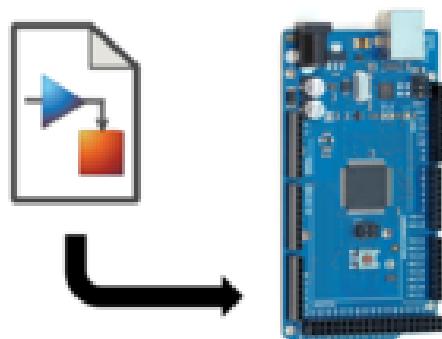


FIGURE 4.1 – Simulink support package for Arduino Hardware :

4.3 Hardware :

4.3.1 La carte Arduino Mega 2560 :

On a choisis la carte Arduino et plus spécifiquement la carte Mega 2560 car c'est est facile à utiliser, fonctionne en mode autonome et est peu coûteux.

La carte Arduino Mega 2560 est basée sur un ATMega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 54 broches numériques d'entrée / sortie dont 14 PWM, 16 entrées analogiques, 4 UART (ports série matériels), un oscillateur en cristal de 16 MHz, d'une connexion USB, une prise d'alimentation, d'une embase ICSP, et un bouton de réinitialisation. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires. Il contient tout le nécessaire pour soutenir le microcontrôleur suffit de le brancher à un ordinateur avec un câble USB ou de l'alimenter avec un adaptateur AC-DC ou une batterie pour démarrer.

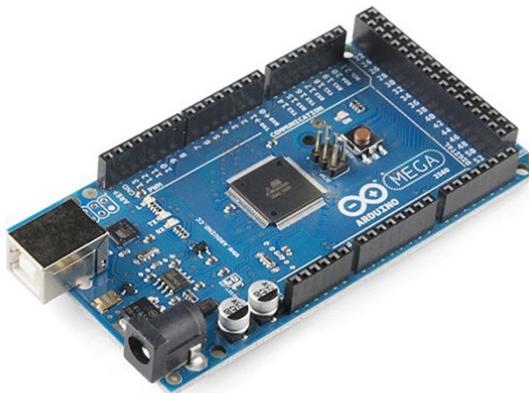


FIGURE 4.2 – Arduino Mega 2560 :

Les caractéristiques d'Arduino Mega 2560 :

Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	54 (dont 14 PWM)
Broches d'entrées analogiques	16 (utilisables en broches E/S numériques)
Intensité de courant par broche E/S	20 mA
Intensité de courant pour la broche 3.3V	20 mA
Mémoire Programme Flash	56 KB dont 8 KB sont utilisés par le boot loader
Mémoire SRAM (volatile)	8 KB
Mémoire EEPROM (non volatile)	4 KB
Vitesse d'horloge	16 MHz

FIGURE 4.3 – Caractéristiques d'une Carte Arduino Mega 2560

4.3.2 Le module L298N :

Le module L298N est un Double Pont-H destiné au piloter un moteur (H-Bridge Motor Driver). Il est basé sur le composant L298N développé spécifiquement pour ce cas d'utilisation.

Il s'agit d'un pilote à double pont capable d'entraîner des charges inductives comme des relais, Solénoides, 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. C'est un module extrêmement utile pour le contrôle de robots et ensembles mécanisés. Il peut contrôler deux moteur courant continu ou un moteur pas-à-pas 4 fils 2 phases.

On a choisi le module L298N car il supporte des tensions plus élevées, des courants importants, il est aussi conçu pour accepter les niveaux logiques TTL standard (basse tension, courant faibles) ce qui est idéal pour un microcontrôleur.

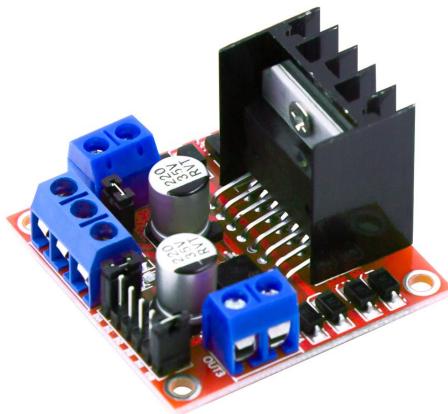


FIGURE 4.4 – Le Module L298N :

Spécifications du Module L298N :

- Composant de contrôle en puissance : L298N
- Alimentation de la charge : de +6V à +35V
- Courant Max (en pointe) : 2A
- Tension de commande logique Vss : de +5 à +7V (alimentation interne de +5V)
- Courant de commande logique : de 0 à 36mA
- Tensions pour contrôle du sens : Low -0.3V 1.5V, high : 2.3V Vss
- Tensions pour contrôle « Enable » : Low -0.3V 1.5V, high : 2.3V Vss
- Puissance Max : 25W (Température 75 [U+2103])
- Température de fonctionnement : de -25 [U+2103] à +130 [U+2103]
- Dimensions : 60mm x 54mm
- Poids : 48g

Les caractéristiques du module L298N :

- Des capacités hors-pair pour contrôler le moteur
- Diodes de protections
- Un dissipateur (pour dissiper la chaleur en cas de forte charge)
- Un sélecteur pour sélectionner la source d'alimentation
- 4 Sélecteurs pour les résistances pull up

- Sortie pour 2 moteurs continu/ 1 moteur pas-à-pas (4 bobines, deux phases)
- Indicateur LED pour sens de rotation moteur
- Indicateur LED pour alimentation 5V
- 4 trous de fixation standard
- Léger

Présentation du module L298N :

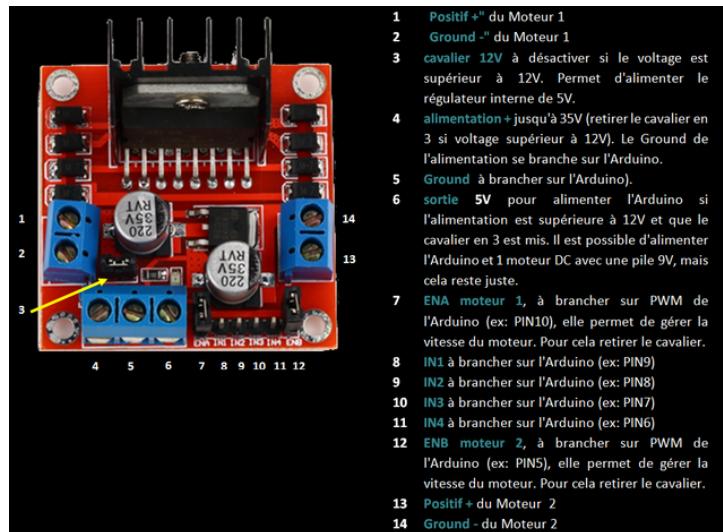


FIGURE 4.5 – Détails Techniques du module L298N :

4.3.3 Motoréducteur (DC,Encodeur 740 ppr) :

On a choisi un moteur MCC avec codeur intégré pour faciliter la phase d'acquisition d'information, dans notre cas la vitesse.



FIGURE 4.6 – MCC avec codeur intégré :

Les fils représentent :

- **Fil Rouge** : Moteur +
- **Fil Noir** : Encodeur -
- **Fil Jaune** : Encodeur phase A
- **Fil Vert** : Encodeur phase B
- **Fil Bleu** : Encodeur +
- **Fil Blanc** : Moteur -

Les caractéristiques du Motoréducteur :

- **Large application** : ce moteur est idéal pour les petites voitures intelligentes, les robots, les imprimantes, les voitures d'équilibrage, les appareils ménagers intelligents, les culbuteurs d'appareil photo, etc.
- **Carreau magnétique haute résistance** : longue durée de vie, puissance élevée, couples plus importants. Structure entièrement fermée pour empêcher la poussière de pénétrer à l'intérieur du moteur, structure simple, entretien pratique.
- **Engrenage entièrement métallique** : la boîte de vitesses est composée d'engrenages métalliques de haute qualité, qui ont une bonne résistance aux chocs et une bonne durabilité prolongeant ainsi la durée de vie de la boîte de vitesses.
- **Brosse en carbone de haute qualité** : sous réserve que d'autres matériaux répondent aux exigences, elle présente de bonnes performances de commutation, une longue durée de vie, une bonne conductivité électrique et thermique et de bonnes performances de lubrification.
- **Bobine en cuivre pur** : le moteur est composé d'un rotor de haute précision et de matériaux de haute qualité, avec une conductivité élevée, une résistance élevée à l'usure et une longue durée de vie.

4.3.3.1 Décodage des signaux de l'encodeur intégré :

Un codeur moteur est un codeur rotatif monté sur un moteur électrique qui fournit des signaux de rétroaction en boucle fermée en suivant la vitesse et / ou la position d'un arbre de moteur. L'encodeur entraîné par le moteur génère deux signaux digital au bornes de ses phases A et B.

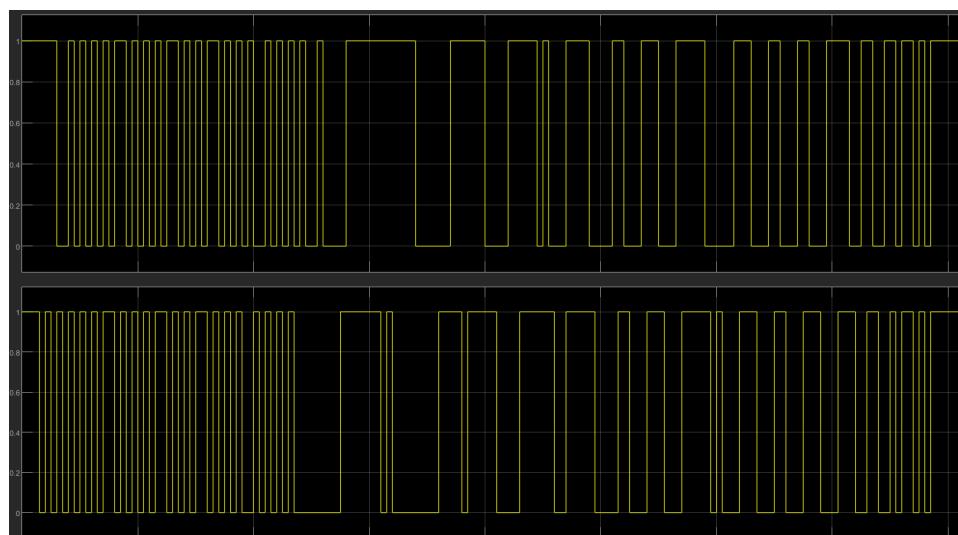


FIGURE 4.7 – Visualisation des signaux aux bornes des phases A et B :

Nous souhaitons estimer la vitesse du moteur en fonction du nombre de l'encodeur en ajoutant les éléments suivants à notre modèle Simulink. Étant donné que les impulsions de l'encodeur indique la position du moteur, nous pouvons approximer la vitesse du moteur sur un intervalle de temps spécifique comme le changement de position du moteur divisé par le changement de temps entre les échantillons. En substance, il s'agit de la vitesse moyenne du moteur sur cet intervalle de temps. Plus précisément, le bloc Différence calcule le changement de position du moteur (en nombres) et le premier bloc Gain se divise par le temps d'échantillonnage. Le bloc de gain (60/740) suivant convertit les unités de impulsions/sec en révolutions/sec, puis de révolutions/sec en révolutions/min.

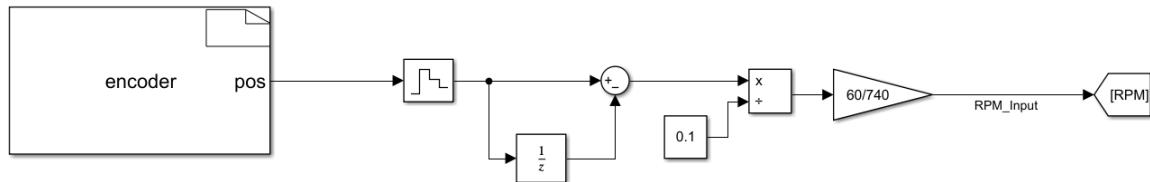


FIGURE 4.8 – Bloc décodage des signaux du codeur :

4.3.4 Branchement du système :

Notre système est composé d'un moteur MCC avec codeur intégré, un pont en H module L298N et une carte Arduino mega 2560 :

- Le moteur est branché au L298N, Moteur (+) à l'emplacement positif du moteur 1, Moteur (-) à l'emplacement positif du moteur 1.
- Les deux phases du codeur sont brancher au deux pin d'interruptions 2 et 3 de la carte Arduino Mega 2560, Phase A est brancher au pin 2, Phase B brancher au pin 3.
- Le codeur est alimenté par 3.3V fournie par la carte Arduino.
- L'alimentation du branchement est faite par un Adaptateur ac/dc de 12V brancher au emplacement +V et Terre du pont en H.

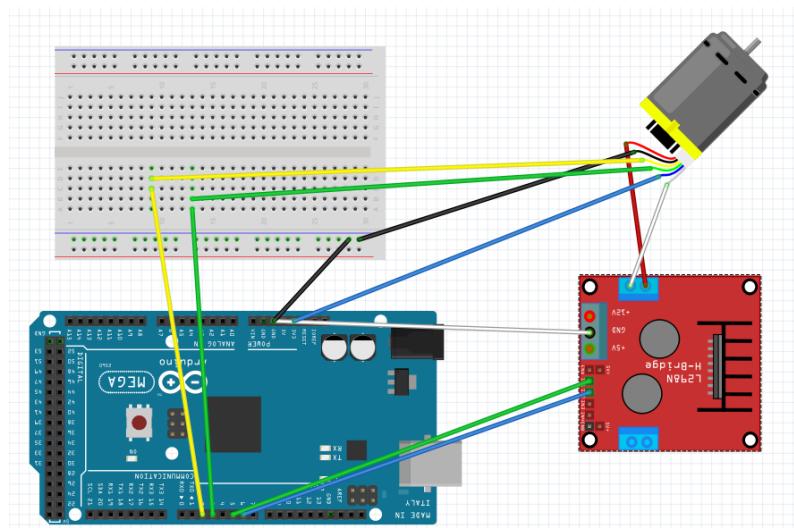


FIGURE 4.9 – Branchement du système sous logiciel Fritzing :

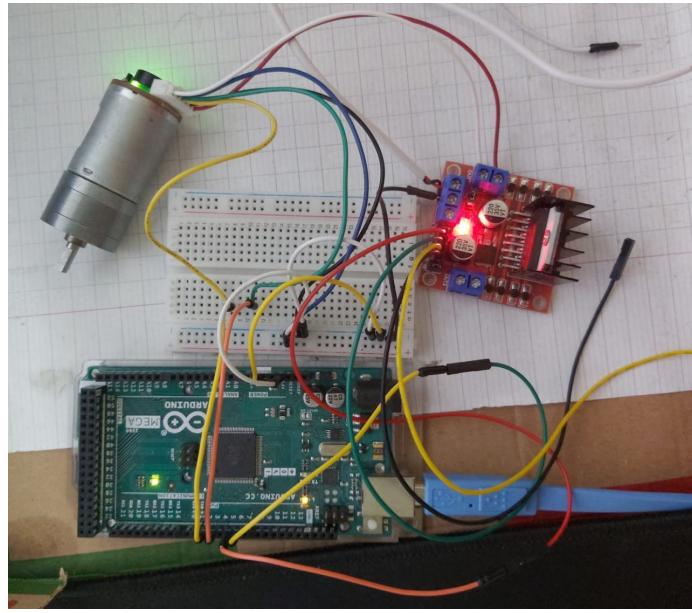


FIGURE 4.10 – Branchement réel du système :

4.4 Réalisation et Résultats :

Toutes les réalisations sont faites à vide.

4.4.1 Boucle Ouverte :

4.4.1.1 Schéma Bloc :

Le Bloc ci dessous représente le système de régulation en boucle ouverte, il nous permet également de surveiller et de régler le système de manière interactive.

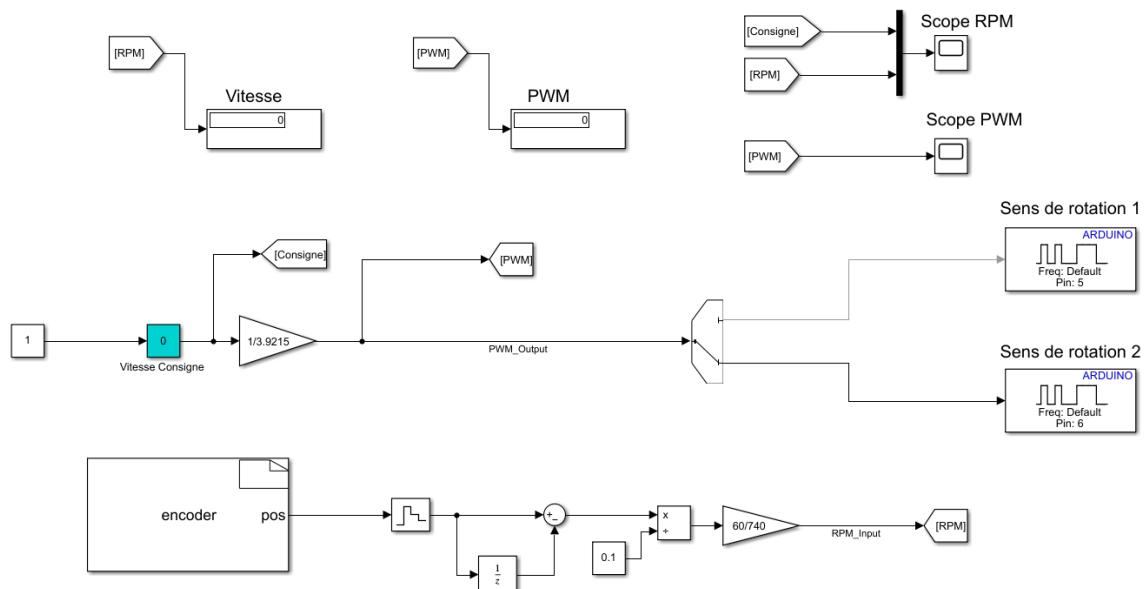


FIGURE 4.11 – Schéma bloc boucle ouverte :

4.4.1.2 Résultats :

On impose une vitesse consigne de 450RPM.
Le résultat de la simulation est représenté sur la figure ci dessous :

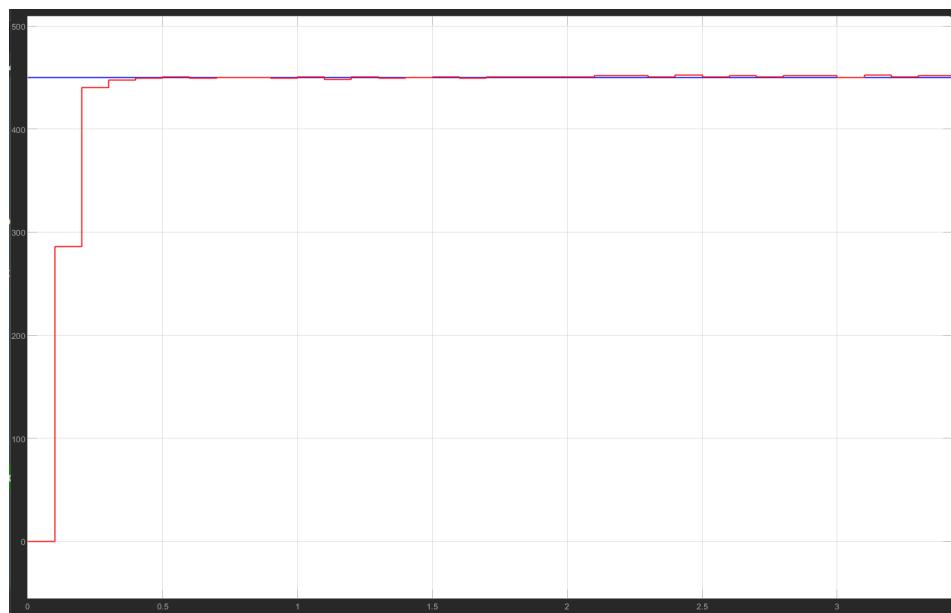


FIGURE 4.12 – Allure de Vitesse :

4.4.2 Boucle fermée :

4.4.2.1 Schéma Bloc :

Le Bloc ci dessous représente le système de régulation en boucle fermée, il nous permet également de surveiller et de régler le système de manière interactive

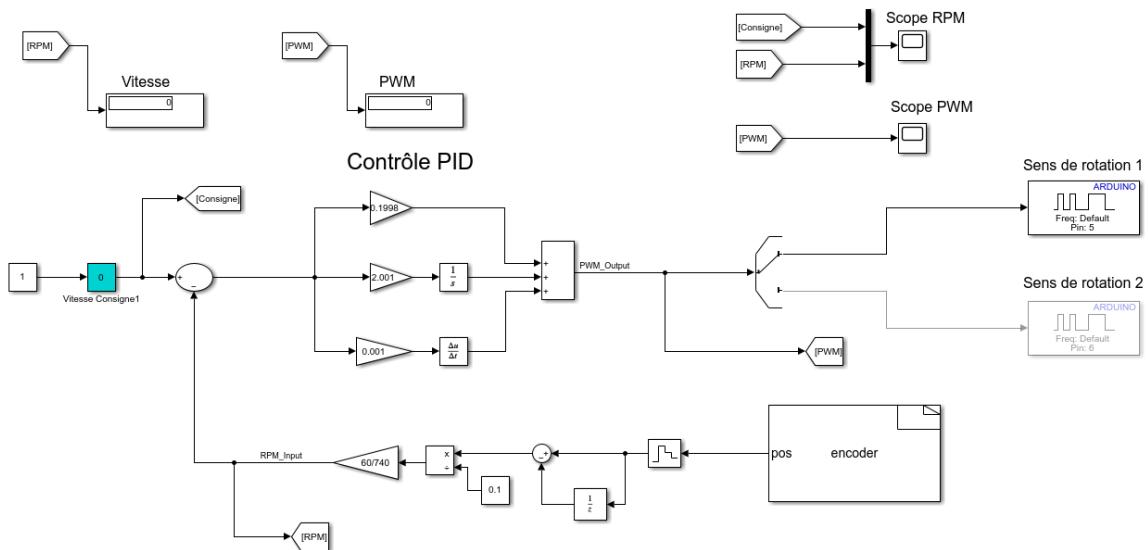


FIGURE 4.13 – Schéma bloc boucle fermée :

4.4.2.2 Résultats :

On impose une vitesse consigne de 350RPM.

Le résultat de la simulation est représenté sur la figure ci dessous :

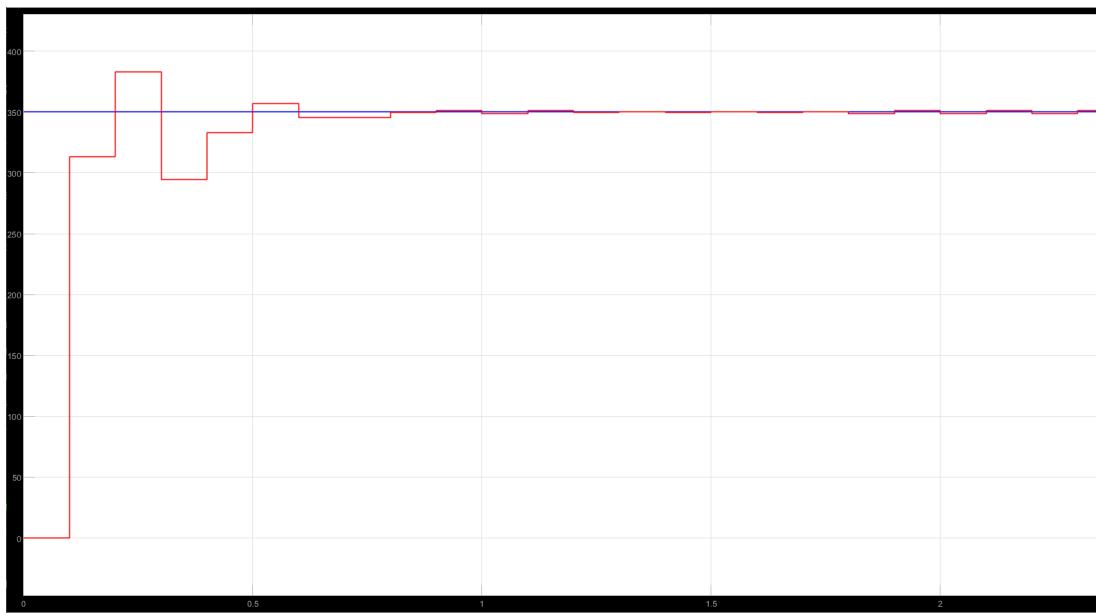


FIGURE 4.14 – Résultats de la simulation boucle fermée :

4.5 Conclusion :

Dans ce chapitre nous avons présenté la réalisation de notre système, en implémentant tout ce que nous avons étudié et réalisée par les Hardware et Software décrits Antérieurement. Ainsi en présentant quelques résultats.

Chapitre 5

Conclusion et Perspectives :

5.1 Conclusion :

Le travail effectué dans ce projet rentre dans le cadre du contrôle et régulation de la vitesse du moteur à courant continu. D'abord, un état d'art a été présenté sur les machines à courant continu.

Ensuite, On a donné un aperçu sur les différentes technologies utilisées (la MLI, les encodeurs,Les convertisseurs (Hacheurs),l'asservissement,les correcteurs et les microcontrôleurs). (constitution, fonctionnement, ainsi que la modélisation).

Durant cette étude projet, on a aussi vu que la commande PID appliquée aux moteurs électriques à courant continu était un régulateur qui pouvait offrir des performances satisfaisantes au niveau de la précision, du temps de réaction, du dépassement, ou de la stabilité, mais le régulateur idéal n'existe pas. Il convient alors de trouver le meilleur compromis en fonction des exigences initiales du cahier de charge.

L'utilisation de la carte Arduino et la plate-forme Simulink à grandement faciliter la réalisation du système et nous a permis d'obtenir un résultat assez concluant.

5.2 les prochaines étapes :

De nombreuses pistes d'amélioration peuvent être suggérées afin de développer de plus notre projet, la plus pertinente d'eux est :

- La création d'une interface graphique de commande pour faciliter cette tâche pour l'utilisateur.

Bibliographie

- [1] <https://fr.wikipedia.org/wiki/Machine_%C3%A0_courant_continu>.
- [2] <http://www.est-usmba.ac.ma/coursenligne/Chapitre6_MCC.pdf>.
- [3] <<https://energieplus-lesite.be/techniques/chauffage10/principes-de-regulation-p-pi-pid/>>.
- [4] <<https://www.alienxarea.fr/comment-piloter-un-moteur-a-courant-continu/>>.
- [5] <<http://tsi.ljf.free.fr/ATS/docs/S2I/CI3A/Hacheur.pdf>>.
- [6] <http://bacstielectronique.free.fr/fichiers_cours_electronique/pont%20H%20_2_.pdf>.
- [7] <<https://www.amazon.fr/Encodeur-Motor%C3%A9ducteur-Encoder-R%C3%A9ducteur-Vitesse/dp/B08626S67Z?th=1>>.
- [8] <<http://for-ge.blogspot.com/2015/07/microcontroleur.html>>.
- [9] <<https://www.gotronic.fr/art-carte-arduino-mega-2560-12421.htm>>.
- [10] <<http://www.planetjeux.net/post/les-avantages-de-la-carte-arduino-mega>>.
- [11] <<https://store.arduino.cc/products/arduino-mega-2560-rev3>>.
- [12] <https://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_DCmotorA#1>.
- [13] <https://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_DCmotorA#1>.
- [14] <<https://fr.mathworks.com/videos/speed-control-of-a-dc-motor-using-pwm-68962.html>>.
- [15] <http://prepatsi.monge.free.fr/robotique/commande_PWM_moteurs.pdf>.