

**République Algérienne démocratique et Populaire**

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

---

Université des Sciences et de la Technologies d'Oran « Mohamed Boudiaf »



**FACULTE DE GENIE ELECTRIQUE**

Département d'Automatique

**PROJET DE FIN D'ETUDE**

---

**COMMANDE D'UN MOTEUR A COURANT CONTINU**

**PAR « ARDUINO » ET «LabVIEW»**

Réalisé par :

- KEBDANI OUSSAMA.
- RAHMOUN LOKMANE NOUR EL ISLEM.
- MOKHTAR BENOUNAN MOHAMED.

Encadré par :

Mr. Z.Bellahcene

Année universitaire 2018/2019

## Table des matières

Table des matières .....	2
Introduction .....	4
Premier chapitre .....	5
Partie 01 : Généralités sur le moteur à courant continu .....	5
1.    fonctionnement du moteur à courant continue : .....	5
2.    Constitution d'un moteur à courant continu : .....	6
3.    les branchements des moteurs à courant continue : .....	6
Partie 02 : Modélisation mathématique du moteur à courant continu EMG30.....	8
1.    Introduction : .....	8
2.    Mise en équation : .....	8
Deuxième chapitre.....	10
Partie 01 : élaboration du circuit de puissance .....	10
1.    introduction sur la Carte ARDUINO : .....	10
2.    Signal de commande PWM : .....	13
3.    Etage de puissance : .....	13
Partie 02 : Elaboration du block diagramme sur LabVIEW .....	18
1.    introduction sur logiciel LabVIEW : .....	18
2.    élaboration du block diagramme sur LabVIEW : .....	18
3.    Réalisation d'une interface graphique de commande : .....	18
Troisième chapitre.....	19
Partie 01 : Etude de la réponse indicielle du moteur.....	20
1.    introduction : .....	20
2.    méthode de STREJC : .....	20
3.    régulateur PID (proportionnel-intégral-dérivé) : .....	22
4.    régulation sous LabVIEW : .....	23

5.	Calcul des paramètres du correcteur (pid) : .....	23
6.	Affichage de la vitesse: .....	26
Conclusion générale .....		27
Annexe .....		28
Bibliographie .....		29

### Introduction :

Les moteurs à courant continue jouent un rôle crucial dans la recherche et les expériences de laboratoire en raison de sa simplicité et de son faible coût. La vitesse du moteur peut être contrôlée par la tension, ou bien la méthode de contrôle par pwm Ici, dans ce projet, on va effectuer la commande par pwm.

Dans ce projet de fin de cycle on est amené à commander un moteur à courant continu à l'aide de deux cartes arduino et LabVIEW dans le but d'étudier son fonctionnement.

Les principaux objectifs de ce projet de fin d'étude sont :

- lire la vitesse de l'encodeur à partir d'une carte arduino.
- la réalisation d'une interface graphique sous LabVIEW dans le but de faire la régulation et la commande du moteur.
- l'identification de la réponse indicielle du moteur et la détermination des paramètres du régulateur pid.
- affichage de la vitesse sous LCD.

Notre travail est structuré de la manière suivante :

- **Premier Chapitre :**  
Généralités sur le moteur à courant continu, et détermination du modèle mathématique.
- **Deuxième Chapitre :**  
Elaboration de la maquette électrique.  
Elaboration de l'interface graphique.
- **Troisième chapitre :**  
Identification des paramètres du moteur par étude de sa réponse indicielle.  
Régulation de la réponse indicielle du moteur, par un Correcteur « Proportionnelle Intégrale dérivée ».
- **Conclusion générale.**

## Premier chapitre

Généralités sur le moteur à courant continu, et détermination du modèle mathématique.

Dans ce chapitre on va comprendre le fonctionnement du moteur à courant continu et ses principales constitutions d'une façon pour faire sa modélisation mathématique.

### Partie 01 : Généralités sur le moteur à courant continu

Le moteur à courant continu (mcc) est une machine tournante qui transforme l'énergie électrique en mécanique, Son fonctionnement est basé sur la loi de Laplace : soit un conducteur  $L$  traversé par un courant  $I$  et placé dans un champ magnétique  $B$  est soumis à une force  $F$  dont le sens est déterminé par la règle des trois doigts de la main droite.

$$F = B \cdot S \cdot I \cdot L \cdot \sin(\alpha)$$

Cette force va tourner le rotor et créer le couple moteur, la construction du moteur est basée sur ce principe.

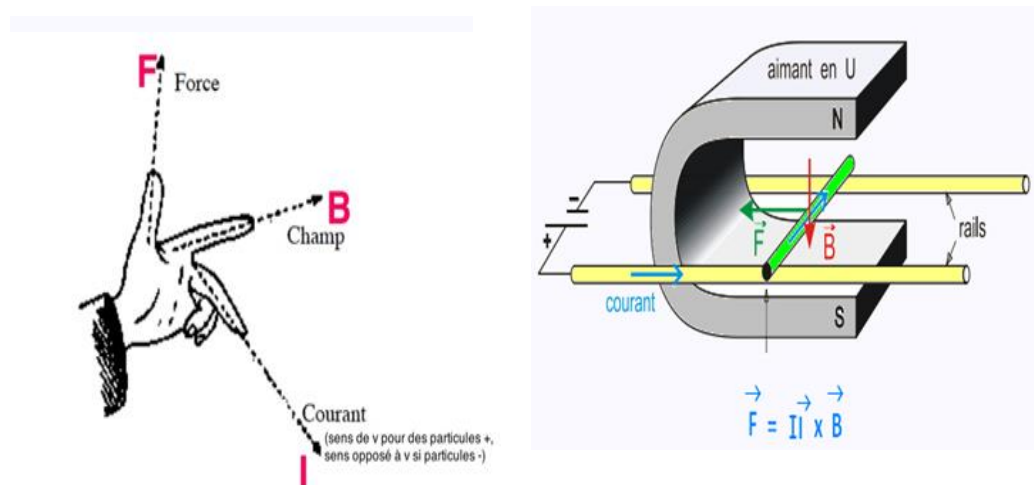


Figure 1.1: Principe de la force LAPLACE

## 2. Constitution d'un moteur à courant continu :

Le moteur à courant continu est constitué principalement par deux ensembles :

Stator (Inducteur) :

C'est l'élément fixe, son rôle est de créer un flux magnétique, cette fonction est assurée par un aimant permanent ou par un courant électrique circulant dans le bobinage.

Rotor (Induit) :

Composé d'un châssis métallique comprenant un certain nombre d'encoches, sur lequel sont placé un certain nombre de bobinage.

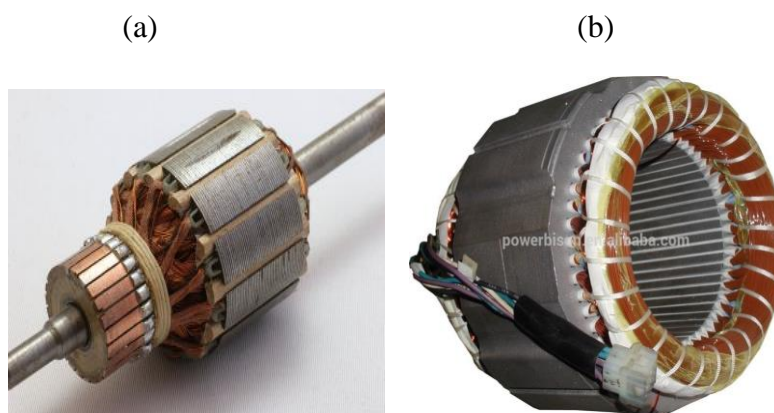


Figure I.2 : (a) Illustration de l'inducteur, (b) Illustration de l'induit.

## 3-les branchements des moteurs à courant continu :

### a-excitation indépendante :

L'inducteur et l'induit sont alimentés de façons indépendantes.

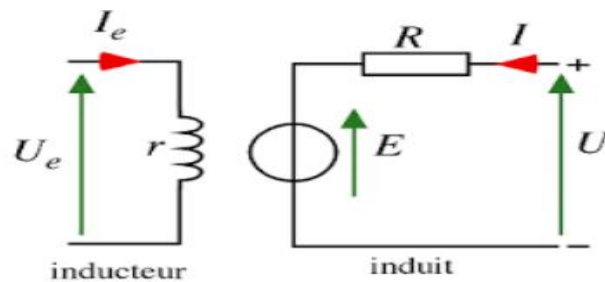


Figure I.3 : Circuit électrique du moteur à excitation Séparée.

**b-excitation série :**

L'inducteur et l'induit sont connectés en série.

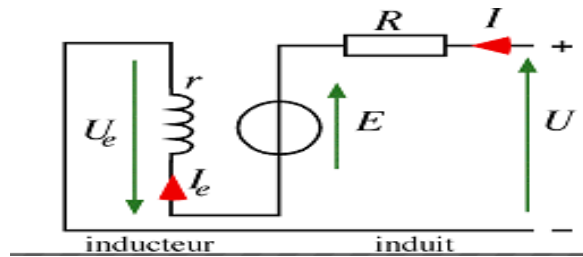


Figure I.4 : Circuit électrique du moteur à excitation Shunt.

**c-excitation parallèle(ou shunt) :**

L'inducteur et l'induit sont connectés en parallèle.

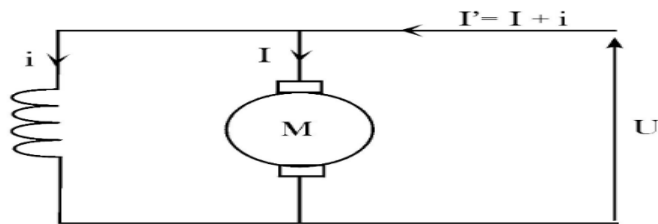


Figure I.5 : Circuit électrique du moteur à excitation Série.

**d-excitation composé :**

Une partie du stator est raccordé en série avec le rotor et un autre est de type parallèle ou shunt. Ce moteur réunit les avantages des deux types de moteur : le fort couple à basse vitesse du moteur série et l'absence d'emballement (survitesse) du moteur shunt.

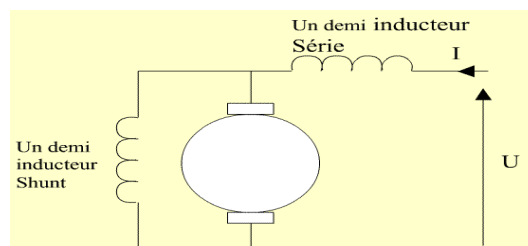


Figure I.6 : Circuit électrique du moteur à excitation Composée.

## Partie 02 : Modélisation mathématique du moteur à courant continu EMG30

## 1. Introduction :

Dans notre système en étude un moteur à courant continu à aimant permanents ce qui entraîne une charge en rotation de couple  $C_c(t)$ .

Ce système contient deux entrées :  $-u(t)$  (tension alimentation)

- Couple de charge  $C_c$  (perturbation)

La sortie correspond à la position du rotor  $\vartheta(t)$  pour un control de position ou bien  $w(t) = \frac{d\vartheta(t)}{dt}$  pour un control en vitesse.

Mise en équation :

Le moteur à courant continu peut être représenté sous le schéma électrique suivant :

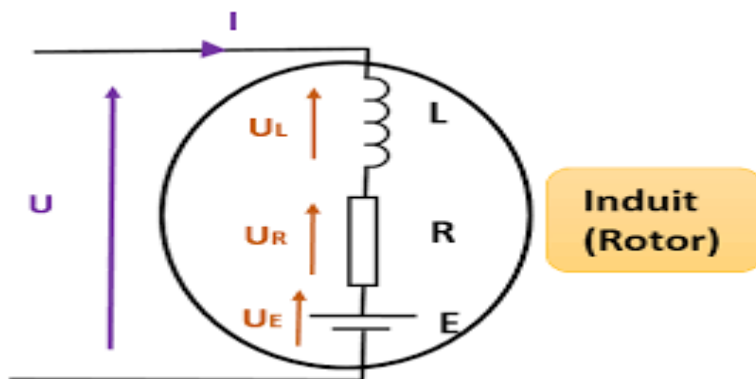


Figure I.7 : Circuit électrique du moteur à courant continu.

$$(1) \quad U(t) = Ri(t) + L \frac{di}{dt} + E(t) \quad \text{loi des mailles}$$

$$E = kw(t) \quad \text{fcem (loi de Lenz)}$$

Avec :

$E(t)$  : force électromotrice (volts).

$K$  : constante de force électromotrice.

$W$  : vitesse angulaire (rad/min).

A l'équilibre on a :

$$C_m = Kci(t) = fw(t) + J \frac{dw(t)}{dt} + C_r \quad (2)$$

Ou

$C_m$  : couple moteur.



$K_c$  : constante du couple moteur.

$F$  : coefficient du frottement (N.m).

$J$  : moment d'inertie d'axe du rotor.

$C_r$  : couple résistant.

On effectue la transformée de Laplace sur les équations précédentes, on obtient :

De (1)

$$U(p) = RI(p) + L.p.I(p) + E(p)$$

$$U(p) = I(p)(R + L.p) + K_e w(p)$$

De (2)

$$I(p) = \frac{f w(p) + J p w(p) + C_r}{K_c}$$

On remplace (2) dans (1) on obtient la fonction de transfert suivante:

$$\frac{w(p)}{U(p)} = \frac{K_c}{(f + Jp)(R + Lp) + K_e K_c}$$

On peut l'écrire sous cette forme :

$$\frac{w(p)}{U(p)} = \frac{\frac{K_c}{fR}}{\left(1 + \frac{J}{f}p\right)\left(1 + p\frac{L}{R}\right) + K_e \frac{K_c}{fR}}$$

On prend :

$$\tau_1 = \frac{J}{f}; \quad \tau_2 = \frac{L}{R}; \quad K = \frac{K_c}{fR}$$

On obtient :

$$\frac{w(p)}{U(p)} = \frac{K}{(1 + \tau_1 p)(1 + \tau_2 p) + K_e K}$$

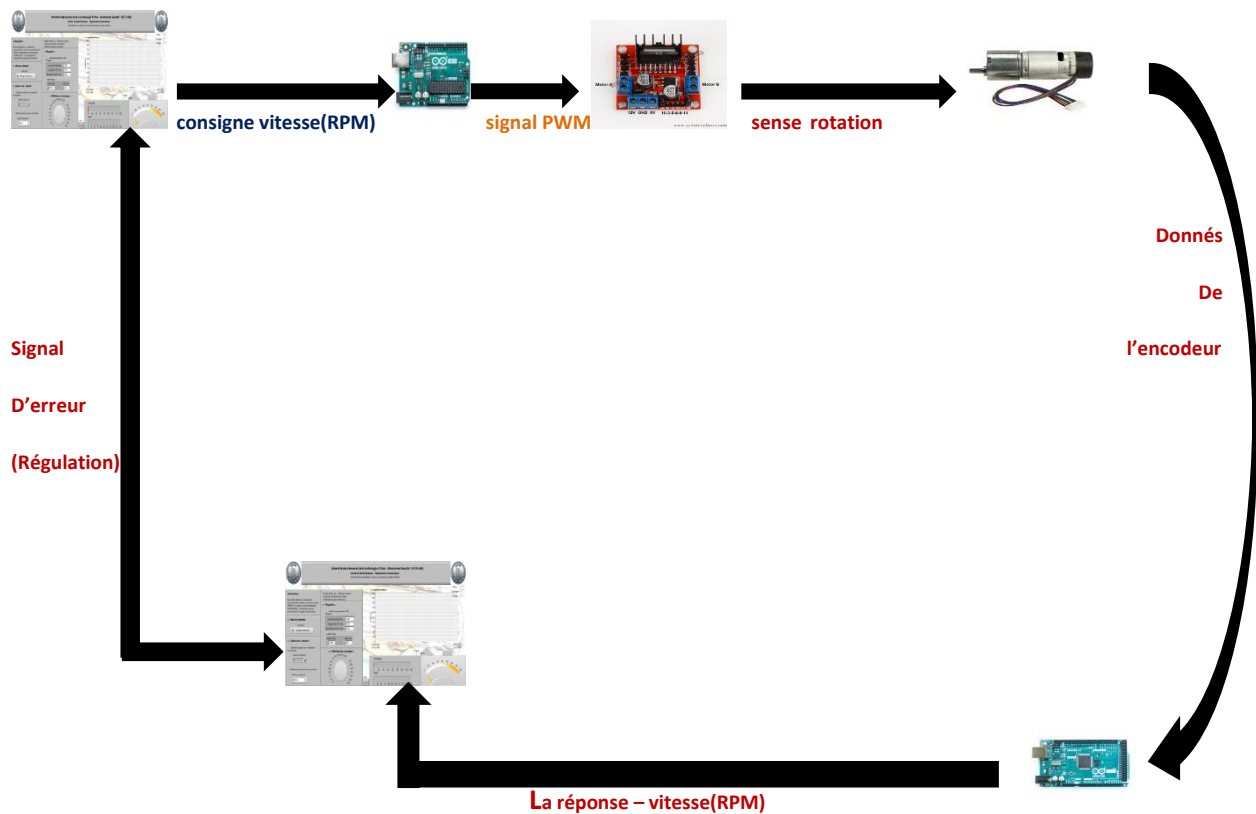
Cette fonction de transfert est d'ordre 2, et ainsi il y'a 2 constantes de temps qui régissent le système :

La première constante de temps est mécanique, tandis que la deuxième est électrique.

### 1. Introduction :

Dans cette session on va commencer à la partie pratique du notre projet on commence par la réalisation du circuit du puissance et de faire une liaison entre la partie logiciel et la partie matériel à travers la carte arduino et logiciel virtuel LabVIEW.

On peut schématiser notre projet sous la forme suivante :



L'interface graphique va donner une consigne vitesse a l'arduino qui reçoit cette consigne et envoie un signal PWM au driver qui donne le sens de rotation au moteur, la deuxième arduino reçoit les données du encodeur pour calculé la vitesse, cette réponse va être affiché sur notre interface et finalement l'interface va comparer entre la consigne donné et la vitesse calculé pour faire la régulation.

### 2. Carte ARDUINO :

L'arduino c'est une plateforme électronique open source basé sur une simple interface entrée/sortie. Les cartes arduino peuvent lire les entrées (lumière sur un capteur, doigt sur un bouton ou message Twitter) et en faire une sortie: activer un moteur, allumer une LED, publier quelque chose en ligne. Pour ce faire, on utilise le langage de programmation arduino (basé sur le câblage) et le logiciel arduino (IDE) basé sur le traitement.

Notre maquettes contient deux cartes arduino, une de type mega 2560 pour la commande du moteur (par le signal PWM) à travers LabVIEW et l'autre uno pour le reçoit des donnés et l'affichage de la vitesse à travers LCD.

## 2-a-arduino mega 2560 :

Arduino Mega 2560 est une carte à microcontrôleur basée sur l'ATmega2560.

Il possède :

54 broches d'entrée / sortie numériques (dont 15 peuvent être utilisées en tant que sorties PWM).

16 entrées analogiques.

4 UART (ports série matériels).

Un oscillateur à cristal de 16 MHz.

Une connexion USB, une prise de courant, un en-tête ICSP, et un bouton de réinitialisation.

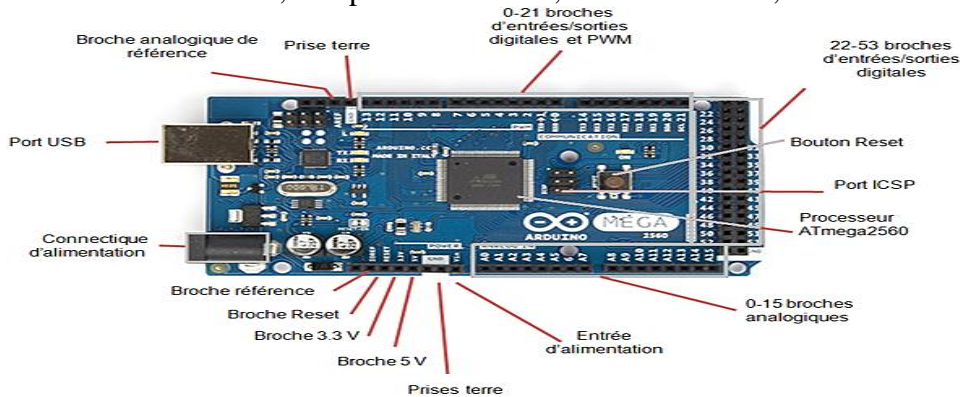


Figure II.1 : Vue réelle sur la carte ARDUINO MEGA

## 2-b-arduino uno :

Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328P. Il possède :

14 broches d'entrée / sortie numériques (dont 6 peuvent être utilisées en tant que sorties PWM).

6 entrées analogiques.

Un quartz 16 MHz.

Une connexion USB, une prise d'alimentation, un en-tête ICSP et un bouton de réinitialisation.

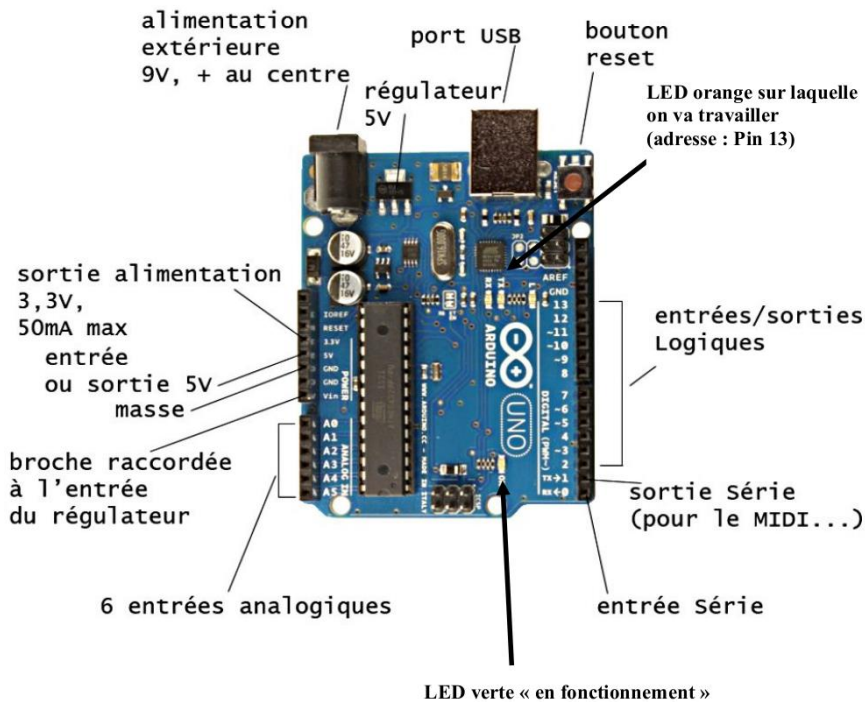


Figure II.2 : Vue réelle sur la carte ARDUINO UNO

## 2-c-Signal PWM (PULSE WIDTH MODULATION):

C'est un signal carré avec un rapport cyclique modulé en fonction d'un signal de commande, il nous permet de varier la vitesse du moteur à courant continu, cette technique est utilisée dans les onduleurs, redresseurs, et les convertisseurs.

Le rapport cyclique (duty cycle) pour notre signal rectangulaire est définie par  $\alpha$  est défini comme étant le rapport entre la durée du temps haut (th) du signal et sa période(T).

$$\alpha = \frac{th}{T} \quad \text{avec} \quad 0 \leq \alpha \leq 1$$

Le rapport cyclique est mesuré en pourcentage comme elle présente la figure ci-dessous :

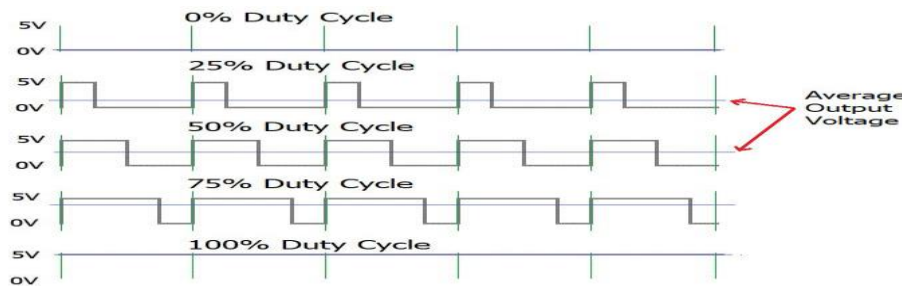


Figure II.3 : Signal PWM avec différents rapports cycliques sur une période « T »

La variation de la vitesse du moteur est effectuée en augmentant ou en diminuant l'intervalle de temps lorsque le signal de commande a la valeur logique 1 ainsi à un rapport cyclique de 100% le moteur aura le régime maximum et dans le cas du rapport cyclique égal à 0% le moteur sera arrêté.

## 3. Etage de puissance :

Le circuit électrique doit comporter les composants clés suivants :

Pont-H Driver (L298N):

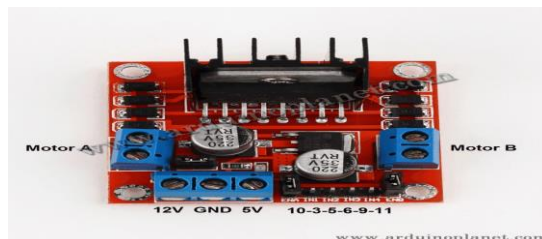


Figure II.4 : Vue réelle sur le moteur EMG30.

Un pont en H est un circuit électronique généralement utilisés pour contrôler la vitesse et la direction des moteurs, mais peuvent être utilisés pour freiner le moteur. Un pont H est un circuit peut conduire un courant dans l'une ou l'autre polarité et être contrôlé par modulation

de largeur d'impulsion (PWM). La modulation de largeur d'impulsion est un moyen de contrôler la durée d'une impulsion électronique.

Alimentation :

On a utilisé un transformateur (220v-12v)

Afficheur LCD (16x2) :



*Figure II.5 : Vue réelle sur le moteur EMG30.*

L'afficheur LCD ("Liquid Crystal Display") est en particulier une interface visuelle, Son rôle est de transmettre les informations utiles d'un système à un utilisateur. Il affichera donc des données susceptibles d'être exploiter par l'utilisateur d'un système.

Alimentation: 5 Vcc

Interface I2C

Contraste ajustable via potentiomètre

Dimensions: 82 x 35 x 18 mm

#### 4. Moteur :

L'EMG30 est un actionneur largement utilisée dans les applications robotique petites ou moyennes. C'est un moteur 12v dotés d'un encodeur effet de HALL comme capteur de position, avec une précision de 360 impulsions par tour et un facteur de réduction égale à 30.



*Figure II.6 : Vue réelle sur le moteur EMG30.*

##### 4. a. Encodeur à effet Hall :

Les senseurs à effet Hall permettaient de mesurer des champs magnétiques continus ou statiques mais nécessitaient que le senseur soit en mouvement. Il est utilisé comme senseur de position, encodeur, détecteur de rotation.

Principe :

L'effet Hall a été découvert par Edwin Hall en 1879.c' est la production d'une différence de tension  $U$  aux bornes d'un conducteur électrique, transversale à un courant électrique  $I_{dc}$  dans le conducteur est soumis à un champ magnétique  $B$  appliqué perpendiculairement au courant.

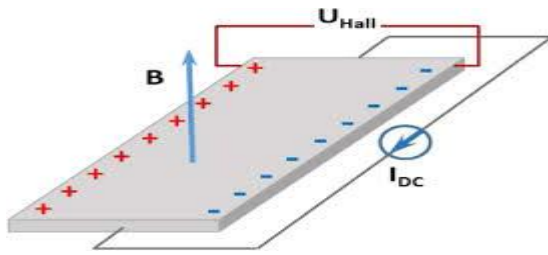


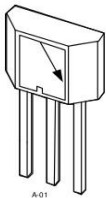
Figure II.7 : principe d'effet de hall.

Le coefficient de Hall est défini comme le rapport du champ électrique induit au produit de la densité de courant et du champ magnétique appliqué. Il s'agit d'une caractéristique du matériau dans lequel est fabriqué le conducteur, car sa valeur dépend du type, du nombre et des propriétés des porteurs de charge qui constituent le courant.

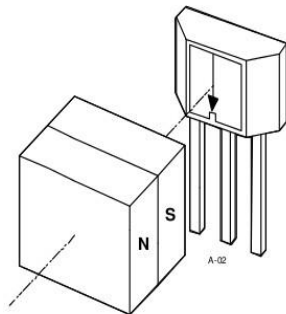
### Fonctionnement du senseur

Le senseur à effet Hall n'est pas activé (OFF) lorsqu'il n'y a pas de champ magnétique. Il s'active (ON) en présence d'un champ magnétique comme présenté sur les deux figures suivantes.

Le champ magnétique terrestre n'influence pas le senseur à effet Hall.



Le senseur est inactif en l'absence de champ magnétique.



Le senseur s'active lorsqu'un **champ magnétique de type SUD** est présenté sur la **face avant du senseur**.



Schéma sous Proteus :

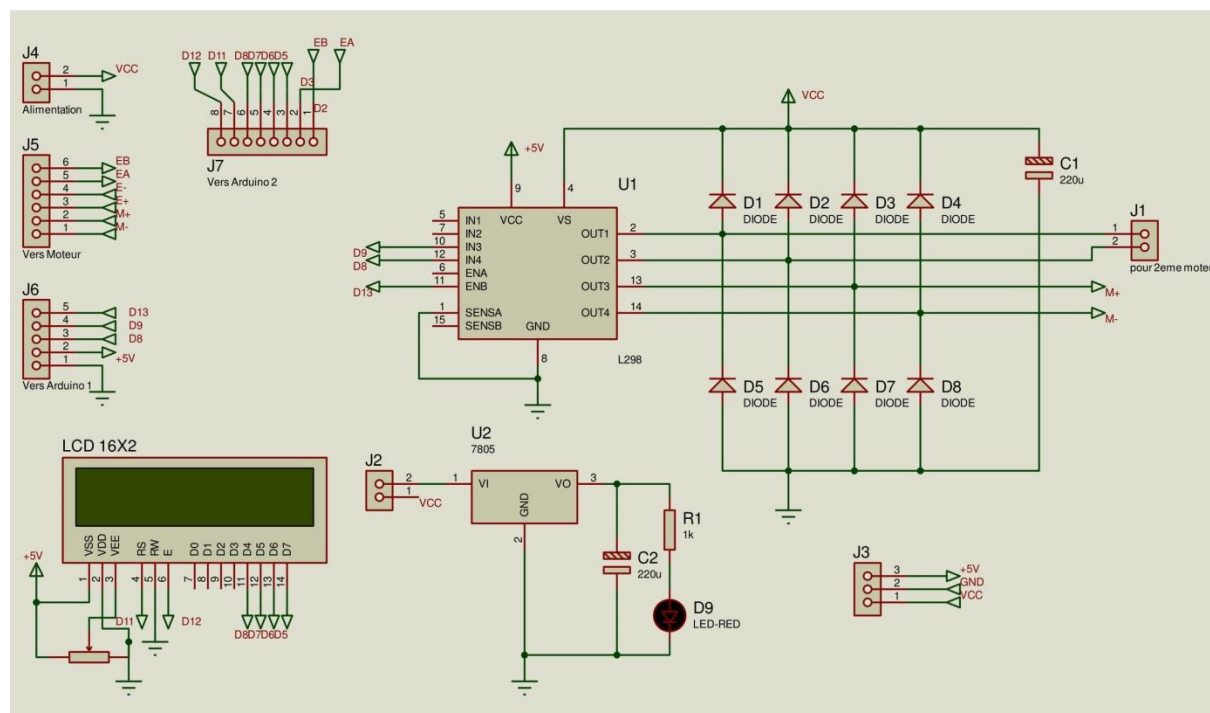


Figure II.8 : Le schéma électrique de l'étage de puissance.

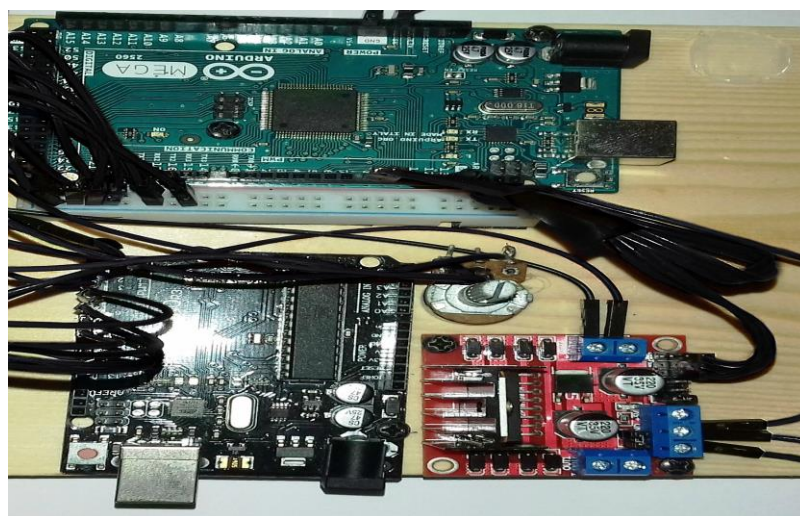
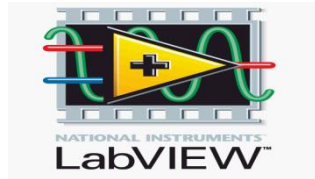


Figure II.9 : Le montage électrique réel de commande du moteur

## Partie 02 : Elaboration du schéma et paramètres de LabVIEW

### 1. Introduction:

LabVIEW (laboratory virtual instrument engineering workbench):



C'est un environnement de développement qu'il s'appuie sur le langage G, créé par National Instruments. Il offre une approche de programmation graphique qui comprend la configuration matérielle, les données de mesure et le débogage. Cette visualisation simplifie l'intégration du matériel de mesure de n'importe quel fournisseur. Il permet de créer des logiciels complexes. Grâce à ses bibliothèques de fonctions dédiées à l'acquisition de données, l'instrumentation, à l'analyse mathématique des mesures.

### 2-élaboration du block diagramme sur LabVIEW :

Avant de connecter notre carte arduino avec LabVIEW il faut installer le pilote linux sur LabVIEW pour effectuer la communication entre arduino et LabVIEW.

LINX fournit des VIs LabVIEW faciles à utiliser pour interagir avec les plates-formes intégrées courantes telles qu'arduino, chip KIT et myRIO. On peut transférer des données sur notre PC en quelques secondes ou utiliser les VIs de périphériques pour accéder aux E / S numériques, E / S analogiques, SPI, I2C, UART, PWM et autres de nos périphériques.

Le block diagramme contient les fonctions, les constantes, les structures et les fils... les données. Après avoir créé la fenêtre du front panel, on ajoute du code à l'aide de la représentation graphique des fonctions qui permettant de contrôler les objets.

Le schéma fonctionnel du système en boucle fermée est présenté ci-dessous:



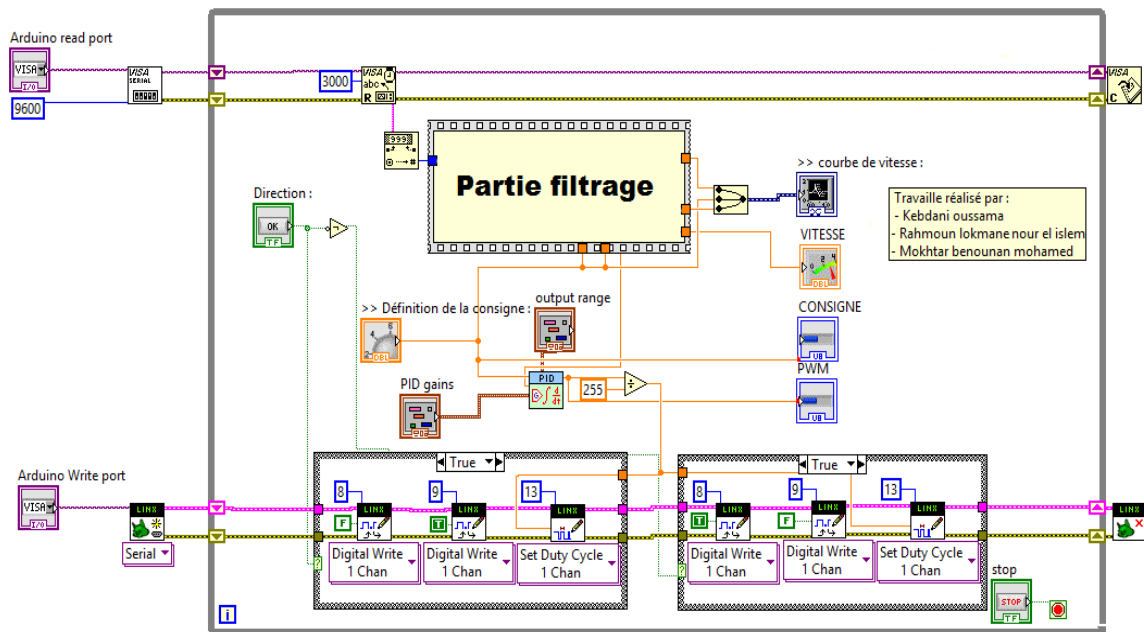


Figure II.10 : Schéma Bloc du système en boucle ouverte.

Ce block contient une partie pour définir les entrées/sortie, Une partie commande qui consiste à donner un signal pwm pour la commande du moteur et une partie filtrage pour avoir une réponse stable sans perturbation.

### 3-la réalisation d'une interface graphique sous LabVIEW :

LabVIEW donne une grande simplicité pour réaliser n'importe quelle interface graphique, on a réalisé cette interface dans le but de faciliter l'étude de la réponse du notre moteur.

On peut déviser notre interface sur trois partie :

- Une partie paramètres pour le choix des entrées /sortie de la carte.
- Une deuxième pour le choix de la consigne et des paramètres de régulation.
- Une dernière partie pour visualiser en temps réel les réponses du système : vitesse, consigne, pwm.

La manipulation sur cette interface est structurée comme suit :

La première des chausses on va définir les ports d'entré sortie de notre carte arduino,

Ensuite on va définir la valeur de notre consigne dans ce temps-là le moteur commence à tourner et on règle les paramètres de notre régulateur pid on va obtenir la réponse désiré.

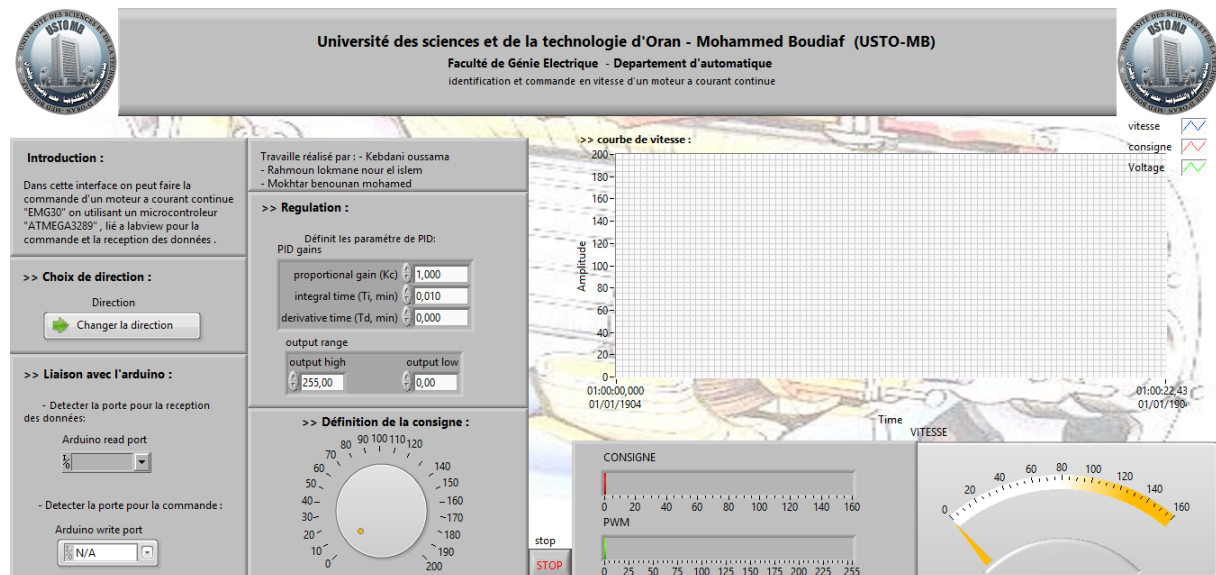


Figure II.11 : Interface graphique sous LabVIEW.

## Troisième chapitre

Identification et régulation de la réponse indicielle du moteur, par un correcteur  
« Proportionnelle intégrale dérivée ».

---

### Partie 01 : Etude de la réponse indicielle du moteur.

#### 1. Introduction :

Dans ce chapitre on va étudier la réponse en vitesse du notre moteur et faire son identification et sa régulation dans le but d'obtenir les paramètres du notre régulateur pid .a la fin on va afficher la vitesse réguler sous l'afficheur lcd.

Dans cette partie on va faire l'identification des paramètres du moteur et la régulation de ca réponse sur la plate-forme LabVIEW.

#### 2-méthode de STREJC :

La méthode de strejc nous permettons d'identifier notre modèle à partir de la réponse à un échelon. La méthode d'identification peut donc être synthétiser comme suit. A partir de la réponse à un échelon, on identifie dans un premier temps le gain statique  $k = \frac{Y}{U}$  où U est l'amplitude de l'échelon d'entrée et Y est l'amplitude de la variation de la mesure. Dans un second temps, on doit identifier l'ordre n du système et la constante de temps .Pour cela, on

trace la tangente à la courbe au point d'inflexion. On relève le temps T1 entre l'instant de déclenchement de l'échelon et l'instant où l'asymptote coupe la valeur initiale de y(t). On relève T2 entre ce dernier instant et l'instant où l'asymptote atteint la valeur finale de y(t). A partir du rapport  $\frac{T1}{T2}$ , on détermine l'ordre du système. Pour la valeur de n choisie, on détermine C à partir de la valeur de  $\frac{T1}{C}$  ou de  $\frac{T2}{C}$ .

Tout se travaille est réaliser sous Matlab. On règle la vitesse à 20tour/min et on obtient  
Après l'identification la réponse suivante :

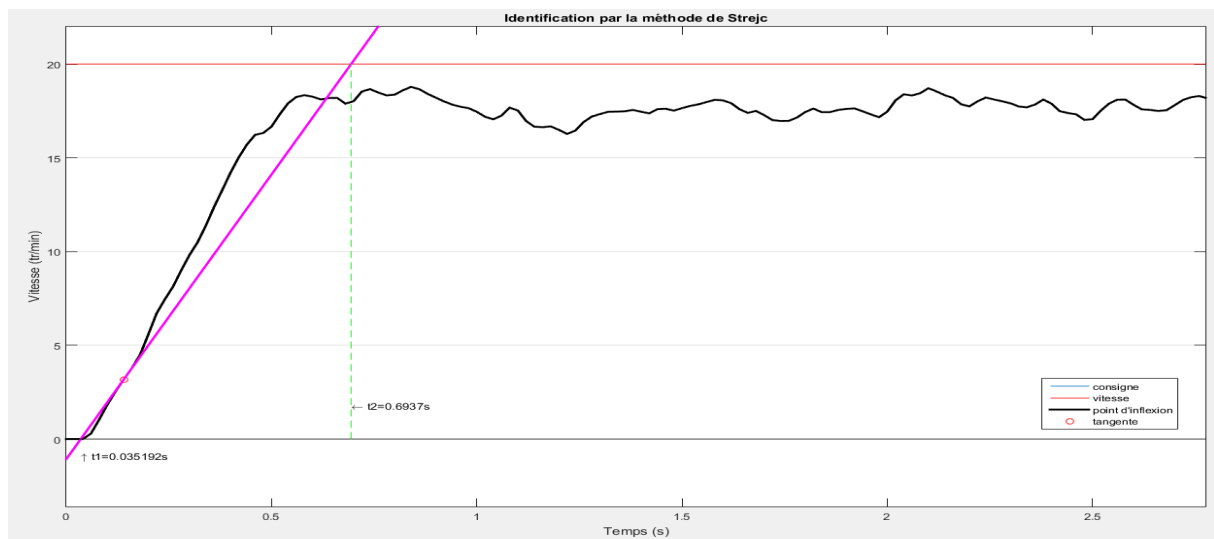


Figure III.1 : Identification par la méthode de STREJC.

Les paramètres obtenu :-t1=0.6937s.

-t2=0.035192s.

La fonction de transfert estimé :  $F(s) = \frac{0.9}{0.6585s + 1}$

Avec un retard de 0.035192s.

Identification par broida :

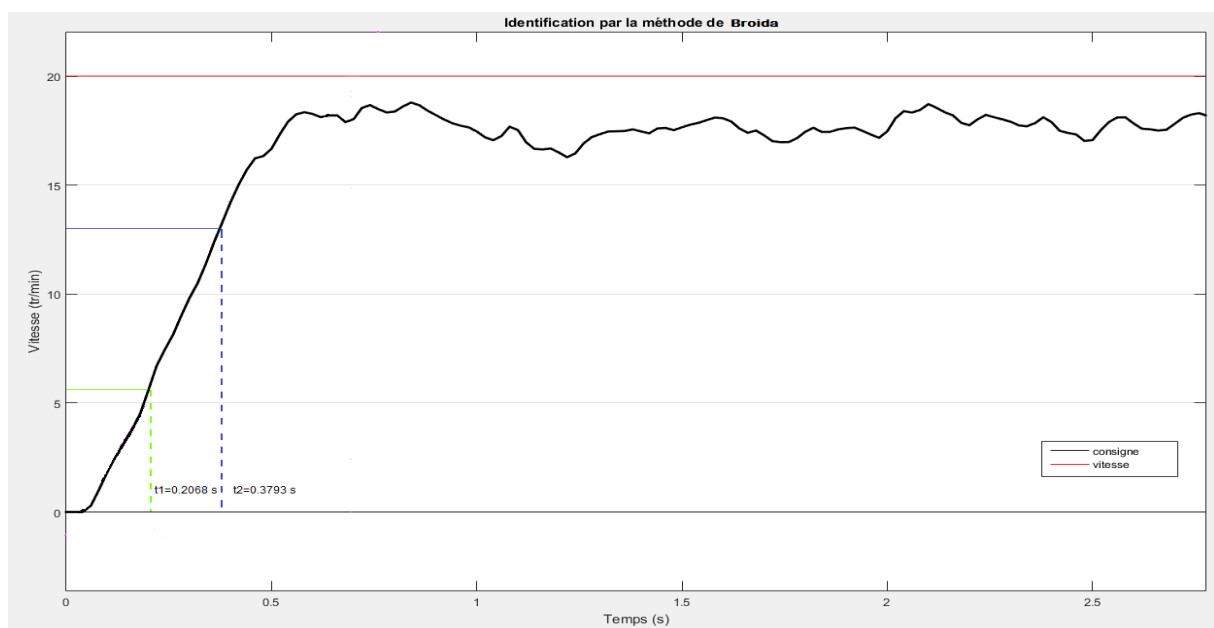


Figure III.2 : Identification par la méthode de Broida.

### 3-régulateur pid (proportionnel-intégral-dérivé) :

C'est un contrôleur largement utilisé dans les systèmes de contrôle industriels et dans diverses autres applications nécessitant un contrôle modulé en continu. Un contrôleur PID calcule en permanence une valeur d'erreur  $e(t)$  en tant que différence entre une valeur de consigne souhaitée (SP) et une variable de processus mesurée (PV) et applique une correction basée sur la proportionnelle, les termes intégraux et dérivés (notés respectivement P, I et D).

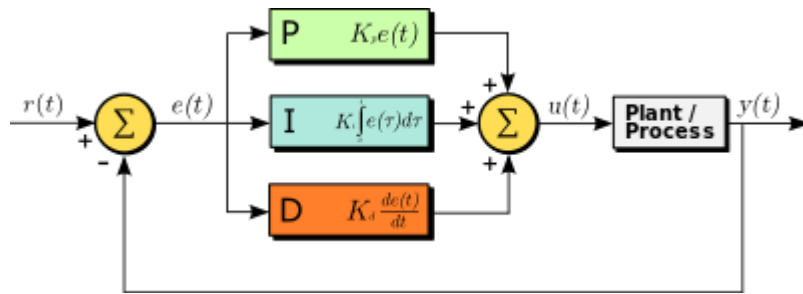


Figure III.3 : régulateur pid dans une boucle fermée.

#### 3-a-Caractéristiques des contrôleurs P, I et D :

Un contrôleur proportionnel ( $K_p$ ) aura pour effet de réduire le temps de montée et réduira mais n'éliminera jamais l'erreur en régime permanent. Une commande intégrale ( $K_i$ ) aura pour effet d'éliminer l'erreur d'état d'équilibre, mais elle risque d'aggraver la réponse transitoire. Un contrôle dérivé ( $K_d$ ) aura pour effet d'accroître la stabilité du système, de réduire les dépassements et d'améliorer la réponse transitoire. Les effets de chacun des contrôleurs  $K_p$ ,  $K_d$  et  $K_i$  sur un système en boucle fermée sont résumés dans le tableau ci-dessous :

paramètres	TEMPS DE MONTÉE	dépassement	Temps d'établissement	erreur
$K_p$	démunie	Augmente	Petit changement	démunie
$K_i$	démunie	Augmente	augmente	éliminé
$K_d$	Petit changement	démunie	démunie	Petit changement

#### 4-régulation sous LabVIEW :

Avant de faire notre régulation on a obtenu une réponse perturbé ce qui nécessite d'utiliser plusieurs filtre pour ajuster notre réponse.la réponse perturbé obtenue est comme celle-ci :

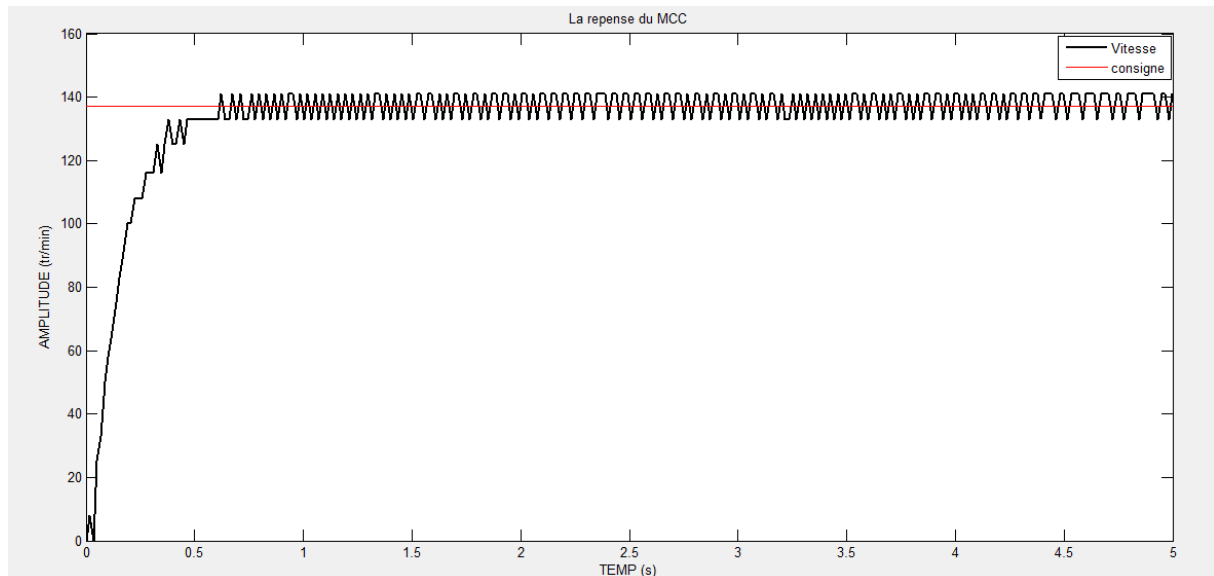


Figure III.4 : réponse perturbé du moteur

Après le filtrage sous LabVIEW on obtient la réponse suivante :

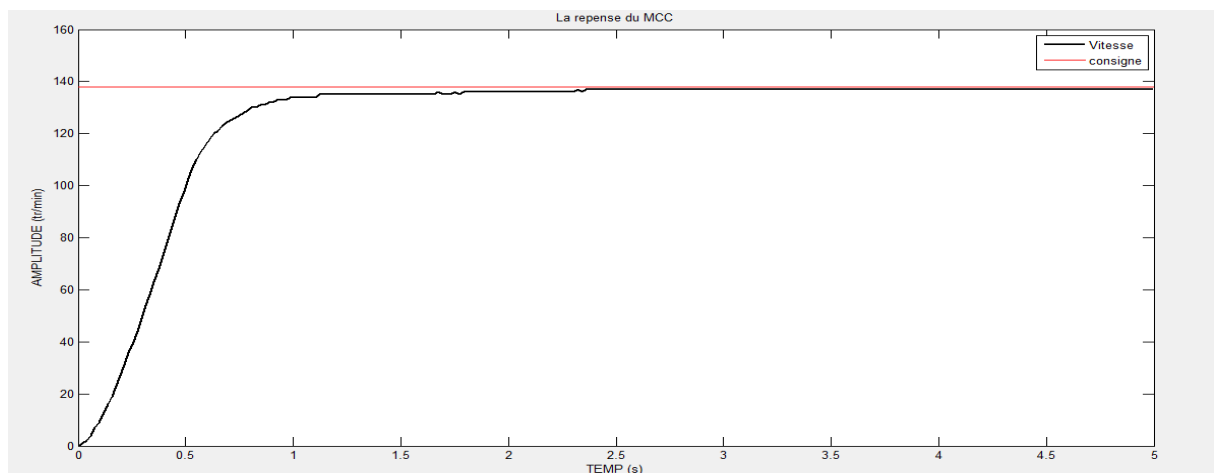


Figure III.5 : réponse du moteur après filtrage.

#### 5-Calcul des paramètres du correcteur (pid) :

Pour calculer nos paramètres on va utiliser la méthode de pompage, Cette opération est effectuée en réglant les gains I (intégral) et D (dérivé) à zéro. Le gain "proportionnel",  $K_p$  est ensuite augmenté (à partir de zéro) jusqu'à atteindre le gain critique  $K_{cr}$  où la sortie de la

boucle de régulation présente des oscillations stables et cohérentes. La période d'oscillation  $T_{cr}$  permettent de définir les gains P, I et D en fonction du type de contrôleur utilisé.

Après le pompage on obtient cette réponse :

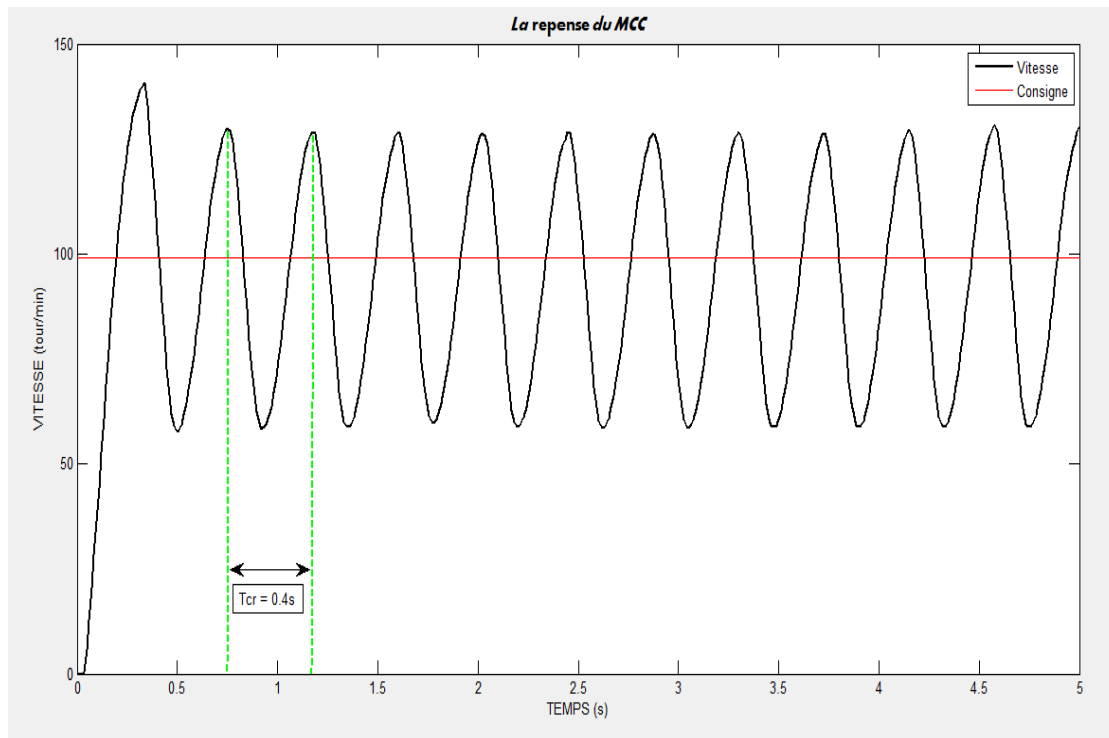


Figure III.6 : réponse du moteur après pompage.

Voici ci-dessous le gain critique  $K_{cr}$  du régulateur ainsi que la période d'oscillation  $T_{cr2}$  de la réponse :

$$K_{cr}=4 \quad , \quad T_{cr}=0.4s$$

On peut maintenant calculer les paramètres du régulateur  $K_p$  et  $K_i$  à l'aide d'un tableau spécial des algorithmes PID sur LabVIEW :

Controller	$K_p$	$T_i$	$T_d$
P	$0.44T_p/\tau$		
PI	$0.4T_p/\tau$	$5.33\tau$	
PID	$0.53T_p/\tau$	$4.0\tau$	$0.8\tau$

Tab 1 : Paramètres PID obtenus à partir du point critique

Et on obtient :

$$Kp = 1$$

$$Ti = 0.01$$

$$Td = 0.002$$

Testons l'efficacité de cette régulation en simulant la réponse indicielle du système en boucle fermée. La régulation sous LabVIEW, conduit à la réponse indicielle en vitesse suivante :

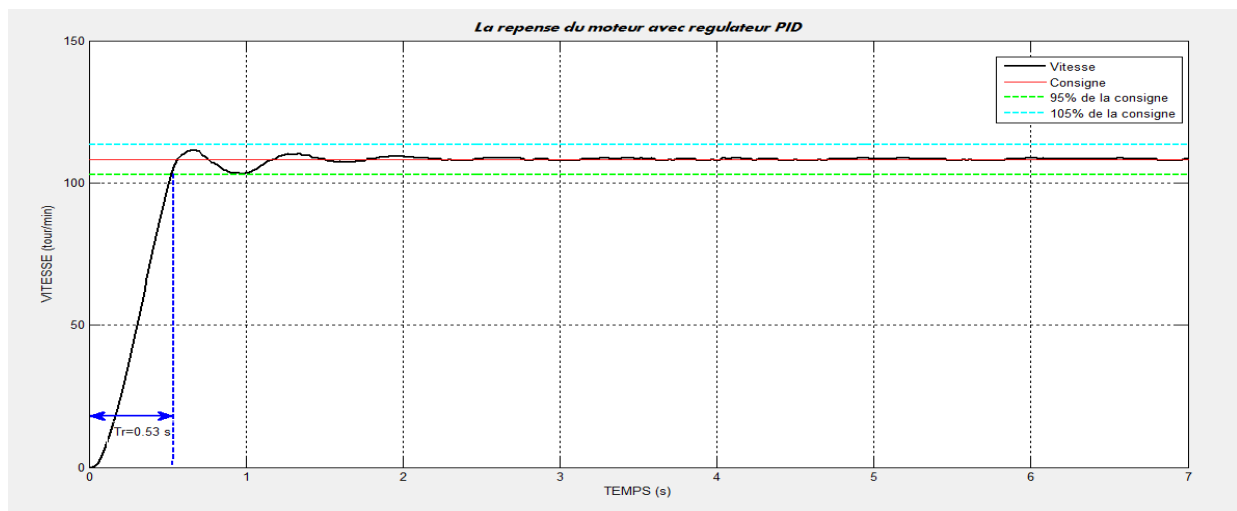


Figure III.7 : réponse du moteur avec régulateur pid.

Pour tester l'efficacité de notre régulateur PID on va introduire des perturbations extérieures et on va visualiser notre réponse, la réponse obtenue et la suivante :

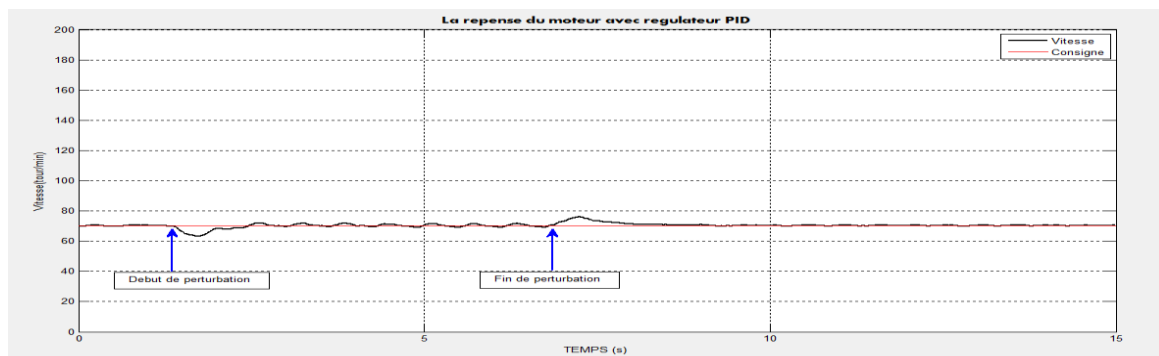


Figure III.6 : réponse du moteur en introduisons des perturbations extérieur.

On peut bien voir que lors de l'applique d'une force la vitesse va diminuer par rapport la consigne mais elle va rapidement revenir à la consigne donnée, pour cela on peut conclure que notre régulateur est efficace.



### 6-Affichage de la vitesse:

Nous devons d'abord activer le fichier d'en-tête (`#include <LiquidCrystal.h>`). Ce fichier d'en-tête contient des instructions qui permettent à l'utilisateur d'interfacer un écran LCD à UNO en mode 4 bits. Avec ce fichier d'en-tête, nous n'avons pas besoin d'envoyer des données à l'écran LCD, nous nous en occupons et nous n'avons pas besoin d'écrire un programme pour l'envoi de données ou une commande à l'écran LCD.

Deuxièmement, nous devons indiquer au tableau le type d'écran LCD que nous utilisons ici. Puisque nous avons tellement de types différents d'affichage à cristaux liquides (comme 20x4, 16x2, ...). Ici, nous allons connecter un écran LCD 16x2 à l'ONU afin d'obtenir `'lcd.begin (16, 2);'`. Pour 16x1, nous obtenons `«lcd.begin (16,1);»`.

Dans cette instruction, nous allons dire au tableau où nous avons connecté les broches. Les broches qui sont connectées doivent être représentées dans l'ordre suivant "RS, En, D4, D5, D6, D7". Ces broches doivent être correctement représentées. Comme nous avons connecté RS à PIN0 et ainsi de suite, comme indiqué dans le schéma de circuit, nous représentons le numéro de broche à carte comme suit: `«LiquidCrystal lcd (13, 12, 5, 4, 3, 2);»`. Les données qui doivent être affichées sur l'écran LCD doivent être écrites comme suit: `"lcd.print (" ... , ..... ");"`.

Lors de l'affichage de la valeur de notre vitesse sur le lcd on a obtenu un affichage instable pour cela on a rajouté le filtre de kalman sur notre programme pour obtenir un affichage stable.

## Conclusion générale

Dans notre mémoire, on présente l'étude du moteur à courant continu, puis la technique de la variation et l'affichage de la vitesse, et enfin sa régulation.

Dans le premier chapitre, nous avons rappelé le principe du fonctionnement de la machine à courant continue et ses constituants ainsi que le modèle mathématique de notre moteur.

Dans le deuxième chapitre, nous avons préparé la maquette d'acquisition et de commande composée de deux cartes ARDUINO, d'un driver, afficheur LCD ainsi que le moteur. Puis nous avons élaboré notre schéma bloc LABVIEW pour le contrôle et la communication avec la carte.

Cette réalisation nous a permis de :

- Montrer que la vitesse de rotation d'un moteur à courant continu dépend de sa tension
- Montrer l'utilité de LABVIEW dans l'acquisition des données et de son traitement avec une grande simplicité avec un traitement de données en temps réel.

Dans le troisième chapitre, après simulation de la réponse indicielle du moteur, nous avons utilisé deux méthodes afin d'identifier les paramètres du moteur, puis nous avons placé notre système en boucle fermée avec un correcteur PID,

Cette régulation nous a permis d'obtenir des performances respectant un cahier de charges imposé.

Les plateformes virtuelles du LabVIEW étaient très pratiques.

L'utilisation du driver nous a permis d'obtenir une valeur fiable de la tension.

L'interface graphique nous a facilité l'étude et la régulation du moteur.

La commande de moteur multiple est réalisée par L298N via Arduino et LabVIEW.

LabVIEW offre une meilleure interface avec les moteurs de la série dc, Arduino et L298N.

Un autre avantage supplémentaire de LabVIEW est qu'il peut être utilisé aussi bien comme dispositif d'affichage que de contrôle dans le système.

## Annexe

### Généralité sur le moteur étudié EMG30 :

Le

EMG30 (codeur, moteur, boîte de vitesses 30 : 1) est un moteur 12v équipé d'un encodeur et d'une boîte de réduction de 30:1. Il est idéal pour les applications robotiques petites ou moyennes, offrant un entraînement et une rétroaction rentables pour l'utilisateur. Il comprend également un condensateur de suppression de bruit standard à travers les enroulements du moteur.

### Connecteur

Les fils de connexions du moteur sont :

- Violet : Encodeur « Pin B » (Vout)
- Bleu : Encodeur « Pin A » (Vout)
- Vert : Encodeur GND (-)
- Brun : Encodeur VCC (+)
- Rouge : Moteur (+)
- Noir : Moteur (-)
- Les couleurs des fils sont du câble réel.
- Le capteur à effet hall accepte des tensions entre 3.5v et 20v.
- Les sorties du capteur sont à collecteur ouvert et nécessitent des résistances de pull-up pour le niveau (voltage) du signal de sortie souhaité.

Dans notre montage, ils sont alimentés à partir de 12v et tiré jusqu'à 5v (fournie par l'ARDUINO) pour les signaux de A et B.

### Spécification

Tension nominale : 12 V

Couple nominal : 1.5 kg / cm

Vitesse nominale : 170 tours/min

Courant nominal : 530 mA

Vitesse à vide : 216 tours/min

Puissance nominale : 4.22W

Compteur de l'encodeur par 1 tour de l'arbre de sortie : 360 comptes.

## Bibliographie

- [1] BENNOURA ABDEL MADJID et DJABRI KHALED mémoire de Master  
Thème : « Etude et simulation d'un variateur de vites d'une Machine à courant continu »  
Universitaire Belhadj Bouchaib d'Ain-Temouchent Institut de Technologie Département de  
Génie Electrique 2017
- [2] Documentation sur le pin PWM :  
[www.arduino.cc/en/Tutorial/SecretsOfArduinoPWM](http://www.arduino.cc/en/Tutorial/SecretsOfArduinoPWM)
- [3] Documentation sur le senseur effet de Hall  
[wiki.mchobby.be/index.php?title=Senseur %C3%A0 Effet Hall](http://wiki.mchobby.be/index.php?title=Senseur_%C3%A0_Effet_Hall)
- [4] Olden. P, "Open Loop motor speed control with LabVIEW", SoutheastCon,  
Proceedings IEEE, pp. 259-262, 2001.
- [5] "PID manual LABVIEW"
- [6] BOUZIANE ABDERRAHIM MOHAMED et BENMAAMAR LAHOUARIA  
LYDIA et INAL NESRINE AMINA mémoire de licence  
  
Thème : CONTROLE D'UN MOTEUR A COURANT CONTINU  
PAR « ARDUINO » ET « MATLAB/SIMULINK »

