

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي و البحث العلمي

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

جامعة سعد دحلب البليدة
UNIVERSITE SAAD DAHLEB DE BLIDA



كلية الهندسة - دائرة الإلكترونيك
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE

MEMOIRE DE PROJET DE FIN D'ETUDES

POUR L'OBTENTION DU DIPLÔME DE
MASTER EN ELECTRONIQUE

OPTION : AUTOMATIQUE

*REGULATION DE NIVEAU D'EAU DANS
UN RESERVOIR ASSISTE PAR
LE LOGICIEL LABVIEW*

Présenté par :

**BENSAADI Ali.*

**ABBAS Saad.*

Proposé par :

**Melle. CHENTIR Amina*

Année Universitaire : 2010 – 2011.



dedicace

Je dédie ce modeste travail :

- ❖ *A mes très chers parents*

- ❖ *A mes sœurs, et frères.*

*Et à ma promotrice **Melle. CHENTIR AMINA** qui m'a aidé pendant toute l'année.*

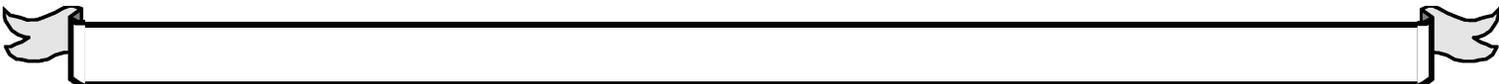
A ma famille

*A mon binôme **ALI***

A mes très chers amis

A tous ceux et celles qui ont contribué de près ou de loin dans la réalisation de ce travail.

SAAD.





dedicace

Je dédie ce Modeste travail

A :

Mes chers parents pour leurs soutiens et leur réconfort

A mes frères et sœurs

*A toute la famille **Bensaadi***

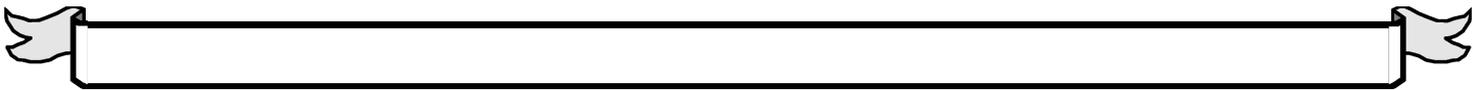
A tous mes amis

A tous mes collègues de la promotion

A toutes les personnes qui me connaissent de près ou de loin...

Et en fin à tous ceux que j'aime.

ALI.





REMERCIEMENTS

*Nous remercions **ALLAH** tout puissant qui nous a donné le courage, la force et la volonté pour réaliser ce modeste travail.*

*Nous remercions tout particulièrement **Melle. CHENTIR AMINA**, pour la proposition de ce sujet, son encadrement, sa disponibilité et son soutien dans notre recherche d'informations et ses conseils lors de l'élaboration du projet.*

*Nous remercions aussi l'ensemble du personnel du laboratoire **LATSI** pour nous avoir permis de réaliser ce projet dans de si bonnes conditions.*

Enfin, nous remercions tous ceux qui ont contribué de près ou de loin, par leurs aides et conseils, à finaliser ce projet.

SAAD & ALI



ملخص

الهدف من مشروع التخرج لهذه الأطروحة هو تنظيم منسوب المياه في خزان. التنظيم يكون بمراقبة مستوى المياه عن طريق جهاز كمبيوتر. تم انجاز ثلاث بطاقات: بطاقة الواجهة بواسطة متحكم « PIC16F877A » ، بطاقة لقياس منسوب المياه و بطاقة للتحكم في المضخة. استعملنا الناقل « RS232 » للربط بين الكمبيوتر و البطاقات التي تم انجازها. وذلك بمساعدة برنامج كتب بواسطة « LABVIEW » ليقوم بعملية تنظيم الجهاز.

Résumé

Ce projet de fin d'étude a pour but de faire une régulation de niveau d'eau dans un réservoir. La régulation consiste à contrôler le niveau d'eau par un ordinateur. Trois cartes ont été réalisées : une carte d'interface à base de « PIC 16F877A », une carte de mesure de niveau d'eau et une carte de commande de la pompe. Nous avons utilisé le bus « RS232 » pour faire la liaison entre le pc et les cartes réalisées et ceci à l'aide d'un programme écrit sous « LABVIEW » pour faire la régulation du procédé.

Abstract

This final project study aims to control a water level in a tank .The regulation is to control the water level by a computer. Three cards were produced: an interface card based on a « PIC 16F877A », a card of water level measurement and a control board of the pump. We used field bus « RS232 » to realize the connection between the PC and the produced cards. All this, using a program written in LABVIEW software to control the process.

INTRODUCTION GENERALE

L'automatique est la science qui traite de l'analyse et de la commande des systèmes dynamiques évoluant avec le temps. En d'autres mots, de l'automatisation de tâches par des machines fonctionnant sans intervention humaine [1]. Le système de la commande peut fonctionner en boucle ouverte à partir d'un signal d'entrée. Cependant, c'est uniquement en boucle fermée qu'on est capable de stabiliser, d'améliorer les performances et de rejeter les perturbations externes des systèmes dynamiques. La loi de commande est générée par un système de commande qu'on appelle correcteur ou régulateur.

De nos jours, grâce aux développements de l'électronique et de l'informatique, la plupart des lois de commande sont implémentées sur des micro-ordinateurs ou processeurs numériques. L'implémentation d'un algorithme de commande sur l'ordinateur en comparaison à une réalisation analogique, offre de nombreux atouts : coût faible, précision élevée, insensibilité au bruit et facilité d'implémentation et souplesse par rapport aux modifications.

Notre travail se place dans ce contexte. Il consiste à réguler le niveau d'eau dans un réservoir assisté par le logiciel LabVIEW. Pour ce faire, une structure de commande basée autour d'un microcontrôleur PIC16F877A et la communication entre les différents organes opératifs sont mis en œuvre.

Le système consiste de deux réservoirs, un réservoir bas c'est la source d'eau et un autre haut pour le remplissage, ceci est gradué avec une règle pour voir le niveau à chaque instant du déroulement de la commande utilisée.

L'opération de remplissage se fait grâce à une pompe électrique commandée, cette dernière contient un moteur à courant continu, sa tension de commande varie de 0 à 12V.

L'objectif de notre projet est donc de réaliser l'interface entre la partie opérative et la partie commande. Cette interface devra contenir le capteur effectuant les mesures nécessaires à la commande du moteur. Elle devra aussi contenir les filtres anti-repliement. Ainsi, bien sur, qu'une alimentation délivrant les tensions nécessaires au fonctionnement de cet ensemble.

La conception de cette interface devra donc passer par une étape d'étude des éléments de mesure afin de répertorier les alimentations nécessaires, ainsi que leur niveau de sortie.

L'implémentation des différents types de régulation : régulation PID et régulation PID flou a été réalisée en utilisant le logiciel LABVIEW.

générale

Autrement dit, ce projet s'articule autour de deux aspects :

Premièrement, l'aspect matériel qui consiste à développer des cartes électroniques capables de gérer les différents organes du système.

Deuxièmement, l'extension du logiciel de contrôle et de commande.

Pour cela, notre mémoire est organisé en quatre chapitres :

- le premier chapitre traite des généralités sur la commande numérique des systèmes ;
- le deuxième chapitre est consacré à la structure du microcontrôleur PIC 16F877A;
- Le troisième chapitre présente la description de la maquette de régulation de niveau d'eau, le protocole de la transmission série et le logiciel de programmation (LABIEW);
- Le quatrième chapitre sera réservé à la réalisation des différents modules et aux tests;
- Et à la fin, une conclusion viendra clôturer notre mémoire.

SOMMAIRE

REMERCIEMENTS

DEDICACES

RESUMES

LISTES DES FIGURES ET TABLEAUX

LISTE DES ABREVIATIONS

SOMMAIRE

INTRODUCTION GENERALE..... 1

CHAPITRE I : GENERALITES SUR LA COMMANDE NUMERIQUE

I-1 Introduction.....	3
I-2 Définition.....	3
I-3 Eléments constitutifs d'une boucle de régulation.....	3
I-4 Principe de régulation.....	3
I-4-1 Comportement en régulation.....	4
I-4-2 Comportement en asservissement.....	4
I-5 Caractérisation d'une régulation.....	5
I-6 Principaux types de boucles de régulation.....	5
I-6-1 Régulation Boucle ouverte.....	6
I-6-2 Régulation Boucle fermée.....	6
I-6-3 Régulation en cascade.....	7
I-6-4 Régulation mixte (boucle fermée et boucle ouverte).....	7
I-6-5 Régulation de rapport.....	8
I-7 Structure et comportement des processus.....	8
I-8 Modèles de procédés.....	9
I-8-1 Méthode rapide pour un procédé intégrateur.....	9
I-9 Principe général de la correction d'un système.....	10
I-10 Les caractéristiques du régulateur PID.....	10
I-10-1 Action Proportionnelle.....	11
I-10-2 Action Intégrale.....	12

I-10-3 Action Dérivée.....	12
I-10-4 Action Proportionnelle-Intégrale (PI).....	12
I-10-5 Action Proportionnelle-Intégrale-Dérivée (PID).....	12
I-11 Le régulateur FLOU.....	13
I-11-1 Historique.....	13
I-11-2 Eléments de base.....	14
a) La variable linguistique.....	14
b) Univers de discours.....	14
c) La fonction d'appartenance.....	14
d) Ensemble flou.....	14
I-11-3 Opérateurs de la logique floue.....	14
a) Union floue.....	14
b) L'intersection floue.....	15
c) Compilateur flou.....	15
I-11-4 Structure d'une commande floue.....	15
a) Base de données.....	15
b) Fuzzification.....	16
c) Défuzzification.....	16
d) La logique de décision.....	16
I-12 Conclusion.....	16

CHAPITRE II : ETUDE DE PIC 16F877A

II-1 Introduction.....	17
II-2 Définition du PIC.....	17
II-3 Les différentes familles de PIC.....	17
II-4 Identification d'un PIC.....	17
II-5 Descriptions générales du PIC 16F877A.....	18
II-6 Les mémoires.....	19
II-6-1 Mémoire programme.....	19
II-6-2 Mémoire EEPROM.....	19
II-6-3 Mémoire RAM.....	19

II-7 Les broches de fonctionnement.....	20
II-7-1 Les broches d'alimentation.....	20
II-7-2 Les broches de quartz.....	20
II-7-3 La broche de réinitialisation.....	21
II-7-4 Les ports d'entrées/ sorties.....	21
a) Port A.....	21
b) Port B.....	21
c) Port C.....	22
d) Port D.....	22
e) Port E.....	22
II-8 Le convertisseur analogique numérique du PIC 16F877A.....	22
II-8-1 La quantification.....	22
II-8-2 Le codage.....	23
II-9 Le bus de communication du PIC 16F877A.....	24
II-10 L'environnement de Programmation MPLAP.....	24
II-11 Présentation du programmeur de PIC.....	25
II-12 Conclusion.....	27

CHAPITRE III : PRESENTATION DU SYSTEME DE REGULATION DE NIVEAU D'EAU

III-1 Introduction.....	28
III-2 Description du système de régulation de niveau.....	28
III-2-1 Présentation de la maquette.....	28
III-2-2 Descriptions des deux cartes d'alimentation stabilisée.....	29
III-2-2-1 1 ^{ère} carte d'alimentation.....	29
a) Choix du transformateur.....	30
b) Choix du pont de redressement.....	30
c) Choix des condensateurs.....	30
d) Choix des régulateurs de tension.....	30
III-2-2-2 2 ^{ème} carte d'alimentation.....	31
a) Choix du transformateur.....	31
b) Choix du transistor 2N3055.....	32

III-2-3 Carte de mesure.....	32
III-2-3-1 les composants de la carte de mesure.....	32
a) Le capteur de pression.....	32
b) Réglage d'offset.....	32
c) Amplificateur d'instrumentation.....	33
d) Filtrage.....	33
III-2-3-2 Choix du gain d'amplification.....	34
III-2-3-3 Etalonnage du capteur.....	34
III-2-4 Carte de commande.....	35
III-2-4-1 La pompe.....	35
III-2-4-2 Amplificateur de puissance.....	36
III-2-5 Présentation de la carte d'interfaçage à base du PIC16F877A.....	36
III-2-5-1 Descriptions de la carte.....	36
III-2-5-2 Microcontrôleur PIC 16F877A.....	37
III-2-5-3 Acquisition de la mesure.....	37
III-2-5-4 La conversion numérique analogique.....	38
a) Le convertisseur de courant.....	38
b) Le DAC 0808.....	39
III-2-5-5 Principe de la transmission série (RS232).....	40
a) Brochage de la prise.....	41
b) Protocole de transmission.....	42
c) Principe d'adaptation PIC- RS232.....	44
III-2-6 Commande et régulation dans LABVIEW.....	45
III-2-6-1 Présentation de LABVIEW.....	45
III-2-6-2 Bibliothèque de commandes.....	45
a) La bibliothèque de contrôle PID.....	45
b) La bibliothèque de logique floue.....	46
III-2-6-3 Transmission série dans Labview.....	47
a) Bibliothèque RS232.....	47
III-2-7 Conclusion.....	48

CHAPITRE IV : REALISATION ET TESTS

IV-1 Introduction.....	49
IV-2 Présentation du logiciel PROTEUS.....	49
IV-3 Implémentation des différentes cartes réalisées sur PROTEUS.....	49
a) Cartes d'alimentations.....	50
b) Carte de mesure.....	51
c) Carte d'interface avec la carte de commande.....	51
IV-4 Application de la régulation PID dans Labview.....	52
IV-4-1 Boucle de régulation.....	52
IV-4-2 Configuration des paramètres.....	53
IV-4-3 Résultats expérimentaux.....	54
A) En asservissement.....	54
B) En régulation.....	55
IV-5 Application de la régulation floue dans Labview.....	56
IV-5-1 Boucle de régulation.....	56
IV-5-2 Configuration des paramètres flous.....	56
IV-5-3 Résultats expérimentaux.....	58
A) En asservissement.....	58
B) En régulation.....	59
IV-6 Conclusion.....	60
CONCLUSION GENERALE.....	61
BIBLIOGRAPHIE.....	62
ANNEXES	
NOMENCLATURE	

LISTE DES ABREVIATIONS

P.I.D	Proportionnel Intégral Dérivée
P.I.C	Peripheral Interface Controller
R.I.S.C	Reduced Instructions Set Computer
E.E.P.R.O.M	Electrical Erasable Programmable Read Only Memory
R.A.M	Random Access Memory
C.A.N	Conversion Analogique Numérique
U.S.A.R.T	Universal Synchronous Asynchronous Reciever Transmitter
D.A.C	Digital Analoge Converter
R.S.232	Recommended Standard 232
Lab.V.I.E.W	Laboratory Virtual Instrument Engineering Workbench

I-1 Introduction

Le premier but de l'automatique est la mise au point de systèmes de commande ou asservissements sur des procédés (ou systèmes) [2].

Dans ce chapitre nous allons rappeler les définitions de différents systèmes de commande (boucle ouverte, boucle fermée, boucle en cascade, boucle mixte ...etc.). Puis la structure et le comportement des processus, enfin les caractéristiques des régulateurs PID et flou.

I-2 Définition

La régulation regroupe l'ensemble des techniques utilisées visant à contrôler une grandeur physique soumise à des perturbations. Cette grandeur physique est alors appelée "grandeur réglée"[3]. Comme exemples de grandeurs physiques, on peut citer : la pression, la température, le débit, le niveau etc...

La consigne : est la valeur que doit prendre la grandeur réglée.

I-3 Eléments constitutifs d'une boucle de régulation

Une boucle de régulation doit comporter au minimum les éléments suivants :

- Un capteur de mesure.
- Un transmetteur souvent intégré au capteur.
- Un régulateur.
- Un actionneur.

Elle est souvent complétée par des enregistreurs, des convertisseurs et des sécurités.

I-4 Principe de régulation

Le contrôle de procédé met en œuvre une chaîne d'outils à plusieurs niveaux. Schématiquement, le lien entre le capteur et l'actionneur peut se représenter comme ci-dessous (figure I-1). Le but étant de maîtriser le procédé et, soit de le piloter, soit de l'empêcher d'évoluer en raison des perturbations [4].

Le système se décompose en trois zones : mesurage, algorithme et commande.

- Le mesurage englobe le capteur, l'acquisition par l'automate et le traitement de la mesure.
- L'algorithme correspond à la partie calcul de la commande, en respectant les consignes.
- La commande représente l'écriture sur l'actionneur.

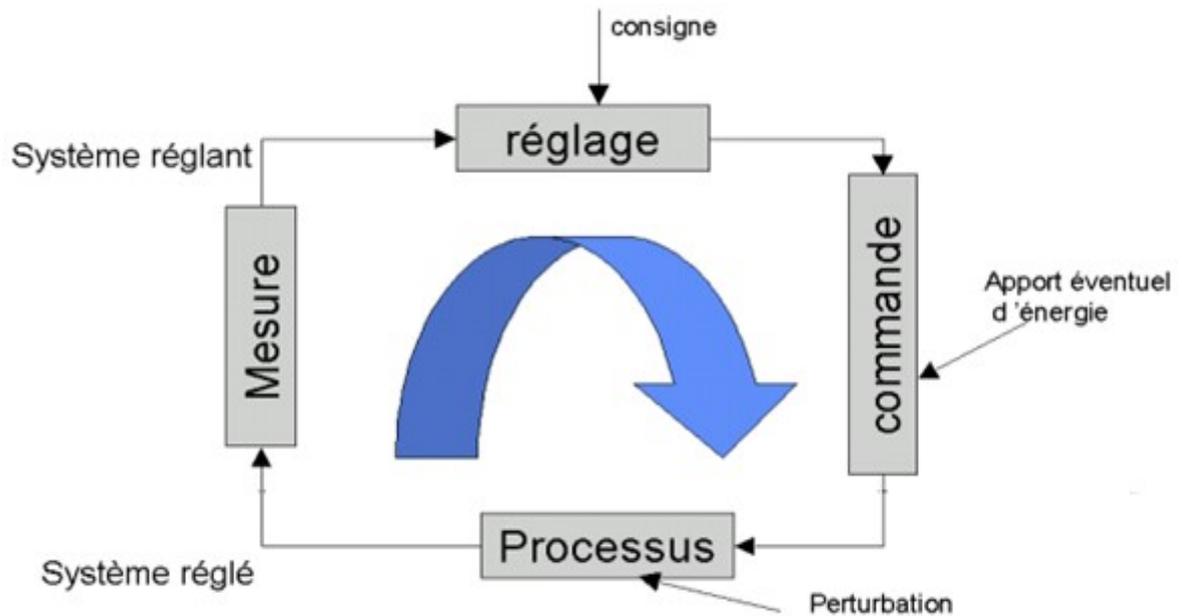


Figure I-1: La chaîne de régulation [4].

Le choix des éléments de la chaîne de régulation est dicté par les caractéristiques du processus à contrôler, ce qui nécessite de bien connaître le processus en question et son comportement.

I-4-1 Comportement en régulation

Le comportement en régulation consiste à minimiser l'influence des perturbations sur la mesure (en régulation, la référence est généralement fixée). Par exemple, réguler la vitesse de rotation d'un moteur (la maintenir constante) quelle que soit la charge (couple résistant) appliquée sur l'axe du moteur (voir figure I-2).

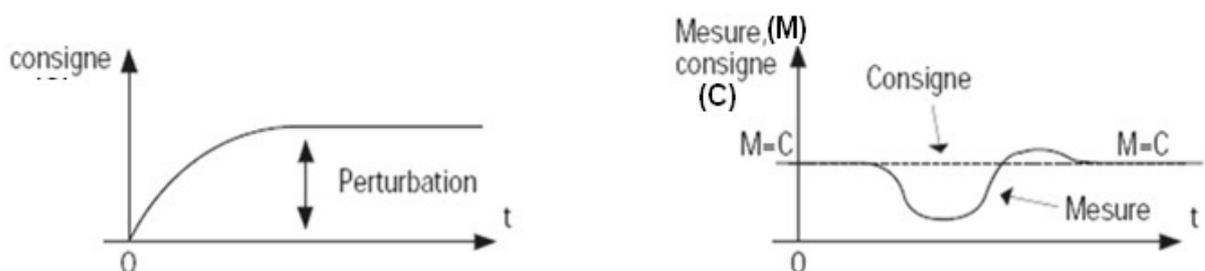


Figure I-2 : Comportement en régulation [5].

I-4-2 Comportement en asservissement

Consiste à faire varier la sortie du procédé par action sur l'entrée de référence (on parle aussi du mode de "poursuite") (figure I-3).

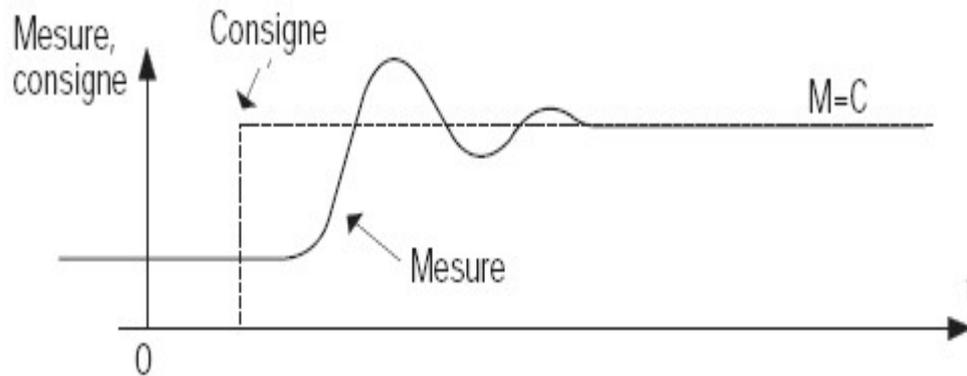


Figure I-3 : Comportement en asservissement [5].

I-5 Caractérisation d'une régulation

Pour qu'une régulation soit correcte, il faut :

- Qu'elle ne mette pas en péril la stabilité du procédé : une instabilité se caractérise par des oscillations excessives.
- Qu'elle assure une bonne précision: l'écart consigne/mesure doit être le plus faible possible.
- Qu'elle corrige rapidement l'influence d'une perturbation : le temps de réponse caractérise l'aptitude de la boucle de régulation à suivre les variations de l'écart consigne/mesure (figure I-4).

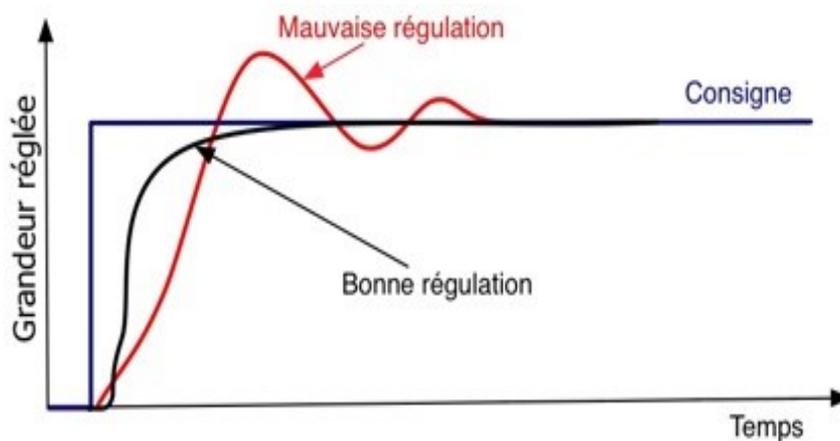


Figure I-4 : caractéristique d'une régulation [3].

I-6 Principaux types de boucles de régulation

Il existe plusieurs types de boucle de régulation, parmi elles :

I-6-1 Régulation Boucle ouverte

La commande en boucle ouverte (figure I-5) d'un système consiste à introduire, à l'entrée de ce système, le signal $e(t)$ permettant d'obtenir à sa sortie, le signal $s(t)$ correspondant à la réponse voulue. Cela nécessite, bien sur la connaissance d'un modèle de fonctionnement du système par exemple, de sa fonction de transfert $G(p)$ [6].

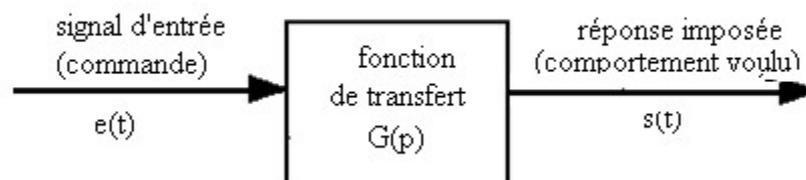


Figure I-5 : Schéma d'un système en Boucle Ouverte [6].

Les inconvénients majeurs de la boucle ouverte sont :

- sensibilité aux perturbations, aux incertitudes sur les paramètres internes ;
- impossibilité de modifier les performances dynamiques ;
- impossibilité de stabiliser un système naturellement instable

I-6-2 Régulation Boucle fermée

Le principe de la commande en boucle fermée (figure I-6) résulte d'une simple question de logique et bon sens : pour mieux maîtriser le fonctionnement d'un système, mesurons son comportement, vérifions que ce comportement correspond bien à ce que l'on attend et utilisons cette information pour adapter le signal de commande [6].

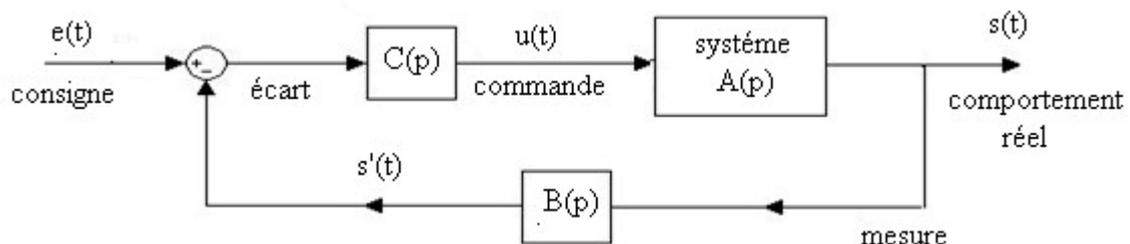


Figure I-6: schéma général d'une boucle fermée [6].

- L'ensemble constitué du système ($A(p)$) et éventuellement du dispositif ($C(p)$) est appelée chaîne directe ou chaîne de commande.

- L'ensemble constitué de la mesure et du dispositif ($B(p)$) est appelée chaîne de retour ou boucle de retour.

Dans certains cas, le dispositif ($B(p)$) peut être inexistant ($B(p)=1$) ; on parle alors de boucle à retour unitaire.

I-6-3 Régulation en cascade

Une régulation en cascade est composée de deux boucles imbriquées. Une mesure intermédiaire est contrôlée par la boucle esclave (figure I-7). La boucle maîtresse contrôle la grandeur réglée de la régulation, sa commande et la consigne de la régulation esclave [7].

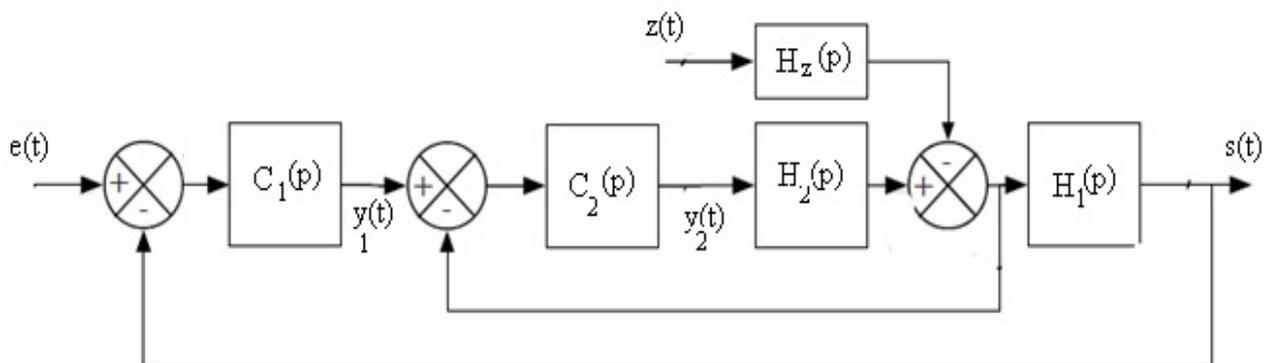


Figure I-7 : exemple de régulation en cascade [7].

I-6-4 Régulation mixte (boucle fermée et boucle ouverte)

Une telle boucle est utile lorsqu'une perturbation a un poids important et que la mesure ne varie pas rapidement suite à cette perturbation. On utilise la mesure d'une perturbation pour compenser ses effets sur la grandeur réglée. Le système peut alors être représenté par la figure I-8:

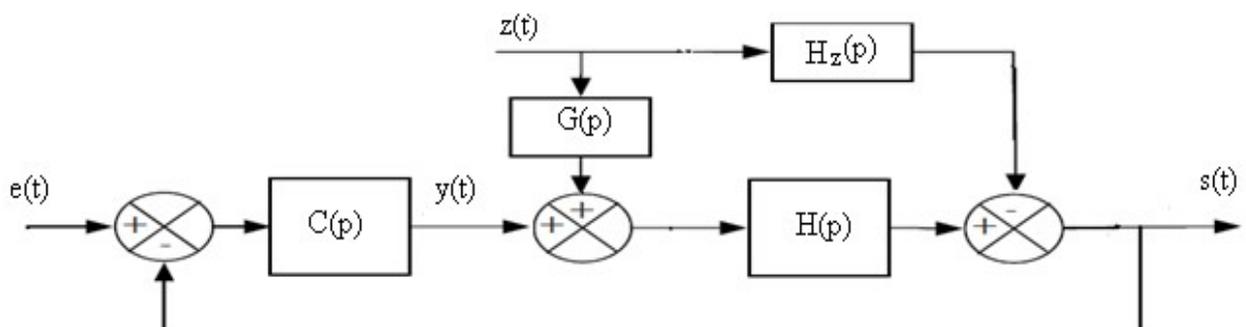


Figure I-8 : exemple de régulation mixte [7].

I-6-5 Régulation de rapport

On utilise une régulation de rapport quand on veut un rapport constant entre deux grandeurs réglées s_1 et s_2 ($s_2/s_1 = \text{constante}$).

Dans l'exemple ci-dessous, la grandeur pilote s_1 est utilisée pour calculer la consigne de la boucle de régulation de la grandeur s_2 (figure I-9) [7].

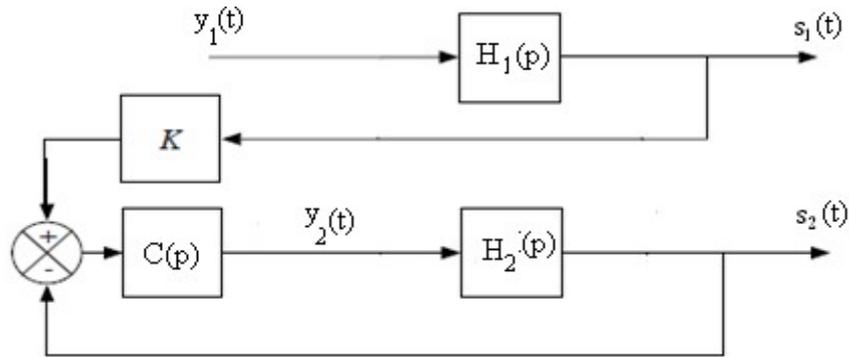


Figure I-9 : exemple de régulation rapport [7].

I-7 Structure et comportement des processus

Si un processus possède une grandeur réglante U et une grandeur réglée S , son comportement peut être représenté soit par une équation différentielle reliant les valeurs de S et de U en fonction du temps, soit par une représentation dite fonction de transfert déduite de la transformation de Laplace appliquée à cette équation différentielle (figure I-10).

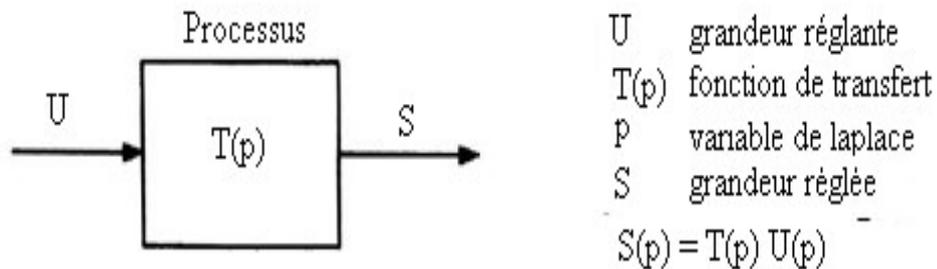


Figure I-10 : fonction de transfert.

Dans ce dernier cas, le comportement du processus est décrit par la relation :

$$S(p) = T(p) U(p) \tag{I-1}$$

Dans laquelle $T(p)$ est la fonction de transfert considérée, p la variable de Laplace.

I-8 Modèles de procédés

Les méthodes de réglage nécessitent souvent de connaître le procédé par sa fonction de transfert. Plusieurs méthodes d'identification en boucle ouverte existent :

- Méthode directe : confrontation de la réponse théorique et expérimentale
- Méthode de Strejc
- Méthode de Broida
- Méthode rapide pour un procédé intégrateur

Pour notre projet, nous nous sommes intéressés à la méthode rapide pour un procédé intégrateur.

I-8-1 Méthode rapide pour un procédé intégrateur

En l'absence de toute perturbation, on envoie un signal d'entrée $u(t)$ connu (impulsion échelon ou rampe) et on enregistre le signal de sortie $y(t)$ qui est analysé ensuite.

La réponse à un échelon d'un procédé intégrateur est une rampe, en régime permanente, L'asymptote de cette réponse est une droite d'équation $y(t) = a(t - t_1)$ de pente a et qui coupe l'axe en t_1 (figure I-11) :

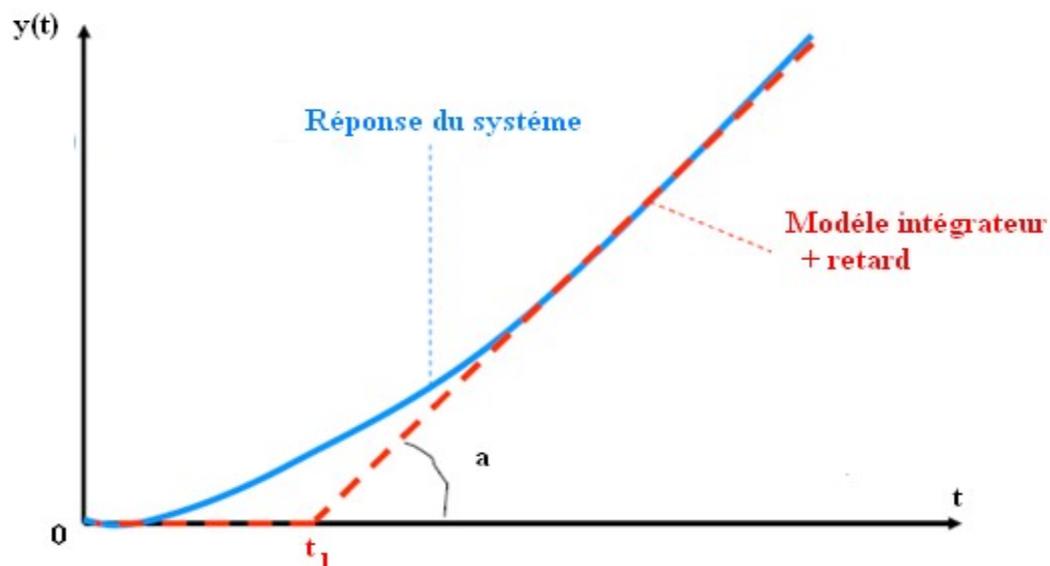


Figure I-11 : réponse à un échelon d'un procédé intégrateur [8].

On identifie la réponse du système réel à la réponse d'un système intégrateur pur avec retard c'est-à-dire de fonction de transfert [8]:

$$H(p) = \frac{ke^{-\tau p}}{p} \quad (I-2)$$

Les paramètres de ce système sont donnés par :

- $\tau = t_1$
- $k = \frac{a}{\Delta u}$ où Δu est l'amplitude de l'échelon appliqué en entrée.

I-9 Principe général de la correction d'un système

L'idée consiste à introduire dans la chaîne directe, en amont du système $A(p)$, un dispositif supplémentaire de fonction de transfert $C(p)$, appelé correcteur et dont le rôle essentiel doit consister à modifier les performances du système initial (figure I-12).

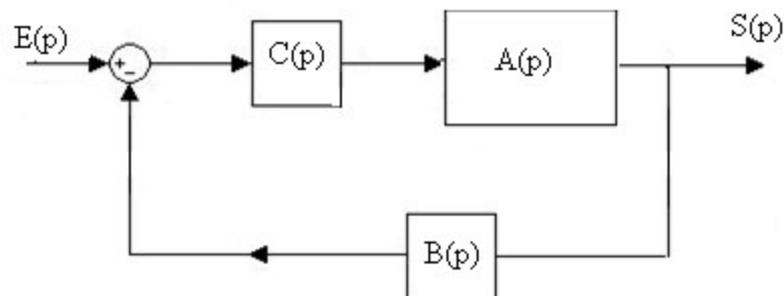


Figure I-12 : Schéma générale d'une boucle de regulation.

I-10 Les caractéristiques du régulateur PID

La commande PID est dite aussi (correcteur, régulateur, contrôleur), se compose de trois termes P, I et D, où le 'P' correspond au terme Proportionnel, 'I' pour le terme Intégral et 'D' pour le terme Dérivée de la commande.

Les régulateurs PID sont probablement les plus largement utilisés dans le contrôle industriel. Même les systèmes les plus complexes de contrôle industriel peuvent comporter un réseau de contrôle dont le principal élément de contrôle est un module de contrôle PID.

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

La PID est insérée dans la chaîne directe de l'asservissement, en série avec le processus, comme indiqué dans la figure I.13. Ce régulateur élabore à partir du signal d'erreur $e(t)$ une commande $u_c(t)$ en fonction de trois actions proportionnelle, intégrale et dérivée.

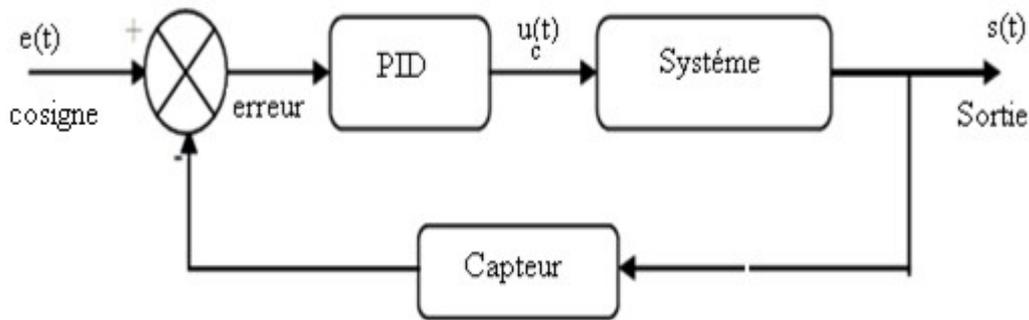


Figure. I-13 : Schéma bloc d'un système avec correcteur (Boucle fermée) [9].

$$u_c(t) = K_p e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt}$$

$$= K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

$$\text{T.L} \Rightarrow U_c(p) = K_p e(p) + K_i \frac{e(p)}{p} + K_d p e(p) \quad (\text{I-3})$$

Où :

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.

I-10-1 Action Proportionnelle

La commande de type 'Proportionnelle' est la plus simple qui soit. Il s'agit d'appliquer une correction proportionnelle à l'erreur corrigeant de manière instantanée, c'est-à-dire, lorsque le signal de commande est proportionnel au signal d'erreur:

$$u(t) = K_p e(t) \Rightarrow U(p) = K_p * e(p) \quad (\text{I-4})$$

Son rôle est d'amplifier l'erreur pour que le système réagisse plus vivement, il permet de diminuer le temps de montée et le temps de réponse. Lorsque l'on augmente le gain ' K_p ', le système réagit plus vite et l'erreur statique s'en trouve améliorée, mais en contre partie le système perd la stabilité. Le dépassement se fait de plus en plus grand, et le système peut même diverger dans le cas d'un ' K_p ' démesuré [9].

I-10-2 Action Intégrale

L'action intégrale est caractérisée par T_i (en s) ou $K_i=1/T_i$. T_i Temps d'intégration

$$u(t) = \frac{1}{T_i} \int_0^t e(\tau) d\tau = K_i \int_0^t e(\tau) d\tau \quad (\text{I-5})$$

L'action intégrale est d'autant plus rapide ou énergétique que T_i est petit. Son rôle est d'annuler l'écart entre mesure et consigne en régime établi, c'est pourquoi elle est toujours appliquée sur l'écart. C'est une action lente ; trop rapide, et a un effet déstabilisant [8].

I-10-3 Action Dérivée

L'action dérivée est caractérisée par T_d (en s). T_d représente le temps au bout duquel la variation de l'entrée est égale à la sortie

$$u(t) = T_d \frac{de(t)}{dt} = K_d \frac{de(t)}{dt} \quad (\text{I-6})$$

L'action dérivée a pour but de stabiliser la boucle [10].

I-10-4 Action Proportionnelle-Intégrale (PI)

Le correcteur de type PI est une régulation de type P auquel on a ajouté un terme intégral, il élabore alors une commande qui peut être donnée par la relation suivante :

$$u_c(t) = K_p e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \iff U_c(p) = K_p e(p) + K_i \frac{e(p)}{p} \quad (\text{I-7})$$

Le terme intégral complète l'action proportionnelle puisqu'il permet de compenser l'erreur statique et d'augmenter la précision en régime permanent. L'intégrale agissant comme un filtre sur le signal intégré, elle permet de diminuer l'impact des perturbations (bruit, parasites), et il en résulte alors un système plus stable.

Malheureusement, un terme intégral trop important peut lui aussi entraîner un dépassement de la consigne et une stabilisation plus lente [9].

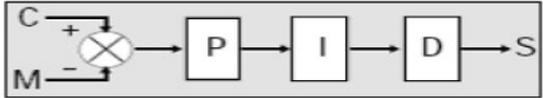
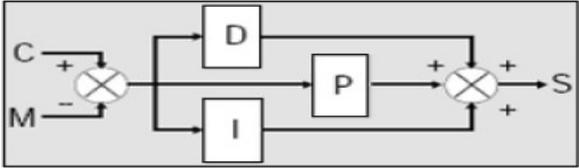
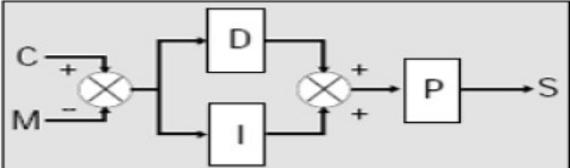
I-10-5 Action Proportionnelle-Intégrale-Dérivée (PID)

L'action conjuguée PID permet une régulation optimale en associant les avantages de chaque action : la composante P réagit à l'apparition d'un écart de réglage, la composante D s'oppose aux variations de la grandeur réglée et stabilise la boucle de régulation et la composante I

élimine l'erreur statique. Et c'est pour cela que ce type de correcteur est le plus utilisé en milieu industriel.

Dans un régulateur PID, il existe plusieurs façons d'associer les paramètres P, I et D. en effet, le correcteur PID peut avoir une structure série, parallèle ou mixte (voir tableau I-1) :

Tableau I-1 : Différentes structures du régulateur PID [4].

Structure du régulateur PID	Schéma et fonction de transfert
Série	 $K_P \left(\frac{T_i + T_d}{T_i} + \frac{1}{pT_i} + pT_d \right)$
Parallèle	 $K_P + \frac{1}{pT_i} + pT_d$
Mixte	 $K_P \left(1 + \frac{1}{pT_i} + pT_d \right)$

I-11 Le régulateur FLOU

Les contrôleurs à logique floue sont l'application la plus populaire de la théorie de la logique floue.

I-11-1 Historique

C'est à l'université de Berkeley Californie, USA en 1963, qu'est née la logique floue ou "Fuzzy logic". Son père créateur est Monsieur Lotfi A. Zadeh, automaticien de réputation internationale, professeur à l'université de Berkeley en Californie, qui a contribué à la modélisation de phénomène sous forme floue. En effet, le principe de la commande floue a été expérimenté pour la première fois en 1974 par l'équipe du professeur E.H. Mamdani. Les résultats obtenus, ont été repris et développés par plusieurs équipes en Europe de l'Ouest. Puis en 1985. La logique floue a été introduite à l'Institut de Technologie de Tokyo et dans quelques autres universités japonaises

grâce au chercheur japonais M. Sugene. Les sociétés japonaises comprirent l'avantage à la fois technique et commercial de la logique floue [11]:

- facilité d'implantation.
- solution de problèmes multi variables complexes.
- robustesse vis à vis des incertitudes.
- possibilité d'intégration du savoir de l'expert.

I-11-2 Eléments de base

a) La variable linguistique

La théorie de la logique floue est basée sur la notion de variable linguistique qui prennent des valeurs linguistiques ou flous. Par exemple : température = {basse, moyenne, élevée}

- Température : la variable linguistique.
- {Basse, moyenne, élevée}: les valeurs linguistiques.

b) Univers de discours :

C'est l'ensemble des valeurs réelles que peut prendre la variable floue, en général l'univers de discours est noté U .

c) La fonction d'appartenance

Les fonctions d'appartenance les plus utilisées sont : Triangulaire, trapézoïdale et gaussienne

d) Ensemble flou

Un ensemble flou est caractérisé par son nom (valeur linguistique) et sa fonction d'appartenance, il est noté $U_{\text{nom}}(u)$.

I-11-3 Opérateurs de la logique floue

Soit 2 ensembles flous (A et B) définis sur les univers de discours v et w avec les fonctions d'appartenance U_A U_B respectivement :

a) Union floue

La fonction d'apparence de l'union $U_{A \cup B}$ est définie par :

$$U_{A \cup B}(u) = \max \{U_A (v), U_B (w)\} \quad (\text{I-8})$$

b) L'intersection floue

La fonction d'appartenance de l'intersection $U_{A \cap B}$ est définie par :

$$U_{A \cap B}(u) = \min \{U_A(v), U_B(w)\} \quad (I-9)$$

c) Compilateur flou

La fonction d'appartenance du compilateur $U_{\bar{A}}$ est définie par :

$$U_{\bar{A}}(u) = 1 - U_A(v) \quad (I-10)$$

I-11-4 Structure d'une commande floue

Un contrôleur flou comporte les différents éléments suivants (figure I -14) [11] :

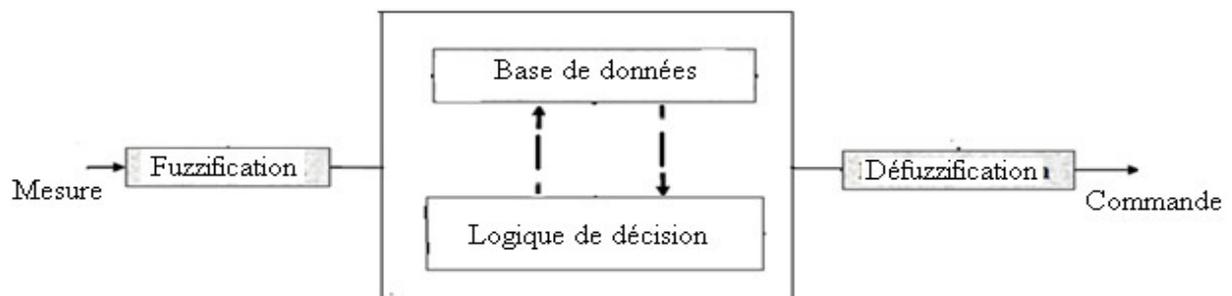


Figure I -14: Structure d'une commande floue [11].

a) Base de données

Contient les données qui permettent d'inférer une sortie numérique à partir d'une entrée numérique, généralement elle contient les éléments suivants :

- **Les variables d'entrée-sortie**

Chaque variable d'entrée sortie d'un contrôleur flou est caractérisée par : son nom, son univers de discours, sa fonction d'appartenance et sa distribution sur l'univers de discours.

- **Base des règles**

C'est l'ensemble des règles linguistiques qui donne la relation entre les variables linguistiques des prémisses (entrées) et les variables linguistiques des conséquences (sorties). Deux types de règles sont utilisés dans le domaine de l'automatique :

- Type Mamdani : dans ce type les conséquences sont floues :

Ri : si X_1 est A_i et X_2 est B_i alors Y_i est C_i

- X_1 et X_2 : les variable des prémisses.
- Y_i : conséquence.
- A_i , B_i et C_i : l'ensemble flou.
- Type Sugeno : dans ce type les conséquences sont des fonctions linéaires :

Ri : si X_1 est A_i et X_2 est B_i alors $Y_i = f(X_1, X_2)$

Le nombre maximal des règles dépend du nombre de variables de prémisses et le nombre de leur ensemble flou.

b) Fuzzification

C'est l'interface numérique floue (linguistique), qui transforme les données numériques en données linguistiques.

c) Défuzzification

C'est l'interface floue réelle, qui transforme les données linguistiques en données numériques.

Les méthodes de défuzzification sont [4]:

- Défuzzification par centre de gravité.
- Défuzzification par valeur maximale.

d) La logique de décision

Consiste à déterminer comment interpréter une règle et comment l'ensemble des règles, d'une façon générale : un ensemble des règles floues peut être interprété par un **ou flou** (FUZZY OR), et une règle floue peut être interprété comme un **et flou** (FUZZY AND).

I-12 Conclusion

Dans ce chapitre, nous avons présenté les principaux éléments des systèmes asservis en boucle ouverte et en boucle fermée, l'utilité de la commande (correction) pour un système donné, une description générale des systèmes industriels, ainsi que quelques méthodes de régulation, qui se résument dans l'utilisation des régulateurs PID. Enfin, nous avons touché la logique floue à travers les régulateurs de type Mamdani et de Sugeno.

Nous présentons dans le chapitre suivant la structure du microcontrôleur à utiliser.

II-1 Introduction

Depuis plusieurs années, le traitement numérique du signal est une technique en plein essor. Cette technique s'appuie sur plusieurs disciplines, citons les principales [12]:

- L'électronique analogique et numérique.
- Les microprocesseurs.
- L'informatique.
- Traitement du signal.

Dans ce chapitre, nous allons d'écrire les principes caractéristiques internes et externes des microcontrôleurs PIC, le logiciel de programmation des PIC et enfin le chargement de programmes dans les PIC.

II-2 Définition du PIC

Le PIC (Peripheral Interface Controller) est un microcontrôleur c'est-à-dire une unité de traitement de l'information de type microprocesseur à laquelle, on ajoute des périphériques internes permettant de réaliser des montages sans nécessité d'ajout de composants externes qui sont des composants dits « RISC ». (Reduced Instructions Set Computer), ou encore (composant à jeu d'instructions réduites).

II-3 Les différentes familles de PIC

Les microcontrôleurs PIC sont divisés en 3 grandes familles [13] :

- La famille « Base-line » utilisant des mots de 12 bits.
- La famille « Mid-range » utilisant des mots de 14 bits (et dont font partie les 16F8xx)
- La famille « High-end » utilisant des mots de 16 bits.

II-4 Identification d'un PIC

Pour identifier un PIC, on utilise simplement son numéro :

16 : indique la catégorie du PIC, c'est un « Mid-range ».

L : indique qu'il fonctionne avec une plage de tension beaucoup plus tolérante.

C : indique que la mémoire programme est un EPROM ou une EEPROM.

CR ou F : indique le type de mémoire ; CR(ROM) ou F (FLASH).

XX : représente la fréquence d'horloge maximale que le PIC peut recevoir.

Une dernière indication qu'on le trouve est le type de boîtier.

Exemple

Un 16f877-20 est un PIC « Mid-range » (16) ou la mémoire programme est de Type FLASH (F) et réinscriptible de type 877 et capable d'accepter une fréquence d'horloge de 20MHz (figure II-1).

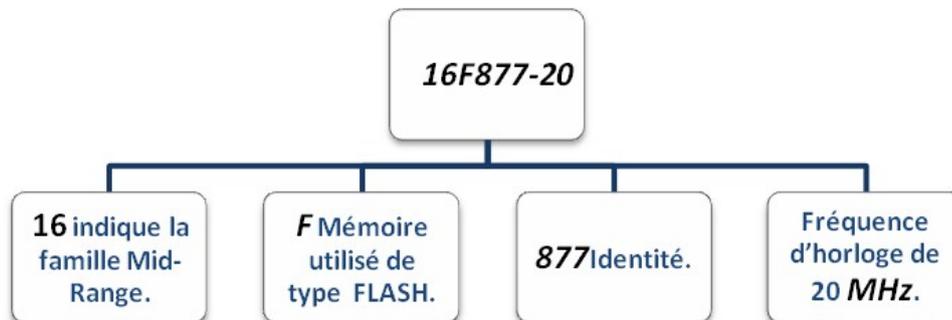


Figure II-1: description de la configuration du PIC 16F877 [5].

II-5 Descriptions générales du PIC 16F877A

La photo de la figure II-2 nous montre le schéma réel du PIC16F877A



Figure II-2 : Schéma réel du 16F877.

Le PIC 16F877A est caractérisé par :

- Une fréquence de fonctionnement élevée, jusqu' à 20 MHz.
- Une mémoire vive de 368 octets.
- Une mémoire EEPROM pour sauver des paramètres de 256 octets.
- Une mémoire morte de type FLASH de 8 K mots (1mot = 14 bits).
- Chien de garde WDT.

- 33 Entrées et sorties.
- Chaque sortie peut sortir un courant maximum de 25 mA.
- 3 Temporisateurs : TIMER0 (8 bits avec pré diviseur), TIMER1 (16 bits avec pré diviseur avec possibilité d'utiliser une horloge externe réseau RC ou QUARTZ) et TIMER2 (8 bits avec pré diviseur et post diviseur).
- 2 entrées de captures et de comparaison avec PWM (Modulation de largeur d'impulsions)
- Un convertisseur analogique numérique 10 bits avec 8 entrées multiplexées maximum.
- Une interface de communication série asynchrone et synchrone (USART/SCI).
- Une interface de communication série synchrone (SSP/SPI et I2C).
- Une tension d'alimentation entre 2 et 5.5 V.

II-6 Les mémoires

Il existe trois blocs de mémoires :

II-6-1 Mémoire programme

La mémoire programme est constituée de 8K mots de 14 bits. C'est dans cette zone que nous allons écrire notre programme et le sauvegarder une fois compilé. Cette mémoire de 1K stocke le programme; l'emplacement de celui-ci peut être à n'importe quel endroit de la mémoire. Notons qu'une instruction est codée sur 1 mot, donc 1K donne mille instructions possibles.

II-6-2 Mémoire EEPROM

La mémoire EEPROM (Electrical Erasable Programmable Read Only Memory), est constituée de 256 octets que nous pouvons lire et écrire depuis notre programme. Ces octets sont conservés après une coupure de courant et sont très utiles pour stocker des paramètres et semi paramètres. Son utilisation implique un procédé spécial, car ce n'est pas une RAM, mais bien une ROM de type spécial. Il est donc plus rapide de la lire que de l'écrire.

II-6-3 Mémoire RAM

La mémoire RAM (Random Access Memory) de 368 octets, contenant les registres spéciaux et les registre à usage général, est divisée en 4 blocs mémoires. C'est la mémoire d'accès rapide, mais volatile c'est-à-dire toutes les données stockées seront perdues lors d'une coupure de courant.

II-7 Les broches de fonctionnement

Les broches de fonctionnement sont les broches qui permettent au microcontrôleur de fonctionner. Elles doivent être obligatoirement connectées pour qu'elles fonctionnent.

Nous les avons classées de la façon suivante (figure II-3) [14]:

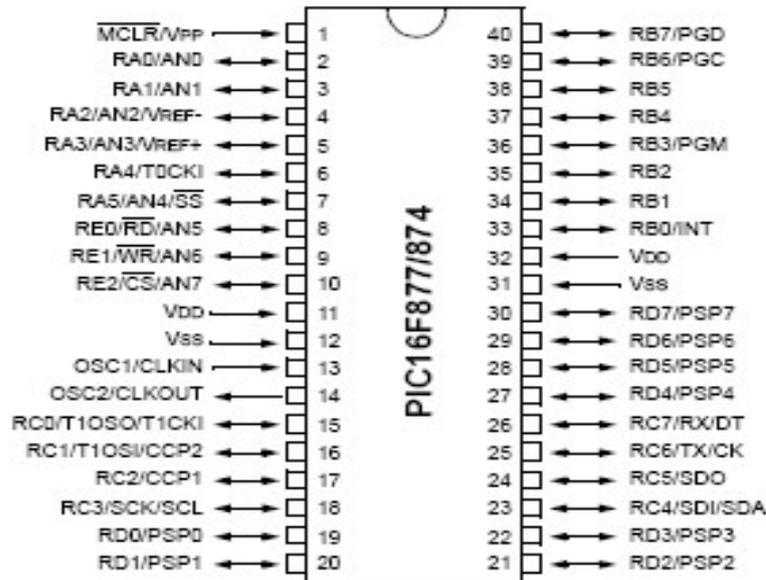


Figure II-3: Les différentes broches du PIC 16F877A [14].

II-7-1 Les broches d'alimentation

Le PIC 16F877A a quatre broches d'alimentation. Les deux pins pour le « VSS » sont le n° 12 et n° 31, et pour le « VDD » sont les pins n° 11 et n° 32. Il lui suffit une alimentation de 5V. On remarque qu'on a 2 connections « VDD » et 2 connections « VSS ». La présence de ces 2 pins s'explique pour une raison de dissipation thermique. Les courants véhiculés dans le pic sont loin d'être négligeables du fait des nombreuses entrées/sorties disponibles. Le constructeur a donc décidé de répartir les courants en plaçant 2 pins pour l'alimentation VSS, bien évidemment, pour les mêmes raisons, ces pins sont situées de part et d'autre du PIC, et en positions relativement centrales.

II-7-2 Les broches de quartz

Dans le cas du 16F877A le quartz utilisé est typiquement un quartz de 4MHz, c'est-à-dire qu'il va fournir 4millions d'impulsions par seconde. Le temps qui s'écoule entre deux impulsions s'appelle un cycle d'horloge.

Le quartz de 4 MHz est connecté aux broches osc1 (n° 13) et osc2 (n° 14).

II-7-3 La broche de réinitialisation

Une broche particulière du microprocesseur est la broche MCLR (Master CLEAR). Cette broche est d'avantage une broche de contrôle que de fonctionnement.

Cette broche a pour effet de provoquer la réinitialisation du microprocesseur lorsqu'elle est connectée à 0. Ceci provoque l'arrêt de programme qui va recommencer à la première instruction. L'ensemble des registres du microprocesseur vont par ailleurs être également remis à leur valeur de départ.

Pour que le processeur fonctionne, il faut donc que cette broche soit connectée à 5V. Un bouton poussoir est prévu pour le reset par l'utilisateur. L'appui sur le bouton poussoir va mettre transitoirement la broche MCLR à (0).

II-7-4 Les ports d'entrées/ sorties

Toutes les autres broches du 16F877A sont des broches de ports.

- Port A : 6 pins E/S numérotées de RA0 à RA5.
- Port B : 8 pins E/S numérotées de RB0 à RB7.
- Port C : 8 pins E/S numérotées de RC0 à RC7.
- Port D : 8 pins E/S numérotées de RD0 à RD8.
- Port E : 3 pins E/S numérotées de RE0 à RE2.

Pour déterminer les modes des ports (E/S), il faut sélectionner leurs registres TRIS :

- Le positionnement d'un bit à '1' place le pin en entrée.
- Le positionnement d'un bit à '0' place le pin en sortie.

a) Port A

Le port A est formé par 6 entrées/sorties numérotées de RA0 à RA5 qui peuvent être utilisées comme des entrées pour le convertisseur analogique numérique ou utilisées pour le TIMER0. On peut utiliser ce port, soit pour la conversion analogique numérique, soit en mode (E/S). Dans notre projet on a utilisé RA0 comme entrée pour le CAN.

b) Port B

Il possède 8 pins d'entrées/sortie numérotés de RB0 à RB7. La seule particularité est qu'on peut utiliser le pin RB0 comme entrée de déclenchement d'interruption externe.

c) Port C

Ce port possède 8 pins d'entrées/sortie numérotée de RC0 à RC7. La particularité de ce port est qu'il permet d'établir une communication série via l'USART par ses pins RC6 et RC7.

d) Port D

Ce port n'est présent que sur les PIC16F877. Il possède 8 pins d'entrées/sortie numérotés de RD0 à RD7. Ces 8 lignes peuvent aussi servir comme port parallèle dit esclave (slave).

e) Port E

Ce port n'est présent que sur les PIC16F877. Il est composé de 3 pins d'entrées/sortie numérotée de RE0 à RE2. Ces 3 pins constituent aussi les 3 entrées de convertisseurs analogiques (AN5 à AN7) et les lignes de commande du bus esclave PSP.

II-8 Le convertisseur analogique numérique du PIC 16F877A

La fonction conversion analogique-numérique consiste à transformer une grandeur électrique en une grandeur numérique exprimée sur N bits. Cette grandeur de sortie représente, dans le système de codage qui lui est affecté, un nombre proportionnel a la grandeur analogique d'entrée.



Figure II-4 : Principe de conversion analogique / numérique [5].

Le CAN intégré dans le 16F877A est un CAN 10 bits qui donne une précision en 5V de 5mV environ, ce qui est une précision tout à fait intéressante. La résolution 10 bits du PIC permet d'attribuer 1024 valeurs numériques à notre signal d'entrée.

D'une manière générale, le principe de conversion analogique/numérique nécessite deux opérations:

II-8-1 La quantification

Opération qui consiste à associer une valeur analogique à la plus petite variation mesurable entre deux valeurs codées distinctes en sortie. Cette valeur est appelée quantum :

$$q = (V_{ref+} - V_{ref-})/2^n \tag{II -1}$$

q : quantum (V), aussi appelé résolution

n : nombre de bits en sortie du convertisseur

II-8-2 Le codage

Opération qui assigne une valeur numérique à chacun de ces niveaux. Les codages les plus couramment utilisés sont :

- Le binaire naturel, pour les nombres non signés,
- Le complément à 2 pour les nombres signés,
- Le code binaire signé.

Le contrôle du module CAN se fait par les deux registres ADCON0 et ADCON1 :

- **ADCON0** : permet de définir l’horloge de conversion et le canal à convertir.
- **ADCON1** : permet la configuration des entrées analogiques et numériques et les tensions de références Vref+ et Vref-.

Dans notre projet la conversion dans le PIC s’effectue par approximations successives dont le fonctionnement est détaillé ci dessous (figure II-5) [15]:

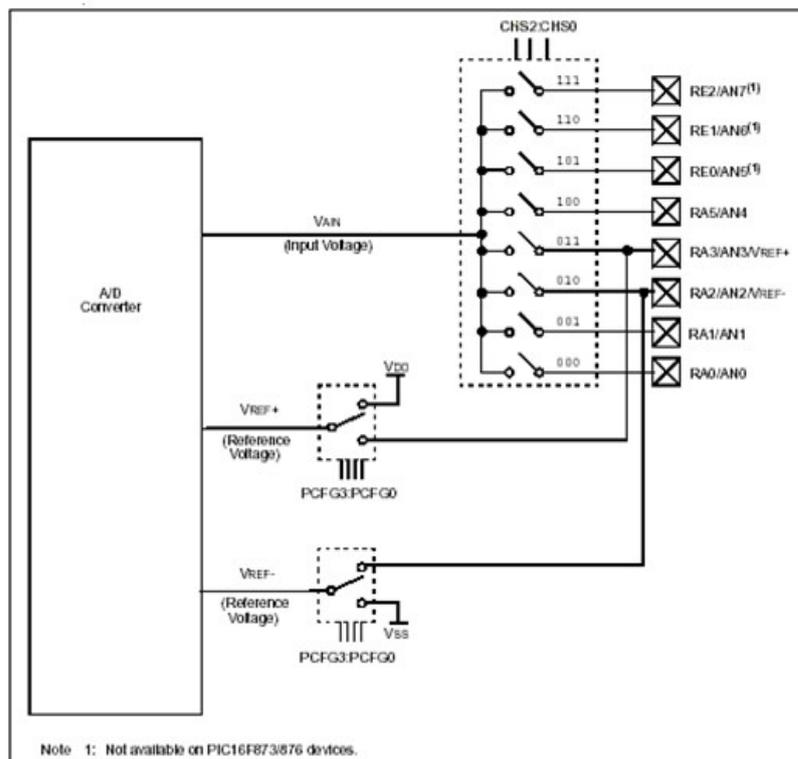


Figure II-5 : schéma interne des entrées du CAN [15].

écrit sur l'éditeur de MPLAB dans son programme source, sera compilé, la compilation délivre plusieurs fichiers, le plus important est le code machine du programme.

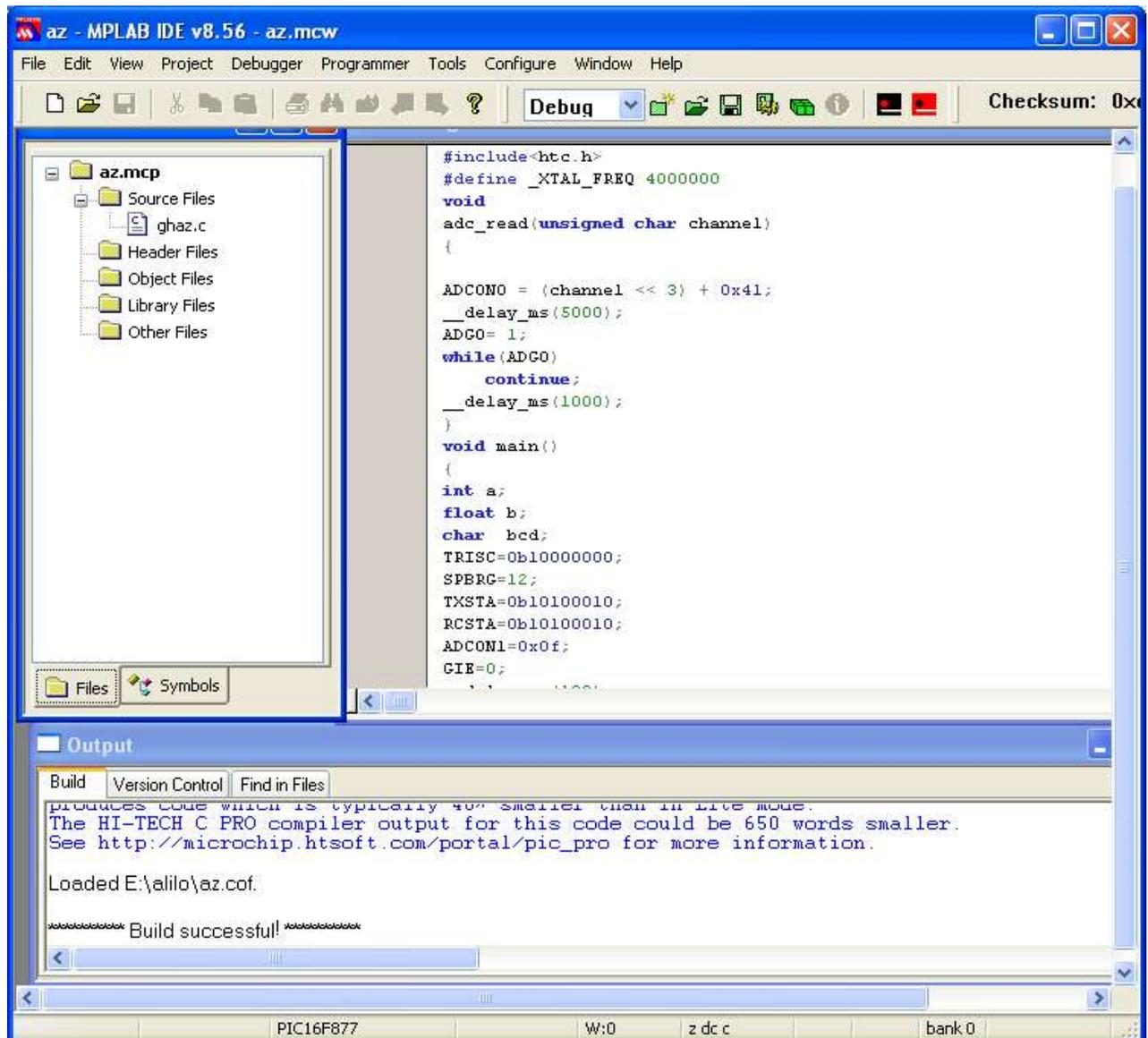


Figure II-7: l'éditeur de MPLAB.

II-11 Présentation du programmeur de PIC

Si le programme ne présente aucune erreur lors de la vérification, on pourra alors le charger dans la mémoire programme du microcontrôleur à l'aide d'un circuit programmeur géré par logiciel de programmation.

Nous avons utilisé une carte de développement pour le chargement (figure II-8) de nos programmes dans le PIC :

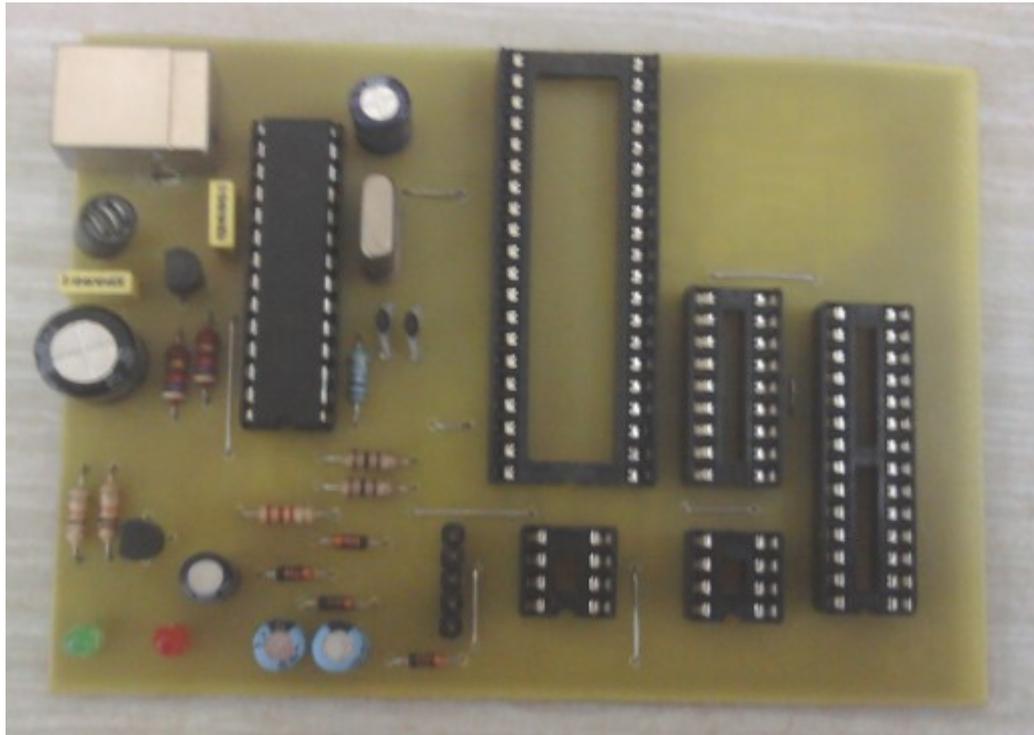


Figure II-8 : Le programmeur du PIC16F877A.

Le logiciel de programmation que nous avons utilisé est le WINPROG (Figure II-9 et II-10), qui permet la liaison entre le PC et le circuit programmeur via le port USB.

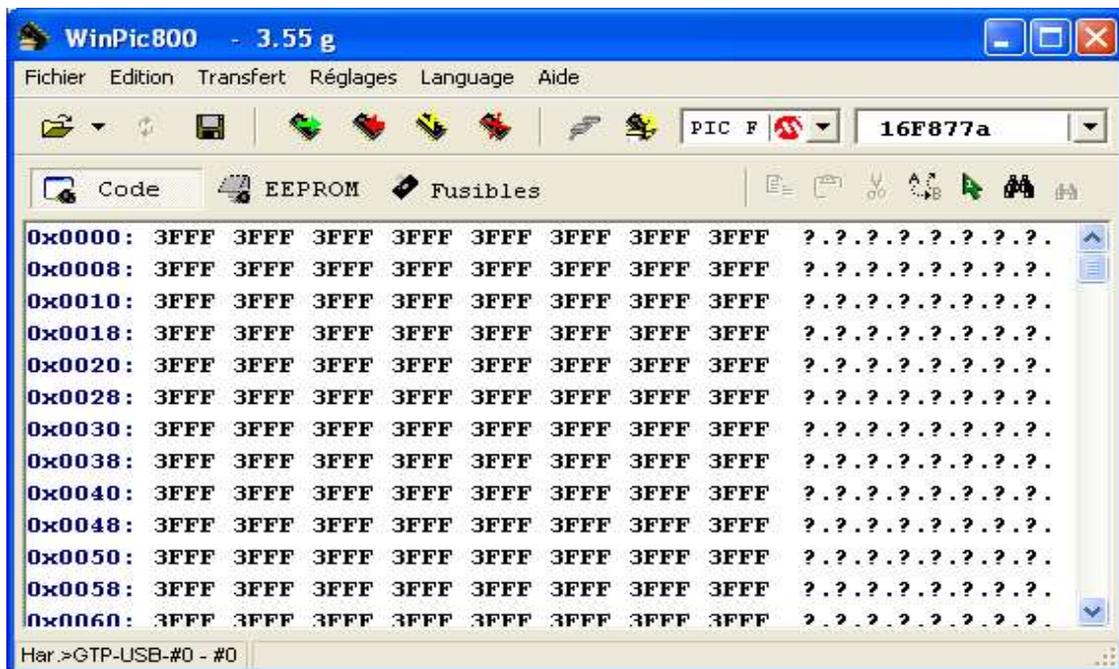


Figure II-9 : Menu principal du logiciel WINPRG.

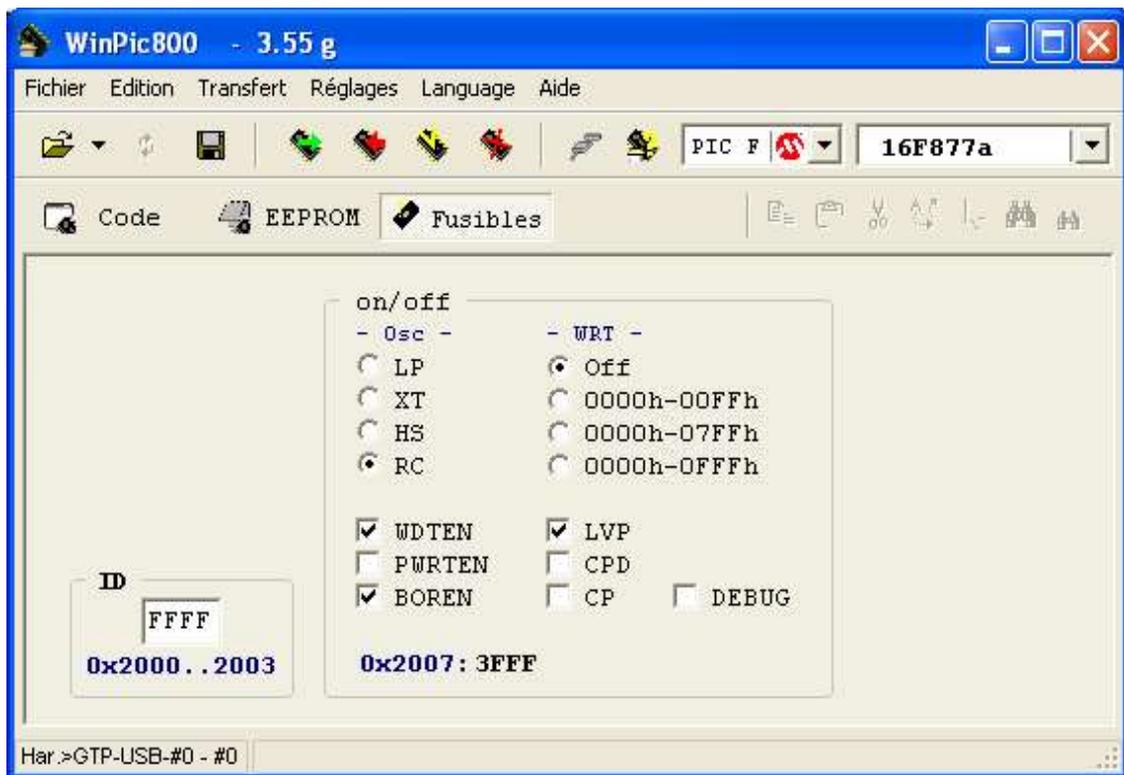


Figure II-10 : Menu principal du logiciel WINPRG.

- Il suffit de cliquer sur «détection PIC » pour représenter le composant qui est sélectionné pour la programmation.
- Pour vérifier le programme contenu dans le PIC, on appuie sur « lire».
- Pour écraser un programme qui est déjà contenu dans le PIC, on appuie sur « effacer».
- Pour programmer le PIC, on doit sélectionner le fichier du notre programme « .hex » en cliquant dans le menu « file » et après on appuie sur « ouvrir ». Puis choisie le type de l'oscillateur et appuie sur « Fusibles ». Après cette opération, on clique sur «programmer».

II-12 Conclusion

Nous remarquons que les microcontrôleurs sont devenus indispensables dans les montages électroniques de part leur complexité et leur rapidité d'exécution de travail.

Nous avons choisit le 16F877A pour sa multitude de fonctions et le nombre considérable d'entrées/sorties selon notre besoin. Cela va nous aider dans la réduction du nombre des composants, de la carte en plus de sa disponibilité sur le marché.

Nous présentons dans le chapitre suivant la maquette réalisée pour réguler le niveau d'eau.

III-1 Introduction

Dans ce chapitre, on présentera la description électronique de la maquette de régulation de niveau d'eau, la carte de mesure et la carte de commande. Ensuite, la structure de la carte d'interfaçage réalisée, les principes de conversion A/N, N/A et le protocole de la transmission série et on terminera par le logiciel utilisé pour la programmation (LABVIEW).

III-2 Description du système de régulation de niveau

III-2-1 Présentation de la maquette

La maquette de régulation est constituée : d'un réservoir haut de remplissage d'eau, d'un réservoir bas (source d'eau), d'un capteur de pression différentielle, d'une pompe, 4 cartes électroniques (deux cartes d'alimentation, une carte d'acquisition ou de mesure et une carte d'interface avec le PC). La régulation est assurée par un programme développé avec le logiciel LABVIEW.

Le schéma synoptique de la structure de la maquette conçue est donné par la figure III-1.

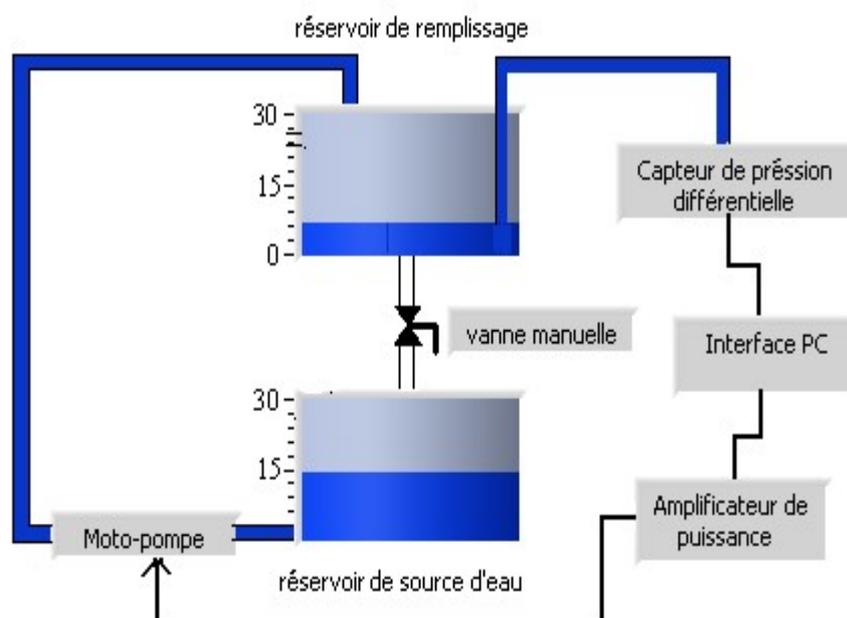


Figure III-1 : Structure de la maquette [5].

- Le réservoir haut est en verre transparent permettant ainsi l'observation de l'évolution du niveau d'eau. Il est de forme parallélépipédique de 35 cm de hauteur et 20cm de largeur, et

de 10 cm de profondeur, une règle graduée permet de lire directement le niveau d'eau. Ce réservoir présente une entrée de remplissage d'eau.

- Le réservoir bas représente la source d'eau. Il est naturellement plus volumineux que le réservoir haut. Il est aussi de forme parallélépipédique, de 20 cm de hauteur, de largeur, et de profondeur. Il a une capacité de remplissage de 8 litres. Ce réservoir est aussi en verre transparent.
- La vanne manuelle placée sur le retour d'évacuation permet de varier le débit de sortie du réservoir haut et d'introduire des perturbations en fonctionnement statique.
- La pompe.
- Le capteur de pression différentielle.

III-2-2 Descriptions des deux cartes d'alimentation stabilisée

III-2-2-1 1^{ère} carte d'alimentation

Nos cartes de mesure et d'interface doivent être alimentées par une alimentation stabilisée qui fournit à la sortie:

- 5V pour alimenter le PIC, le MAX232 et le DAC0808.
- $\pm 12V$ pour l'alimentation des circuits TTL (LF351N, TL081, TL082 et LM324), ainsi que 12V pour alimenter le capteur.
- La masse (GND).

La carte réalisée contient (Figure III-2) :

- Un transformateur abaisseur, qui fournit sur son secondaire une tension alternative très inférieure à celle du secteur (220V/12V).
- Un pont redresseur (diodes en pont de Graëtz), qui fournit en sortie une tension non plus alternative mais redressée.
- 5 capacités de filtrage, qui réduisent l'ondulation de la tension issue du pont redresseur.
- 3 régulateurs de tension, dont le rôle est de stabiliser le potentiel de sortie par une certaine valeur (5V, 12V et -12V). En fin, pour avoir une tension de 5V, on a choisi l'un des régulateurs de tension les plus utilisés. Le 7805, également pour les tensions 12V et -12V, on a choisi respectivement le 7812 et le 7912(voir annexe1).
- 2 capacités de découplage.

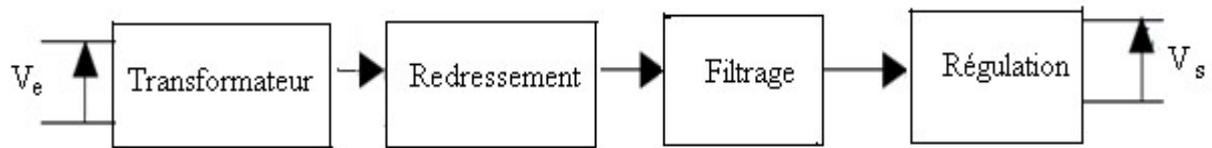


Figure III-2: schéma synoptique de 1ere alimentation.

a) Choix du transformateur

Notre Carte de développement nécessite une tension d'alimentation de (5V, 12V et -12V) et un courant de 0.5A. Ainsi, nous devons choisir un transformateur à point milieu dont la tension au secondaire est à 12V et un courant de 1A.

b) Choix du pont de redressement

Le choix de pont de diode est basé essentiellement sur :

- La tension inversée maximale de diode.
- Le courant moyen direct.

c) Choix des condensateurs

On utilise les condensateurs dans le montage pour deux raisons, le filtrage et le découplage.

1- Choix des condensateurs de filtrage:

Pour obtenir une tension presque constante, il faut brancher un ou plusieurs condensateurs en parallèle juste après le pont redresseur, plus la valeur de la capacité est élevée plus le filtrage sera meilleur. Les deux principaux critères à considérer dans le choix d'un condensateur sont :

- sa capacité.
- sa tension de service.

2- Choix des condensateurs de découplage :

On les choisit de telles sortes qu'ils servent à améliorer la stabilité du régulateur.

d) Choix des régulateurs de tension

Un régulateur de tension est un composant à semi-conducteur dont le rôle consiste à rendre quasi continue une tension qui présente une ondulation issue d'un pont redresseur et à stabiliser sa valeur.

La tension de sortie V_{out} est le principal critère de choix, puisqu'elle correspond à la tension désirée. Ainsi, on choisira un LM78xx ou 79xx.

III-2-2-2 2^{ème} carte d'alimentation

La deuxième carte d'alimentation est conçue pour alimenter l'actionneur (la pompe). Cette carte a pour rôle de donner la puissance nécessaire.

La carte réalisée contient :

- Un transformateur abaisseur, qui fournit sur son secondaire une tension alternative très inférieure à celle du secteur (220V/12V).
- Un pont redresseur (diodes en pont de Graëtz) qui fournit en sortie une tension non plus alternative mais redressée.
- 5 capacités de filtrage, qui réduisent l'ondulation de la tension issue du pont redresseur.
- Un régulateur de tension, son rôle est de stabiliser le potentiel de sortie à la valeur 8V, on a choisi le régulateur de tension 7808(voir annexe1).
- Transistor 2N3055 (voir annexe1).
- Une diode IN5408.

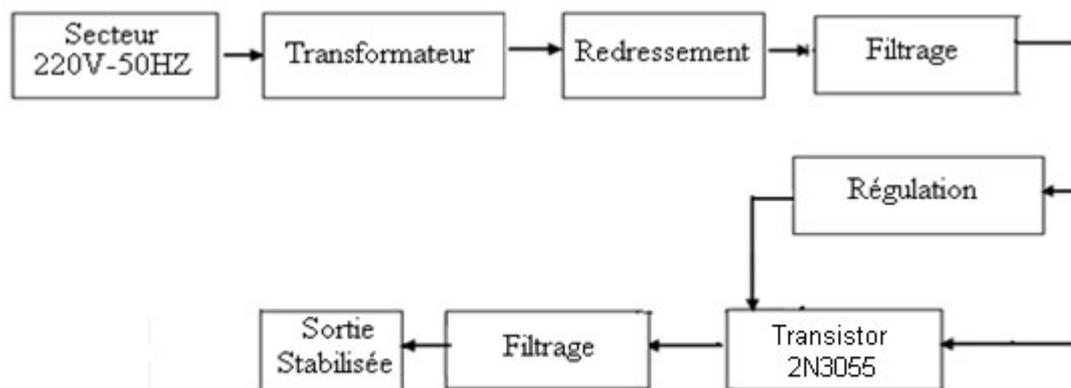


Figure III-3: schéma synoptique de la 2^{ème} alimentation.

a) Choix du transformateur

Notre Carte nécessite une tension d'alimentation de 8V et un courant de 1.5 A au maximum, ainsi nous devons choisir un transformateur à point milieu dont la tension au secondaire est à 12V et courant de 1.5 A.

b) Choix du transistor 2N3055

On a choisit ce type du transistor grâce à sa grande puissance et ses caractéristiques qui sont compatibles avec notre projet.

III-2-3 Carte de mesure

Pour pouvoir réguler le processus, le régulateur doit avoir une tension image bien adaptée.

III-2-3-1 les composants de la carte de mesure

Cette carte se compose de 4 parties :

a) Le capteur de pression

Le capteur, référence **HONEYWELL 26PCAFA6D** (voir annexe 1), mesure la pression de fond du réservoir haut par rapport à la pression Atmosphérique. Un tube en plastique permet de transmettre hermétiquement la pression de fond du réservoir au capteur. On peut représenter celui-ci par son schéma équivalent (figure III-4):

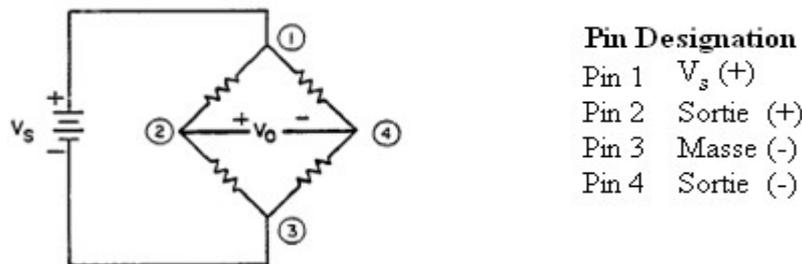


Figure III-4: Schéma équivalent du capteur de pression.

On fournira à ce capteur une alimentation de 12V (bornes 1 et 3). Lors d'une pression non nulle, on verra apparaître, à ses bornes 2 et 4, une différence de potentiel proportionnelle au mesurande. C'est cette différence de potentiel, image de la pression, que l'on viendra mesurer.

b) Réglage d'offset

Le bloc formé par l'amplificateur TL081 (voir annexe1) et une résistance variable, est un montage potentiométrique qui permet un réglage de l'offset. En effet, une tension apparaît à la sortie du capteur à vide à cause d'une pression atmosphérique ce qui fausse la lecture du niveau d'eau. Pour éliminer cette tension, On branche la borne 4 du capteur à l'entrée non inverseur du TL081 (borne 3), et les 2 bornes de la résistance variable aux bornes 1 et 5 de TL081. Ainsi, on mettra les 2 bornes du voltmètre, l'une sur la sortie de TL081 et l'autre sur la borne 2 du capteur, et on fait varier la résistance variable jusqu'à lire la tension 0 V sur le voltmètre.

c) Amplificateur d'instrumentation

La grandeur image de la mesurande aux bornes de notre capteur de pression est de faible valeur. Pour travailler avec une bonne précision, il est nécessaire d'amplifier cette tension avant de la mesurer. On a choisi donc un montage amplificateur à fort taux de rejection de mode commun afin d'amplifier que le signal utile. Le montage choisi porte le nom d'amplificateur d'instrumentation. Celui-ci est constitué de 3 amplificateurs opérationnels et de plusieurs résistances. Son schéma de montage est le suivant (Figure III-5) :

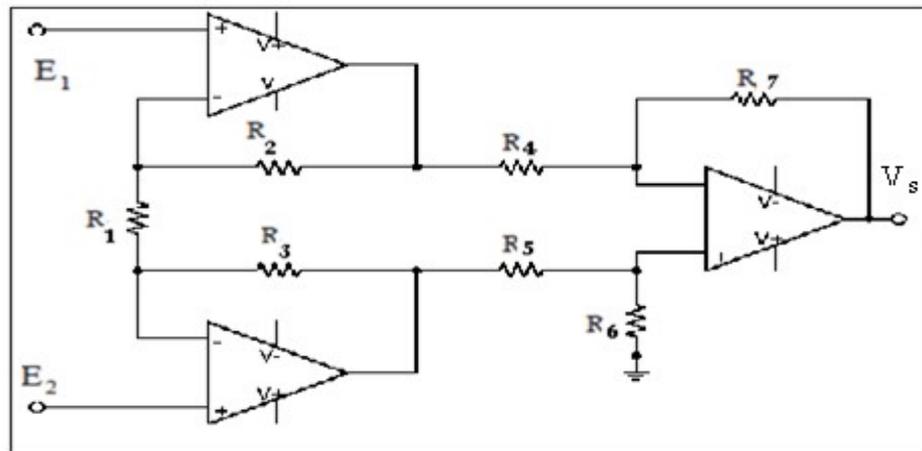


Figure III-5 : Amplificateur d'instrumentation [16].

La tension de sortie est donnée par la relation suivante :

$$V_s = G_T (E_2 - E_1) \tag{III-1}$$

Avec :

$$G_T = \left(1 + \frac{R_2 + R_3}{R_1}\right) \left(\frac{R_7}{R_4}\right)$$

Les entrées E_1 et E_2 représentent respectivement la sortie du TL081 et la borne 3 du capteur.

G_T : le gain d'amplification.

d) Filtrage

La dernière partie du montage représente un filtre qui réalise une opération de traitement du signal. Il atténue certaines composantes du signal (parasites, bruits) et laisse passer d'autres (signal utile). Pour ce montage on a utilisé un filtre passe-bas.

III-2-3-2 Choix du gain d'amplification

Le capteur est alimenté par 12 V et délivre une sortie variant de 0 à 8.3 mV. Il fait correspondre une tension nulle à une hauteur nulle et une tension de 8.3 mV à une hauteur de 30 cm. Par contre, le microcontrôleur a une pleine échelle égale à 5 V. D'où, on peut calculer le gain d'amplification :

$$G = \frac{5}{8.3 \cdot 10^{-3}} = 602 \quad (\text{III-2})$$

Pour avoir ce gain en tension, nous prenons les valeurs des résistances suivantes :

$R_1 = 5 \text{ k}\Omega$; $R_2 = R_3 = 1500 \text{ k}\Omega$ et $R_4 = R_5 = R_6 = R_7 = 100 \text{ k}\Omega$.

III-2-3-3 Etalonnage du capteur

L'étalonnage nous permet de se renseigner sur la sensibilité du capteur et sur la linéarité de sa réponse ou non par rapport à la variation de la pression. En variant le niveau d'eau, on a relevé les valeurs suivantes de la tension de sortie du capteur (voir tableau III-1) :

Tableau III-1 : Variation de la mesure en fonction du niveau.

H (cm)	0	2	4	6	8	10	12	14	16	18	19	20	21	24	28
V(volts)	0	0.32	0.66	1	1.33	1.67	2	2.32	2.66	3	3.18	3.34	3.5	4	4.69

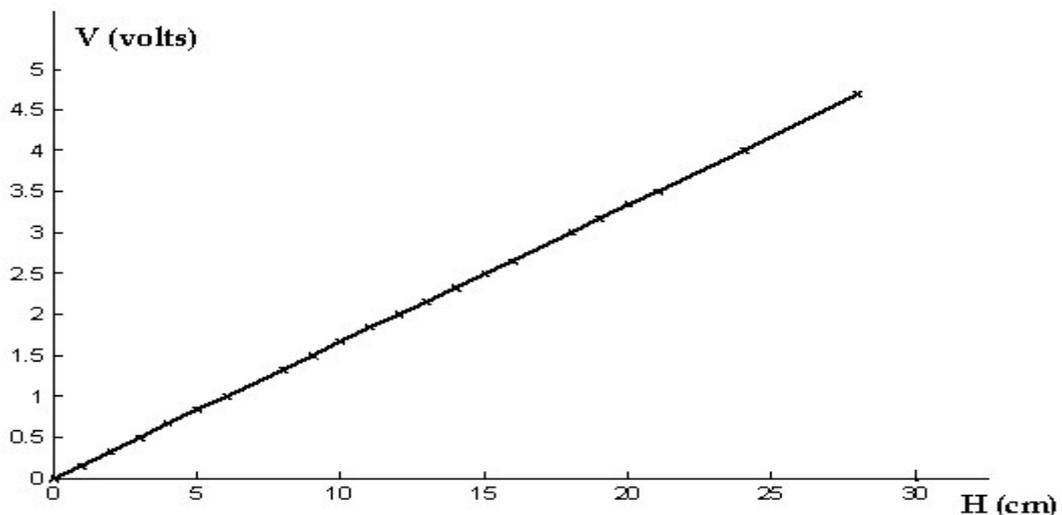


Figure III-6 : courbe d'étalonnage.

La courbe d'étalonnage est linéaire, sous forme d'un polynôme simple de la forme :

$$V = a * H + b \quad (\text{III-3})$$

- Avec :
- $a = \frac{\Delta V}{\Delta H} = 0.17$ (la pente de la droite).
 - $b = 0$ (la valeur initiale).

On peut donc varier la commande proportionnellement au niveau d'eau choisi sans compenser la mesure délivrée par le capteur.

III-2-4 Carte de commande

Notre carte de commande est constituée d'une pompe et d'un amplificateur de puissance :

III-2-4-1 La pompe

La pompe utilisée est une pompe d'essuie glace d'automobile permet le remplissage du réservoir haut, elle est composée d'une entrée d'aspiration de 5 mm de diamètre, d'une sortie de refoulement de 5 mm de diamètre. La pompe est alimentée par une tension continue variable de 0 à 12 V, elle consomme en fonctionnement nominal un courant de 1 A.

Dans notre projet on a limité la tension d'alimentation de la pompe par 4.6V. (0 à 4.6V)

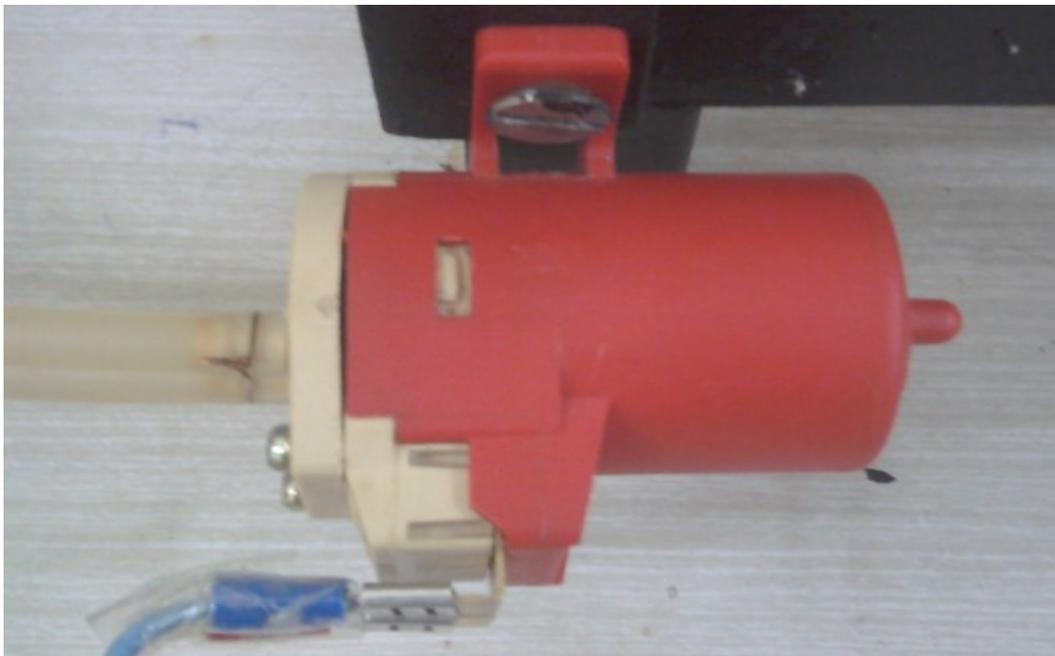


Figure III-7 : La pompe.

III-2-4-2 Amplificateur de puissance

Pour commander directement un moteur, on a besoin d'un amplificateur de puissance pour délivrer la puissance nécessaire.

La puissance nécessaire au système à commander (de type électrique, mécanique, thermique,...) n'est pas fournie par l'élément de commande. En fait, celui-ci module, en fonction de la commande, l'énergie empruntée à une source d'énergie qui alimente l'actionneur (pompe).

Le montage de l'amplificateur de puissance utilisé est donné par le schéma de la figure III-8 :

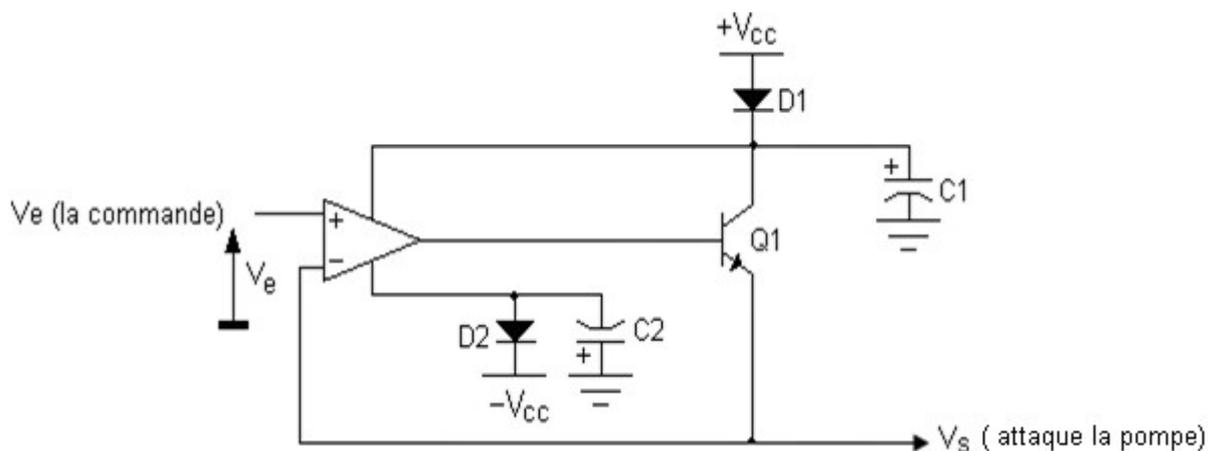


Figure III-8 : Schéma de l'amplificateur de puissance.

III-2-5 Présentation de la carte d'interfaçage à base du PIC16F877A

L'interfaçage permet l'échange d'informations entre deux ou plusieurs périphériques. Pour notre cas, on s'intéresse à l'interfaçage entre un PC et un réservoir rempli d'eau, pour cela on a réalisé une carte d'interfaçage à base du PIC 16F877A.

III-2-5-1 Descriptions de la carte

Pour pouvoir commander le procédé (dans notre cas c'est un niveau d'eau dans un réservoir) avec une loi de commande numérique à travers un PC, il est nécessaire d'utiliser une carte d'interfaçage qui a pour but de transformer les signaux numériques en des signaux analogiques équivalents et inversement.

La carte réalisée contient :

- Un PIC16F877A
- Un DAC0808
- Un AOP LF351

- MAX232
- Port série

Nous décrivons ci-dessous les caractéristiques de ces composants un par un :

III-2-5-2 Microcontrôleur PIC 16F877A

Il s'agit d'une carte à base d'un microcontrôleur 16F877A. Autour de ce dernier des connecteurs donnent accès aux ports d'entrées/sorties. Elle est composée de plusieurs étages que nous avons définies précédemment, étage de l'alimentation, l'horloge, l'interface série, le reset et le CAN, (voir chapitre II).

III-2-5-3 Acquisition de la mesure

Cet organigramme (Figure III.9) représente les étapes d'acquisition d'un signal analogique appliqué sur les pins du CAN.

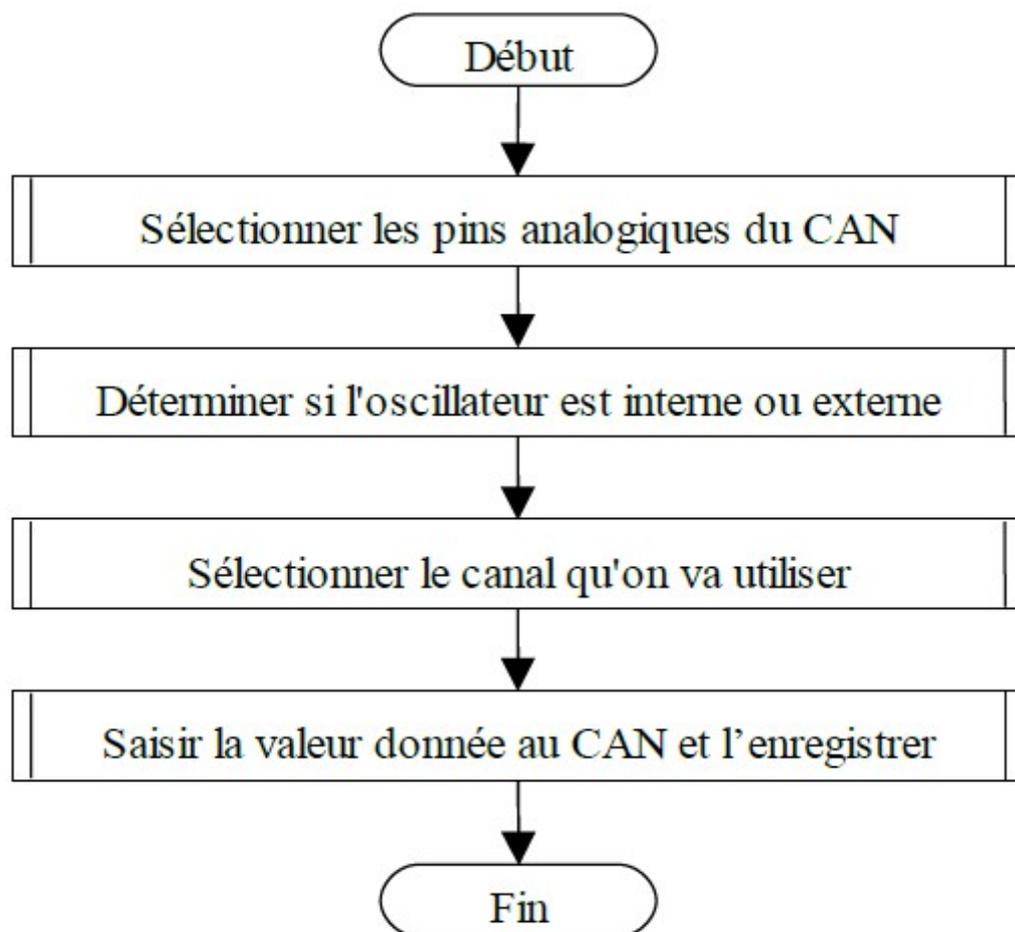


Figure III-9 : L'organigramme d'acquisition de la mesure [5].

III-2-5-4 La conversion numérique analogique

On dispose d'un mot numérique de n bits, que l'on voudrait convertir en une tension analogique, en considérant un code binaire :

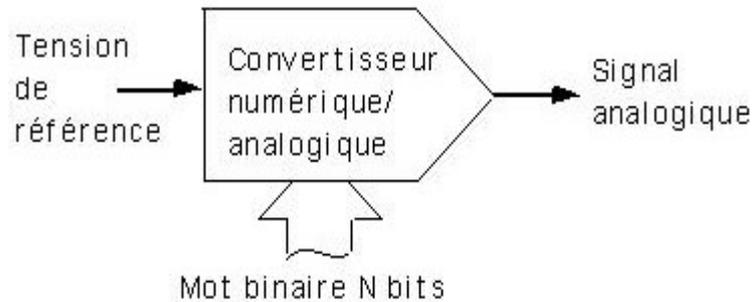


Figure III-10: Schéma d'un convertisseur numérique analogique [5].

a) Le convertisseur de courant

Ce convertisseur fournit un courant qui dépend de la tension référence V_{ref} ainsi que les états des interrupteurs suivant la formule suivante :

$$I_0 = I_{ref} \left(A_0 + \frac{A_1}{2} + \frac{A_2}{4} + \dots + \frac{A_{n-1}}{2^{n-1}} \right) \tag{III-4}$$

Avec : $I_{ref} = \frac{V_{ref}}{R}$

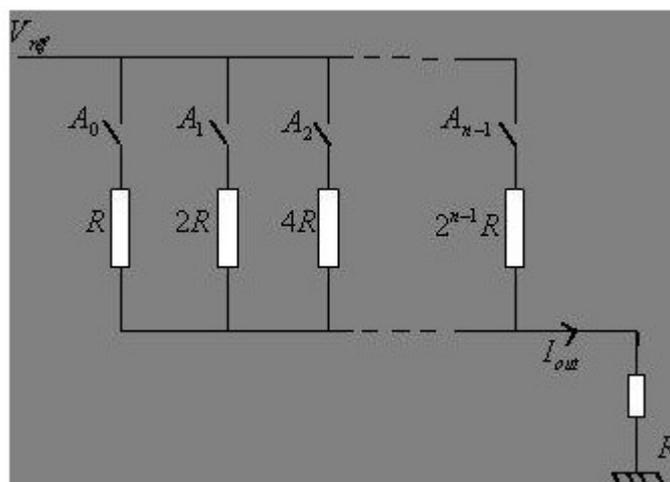


Figure III-11: Schéma de principe d'un CNA de courant [5].

Dans notre projet nous avons utilisé un convertisseur numérique analogique de courant "le DAC0808" suivi d'un convertisseur courant/tension à base d'un amplificateur opérationnel.

b) Le DAC 0808

Les DAC0808 sont des circuits monolithiques convertisseurs D/A 8 bits fournissant un courant plein échelle de 150 mA et ne dissipant que 33 mW avec une alimentation égale à 5 V.

Les DAC0808 s'interfacent directement avec la logique TTL, DTL ou CMOS.

Concernant le brochage du DAC0808, il est donné par le schéma suivant :

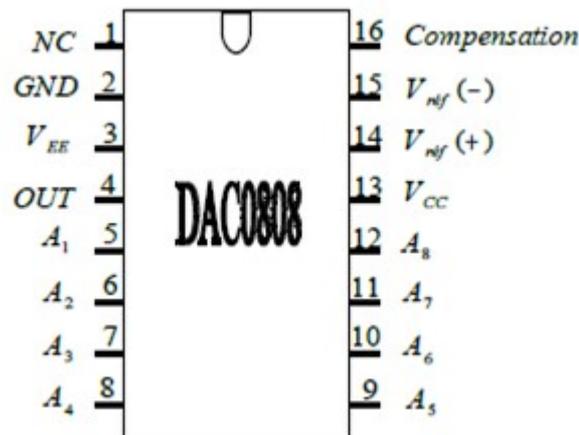


Figure III-12 : Brochage du DAC0808 [5].

Le tableau ci-dessous donne un aperçu sur les valeurs limites d'un CNA de type DAC0808 :

Tableau III-2 : Caractéristiques du DAC0808.

V _{CC}	+18V DC
V _{EE}	-18V DC
Tension d'une entrée digitale, V5-V12	-10 VDC à +18 VDC
Tension de sortie, V _o	-11 VDC à +18 VDC
Courant de référence, I14	5 mA
Tension de référence V14, V15	V _{CC} et V _{EE}
Puissance dissipée	1000 mW
Décote au-dessus de 25 °C	6.7 mW/°C
T° de fonctionnement DAC0808L	-55°C ≤ TA ≤ +125°C
T° de fonctionnement DAC0808LC séries	0 ≤ TA ≤ +75°C
Plage de températures de stockage	-65°C à +150°C

L'application typique du DAC0808 (voir annexe1) utilisée pour notre carte est de convertir le courant issu du convertisseur en une tension proportionnelle en utilisant le circuit LF351 (voir annexe1). La figure III-13 illustre ce fonctionnement :

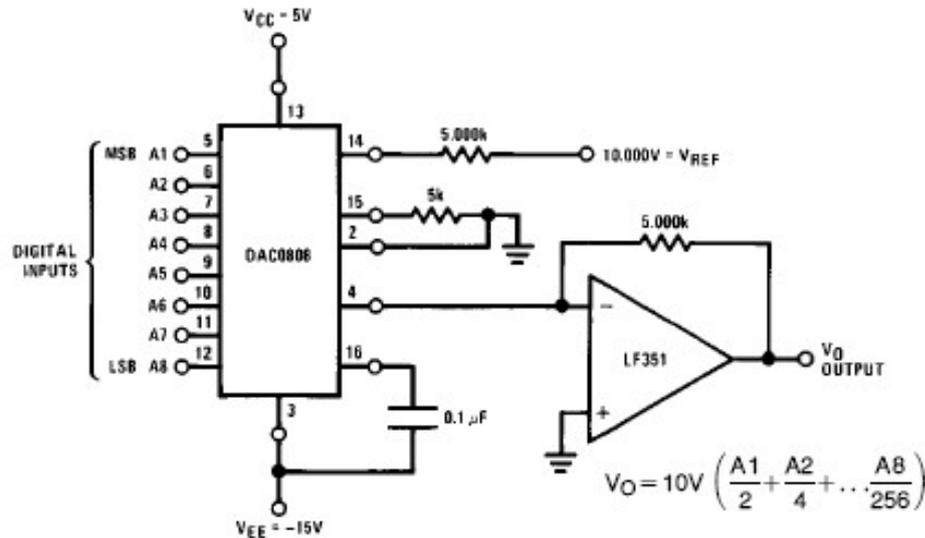


Figure III-13 : Schéma du câblage du DAC0808 avec l'amplificateur LF351 [5].

III-2-5-5 Principe de la transmission série (RS232)

La liaison série aux normes RS 232 est utilisée dans tous les domaines de l'informatique. Elle permet la réalisation d'une liaison simple à mettre en œuvre entre 2 équipements. Elle est de type asynchrone, Ils doivent par ailleurs utiliser le même protocole.

Elle est caractérisée par :

- Un niveau de tension de ± 25V.
- La gamme de débit est (100bits /sec à 20Kbits/sec).
- Distances utiles sont de 1m à 15m.
- Un émetteur par liaison.
- Un récepteur par liaison.



Figure III-14 : schéma fonctionnel d'une liaison série asynchrone de la norme RS232.

a) Brochage de la prise

Les prises RS232 sont aux formats DB9 ou DB25, nous nous intéresserons uniquement au format DB9. La figure ci-dessous présente des SUB-D 9 male et femelle. (Figure III-15)

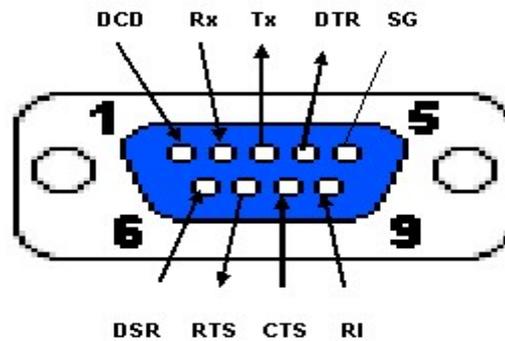


Figure III-15 : Les prises RS232.

Ces différentes broches sont décrites dans le tableau suivant :

Tableau III-3 : différentes broches des prises RS232.

N°	Signal	E/S	Utilisation
1	DCD	E	Carrier Detect : Annonce que l'autre équipement reçoit une réponse.
2	Rx	E	Received Data : Entrée de réception des données.
3	Tx	S	Transmitted Data : Sortie d'émission des données.
4	DTR	S	Data Terminal Ready : Indique à l'autre équipement que l'on souhaite communiquer.
5	SG		Signal Ground : Masse de référence des signaux (0V).
6	DSR	E	Data Set Ready : Indique que l'équipement opposé est prêt.
7	RTS	S	Request To Send : Demande à l'équipement opposé de se tenir prêt à recevoir.
8	CTS	E	Clear To Send : Indique que l'équipement opposé est prêt à recevoir.
9	RI	E	Ring Indicator : Annonce que le modem reçoit un appel.

Les deux broches principales sont les broches de transmission des données 2 et 3, la broche d'émission d'un équipement est reliée à la broche de réception de l'autre (câble croisé). Les autres broches sont utilisées pour les protocoles de transmission comme le montre la (figure III-16).

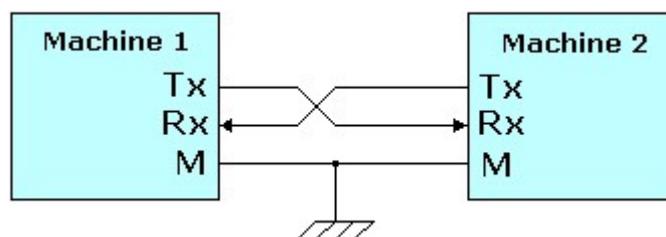


Figure III-16 : le branchement entre deux machines.

b) Protocole de transmission

Afin que les éléments communicants puissent se comprendre, il est nécessaire d'établir un protocole de transmission. Ce protocole devra être le même pour les deux éléments afin que la transmission fonctionne correctement.

Paramètres rentrant en jeu :

- Longueur des mots : 7 bits (ex : caractère ascii) ou 8 bits
- La vitesse de transmission : les différentes vitesses de transmission son réglables à partir de 110 bauds (bits par seconde) de la façon suivante : 110 bds, 150 bds, 300 bds, 600 bds, 1200 bds, 2400 bds, 4800 bds, 9600 bds.
- Parité : le mot transmis peut être suivi ou non d'un bit de parité qui sert à détecter les erreurs éventuelles de transmission. Il existe deux types de parité (parité paire et parité impaire).

- Bit de Start : la ligne au repos est à l'état logique 1 pour indiquer qu'un mot va être transmis. La ligne passe à l'état bas avant de commencer le transfert. Ce bit permet de synchroniser l'horloge du récepteur.
- Bit de stop : après la transmission, la ligne est positionnée au repos pendant 1,2 ou 1,5 périodes d'horloge selon le nombre de bits de stop.

Le bit de Start apparait en premier dans la trame puis les données (poids faible en premier), la parité éventuelle et le (les) bit(s) de stop.

Exemple

Soit à transmettre en parité paire, avec 2 bits de stop, le caractère B dont le codage ascii est 1000010_2 la trame sera la suivante (Figure III-17) :

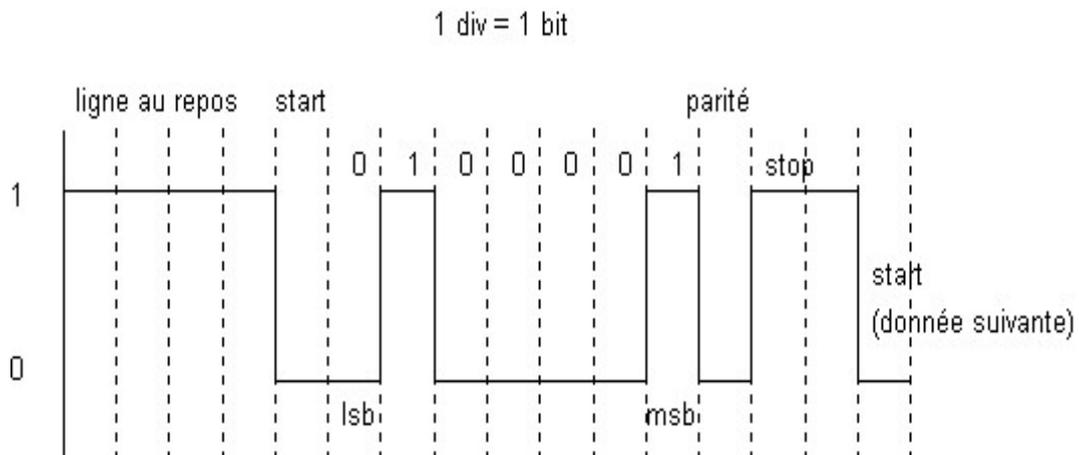


Figure III-17 : Exemple de transmission série [5].

Les deux organigrammes ci-dessous présenteront le principe de la transmission et la réception série que nous avons utilisé pour la programmation de notre PIC.

Évidemment le même principe sera utilisé pour le programme implanté sur Labview.

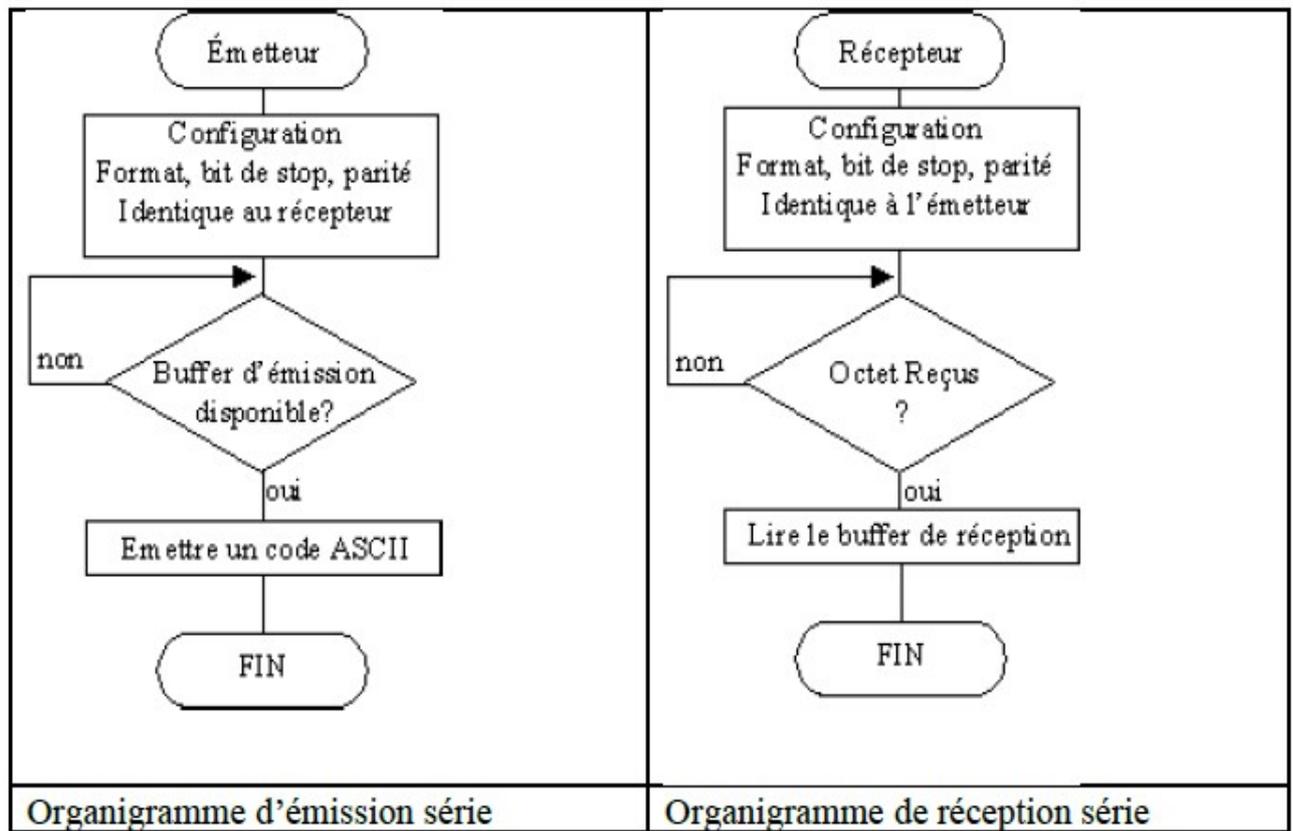


Figure III-18 : Organigramme de la transmission série [5].

c) Principe d'adaptation PIC- RS232

Le PIC utilise les niveaux 0V et 5V pour définir respectivement des signaux « 0 » et « 1 ». La norme RS232 définit des niveaux de +12V et -12V pour établir ces mêmes niveaux.

Nous aurons donc besoin d'un circuit chargé de convertir les niveaux des signaux entre PIC et PC. Le pin TX du PIC émettra en 0V/5V et sera convertie en +12V/-12V vers notre PC. La ligne RX du PIC recevra les signaux en provenance du PC, signaux qui seront convertis du +12V/-12V en 0V/5V par notre circuit de pilotage du bus.

Nous utiliserons le célèbre circuit MAX232 (voir annexe 1) pour effectuer cette adaptation de niveaux.

Ce circuit contient un double convertisseur à double direction. Autrement dit, il dispose de (figure III-19):

- 2 blocs, dénommés T1 et T2, qui convertissent les niveaux entrées en 0V/5V en signaux sorties sous +12V/-12V. En réalité, on n'a pas tout à fait +12V et -12V, mais plutôt de l'ordre de +8,5V/-8,5V ce qui reste dans la norme RS232.

- 2 blocs, dénommés R1 et R2, qui convertissent les niveaux entrées en +12V/-12V en signaux sorties sous 0V/5V.

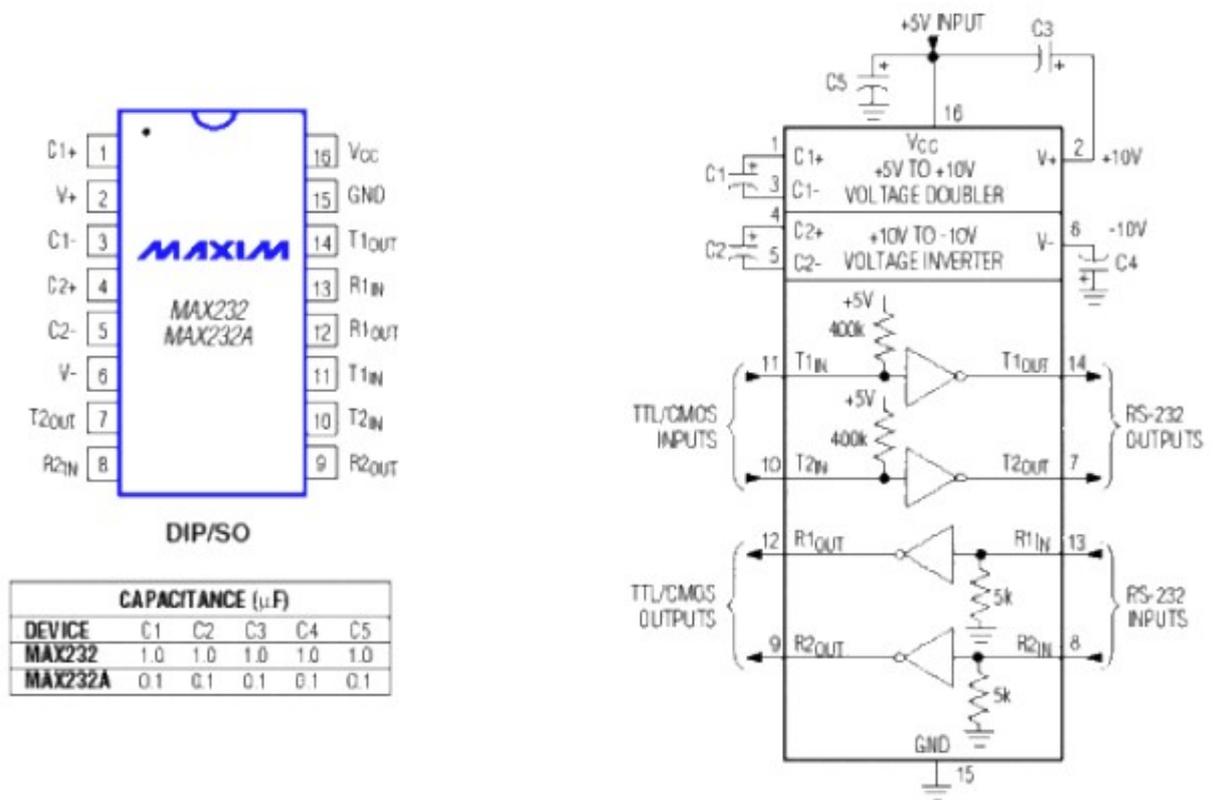


Figure III-19 : Circuit intégré MAX232 [5].

III-2-6 Commande et régulation dans LABVIEW

III-2-6-1 Présentation de LABVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) est un langage de programmation dédié au contrôle d'instruments et l'analyse de données. En Labview, on n'écrit pas de lignes de programme dans un langage textuel comme Pascal ou C. On manipule des objets graphiques. Ces objets représentent à la fois les variables du programme, ainsi que des fonctions qui vont réaliser des actions portant sur ces variables. La programmation en Labview consiste simplement à concevoir le traitement de l'information, organiser et relier les variables avec les fonctions au moyen des fils.

Lorsqu'on lance Labview, on obtient deux fenêtres de base (voir annexe 2) [17]:

- Fenêtre face avant : constituée de l'interface utilisateur.
- Fenêtre de diagramme : qui contient l'algorithme du programme décrit sous la forme d'un diagramme en langage graphique.

Pour écrire un programme sur Labview, on a besoins des « Palettes » qui nous offrent la possibilité de modifier la face avant et le digramme de Labview. On trouve trois palettes (voir annexe 2):

- Palette d'outils ;
- Palette de commandes ;
- Palette de fonctions .

III-2-6-2 Bibliothèque de commandes

Labview possède des bibliothèques de fonctions spécialisées dans le domaine de la mesure, du test et du contrôle-commande. Ces bibliothèques de fonctions peuvent être réalisées soit à partir des fonctions et operateurs de base, soit directement par des programmes écrits en langage de haut niveau, compilés et ensuite intégrés sous la forme d'un nœud graphique.

Dans notre projet, nous avons utilisé des bibliothèques nommées « PID control toolset » et « fuzzy control toolset » pour la commande et la régulation de notre système.

a) La bibliothèque de contrôle PID

Elle ajoute aux fonctions standards de Labview un ensemble d'algorithmes de régulation P, PI, PD et PID, avec toutes les options que l'on trouve généralement dans les systèmes de régulation complets.

Tous les régulateurs dans cette bibliothèque ont une structure parallèle de la forme :

$$C(p) = k_p + \frac{1}{pT_i} + pT_d \quad (III-5)$$

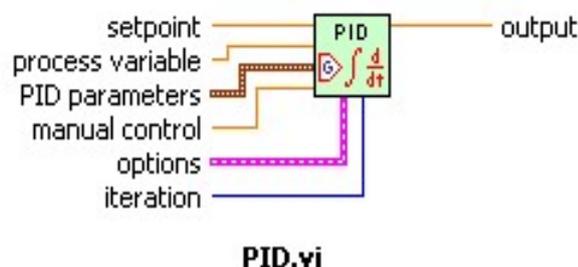


Figure III-20 : Exemple de bibliothèque de régulation PID [5].

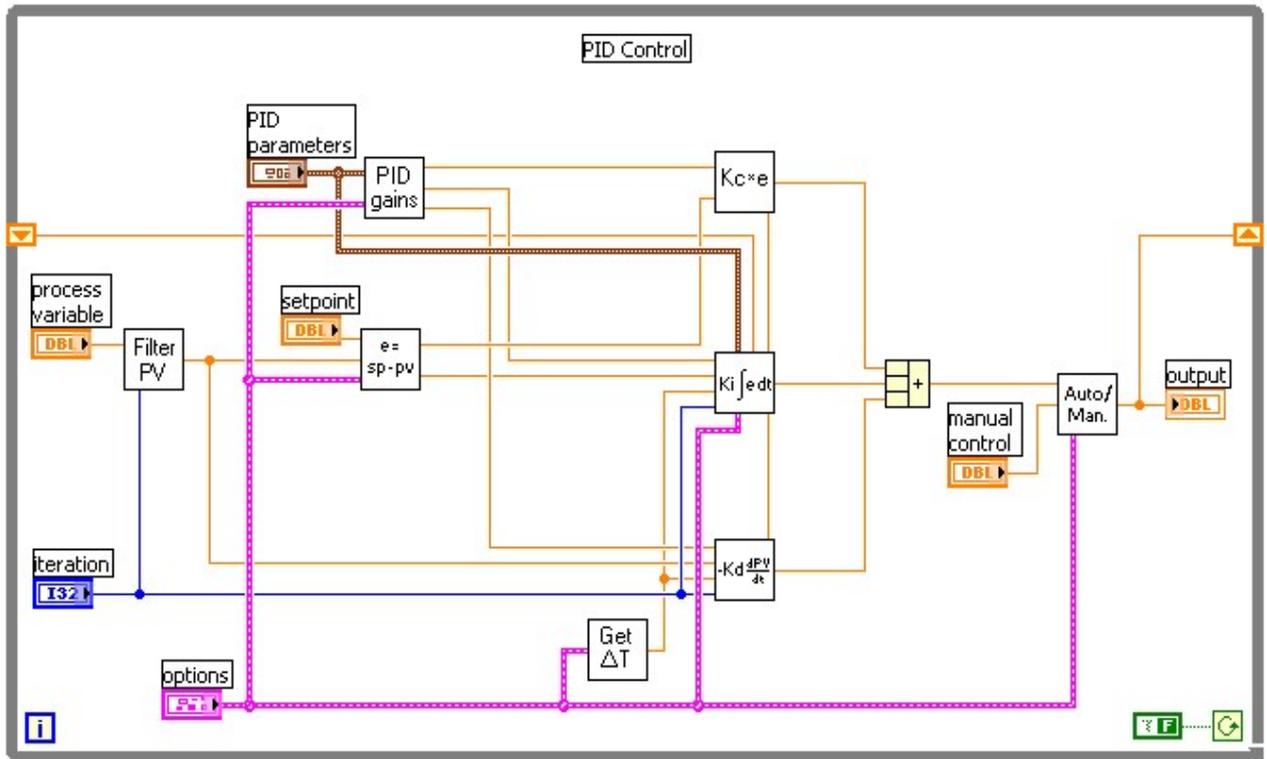


Figure III-21 : structure interne de PID [5].

b) La bibliothèque de logique floue

Elle est destinée à accélérer le développement d'applications de contrôle pour les systèmes non linéaires ou complexes. Elle propose une interface graphique pour la conception de contrôleurs de logique floue et des instruments virtuels (VI) pour créer ces contrôleurs dans Labview.

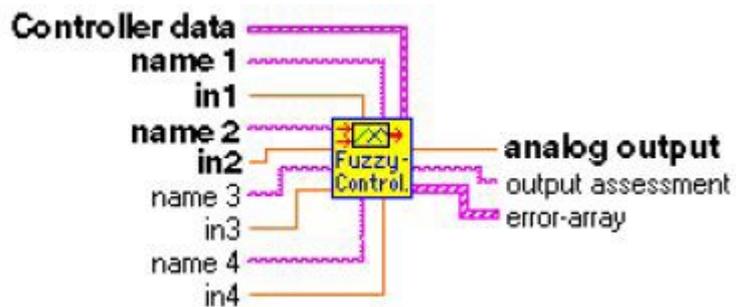


Figure III-22 : Exemple de Bibliothèque de la commande floue [5].

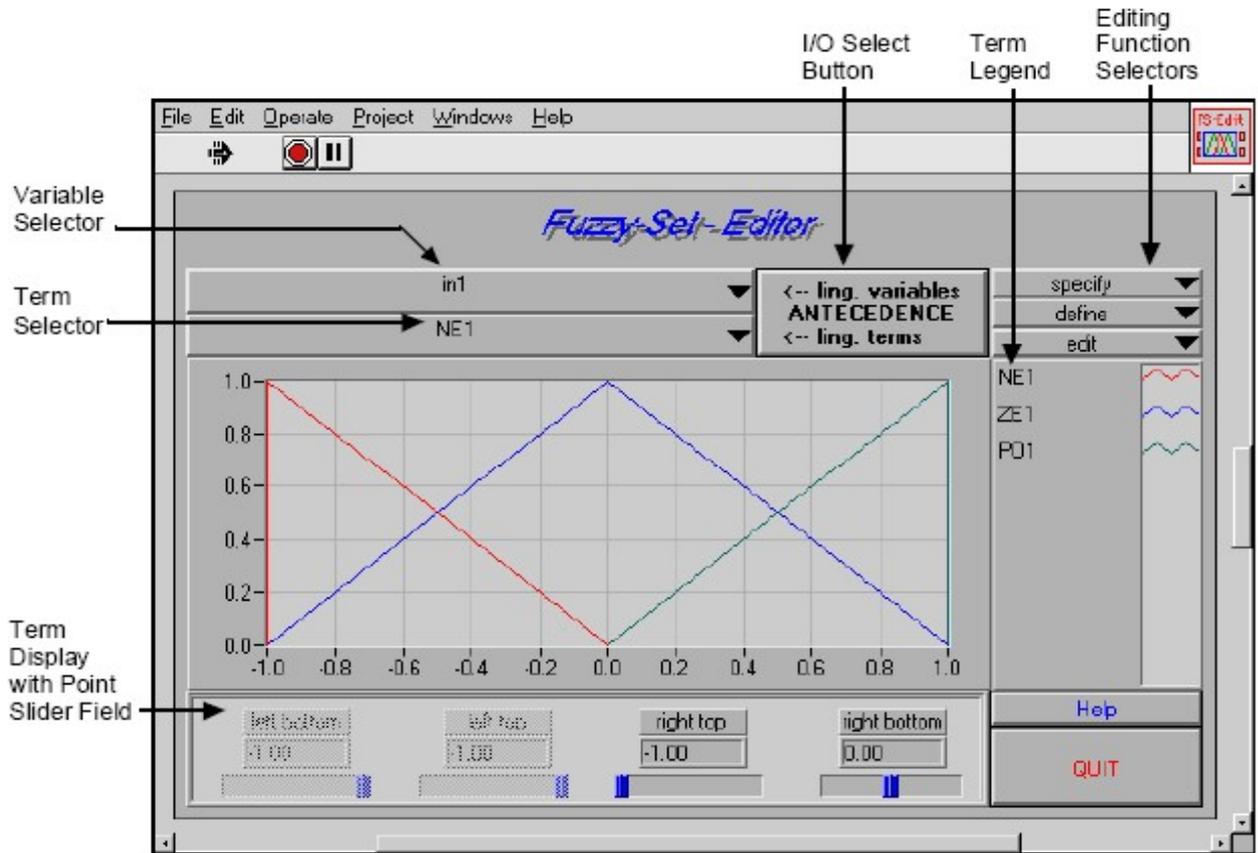


Figure III-23 : Interface graphique pour la conception d'une commande floue [5].

III-2-6-3 Transmission série dans Labview

a) Bibliothèque RS232

La liaison série utilise un émetteur pour envoyer les données les unes derrière les autres, bit par bit, sur une ligne de communication unique, à destination d'un récepteur. Cette technique de communication est classique pour les transferts à faible vitesse ou sur de longues distances.

Etant donnée la simplicité de la liaison série, la bibliothèque pour le port série se résume à cinq instruments virtuels (VI): Serial Port Init, Serial Port Write, Port Read, Bytes at Serial Port et Serial Port close.

Un programme de gestion de port série, réalisé en langage graphique dans l'environnement Labview est donc très simple puisqu'il consiste à mettre en séquence les différents VI de base.

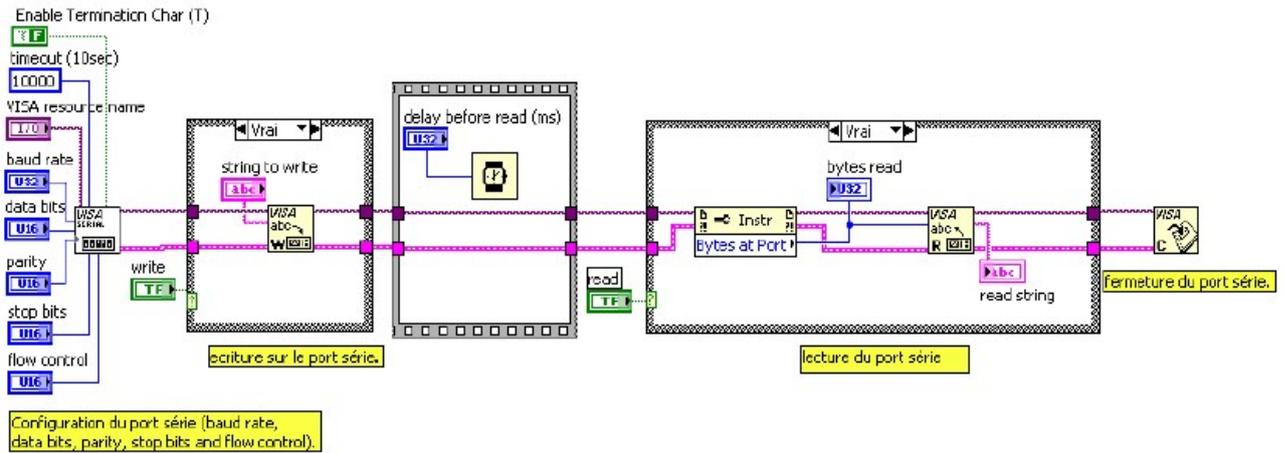


Figure III-24 : Bibliothèque du port série sur Labview [5].

III-2-7 Conclusion

Après la description des modules de contrôle, la régulation de niveau d'eau peut être réalisée. Le chapitre suivant sera donc consacré à la réalisation des différentes cartes électroniques nécessaires pour notre projet et à la programmation.

IV-1 Introduction

Après avoir présenté au chapitre III le système de régulation de niveau d'eau, nous allons dans ce chapitre aborder la réalisation pratique de différentes cartes électroniques qui composent notre système, ainsi qu'on va appliquer les différents modules de régulation (PID et flou) sur Labview.

IV-2 Présentation du logiciel PROTEUS

Dans cette partie nous allons simuler notre projet à l'aide du logiciel spécialisé « ISIS » (figure IV-1). En fait ce logiciel nous permet de faire fonctionner nos cartes de manière virtuelle grâce à son système performant. Il permet aussi de faire le schéma imprimé des cartes réalisées grâce à l'éditeur ARES trouvé sur son menu.

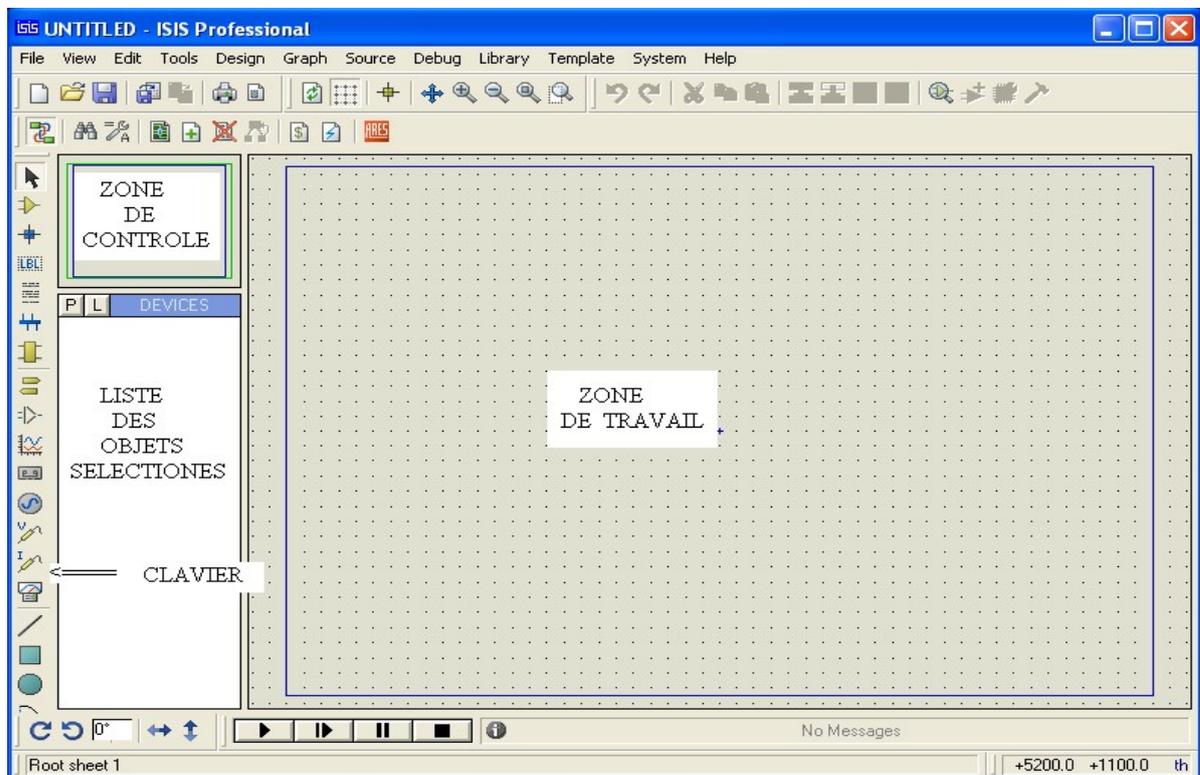


Figure IV-1: menu de proteus.

IV-3 Implémentation des différentes cartes réalisées sur PROTEUS

Les figures suivantes représentent les différentes cartes réalisées sur Proteus suivant la description faite au chapitre III. Le schéma des circuits imprimés de ces cartes et d'implémentations réelles sont données dans l'annexe 3.

a) Cartes d'alimentations

La Figure IV-2 représente la première carte d'alimentation dont le rôle est d'alimenter les différents composants de la carte d'interface et la carte de mesure.

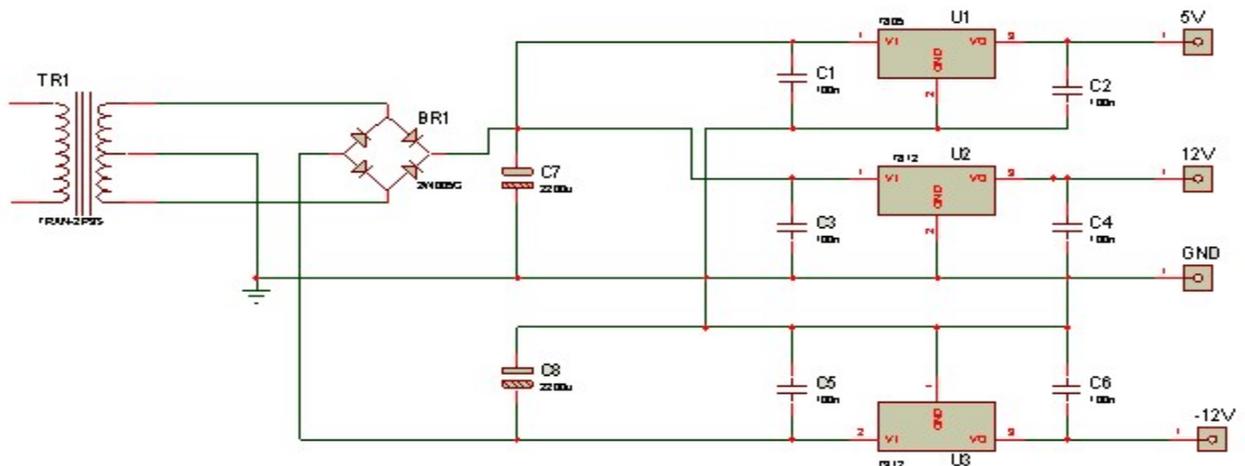


Figure IV-2: 1^{ère} carte d'alimentation.

La deuxième carte d'alimentation a été conçue pour donner la puissance nécessaire à l'actionneur (Pompe). (Figure IV-3)

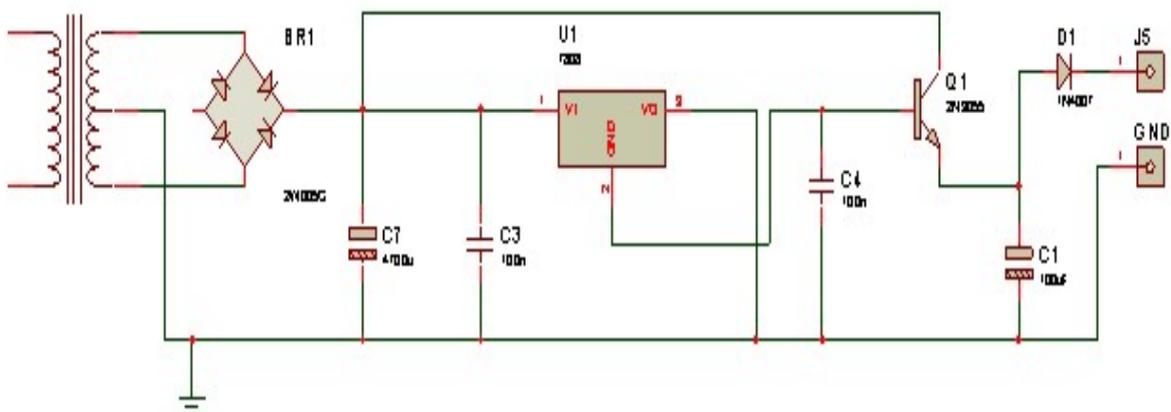


Figure IV-3: 2^{ème} carte d'alimentation.

b) Carte de mesure

Cette carte (Figure IV-4) contient 4 parties (capteur, réglage d'offset, amplification et filtrage) qui sont expliquées dans le chapitre précédent.

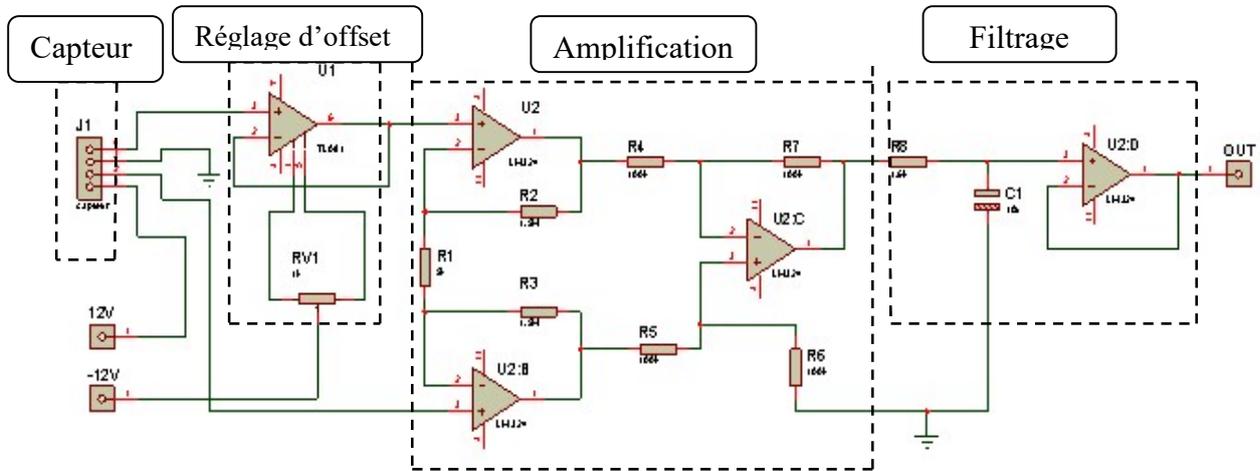


Figure IV-4: carte de mesure.

c) Carte d'interface avec la carte de commande

Cette carte (Figure IV-5) assure la liaison entre le PC et notre système.

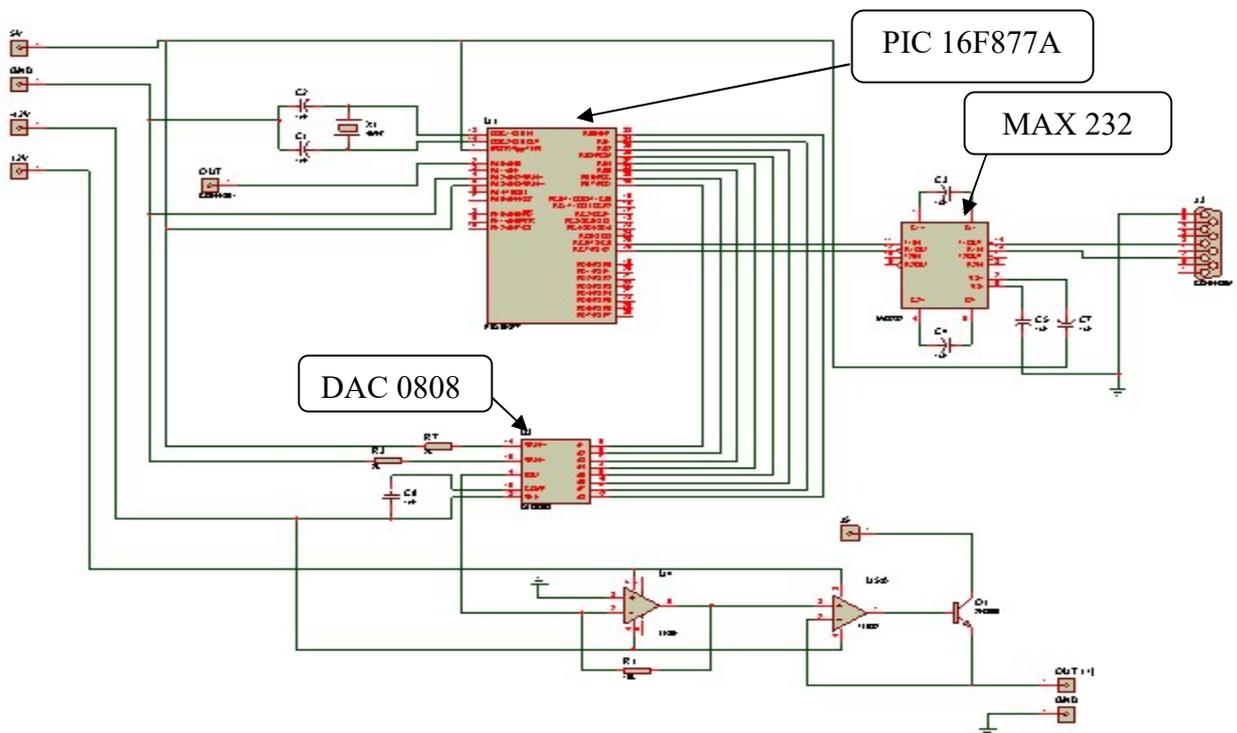


Figure IV-5: Carte d'interface avec la carte de commande.

IV-4 Application de la régulation PID dans Labview

IV-4-1 Boucle de régulation

Pour expliquer bien le programme conçu pour cette commande, on a fait un schéma décrivant ses différentes étapes : (Figure VI-6).

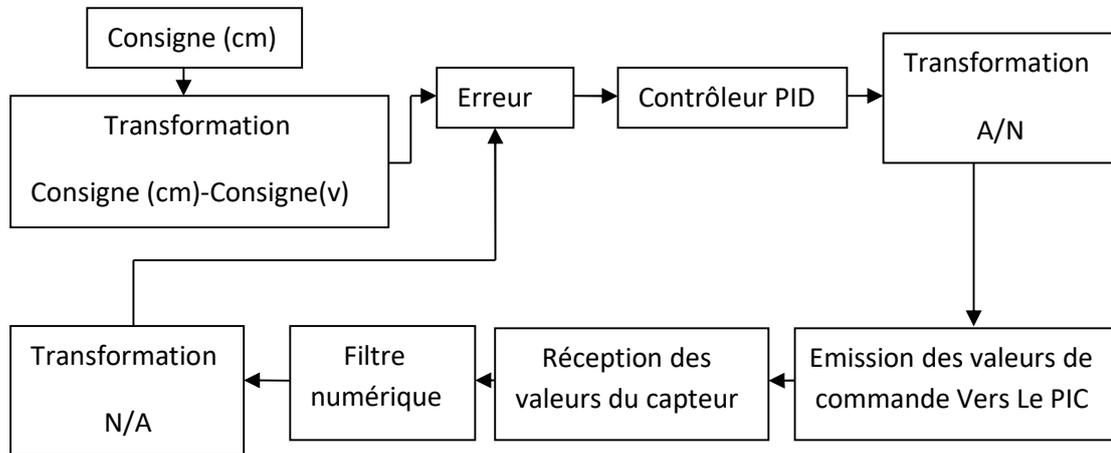


Figure IV-6: schéma descriptif du programme.

Le diagramme final pour la commande PID dans Labview est le suivant : (Figure VI-7).

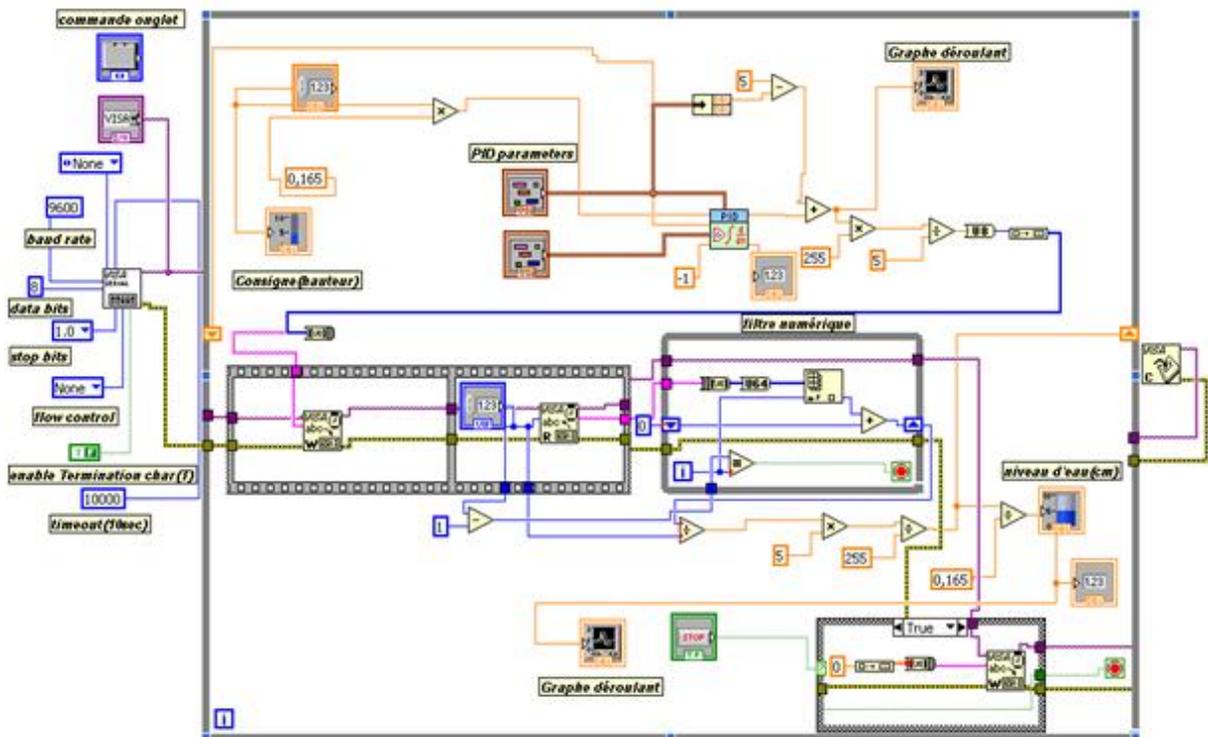


Figure IV-7 : Diagramme de commande PID.

tests

Dans notre programme il y a deux interfaces, l'une pour saisir les entrées (Figure VI-9) et l'autre pour afficher les informations (Figure VI-8).

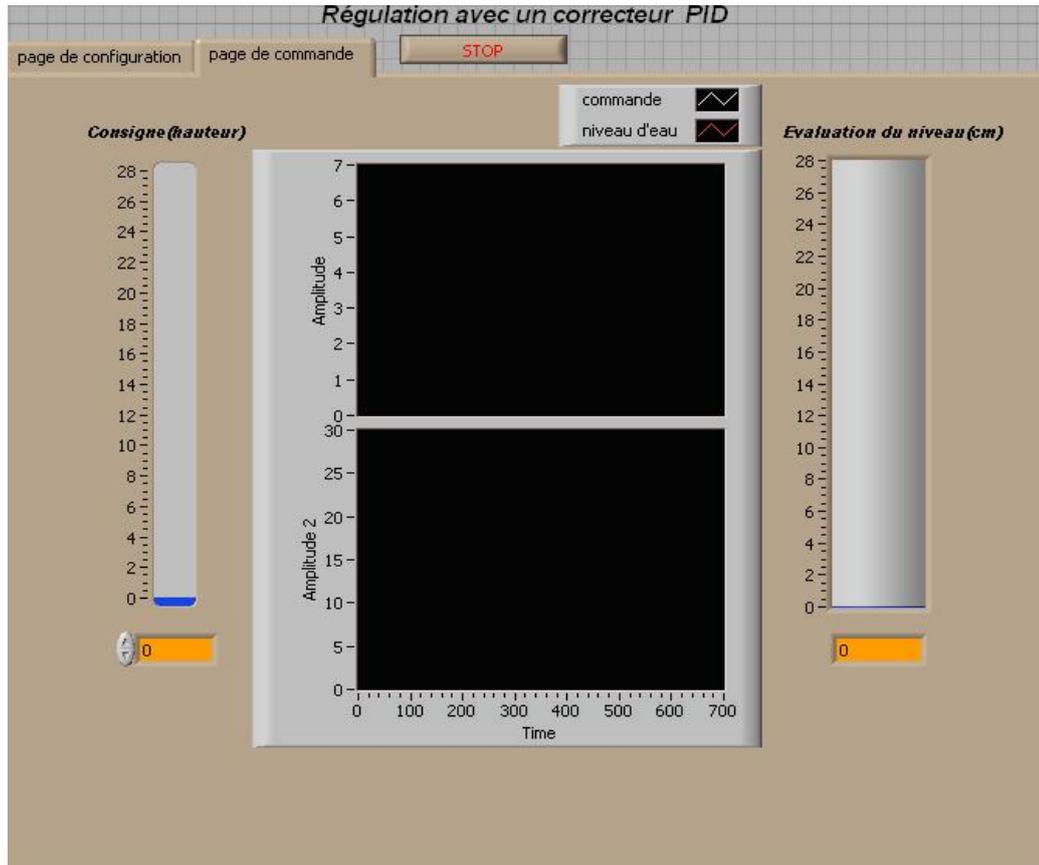


Figure IV-8 : Menu d'affichage des informations

IV-4-2 Configuration des paramètres

On fixe la consigne désirée et les paramètres du PID. Pour notre application, on a choisit $K_p=10$, $T_i=0.5$ min, $T_d=0$ min, et le nombre d'échantillons pour le filtre numérique $n=40$. La détermination de ces paramètres a été faite grâce à la méthode des approximations successives. Cette méthode consiste à faire des tests sur chacune des actions du régulateur indépendamment des autres actions.

Nous avons configuré le port série comme suit :

- Port de communication : port1 ;
- Vitesse de transmission : 9600 baud ;
- Nombre de bit à transmettre : un octet ;
- Sans bit de parité ;
- Un seul bit de stop ;
- Sans contrôle de flux.

Nous avons conçu une page spéciale pour mettre ces informations (Figure VI-9).

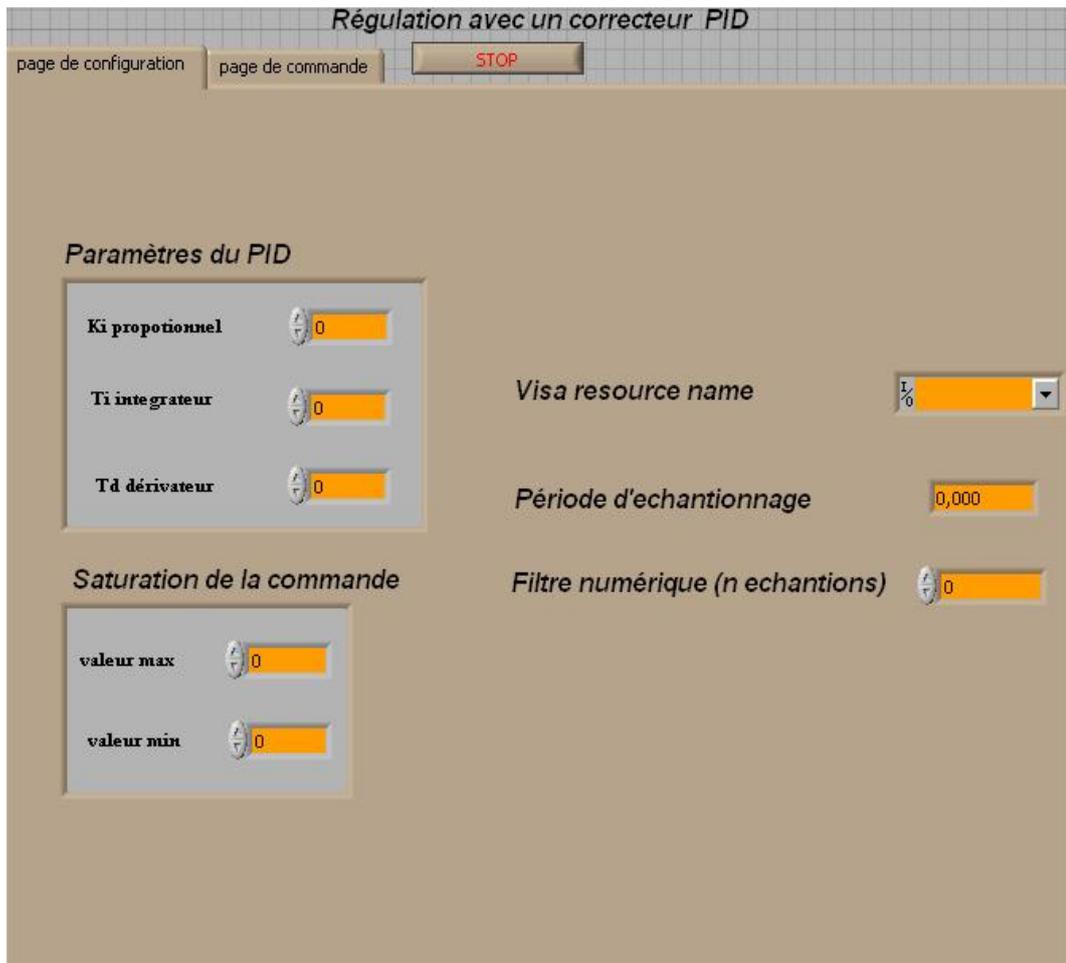


Figure IV-9 : Page de configuration des paramètres du contrôleur PID.

IV-4-3 Résultats expérimentaux

Prenons l'exemple suivant :

On fixe la consigne sur la valeur 10cm et on fait les deux tests de l'asservissement et de régulation (c'est-à-dire, on introduit une perturbation).

A) En asservissement

- Le 1^{er} cas présente un mode d'asservissement normal (suivi de la consigne initiale) (Figure VI-10).
- Le 2^{ieme} cas présente un changement de consigne de 10 cm à 14 cm. On remarque que la commande change au même temps qu'il y a changement de la consigne (Figure VI-11).

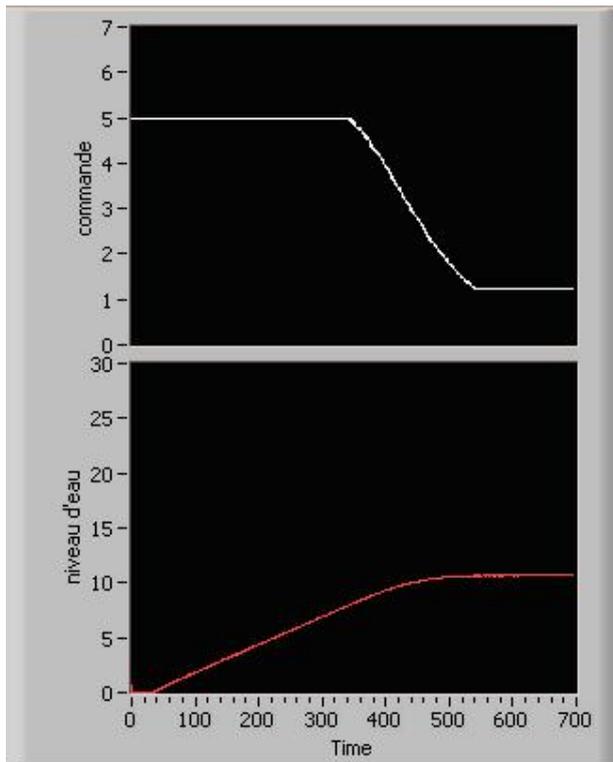


Figure IV-11 : Réponse du système après une consigne 10cm.

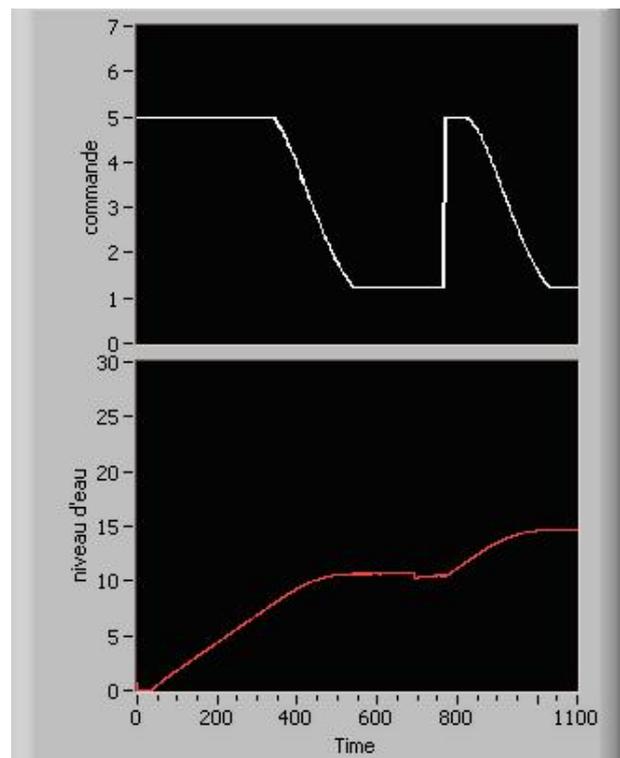
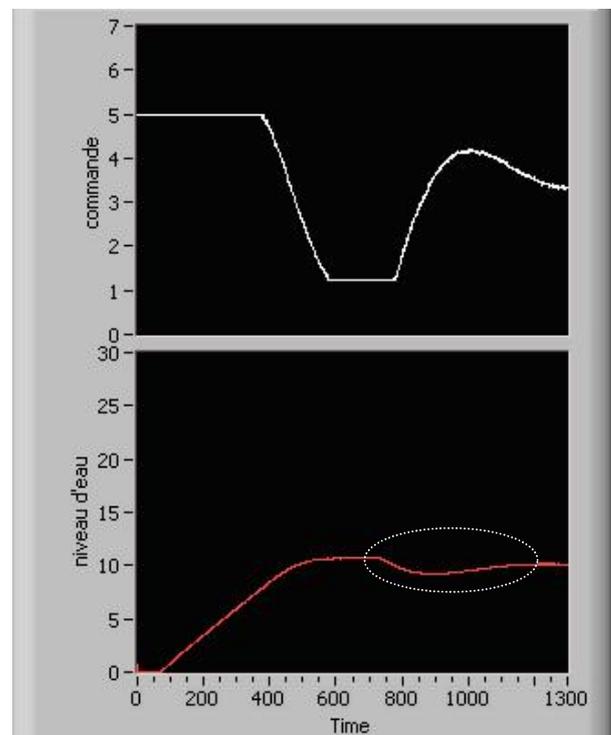


Figure IV-10 : Réponse du système à un changement de consigne de 10 à 14 cm.

B) En régulation

- Dans ce cas, on a procédé une fois que l'état stable de la commande obtenu, à ouvrir la vanne (de 50 à 100%) pour vider le réservoir du haut. On effectue donc une perturbation sur le système.
- On remarque que la commande augmente pour corriger cette perturbation et garder le même état initial (Figure VI-12).

Figure IV-12 : Réponse du système après Une perturbation sur la vanne.



tests

- On observe pour le premier résultat (figure VI-10) qu'il y a un dépassement dû à l'action intégrale (un inconvénient de l'action intégrale). Ce dernier varie de 0.1 à 0.4 cm.
- La rapidité du système est acceptable pour ce type de procédé, la réponse du système à 5% est 144 seconde ($480 \times T_e$). Avec $T_e = 0.3$ s.
- Le système réagit rapidement pour une autre consigne (figure VI-11), avec même erreur statique.
- Dans le cas de la régulation, on remarque une annulation d'erreur statique, c'est l'avantage de l'action intégrale.

IV-5 Application de la régulation floue dans Labview

IV-5-1 Boucle de régulation

L'application de cette structure dans Labview donne le diagramme suivant : (Figure VI-13)

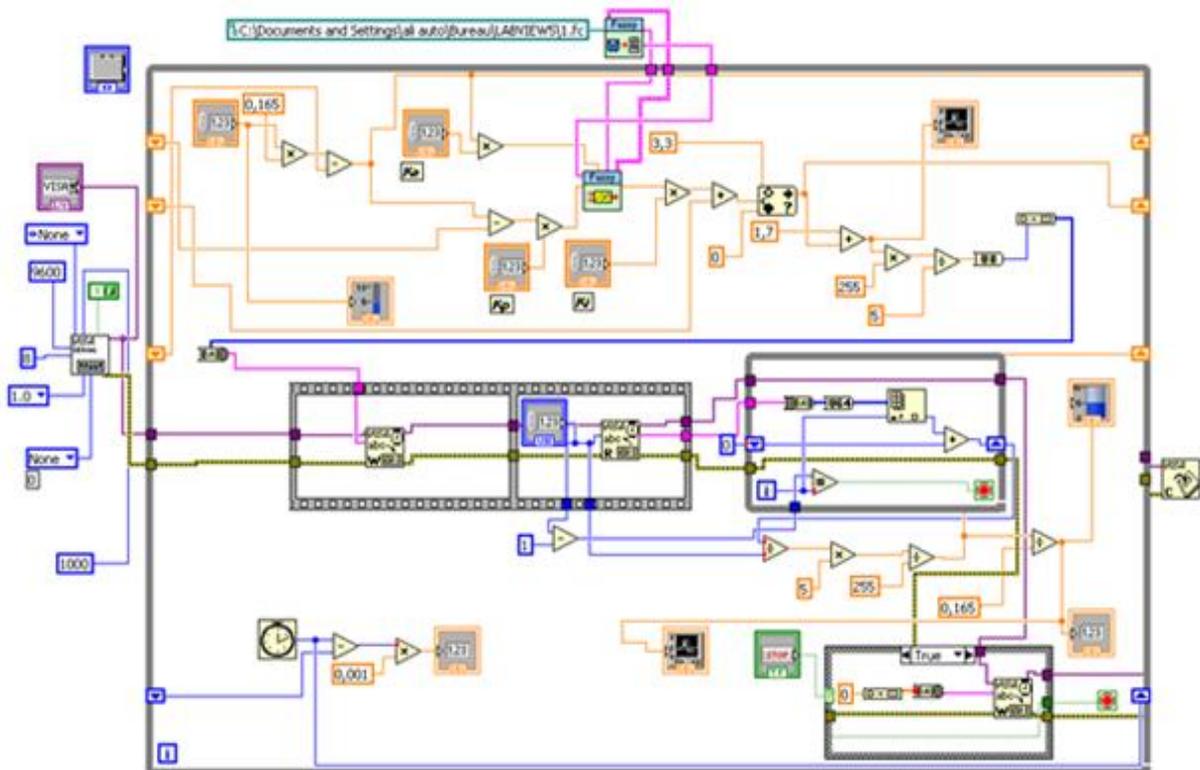


Figure IV-13 : Diagramme de la commande floue sur Labview.

IV-5-2 Configuration des paramètres flous

Les fonctions des appartenances utilisées dans cette commande sont les suivantes : (Figure IV-14, IV-15 et IV-16).

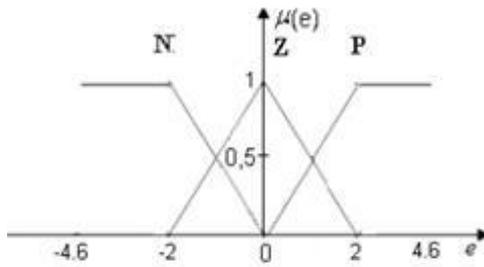


Figure IV-14 : Fonction d'appartenance

Pour l'erreur

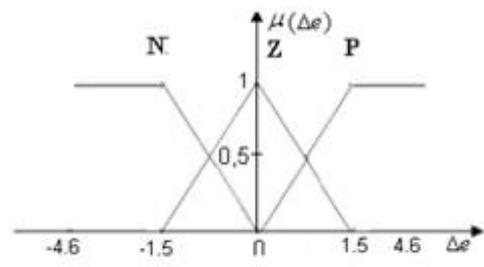


Figure IV-15 : Fonction d'appartenance

Pour la variation d'erreur

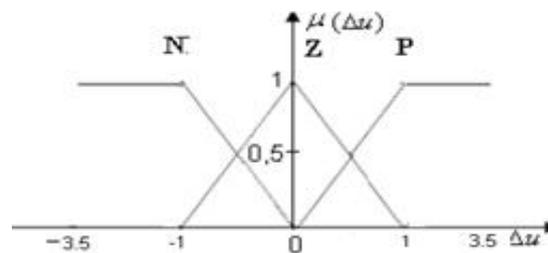


Figure IV-16: Fonction d'appartenance pour la variation de la commande.

Avec :

- $\mu(e)$: fonction d'appartenance pour l'erreur
- $\mu(\Delta e)$: fonction d'appartenance pour la variation d'erreur
- $\mu(\Delta u)$: fonction d'appartenance pour la variation de la commande
- e : l'erreur
- Δe : la variation d'erreur
- Δu : la variation de la commande

La forme triangulaire est choisie pour la fonction d'appartenance de chaque entrée et même pour la sortie Δu (Figure IV-14, IV-15 et IV-16). La méthode de défuzzification est celle du centre de gravité. On résume la base des règles dans le tableau ci-dessous : (Tableau IV-1).

Tableau IV-1: base des règles.

$e / \Delta e$	N	Z	P
N	N	N	Z
Z	N	Z	P
P	Z	P	P

Avec :

- N : Négative
- Z : Zéro
- P : Positive

tests

La configuration du port série est la même pour le contrôleur PID. L'interface graphique sur Labview qui nous permet de saisir ces configurations est la suivante : (Figure IV-17). Dans notre projet on a choisit les gains suivants :

$$K_e = 0.2, K_{\Delta e} = 3, K_{\Delta u} = 0.8$$

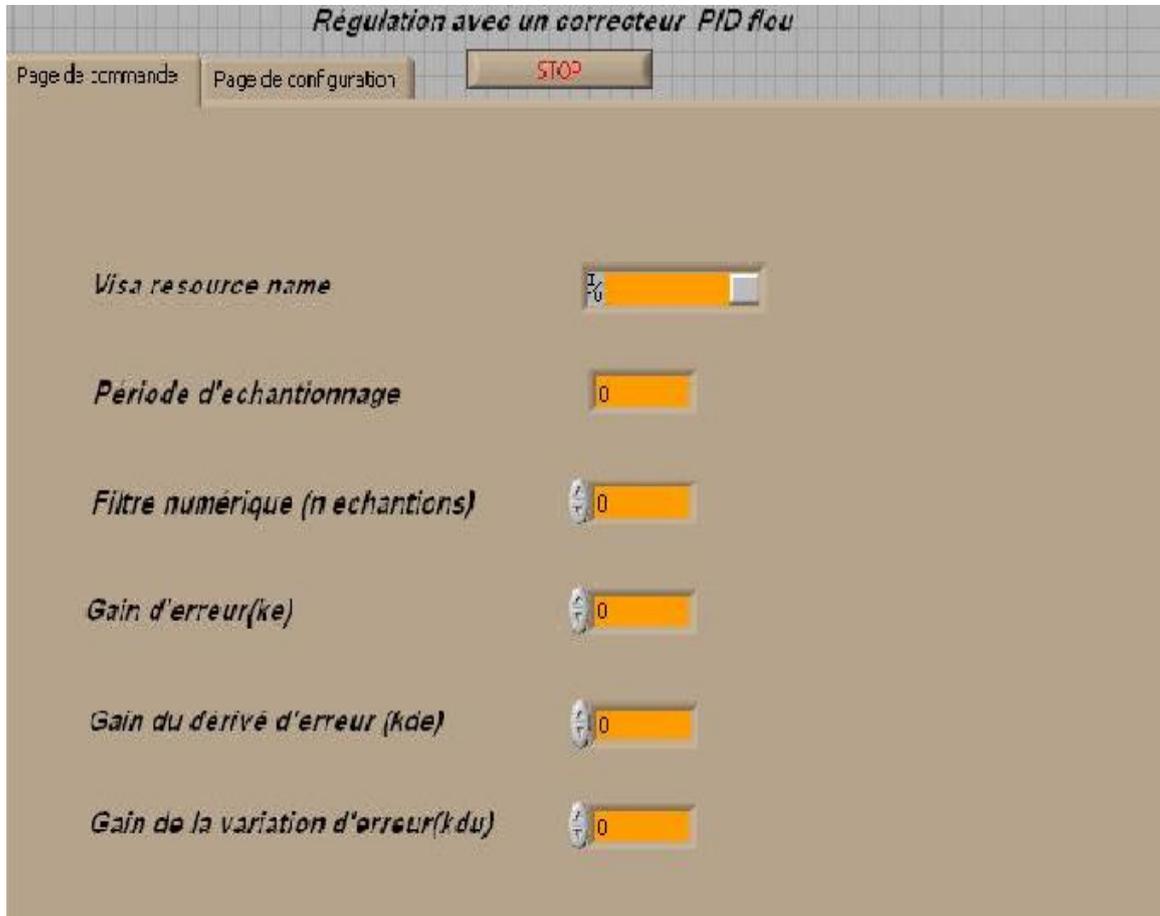


Figure IV-17: Page de la commande floue sur Labview.

IV-5-3 Résultats expérimentaux

Prenons le même exemple que la régulation PID. Les résultats obtenus sont les suivants :

A) En asservissement

On fait les mêmes étapes pour cette commande :

- Le 1^{er} cas présente un mode d'asservissement normal. (Figure IV-18).
- Le 2^{ième} cas présente un changement de consigne de 10 cm à 14 cm. (Figure IV-19).

tests

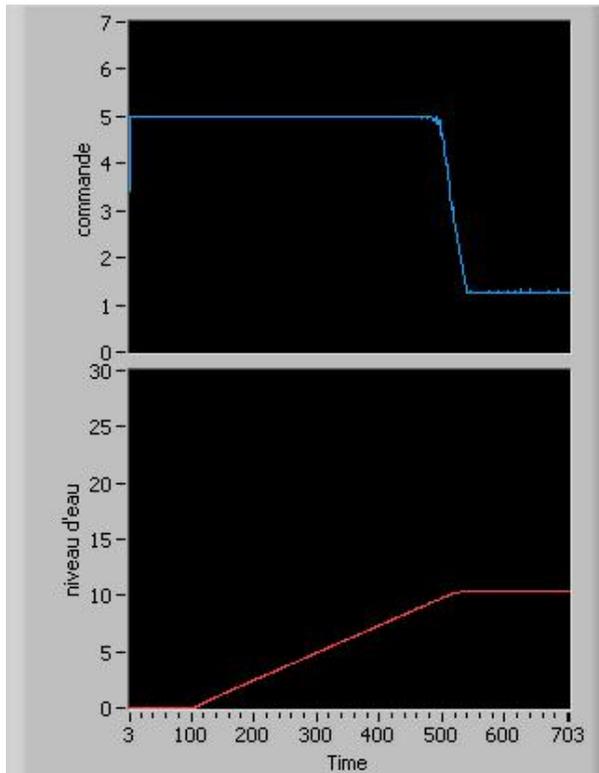


Figure IV-18: Réponse du système après après une consigne 10cm.

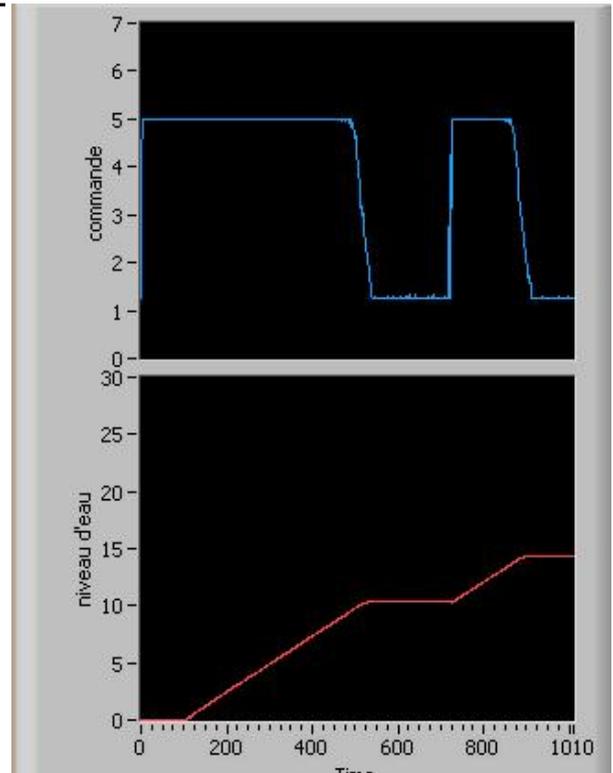
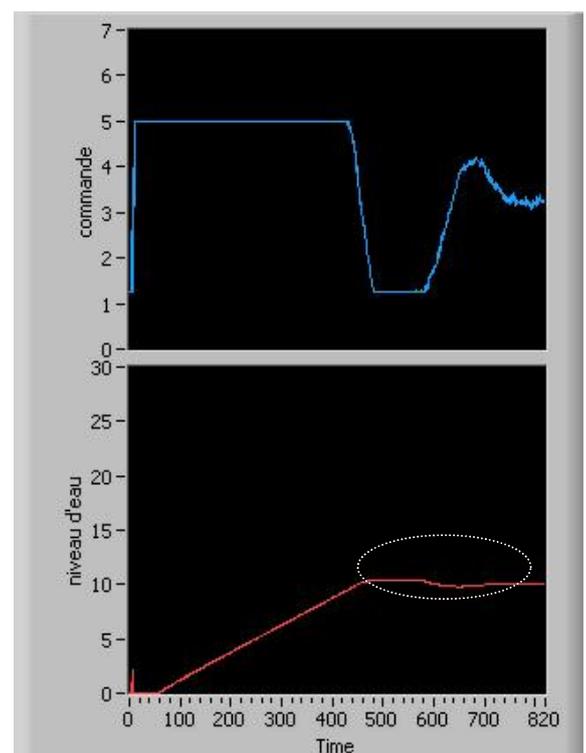


Figure IV- 19: Réponse du système un changement du consigne de 10 à 14 cm

B) En régulation

- Dans ce cas, on a procédé une fois que l'état stable de la commande obtenu, à ouvrir la vanne (de 50 à 100%) pour vider le réservoir du haut. On effectue donc une perturbation sur le système.
- On remarque que la commande augmente pour corriger cette perturbation et garder le même état initial (Figure IV-20).

Figure IV-20 : Réponse du système après Une perturbation sur la vanne.



Les remarques qu'on peut tirer de cette régulation pas rapport à celle de PID sont les suivantes :

- Une petite erreur statique ;
- Système plus rapide;
- Dans le cas d'asservissement, le système réagi mieux ;
- La réaction du système avec la perturbation est plus rapide ;
- La commande FLOUE est donc plus robuste que la PID mais la détermination de ces paramètres nécessite beaucoup de tests.

IV-6 Conclusion

La réalisation de notre montage a été faite à 100%, le montage est fonctionnelle et répond à la consigne désirée selon la commande choisie.

La réalisation a été faite partie par partie dans le but d'obtenir un bon pourcentage de précision, d'une grande stabilité malgré les obstacles rencontrés, et de connaître facilement l'emplacement de l'erreur en cas de dysfonctionnement (et le corriger). Les tests seront alors plus faciles à appliquer afin d'avoir un gain en temps.

La partie pratique doit être précise :

- Partie électronique : circuits imprimés et connexion.
- Partie matérielle : fixation des réservoirs, la règle, et la pompe, ...
- Partie soft : qui a une grande importance pour la réalisation des différents objectifs

On a présenté dans ce chapitre les différents essais pratique qu'on a faits (des essais avec régulateur PID et autres avec régulateur PID flou) et les graphes obtenus à travers ces essais.

Finalement, on peut tirer de ce chapitre que la commande PI flou corrige mieux la perturbation et elle répond plus rapidement que la commande PI classique.

CONCLUSION GENERALE

Dans ce projet, nous avons fait une régulation de niveau d'eau dans un réservoir assisté par le logiciel LABVIEW, ce qui permet la mise au point d'un système de contrôle de niveau d'eau.

Nous avons réalisé un modèle intelligent qui permet le contrôle en temps réel des différents paramètres rentrant dans le bon fonctionnement de ce procédé. Ce modèle est relié par le bus RS 232 qui permet l'acheminement des données jusqu'à l'ordinateur.

D'autre part nous avons appris tout au long de la réalisation de ce travail à programmer le PIC par le compilateur MPLAB, à maîtriser le bus RS 232 et à manipuler les logiciels PROTEUS et LABVIEW.

Une suite de ce travail peut être envisagée par l'automatisation de la technique et ceci par l'ajout d'un capteur de pression qui effectue les mesures nécessaires de niveau d'eau, et une pompe électrique commandée pour le remplissage de l'eau de réservoir bas au réservoir haut.

Finalement, Ce travail a été une occasion de concrétiser et de mettre en évidence nos compétences acquises au cours de notre formation académique avec notamment le parcours de thématique à savoir l'électronique, l'automatique, l'instrumentation et de consolider ainsi notre formation.

LISTES DES FIGURES ET TABLEAUX

Liste des figures

Figure I-1: La chaîne de régulation.....	4
Figure I-2 : Comportement en régulation.....	4
Figure I-3 : Comportement en asservissement.....	5
Figure I-4 : caractéristique d'une régulation.....	5
Figure I-5 : Schéma d'un système en Boucle Ouverte.....	6
Figure I-6: schéma général d'une boucle fermée.....	6
Figure I-7 : exemple de régulation en cascade.....	7
Figure I-8 : exemple de régulation mixte.....	7
Figure I-9 : exemple de régulation rapport.....	8
Figure I-10 : fonction de transfert.....	8
Figure I-11 : réponse à un échelon d'un procédé intégrateur.....	9
Figure I-12 : Schéma générale d'une boucle de regulation.....	10
Figure I-13 : Schéma bloc d'un système avec correcteur (Boucle fermée).....	11
Figure I-14: Structure d'une commande floue.....	15
Figure II-1: description de la configuration du PIC 16F877.....	18
Figure II-2 : Schéma réel du 16F877.....	18
Figure II-3: Les différentes broches du PIC 16F877A.....	20
Figure II-4 : Principe de conversion analogique / numérique.....	22
Figure II-5 : schéma interne des entrées du CAN.....	23
Figure II-6 : schéma des registres rentrant dans la transmission USART.....	24
Figure II-7: l'éditeur de MPLAB.....	25
Figure II-8 : Le programmeur du PIC16F877A.....	26
Figure II-9 : Menu principal du logiciel WINPRG.....	26
Figure II-10 : Menu principal du logiciel WINPRG.....	27
Figure III-1 : Structure de la maquette.....	28
Figure III-2: schéma synoptique de 1ere alimentation.....	30
Figure III-3: schéma synoptique de 2 ^{ème} alimentation.....	31

Figure III-4: Schéma équivalent du capteur de pression.....	32
Figure III-5 : Amplificateur d'instrumentation.....	33
Figure III-6 : courbe d'étalonnage.....	34
Figure III-7 : La pompe.....	35
Figure III-8 : Schéma de l'amplificateur de puissance.....	36
Figure III-9 : L'organigramme d'acquisition de la mesure.....	37
Figure III-10: Schéma d'un convertisseur numérique analogique.....	38
Figure III-11: Schéma de principe d'un CNA de courant.....	38
Figure III-12 : Brochage du DAC0808.....	39
Figure III-13 : Schéma du câblage du DAC0808 avec l'amplificateur LF351.....	40
Figure III-14 : schéma fonctionnel d'une liaison série asynchrone de la norme RS232.....	40
Figure III-15 : Les prises RS232.....	41
Figure III-16 : le branchement entre deux machines.....	42
Figure III-17 : Exemple de transmission série.....	43
Figure III-18 : Organigramme de la transmission série.....	43
Figure III-19 : Circuit intégré MAX232.....	44
Figure III-20 : Exemple de bibliothèque de régulation PID.....	46
Figure III-21 : structure interne de PID.....	46
Figure III.22 : Exemple de Bibliothèque de la commande floue.....	47
Figure III-23 : Interface graphique pour la conception d'une commande floue.....	47
Figure III-24 : Bibliothèque du port série sur Labview.....	48
Figure IV-1: menu de proteus.....	49
Figure IV-2: 1 ^{ere} carte d'alimentation.....	50
Figure IV-3: 2 ^{eme} carte d'alimentation.....	50
Figure IV-4: Carte de mesure.....	51
Figure IV-5: Carte d'interface avec carte de commande.....	51
Figure IV-6: Schéma descriptif du programme.....	52
Figure IV-7 : Diagramme de commande PID.....	52
Figure IV-8 : Menu d'affichage des informations	53
Figure IV-9 : Page de configuration des paramètres du contrôleur PID.....	54
Figure IV-10 : Réponse du système à une consigne 10cm.....	55
Figure IV-11 : Réponse du système après un changement du consigne de 10 à 14 cm.....	55

Figure IV-12 : Réponse du système après une perturbation sur la vanne.....	55
Figure IV-13 : Diagramme de la commande floue sur Labview.....	56
Figure IV-14: Fonction d'appartenance Pour l'erreur.....	57
Figure IV-15 : Fonction d'appartenance Pour la variation d'erreur.....	57
Figure IV-16: Fonction d'appartenance Pour la variation de la commande.....	57
Figure IV-17: Page de la commande floue sur Labview.....	58
Figure IV-18: Réponse du système à une consigne 10cm.....	59
Figure IV-19 : Réponse du système après un changement du consigne de 10 à 14 cm.....	59
Figure IV-20: Réponse du système après une perturbation sur la vanne.....	59

Liste des tableaux

Tableau I-1 : Différentes structures du régulateur PID.....	13
Tableau III-1 : Variation de la mesure en fonction du niveau.....	34
Tableau III-2 : Caractéristiques du DAC0808.....	39
Tableau III-3 : différentes broches des prises RS232.....	41
Tableau IV-1: base des règles.....	57

NOMENCLATURE

1ere carte d'alimentation

TR1 : Transformateur 220V/12V et courant de 1A ;

BR1 : Pont redresseur ;

C1=C2=C3=C4=C5=C6=100 nF ;

C7=C8=2200 uF ;

U1 :7805 ;

U2 : 7812 ;

U3 : 7912.

2eme carte d'alimentation

TR1 : Transformateur 220V/12v et courant 1.5A ;

BR1 : Pont redresseur ;

C1=C4=C3=100 nF ;

C7=4200 uF ;

U1 : 7808 ;

Q1 : Transistor 2N3055 ;

D1 : Diode IN5408.

Carte de mesure

J1 : Capteur de pression différentielle **HONEYWELL 26PCAFA6D** ;

U1 : TL081 ;

U2, U2b, U2c, U2d : LM324 ;

RV : Resistance variable $R=100\text{ K}\Omega$;

$R1=5\text{ K}\Omega$;

$R2=R3=1.5\text{ M}\Omega$;

$R4=R5=R6=R7=100\text{ K}\Omega$;

$R8=1.6\text{ K}\Omega$;

$C1=10\text{ uF}$.

Carte d'interface

U1 : PIC 16F877A ;

X1 : Horloge 4MHZ ;

$C1=C2=22\text{ pF}$;

U2 : MAX232 ;

$C3=C4=C5=C7=1\text{ uF}$;

J3 : Porte DB9 mal ;

U3 : DAC0808 ;

$R7=R2=2\text{ K}\Omega$;

$C8=100\text{ nF}$;

U4 : LF351 ;

$R1=10\text{ K}\Omega$;

U5 : TL082 ;

Q1 : Transistor 2N3055.

REFERENCES BIBLIOGRAPHIQUES

- [1] : B. Bouamama, *Cour Introduction à l'automatique*. [http:// polytech-lille.fr/cours-régulation-automatique/](http://polytech-lille.fr/cours-régulation-automatique/).
- [2] : G. Binet, *Commande numérique par placement de pôles*. Université de Caen, Paris, 2010-2011
- [3] : M. Achaintre, L. Desgranges, *Boucle de régulation de pression*. http://aclyon.fr/etab/lycees/lyc-69/descartes/IMG/pdf/09_Boucle de régulation
- [4] : E.Vitte, *Conseil en automatisme, Techniques et Innovations*. Le magazine Schneider Electric de l'enseignement technologique et professionnel, Juin2004.
- [5] : M. Sakli, *Régulation de niveau d'eau dans un réservoir assisté par le logiciel LabVIEW*. Thèse de fin d'études, Ecole Nationale d'Ingénieurs de GABBES, TUNISIE, 2006-2007.
- [6] : Y. Grabjon, *Automatique système linéaires, non linéaires, à temps continu, à temps discret, représentation d'état*. Edition DUNOD, 2003.
- [7] : P. Gatt, *Ensemble complexe*. Cours BTS CIRA, 2009-2010. <http://btscira.perso.sfr.fr/>.
- [8] : M. Rabi, *Régulation analogique industrielle*. Cours Ecole Supérieure de Technologie de Fès, Maroc, 1999.
- [9] : M. Brahim, H. Bouiche, *Commande PID d'un moteur électrique à courant continu*. Thèse de fin d'études, Université Abderrahmane Mira, Bejaia, 2009-2010.
- [10] : D. Lequesne, *Régulation P.I.D analogique-numérique-flou*. Edition LAVOISIER, 2006.
- [11] : M.L. Fas, *Cours automatique avancée*. Cours master 2 Automatique. Département d'électronique, USDB, 2010-2011.
- [12] : K. Boudjit, *Le processeur de signal*. Cours master 1 Automatique. Département d'électronique. USDB, 2009-2010.
- [13] : Bigonoff, *La programmation des Pcs*. Cours Première Partie-Révision 22. La Gamme Mid-Range par l'Etude des 16f87x (16F876-16F877), 2003.
- [14] : Noxyben, *Programmation des PIC en C partie 1*, 2007. [http:// noxyben.fr/Electronique/PIC/Programmation_des_PICen_part1](http://noxyben.fr/Electronique/PIC/Programmation_des_PICen_part1)
- [15] : *Traitement programme de l'information - partie6 – Le microcontrôleur PIC 16F877*. [http:// la-fayette.fr/STI_EL/Ressources/traitement programme de l'information partie6](http://la-fayette.fr/STI_EL/Ressources/traitement programme de l'information partie6)
- [16] : O. Français, *Amplificateur d'instrumentation*. Edition ESIEE, 2000.
- [17] :N. Pousset, *Cours d'initiation au logiciel labVIEW*. http://nicolas_pousset.perso.neuf.fr/Enseignement/labVIEW/.