

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri De Tizi-Ouzou



Faculté De Génie Electrique Et d'Informatique
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etude
de MASTER PROFESSIONNEL
Filière: **Automatique et Informatique Industrielles**

Présenté par
SEDDAR Yahia
GHEZAL Katia

Mémoire dirigé par **M. CHARIF Moussa**

Thème
Contribution à l'Implémentation d'un
Régulateur Flou sous un API S7-300

Mémoire soutenu publiquement le 03 juillet 2017 devant le jury composé de :

Mr MOULA
Grade, Lieu d'exercice, Président

Mr CHARIF Moussa
Grade, Lieu d'exercice, Rapporteur

Mr ALLAD Mourad
Grade, Lieu d'exercice, Examineur

Mr HADOUCHE Rezki
Grade, Lieu d'exercice, Examineur

Remerciement

Nous tenons tout d'abord à remercier Dieu le tout puissant, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

Nous tenons à remercier aussi notre encadreur Mr CHARIF Moussa pour ses précieux conseils et son aide durant toute la période du travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Je dédie ce travail

A mes chers parents

Pour leur soutien, leur patience, leur sacrifice

et leur amour, ils méritent tout éloge,

eux qui ont fait de moi ce que je suis maintenant.

J'espère être l'image que vous avez fait de

moi, que dieu vous garde et vous bénisse.

Je dédie aussi ce travail à mon cher frère et à mes sœurs,

pour leur affection et leur encouragement qui ont toujours

été pour moi des plus précieux.

Que ce travail soit pour vous le gage de notre profond amour

A tout nos amis

A tous ceux qui m'ont aidé.

A tous ceux qui m'aime

S. YAHIA

Je dédie ce travail

A mes chers parents

Pour leur soutien, leur patience, leur sacrifice

et leur amour, ils méritent tout éloge,

eux qui ont fait de moi ce que je suis maintenant.

J'espère être l'image que vous êtes fait de

moi, que dieu vous garde et vous bénisse.

Je dédie aussi ce travail à mon cher frère et à mes sœurs,

pour leur affection et leur encouragement qui ont toujours

été pour moi des plus précieux.

Que ce travail soit pour vous le gage de notre profond amour

A tout nos amis

A tous ceux qui m'ont aidé.

A tous ceux que m'aime

GH. KATIA

Notations et symboles

Symbole	Description	Unité
W(t)	consigne	% ou mA ou V
X(t)	Mesure	% ou mA ou V
Y(t)	Commande	% ou mA ou V
E(t)	Erreur (W-X)	% ou mA ou V
G_s	Gain statique	Sans
D	précision dynamique	%
K_p	Gain proportionnel	
T_i	Temps d'intégration	seconde
T_d	Temps dérivatif	seconde
p	Opérateur de laplace	
Z(t)	perturbation	
P	Régulateur proportionnel	
PI:	Régulateur proportionnel Intégral	
G_r	Gain de régulateur	
Θ	constante de temps	seconde
τ	retard	seconde
G(p)	fonction de transfert	
K_{rc}	gain critique	
T_{osc}	periode oscillation	seconde
T_r	Temps de réponse	seconde
PID	Régulateur proportionnel Intégral Dérivé	
λ	constante	
μ_A	Fonction d'appartenance	

LISTE DES FIGURES

Chapitre 1

Figure 1.1: Schéma de principe d'une régulation.....	4
Figure 1.2: Système en boucle ouverte.....	4
Figure 1.3: Schéma fonctionnel d'un système en boucle fermée.....	5
Figure 1.4 : Réponse d'un procédé régulé à un échelon de perturbation.....	6
Figure 1.5: Les trois structures de régulateur PID.....	9
Figure 1.6 : Schéma fonctionnel d'un régulateur.....	10
Figure 1.7: Réponse en boucle ouverte du procédé à un échelon de commande.....	11
Figure 1.8: Mise en oscillation juste entretenues d'un système.....	12
Figure 1.9: Réponse idéale d'un système.....	13
Figure 1.10 : Réponse indicielle d'un système du second ordre.....	15
Figure 1.11: Schéma fonctionnel d'une boucle de régulation en cascade.....	15

Chapitre 2

Figure 2.1 : Station de niveau PUL-2.....	19
Figure 2.2 : Schéma P&ID de la station de niveau.....	19
Figure 2.3 : Bloc conditionneur et amplificateur de puissance.....	21
Figure2.4 : Vérification d'état du capteur-transmetteur de niveau.....	22
Figure 2.5 : Structure interne du débitmètre à aubes.....	22
Figure2.6: Vérification d'état du du capteur-transmetteur de débit.....	23
Figure 2.7: Hystérésis de la vanne proportionnelle de régulation.....	24
Figure 2.8: Tracé de la caractéristique statique du processus.....	25
Figure2.9 : Identification des paramètres du modèle.....	26
Figure2.10 : Identification des paramètres du modèle.....	27

Chapitre 3

Figure 3.1: Représentation des ensembles	28
Figure 3.2: Structure générale d'une commande floue.....	29
Figure. 3.3: Exemple de Fuzzification.....	29
Figure 3.4: Fonctions d'appartenances	31
Figure 3.5: Exemple d'inférence Max-produit.....	32
Figure 3.6: Exemple d'inférence Max-min.....	33

Figure 3.7 : Exemple d'inférence Somme-Produit.....	34
Figure 3.8 : Défuzzification par la méthode moyenne des maximums.....	35
Figure 3.9 : Défuzzification par centre de gravité.....	36
Figure 3.10 : Structure régulateur floue de type Mamdani.....	37
Figure 3.11 : Exemple de logique floue en Ladder.....	38

Chapitre 4

Figure 4.1 : API (MODICON 084).....	40
Figure 4.2 : API d'un type compact (siemens-S200).....	41
Figure 4.3 : API d'un type modulaire (siemens S7).....	41
Figure 4.4 : Structure interne des automates.....	42
Figure 4.5 : Cycle de fonctionnement d'un API.....	43
Figure 4.6 : SIMATIC S7-300.....	45
Figure. 4.7 : Schéma fonctionnel de bloc PID (FB41).....	47
Figure 4.8 : PC Adapter USB-MPI.....	50
Figure 4.9 : Interface (port) MPI.....	50
Figure 4.10 : Modules fonctionnels d'un système de supervision.....	51

Chapitre 05

Figure 5.1 : Paramétrage du régulateur sous step7 (FB41).....	55
Figure 5.2 : Réponse du système régulé par automate.....	56
Figure 5.3 : Programme de la régulation en cascade Niveau/Débit.....	57
Figure 5.4 : Réponse de système en régulation cascade.....	58
Figure 5.5 : Fuzzification de l'erreur.....	59
Figure 5.6 : fonction d'appartenance de la dérivée de l'erreur.....	60
Figure 5.7 : Fuzzification de la sortie.....	60
Figure 5.8 : Programmation de la fonction d'appartenance.....	61
Figure 5.9 : Exemple de programme de calculateur de signal de commande.....	62
Figure 5.10 : Exemple de programme de conversion et de chargement.....	63
Figure 5.11 : Réponse expérimentale Contrôleur flou	63
Figure 5.12 : Vue d'accueil du système de contrôle de processus PUL-2.....	64
Figure 5. 13 : Interface de régulation PID avec le bloc FB41	65
Figure 5. 14 : Interface de régulation avec le régulateur flou....	66

LISTE DES TABLEAUX

Chapitre 1

Tableau 1.1: Influence des actions PID sur le processus.....	8
Tableau 1.2: Tableau pour le calcul des paramètres PID en boucle ouverte.....	11
Tableau 1.3 : Réglages préconisés par Ziegler et Nichols en boucle fermé.....	12

Chapitre 2

Tableau 2.1: Caractéristique du transducteur de niveau régulation.....	21
Tableau 2.2: Valeurs expérimentales du comportement de la vanne proportionnelle.....	24
Tableau 2.3 : Relevé de la caractéristique statique.....	25

Chapitre 3

Tableau 3.1 : Opérateurs entre ensembles flous.....	30
--	----

Chapitre 4

Tableau 4.1: Description des différents éléments de l'API S7 300.....	45
Tableau 4.2: Les différents types de régulation PID sous step7.....	46
Tableau 4.4: Paramètres de sorties.....	48
Tableau 4.3: Paramètres d'entrée.....	49

Chapitre 5

Tableau 5.1: Table de règles.....	60
--	----

Sommaire

Introduction générale	1
------------------------------------	---

Chapitre 1 : Notions de Régulation de Processus

1.1. Introduction.....	3
1.2. Définition de la régulation.....	3
1.3. Objectif de régulation.....	3
1.4. Principe général de la régulation	4
1.5. Régulation en boucle ouverte/ en boucle fermée.....	4
1.6. Les qualités attendues d'une régulation.....	5
1.6.1. La stabilité.....	5
1.6.2. La précision.....	6
1.6.3. La rapidité.....	6
1.7. Régulateur PID	7
1.8. Les actions proportionnelle, intégrale et dérivée (PID).....	7
1.9. Les différentes structures du régulateur PID.....	9
1.10. Le régulateur industriel	10
1.11. Paramétrage de régulateur.....	10
1.12. Synthèse des paramètres des actions PID	11
1.12.1. Méthodes en boucle ouverte de Ziegler-Nichols.....	11
1.12.2. Méthode de Ziegler Nichols en boucle fermée.....	12
1.13. Critère de choix.....	13
1.13.1. Le critère idéal.....	13
1.13.2. Critère parfait.....	14
1.14. Régulation Cascade.....	15
1.14.1. Définition de la régulation cascade	15
1.14.2. Rôle de la régulation cascade	16
1.14.3. Cascade sur la grandeur intermédiaire.....	16
1.14.4. Cascade sur grandeur réglante.....	16
1.15. Conclusion.....	17

Chapitre 2 : Description du Système de Régulation de Niveau d'Eau

2.1. Introduction	18
2.2. Description de l'unité de niveau PUL-2.....	18

2.3. Le capteur de niveau	20
2.4. Méthode de mesure de niveau	20
2.4.1. Emploi de flotteurs.....	20
2.4.2. Variation de la capacité électrique des matériaux.....	20
2.4.3. Au moyen d'un faisceau d'ultrasons.....	20
2.5. Conditionnement du signal du capteur.....	20
3.6. Amplificateur de puissance de la vanne proportionnelle.....	21
2.7. Transducteur de niveau de la station PUL-2.....	21
2.7.1. Etalonnage du capteur de niveau.....	21
2.8. Débitmètre à aubes.....	22
2.8.1 Etalonnage du capteur de débit.....	23
2.9. Vanne proportionnelle.....	23
2.10. Contrôle automatique de la vanne proportionnelle.....	23
2.11. Comportement de la vanne proportionnelle	24
2.12. La caractéristique statique du processus.....	25
2.13. Identification du système.....	26
2.14. Identification de système de débit.....	26
2.15. Conclusion	27

Chapitre 3 : Présentation de la Logique Floue

3.1. Introduction	28
3.2. La logique floue.....	28
3.3. La régulation par la logique floue.....	28
3.4. Principe d'une commande floue.....	29
3.4.1. Fuzzification.....	29
3.4.1.1. L'univers de discours.....	30
3.4.1.2. Ensemble flou.....	30
3.4.1.2.1. Opérateurs entre ensembles flous.....	30
3.4.1.3. La fonction d'appartenance.....	31
3.4.1.4. Différentes formes pour les fonctions d'appartenance.....	31
3.4.2. Moteur d'inférence floue.....	31
3.4.2.1. Méthode d'inférence Max-produit.....	32
3.4.2.2. Méthode d'inférence Max-min.....	33
3.4.2.3. Méthode d'inférence Somme-Produit.....	33
3.4.3.4. Base de connaissances.....	34
3.4.3. Défuzzification.....	35
3.4.3.1. Méthode moyenne des maximums (MM).....	35

3.4.3.2. Méthode du centre de gravité (CG).....	35
3.5. Différents types de régulateurs flous.....	37
3.5.1. Le régulateur de type Mamdani.....	37
3.5.2. Régulateur de type Sugeno.....	37
3.6. Transcription de la base floue en langage Ladder.....	37
3.7. Avantages et inconvénients du régulateur flou.....	38
3.8. Conclusion.....	39

Chapitre 4 : Développement de la solution programmable de supervision sous environnement SIMATIC-S / WINCC flexible

4.1. Introduction.....	40
4.2. Historique sur l'automate programmable industriel (API).....	40
4.3. Définition.....	40
4.4. Structure de l'automate programmable industriel.....	41
4.4.1. Aspect extérieur.....	41
4.4.1.1. Automate de type compact.....	41
4.4.1.2. Automate De type modulaire.....	41
4.4.2. Structure interne de l'automate programmable industriel.....	42
4.5. Principe de fonctionnement d'un automate programmable industriel.....	43
4.6. Les types des automates programmables industriels.....	43
4.6.1. Automates de petite gamme.....	43
4.6.2. Automates de moyenne gamme.....	44
4.6.3. Automates de haute gamme.....	44
4.7. Nature des informations traitées par l'automate.....	44
4.8. Choix d'un automate programmable industriel.....	44
4.9. Avantages et Inconvénients des APIs.....	44
4.10. Automate S7-300 CPU314.....	45
4.11. Langages de programmation	45
4.11.1. La liste d'instructions (LIST).....	46
4.11.2. Le logigramme (LOG).....	46
4.11.3. Le schéma à contacts (CONT, langage LADDER).....	46
4.12. Programmation de la régulation PID en utilisant le bloc FB 41 intégré.....	46
4.12.1. Fonctionnement de bloc PID (FB41).....	47
4.12.2. Paramétrage du régulateur PID.....	48
4.11.3. Mise en œuvre d'un régulateur PID sous step7.....	50

4.13. Interface MPI et PC Adapter USB.....	50
4.14. Superviseur HMI SIMATIC WinCC.....	50
4.14.1. Les principales étapes suivies pour créer notre application sous WINCC.....	51
4.14.2. Communication entre WINCC et automate programmable industriel (API).....	51
4.14.3. Caractéristiques techniques de WINCC.....	52
4.14.4. WinCC flexible Runtime.....	52
4.14.5. Création de l'interface WinCC.....	52
4.14.6. Les Alarmes.....	53
4.14.6.1. Alarmes TOR.....	53
4.14.6.2. Alarmes analogiques.....	53
4.15. Conclusion.....	53

Chapitre 5 : Teste de validation et résultats expérimentaux

5.1. Introduction.....	54
5.2. Synthèse du régulateur PID.....	54
5.2.1. Synthèse du correcteur de niveau par la méthode du critère idéal.....	54
5.3. Régulation de niveau par automate SIMATIC S7-300.....	55
5.3.1. Programme et paramétrage du régulateur PID sous step7.....	55
5.3.2. Résultats expérimentaux de l'application du correcteur PI.....	56
5.3.3. Test en régulation en Cascade	56
5.3.4. Paramétrage et intégration du régulateur PI en cascade sous step7.....	57
5.3.5. Résultats expérimentaux PI en cascade (Niveau- Débit).....	58
5.4. Programmation du régulateur flou	59
5.4.1. Entrées / Sorties du régulateur flou	59
5.4.2. Fonctions d'appartenance.....	59
5.4.3. L'implémentation du correcteur flou sous automate S7-300.....	61
5.4.3.1. Programmation des fonctions d'appartenance.....	61
5.4.3.2. Programmation de la commande floue par la méthode du centre de gravité	62
5.4.4. Résultat expérimentaux de régulateur flou.....	63
5.5. Interface de commande PID et FLOU de la station PUL-2/EV.....	64
5.5.1. Constitution de la vue.....	65
5.7. Conclusion.....	66
Conclusion générale.....	67

Introduction générale

Pour être compétitif, un procédé industriel doit être obligatoirement automatisé. En effet, la compétitivité exige de le maintenir le plus près possible de ses conditions optimales de fonctionnement prédéfini par un cahier des charges (conditions et performances imposées), telles que la qualité des produits fabriqués, la flexibilité de la production, la sécurité du personnel et des installations, l'économie de l'énergie et le respect de l'environnement. La régulation occupe une place importante dans l'automatisation d'un procédé industriel.

Les méthodes classiques de l'automatique ont été largement appliquées dans de nombreux systèmes de régulation industrielle. Cependant, la plupart de ces systèmes physiques présentent des non-linéarités et leurs paramètres sont souvent mal connus ou variables dans le temps. Pour commander de telles classes de systèmes, les méthodes conventionnelles de l'automatique ont montré leurs limites en termes de stabilisation et performances. Suite aux développements des calculateurs numériques, les automaticiens se sont intéressés aux nouvelles approches de commande telles que la commande adaptative, la commande prédictive, la commande robuste, ainsi que les techniques basées sur l'intelligence artificielle. Parmi ces dernières; la commande par logique floue. L'implémentation de cette technique de commande est actuellement l'une des préoccupations de nombreux travaux de recherches.

Les systèmes d'inférence flous possèdent deux points forts ; le premier est qu'ils sont généralement construits à partir de la connaissance humaine, et le deuxième réside dans leur capacité descriptive due à l'utilisation des variables linguistique. C'est dans ce contexte que s'inscrit notre projet de fin d'étude. Par conséquent, Notre objectif est d'implémenter un régulateur flou sur un API s7-300 en utilisant le langage de programmation Ladder.

Pour atteindre notre objectif, nous avons envisagé d'une part, de concevoir un régulateur flou pour la régulation de niveau à fin de comprendre l'approche de contrôle basé sur la logique floue et d'autre part, de trouver un moyen de matérialiser ce régulateur basé sur la logique floue et de l'intégrer sur un API S7-300 afin de réguler le niveau de la station de PUL2/EV.

Notre étude suivra le plan suivant :

Nous nous intéressons dans le premier chapitre à la régulation, en donnant quelques généralités et définitions sur la régulation industrielle et les différentes techniques d'identification des systèmes.

Nous présenterons dans le second chapitre, la description de l'unité de régulation de niveau PUL2/EV.

Dans le troisième chapitre nous allons présenter le concept général de contrôle par la logique floue et ces principes de bases, pour mieux nous familiariser avec la commande floue.

Dans le quatrième chapitre nous allons présenter l'automate programmable SIEMENS, les régulateurs intégrés ainsi que l'outil de développement des écrans de communication et de contrôle WINCC flexible 2008 qui est utilisé pour les HMI.

Le dernier chapitre présente les techniques d'implémentation des différents régulateurs (PI et Flou) ainsi que les différents résultats expérimentaux.

Enfin nous terminons notre travail avec une conclusion générale.

Chapitre 1

Notions de Régulation de Processus

1.1. Introduction

Le développement massif des outils mathématique, informatique et technologiques; ont permet des nouvelles techniques avancées de commande, d'analyse et de supervision des processus. Le contrôle des processus doit assurer un fonctionnement optimal sous des impératifs (des contraintes) pratiques prédéfinis par les concepteurs selon un cahier de charge spécifique aux processus tel que la stabilité du système, rapidité et la précision de la réponse dynamique.

1.2. Définition de la régulation

La régulation regroupe l'ensemble de techniques et moyens industriels utilisés visant à contrôler une grandeur physique soumise à des perturbations et de la maintenir égale à une valeur désiré appelée consigne cette grandeur physique est appelé grandeur réglée, on peut citer comme exemple la régulation de niveau dans un réservoir dépendant de plusieurs débits d'alimentation et de soutirage.

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.

1.3. Objectif de la régulation [4]

L'objectif d'une régulation est d'assurer le fonctionnement d'un procédé selon les critères prédéfinis par un cahier de charges. Son objectif principale est de contrôler une ou plusieurs grandeur mesurés quelle que soit les perturbations qui agissent sur le procédé toute en assurant la stabilité, la précision et la robustesse de la loi de commande.

1.4. Principe général de la régulation

Pour chaque boucle de régulation on dispose de trois maillons indispensables qui sont l'organe de mesure, le régulateur et l'organe de contrôle. Il faut donc commencer par mesurer les principales grandeurs servant à contrôler le système à régulé. Le comparateur récupère ces mesures et les compare aux valeurs souhaitées (consigne). En cas de non concordance des valeurs de mesure et des valeurs de consigne, Le régulateur envoie un signal de commande à l'organe de contrôle (vanne, moteur, etc.), afin que celui-ci agisse sur le système. Les paramètres qui régissent le processus sont ainsi stabilisés en permanence à des niveaux souhaités.

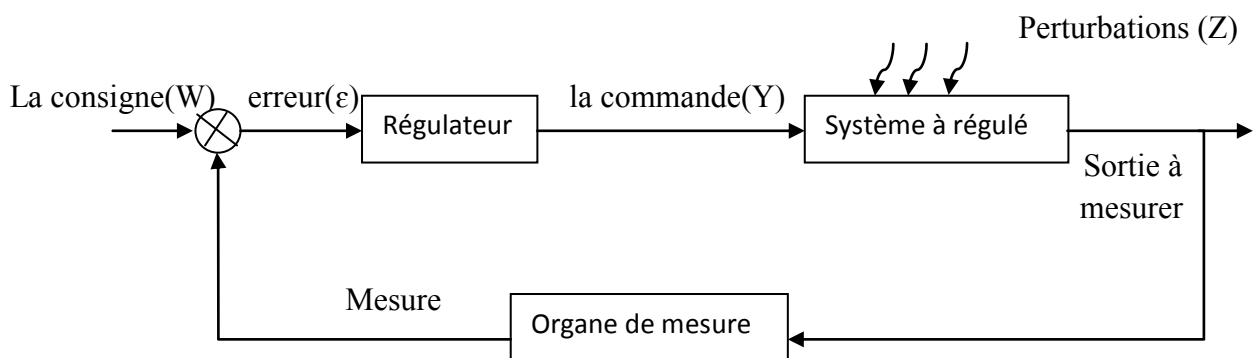


Figure 1.1 : Schéma de principe d'une régulation

1.5. Régulation en boucle ouverte/ en boucle fermée

a. Régulation en boucle ouverte [2]

Un système en boucle ouverte est un système qui ne comporte pas de contre-réaction (feedback) entre la sortie et l'entrée. Classiquement, il est composé du processus physique, d'un capteur pour en mesurer la sortie et d'un actionneur pour agir sur la grandeur d'entrée du processus. Pour une variation manuelle d'amplitude finie de la commande, on a une variation de la grandeur à maîtriser ou à réguler. La régulation en boucle ouverte ne peut être mise en œuvre que si l'on connaît la loi régissant le fonctionnement du processus (autrement dit, il faut connaître la corrélation entre la valeur mesurée et la grandeur réglante).

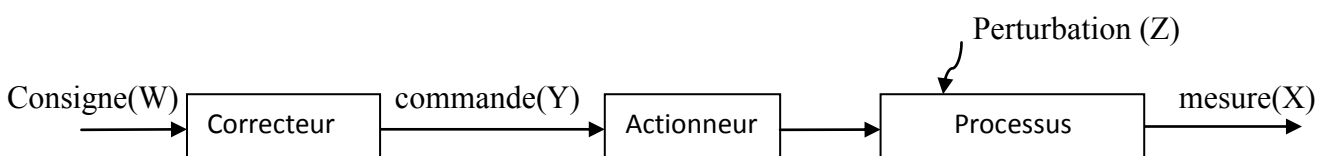


Figure 1.2 : Système en boucle ouverte

b. Régulation en boucle fermée

Dans une régulation en boucle fermée, une bonne partie des facteurs perturbateurs, y compris les dérives propres de certains composants de la boucle, sont automatiquement compensés par la contre-réaction à travers le procédé. Autre avantage, il n'est pas nécessaire de connaître avec précision les lois, le comportement des différents composants de la boucle, et notamment du processus, bien que la connaissance des allures statistiques et dynamiques des divers phénomènes rencontrés soit utile pour le choix des composants.

Parmi les inconvénients d'une régulation en boucle fermée, il faut citer le fait que la précision et la fidélité de la régulation dépendent de la fidélité et de la précision sur les valeurs mesurées et sur la consigne.

La grandeur réglante exerce une influence sur la grandeur réglée, pour la maintenir dans des limites définies malgré les perturbations

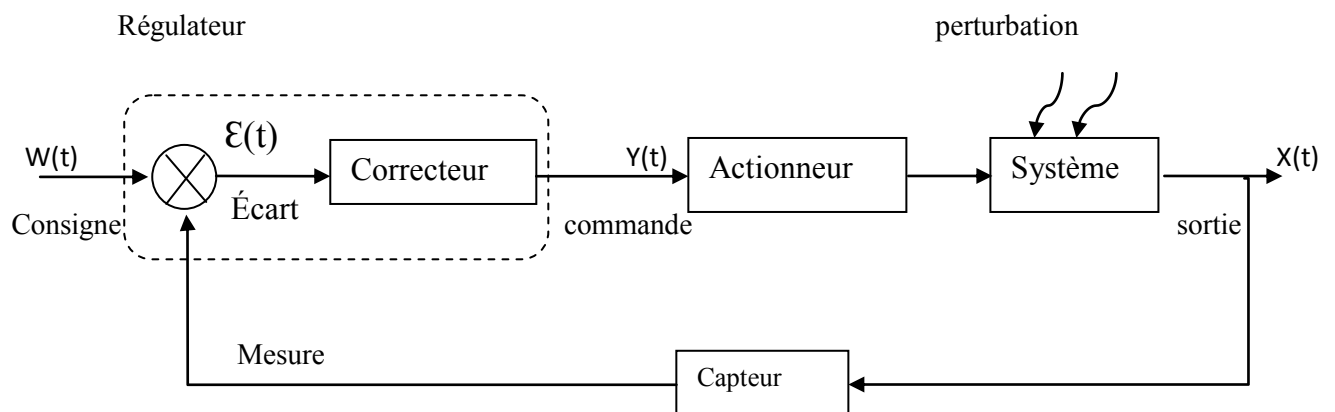


Figure 1.3: Schéma fonctionnel d'un système en boucle fermée

1.6. Les qualités attendues d'une régulation [1]

Les qualités exigées les plus rencontrées industriellement sont la stabilité, la précision et la rapidité de la grandeur à réguler.

1.6.1. La stabilité

La qualité essentielle pour un système de régulation est la stabilité. Dans une approche simplifiée, un système est considéré comme stable si, pour une grandeur à maîtriser se stabilise à une valeur finie. Plus le régime transitoire d'un système soumis à une telle variation est amorti plus il est stable. Le degré de stabilité est alors caractérisé par l'amortissement de ce régime transitoire.

1.6.2. La précision

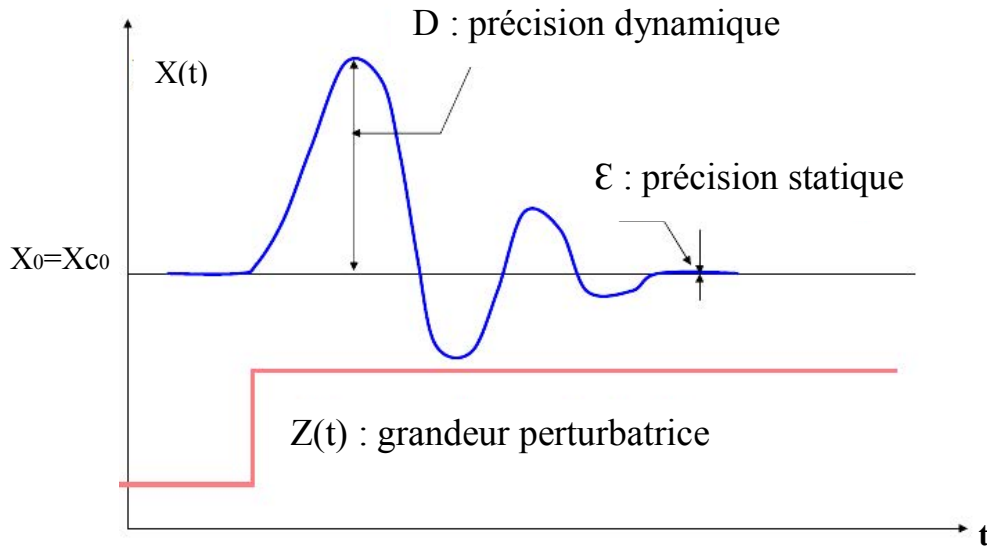


Figure 1.4 : Réponse d'un procédé régulé à un échelon de perturbation [1]

a. Précision statique

Il est naturel d'évaluer la précision d'un système régulé en comparant l'objectif atteint par rapport à celui exigé. La précision d'un système régulé se mesure donc à l'écart entre la consigne demandée et la mesure en régime permanent ; on parle alors de précision statique. Plus l'écart statique est petit, plus le système est précis. L'évaluation de la précision statique s'effectue en réalisant une variation rapide de consigne en amplitude et en mesurant la variation d'amplitude finalement obtenue de la mesure.

b. Précision dynamique

La précision dynamique est caractérisée par le dépassement "D" lors du régime transitoire de la réponse de la grandeur réglée suite à un échelon de consigne ou de perturbation. Cette précision est liée directement au degré de stabilité du procédé; c'est un critère de performance qui peut être défini par les marges de gain et de phase. Elle est donc à prendre en compte lors des réglages des régulateurs.

1.6.3. La rapidité

La rapidité d'un système régulé s'évalue par le temps nécessaire à la mesure pour entrer dans une zone $\pm 5\%$ de sa valeur finale (soit entre 95% et 105%). Le système régulé est d'autant plus rapide que le temps de réponse à 5% est court.

1.7. Régulateur PID [3]

Le régulateur standard le plus utilisé dans l'industrie est le régulateur PID (proportionnel intégral dérive), car il permet de régler à l'aide de ses trois paramètres les performances (amortissement, temps de réponse, ...) d'un processus, Par conséquent, le régulateur PID est bien adapté à la plupart des processus de type industriel et il est relativement robuste par rapport aux variations des paramètres du procédé.

En pratique, à une catégorie donnée de systèmes à réguler correspond un type de correcteur adopté. Pour effectuer un choix judicieux, il faut connaître les effets des différentes actions de régulateur PID (proportionnelle, intégrale et dérivée).

Un régulateur PID est obtenu par l'association de ces trois actions et il remplit essentiellement les trois fonctions suivantes :

-Il fournit un signal de commande en tenant compte de l'évolution du signal de sortie par rapport à la consigne

- Il élimine l'erreur statique grâce au terme intégrateur

-Il anticipe les variations de la sortie grâce au terme dérivateur.

En régulation des procédés, on préfère utiliser une structure mixte du régulateur PID

$$C(t) = K_p \left[\varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(t) dt + T_d \frac{d\varepsilon(t)}{dt} \right] \quad (1.1)$$

1.8. Les actions proportionnelle, intégrale et dérivée (PID)

Les paramètres du régulateur PID sont le gain proportionnel K_p , le temps intégral T_i et le temps dérivatif T_d .

- L'action proportionnelle :

$$P = K_p \varepsilon(t) \quad (1.2)$$

Le rôle de l'action proportionnelle est de minimiser l'écart ε entre la consigne et la mesure et elle réduit le temps de monter et le temps de réponse. On constate qu'une augmentant du gain K_p du régulateur entraîne une diminution de l'erreur statique et permet d'accélérer le comportement

global de la boucle fermée. On serait tenté de prendre des valeurs de gain élevées pour accélérer la réponse du procédé mais on est limité par la stabilité de la boucle fermée. En effet, une valeur trop élevée du gain augmente l'instabilité du système et donne lieu à des oscillations.

- L'action intégrale

$$I = \frac{1}{T_i} \int_0^t \varepsilon(t) dt \quad (1.3)$$

L'action intégrale agit proportionnellement à la surface de l'écart entre la consigne et la mesure, et elle poursuit son action tant que cet écart n'est pas nul. On dit que l'action intégrale donne la précision statique (Elle annule l'erreur statique). L'action intégrale est conditionnée par le temps d'intégrale T_i .

- Le terme dérivatif :

$$D = T_d \frac{d\varepsilon(t)}{dt} \quad (1.4)$$

Elle est une action qui tient compte de la vitesse de variation de l'écart entre la consigne et la mesure, elle joue aussi un rôle stabilisateur, contrairement à l'action intégrale. En effet, elle délivre une sortie variant proportionnellement à la vitesse de variation de l'écart ε

Le comportement des systèmes change en fonction du changement des actions de régulateur. Le tableau (1.1) résume l'influence sur les performances du système, par rapport aux paramètres K_p , T_i et T_d .

Coefficient	Rapidité	Stabilité	Précision
K_p	Augmente	Diminue	Augmente
T_i	Diminue	Augmente	Reste correcte
T_d	Augmente	Diminue	Ne bouge pas

Tableau 1.1 : Influence des actions PID sur le processus

1.9. Les différentes structures du régulateur PID [4]

Il existe trois types d'algorithme PID. Le PID série, le PID parallèle et le PID mixte. Ils traduisent l'organisation interne des modules de calcul du régulateur et caractérisent la loi de commande.

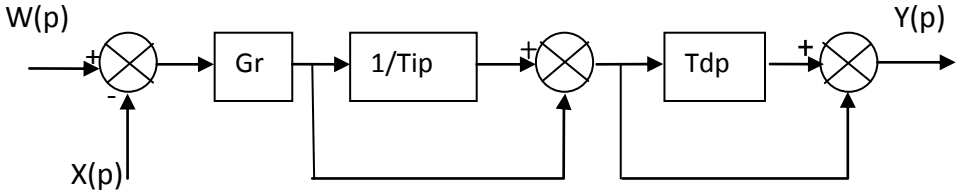
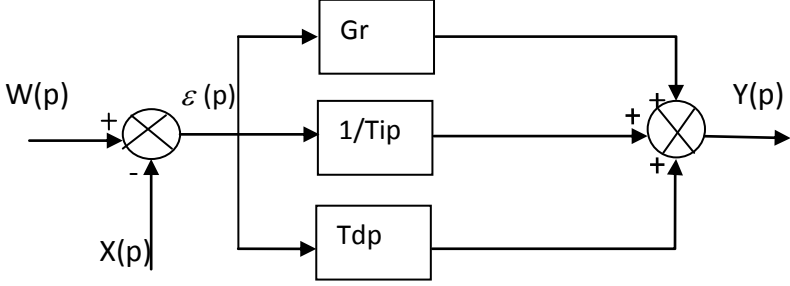
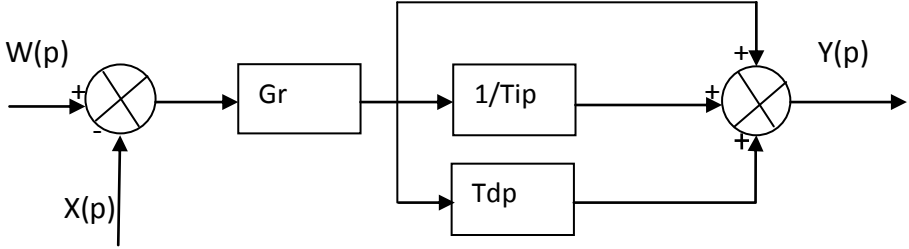
Structure de régulateur	Schéma fonctionnels
<p style="text-align: center;">Série</p>	 <p>Fonction de transfert : $C(P) = Gr(1 + \frac{1}{T_i})(1 + T_d P)$</p>
<p style="text-align: center;">Parallèle</p>	 <p>Fonction de transfert : $C(P) = Gr + \frac{1}{T_i P} + T_d P$</p>
<p style="text-align: center;">Mixte</p>	 <p>Fonction de transfert : $C(P) = Gr(1 + \frac{1}{T_i P} + T_d P)$</p>

Figure 1.5: Les trois structures du régulateur PID

1.10. Le régulateur industriel [1]

Le régulateur industriel est un appareil qui a pour rôle essentiel de contrôler le procédé, c'est-à-dire de garantir les comportements dynamiques et statiques du procédé conformes au cahier des charges défini. Ceci est réalisé par réglage et adaptation des paramètres de sa fonction de transfert au procédé à contrôler.

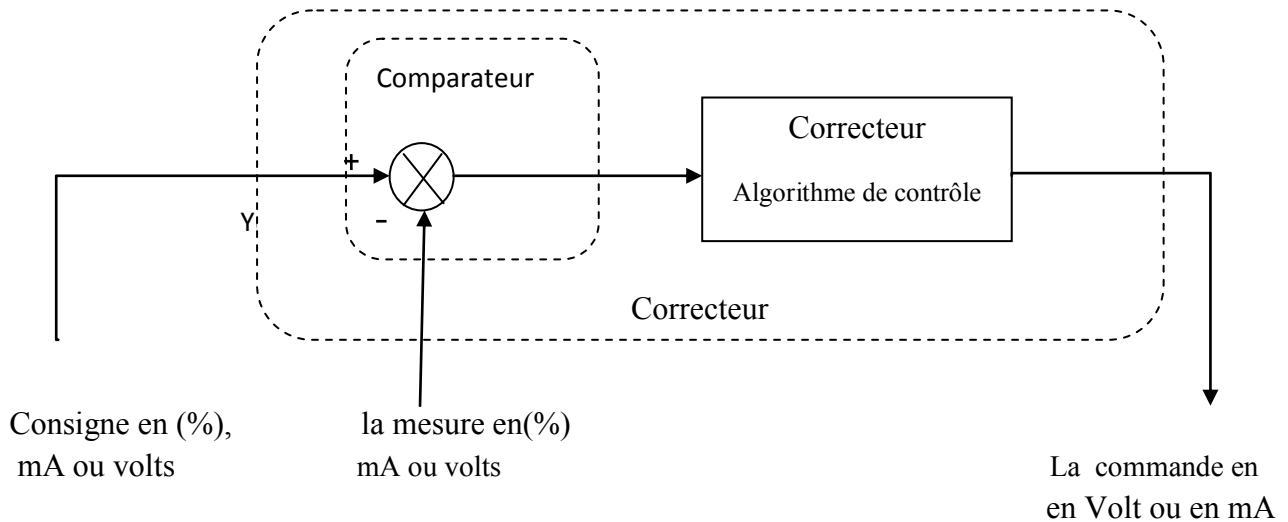


Figure 1.6 : Schéma fonctionnel d'un régulateur.

1.11. Paramétrage de régulateur [1]

Le réglage d'une boucle de régulation (choix de structure et calcul des paramètres du régulateur) doit permettre de répondre au plus grand nombre de contraintes exigées par le **cahier des charges** du procédé à réguler. De nombreuses méthodes de réglage d'une boucle sont possibles selon les besoins en régulation ou en asservissement. Les exigences du cahier des charges sont décrites soit dans le domaine temporel, soit dans le domaine fréquentiel. Le critère de réglage est alors fixé à partir soit de la forme de la réponse temporelle souhaitée pour un type d'entrée (par exemple un échelon), soit à partir des marges de stabilité (marge de gain et de phase). Le critère de précision est, bien entendu, intrinsèquement lié à celui du réglage. Le bon réglage est celui qui répondra au meilleur compromis global du cahier des charges.

1.12. Synthèse des paramètres des actions PID

1.12.1 Méthodes en boucle ouverte de Ziegler-Nichols [1]

C'est la méthode la plus ancienne (1942). Elle a pour objectif la détermination du réglage d'un régulateur PID à partir de la réponse à un échelon du procédé.

On dispose de la réponse $X(t)$ (variation de la sortie) suite à un échelon d'entrée $Y(t)=\Delta y$.

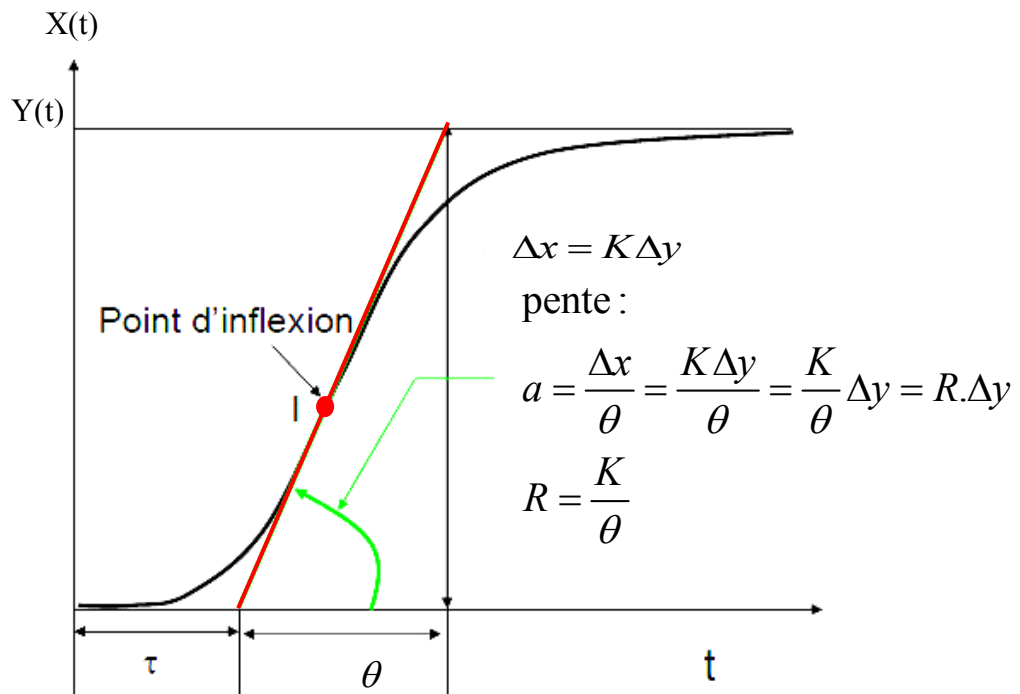


Figure 1.7: Réponse en boucle ouverte du procédé à un échelon de commande Δy

Les valeurs des paramètres PID en boucle ouverte selon Ziegler Nichols sont données sur le tableau ci-dessous.

Type de régulateur	Gain K_p	T_i	T_d
Proportionnel(P)	$\frac{1}{R \cdot \tau}$	0	0
Proportionnel intégrale(PI)	$\frac{0.9}{R \cdot \tau}$	3.3τ	0
Proportionnel intégrale dérivée(PID)	$\frac{1.27}{R \cdot \tau}$	2τ	0.5τ

Tableau 1.2: Tableau pour le calcul des paramètres PID en boucle ouverte

1.12.2. Méthode de Ziegler Nichols en boucle fermée [1]

Cette méthode est la plus connue des méthodes pratiques de réglage des boucles de régulation. L'avantage de cette méthode est qu'il n'y a pas besoin de connaître la fonction de transfert du procédé et que le réglage se fait directement sur le site en chaîne fermée. Après avoir porté la mesure près du point de consigne désiré (manuellement ou automatiquement), le régulateur est réglé en action proportionnelle seule. Le gain G_r , est alors augmenté progressivement jusqu'à obtention du pompage sans que l'organe réglant ne soit jamais en saturation. Le gain critique G_{rc}

du régulateur est le plus petit gain qui permet l'entretien des oscillations. La période d'oscillation est mesurée sur l'enregistrement (figure 1.8).

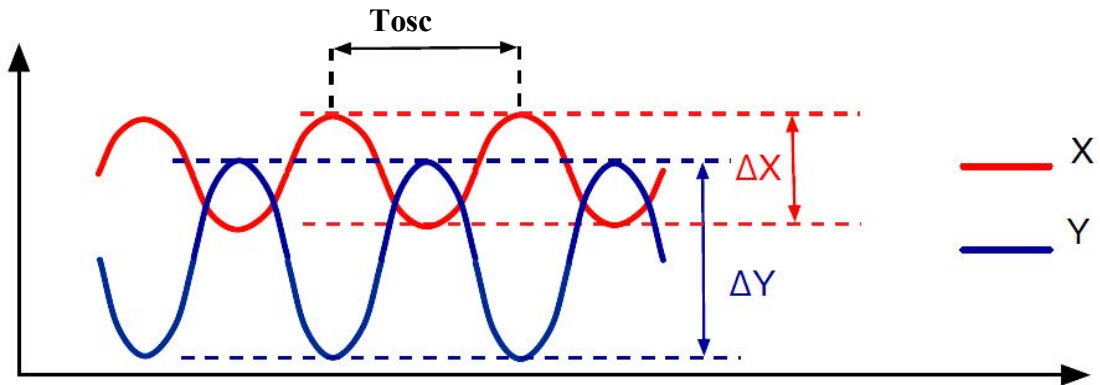


Figure 1.8: Mise en oscillation juste entretenues d'un système

Ziegler et Nichols proposent alors les valeurs de réglage du tableau suivant :

Type de régulateur	P	PI	PI Parallèle	PID série	PID Parallèle	PID mixte
Grc	$\frac{Grc}{2}$	$\frac{Grc}{2.2}$	$\frac{Grc}{2.2}$	$\frac{Grc}{3.3}$	$\frac{Grc}{1.7}$	$\frac{Grc}{1.7}$
Ti	Ti maxi ou annulée	$\frac{Tosc}{1.2}$	$\frac{2 \cdot Tosc}{Grc}$	$\frac{Tosc}{4}$	$\frac{0.85Tosc}{Grc}$	$\frac{Tosc}{2}$
Td	0	0	0	$\frac{Tosc}{4}$	$\frac{Grc \cdot Tosc}{13.3}$	$\frac{Tosc}{8}$

Tableau 1.3 : Réglages préconisés par Ziegler et Nichols en boucle fermée

1.13. Critère de choix [5]

Le choix d'un régulateur s'impose sur un (ou plusieurs) critère possible, simple à utiliser et d'autre part compatible avec la technologie analogique classique. Nous allons donc admettre deux critères de réglage que nous appellerons l'un idéal et l'autre parfait. C'est en basant sur cette méthode qu'on va choisir les paramètres du notre régulateur.

1.13.1. Le critère idéal

La régulation est dite idéale si :

$$G(p) = \frac{\lambda}{Tp} \cdot \frac{1}{Hr(p)} \tag{1.5}$$

D'après la formule (1.6), nous pouvons écrire que $G(p) \cdot Hr(p) = \frac{\lambda}{Tp}$ et les fonctions de transfert des schémas fonctionnels de la figure (a) et (b) suivante sont alors identiques.

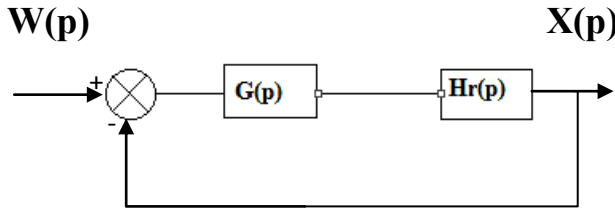


Figure : 1.9.a

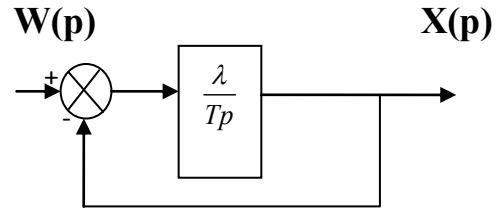


figure : 1.9.b

Soit :

$$FT = \frac{X(p)}{W(p)} = \frac{\frac{\lambda}{Tp}}{1 + \frac{\lambda}{Tp}} = \frac{1}{1 + p\frac{T}{\lambda}} \tag{1.6}$$

Si nous donnons un échelon unité sur W(p), nous obtenons :

$$W(p) = u \cdot \frac{1}{p}$$

$$X(p) = \frac{u}{p} \cdot \frac{1}{1 + p\frac{T}{\lambda}} \tag{1.7}$$

Soit : $L^{-1}X(p) = u (1 - e^{-\frac{\lambda t}{T}})$ cette réponse est dite idéale.

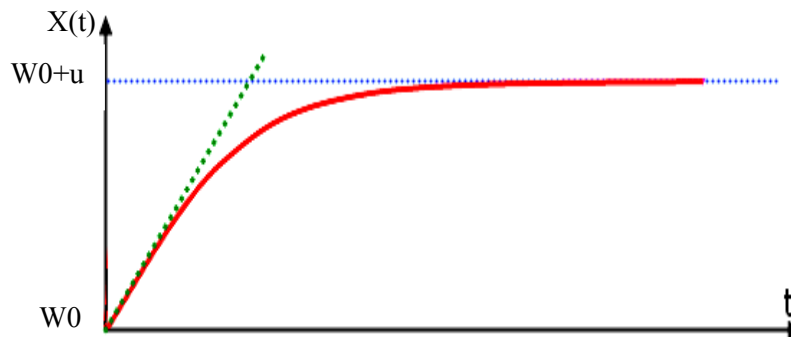


Figure 1.10: Réponse idéale d'un système

On peut diminuer le régime transitoire on augmentant λ sans limite de stabilité. On dira que la stabilité est absolue et le système bouclé apériodique (dénominateur de FT à racines réelles négatives). en résumé, on dira qu'une régulation est idéale si :

- La stabilité est absolue
- La fonction de transfert est apériodique.

1.13.2. Critère parfait

La régulation est dite parfaite si :

$$G(p) = \frac{\lambda}{Tp} \cdot \frac{1}{(\theta P+1)} \cdot \frac{1}{Hr(p)} \tag{1.8}$$

Explicitons de cette formule.

Nous pouvons écrire, d'après cette formule que :

$$G(p) \cdot Hr(p) = \frac{\lambda}{Tp} \cdot \frac{1}{(\theta P+1)} \tag{1.9}$$

Et les fonctions de transfert des schémas fonctionnels des figures (c) et (d) suivantes sont identiques:

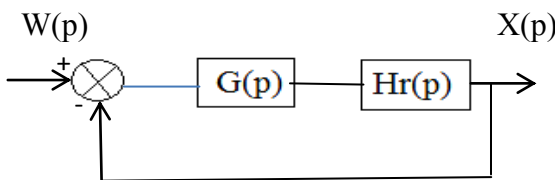


Figure : 1.9.c

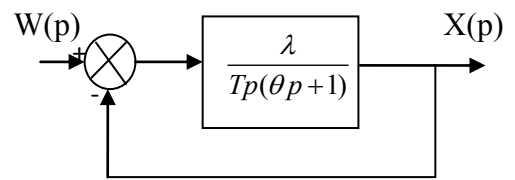


Figure : 1.9.d

Soit:

$$FT = \frac{X(p)}{W(p)} = \frac{\frac{\lambda}{Tp(\theta p+1)}}{1 + \frac{\lambda}{Tp(\theta p+1)}} = \frac{\lambda}{\theta Tp^2 + Tp + \lambda}$$

Donnons un échelon unité sur W(p), soit W(p) = u

$$X(p) = \frac{u}{p} \cdot \frac{\lambda}{\theta Tp^2 + Tp + \lambda} \tag{1.10}$$

Deux types de réponses peuvent être créés suivant le signe du discriminant Δ du dénominateur, Quant à la stabilité elle sera absolue puisque leur partie réelle α est négative ($\alpha = -\frac{1}{\theta}$)

- **Premier type** $\Delta > 0$

Soit $T^2 - 4\lambda\theta T > 0 \rightarrow$ système apériodique

- Deuxième type $\Delta < 0$

Soit $T^2 - 4\lambda\theta T < 0 \rightarrow$ système pseudopériodique.

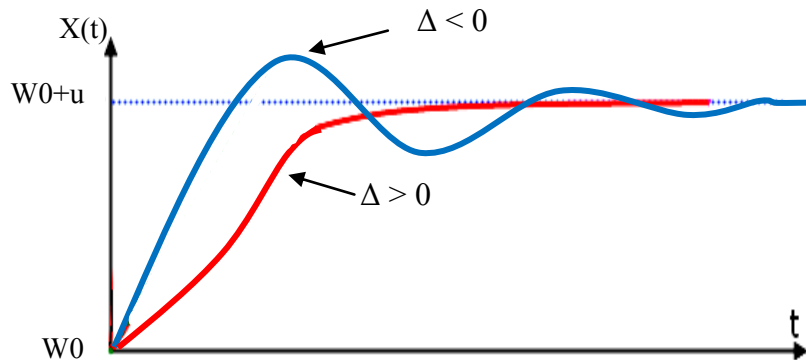


Figure 1.11 : Réponse indicielle d'un système du second ordre

En résumé, une régulation sera dite parfaite si sa réponse indicielle est du second ordre (apériodique ou pseudo-apériodique) quelques soient les réglages du régulateur.

1.14. Régulation Cascade

1.14.1. Définition de la régulation cascade [16]

Une régulation cascade est composée de deux boucles imbriquées. Le système peut être décomposé en deux sous systèmes liés par une grandeur intermédiaire mesurable. Une première boucle esclave. La deuxième boucle, est la boucle maîtresse. La grandeur réglée de la régulation cascade commande la consigne de la régulation esclave.

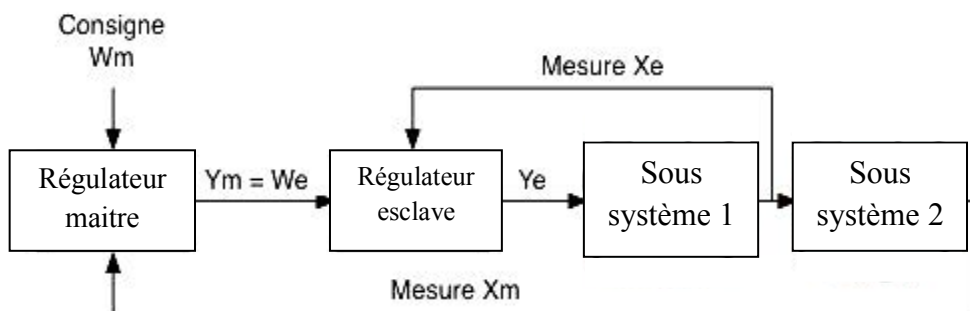


Figure 1.12: Schéma fonctionnel d'une boucle de régulation en cascade

1.14.2. Rôle de la régulation cascade

L'objectif d'une régulation en cascade est de minimiser les effets d'une ou de plusieurs grandeurs perturbatrices qui agissent soit sur la variable réglante, soit sur une grandeur intermédiaire qui se trouve en amont de la variable à réguler. Ce type de régulation est intéressant lorsque l'on a affaire à des procédés à grande constante de temps. En effet, quand une perturbation se manifeste, il est nécessaire d'attendre que son influence se ressente au niveau de l'organe de mesure placé en sortie de chaîne de la régulation. Si le temps de réponse est long, la correction n'intervient donc que très tardivement, par rapport à la cause qui l'a produite ce retard d'action entraîne un déphasage entre la grandeur réglée et la grandeur réglante, comme s'il s'agit de l'inversion de sens d'action de la régulation. Ce qui conduira à des oscillations entretenues et donc une instabilité de procédé.

L'inconvénient majeur de la régulation en cascade est qu'elle n'apporte aucune amélioration si la grandeur perturbatrice se produit en aval de la mesure intermédiaire. Pour que la cascade soit justifiée, il faut que la boucle interne soit beaucoup plus rapide que la boucle externe.

On distingue deux types de la régulation en cascade : cascade sur grandeur réglante et cascade sur grandeur intermédiaire.

1.14.3. Cascade sur une grandeur intermédiaire

Dans ce type de cascade, la boucle interne régule une grandeur intermédiaire de même nature que la grandeur réglée les deux boucles sont soumises aux mêmes perturbations.

La position de la grandeur intermédiaire est telle qu'elle subit les perturbations avant la grandeur réglée.

L'intérêt de la cascade sur la grandeur intermédiaire, par rapport à celle sur la grandeur réglante, est le fait qu'elle corrige un plus grand nombre de perturbations.

1.14.4. Cascade sur la grandeur réglante

Dans ce type de cascade, la boucle interne régule une grandeur réglante de différente nature que la grandeur réglée. Ce type de régulation cascade est efficace uniquement sur les perturbations affectant la grandeur réglante.

1.15. Conclusion

Dans ce chapitre nous avons présenté brièvement quelques définitions des systèmes de régulation ainsi que les régulateurs standards (PID).

Le correcteur qui est l'élément principal d'un système de régulation doit respecter au maximum les contraintes imposées sur la robustesse, en termes de dépassement, temps de réponse et en précision statique sous l'effet d'un changement de consigne ou de présence de perturbation.

La synthèse paramétrique du correcteur PID utilise des méthodes précises qui permettent de déterminer les coefficients du correcteur.

Chapitre 2

Description du Système de Régulation de Niveau PUL-2

2.1. Introduction

On se propose dans ce chapitre de présenter le système de régulation du niveau ainsi que la vérification des différents composants. Ceci nous permettra de mettre en évidence les différents composants d'une boucle de régulation et de mieux comprendre le fonctionnement de notre station.

2.2. Description de l'unité de niveau PUL-2 [7]

La station de régulation de niveau / débit PUL-2 (figure 2.1) est constituée d'une cuve graduée pour les lectures de niveau (La plage du niveau va de 0 à 500mm) et d'un réservoir de stockage servant à extraire le liquide et le recueillir dans la cuve de régulation, une pompe à palette actionnée par un moteur électrique à courant continu (30 Vcc) qui fournit de liquide (eau) nécessaire pour atteindre le niveau souhaitée et le maintenir.

Le système possède aussi une électrovanne proportionnelle à commande électrique qui fonctionne entre [0-24v] et une vanne **TOR** (tout ou rien) qui est alimentée avec 24(Vcc).

Un transducteur de niveau contient un capteur de pression situé au fond de la cuve de processus. L'unité PUL-2 est équipée aussi de deux vannes manuelles, l'une agit sur le débit d'entrée et l'autre sur celui de sortie, permettent d'imposer la "charge" et ses variations.

L'unité de processus PUL-2 est équipée de plus, d'un débitmètre à aubes, permettant de réaliser des études sur les paramètres débit de liquide de l'unité. La cuve est transparente pour des raisons didactiques.

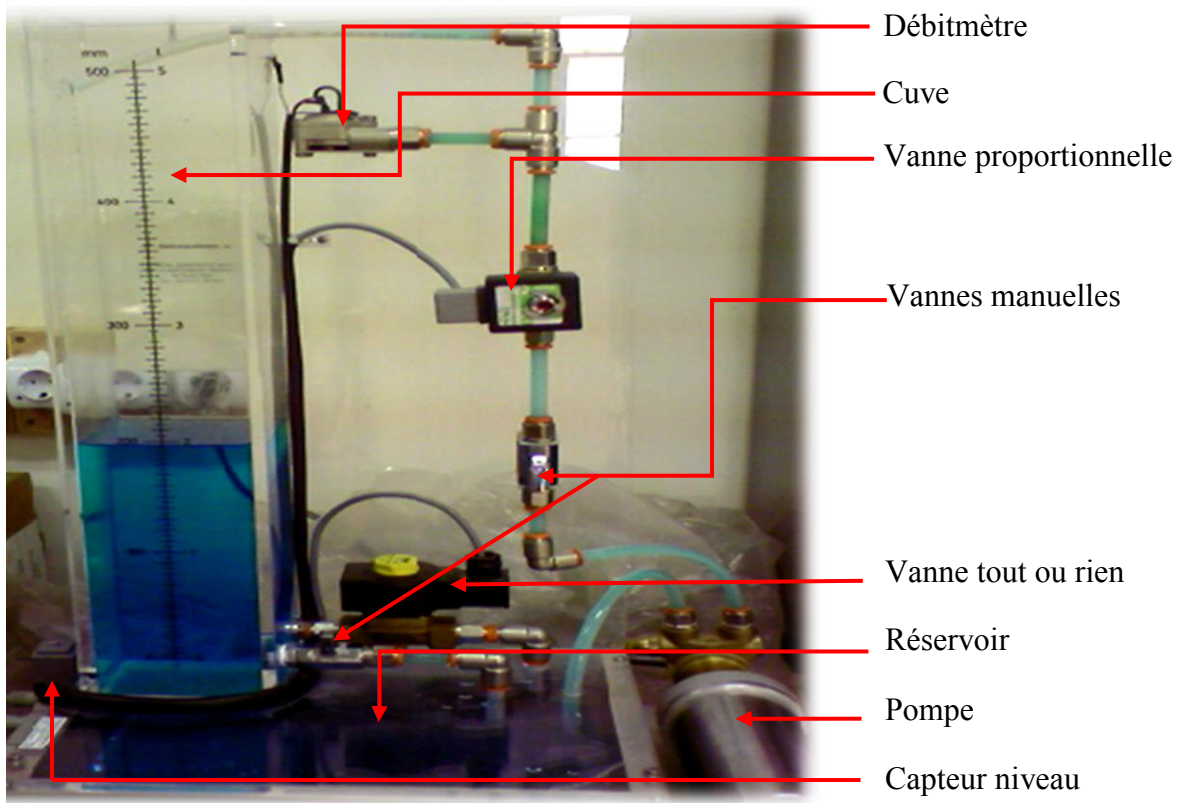


Figure2.1 : Station de niveau PUL-2

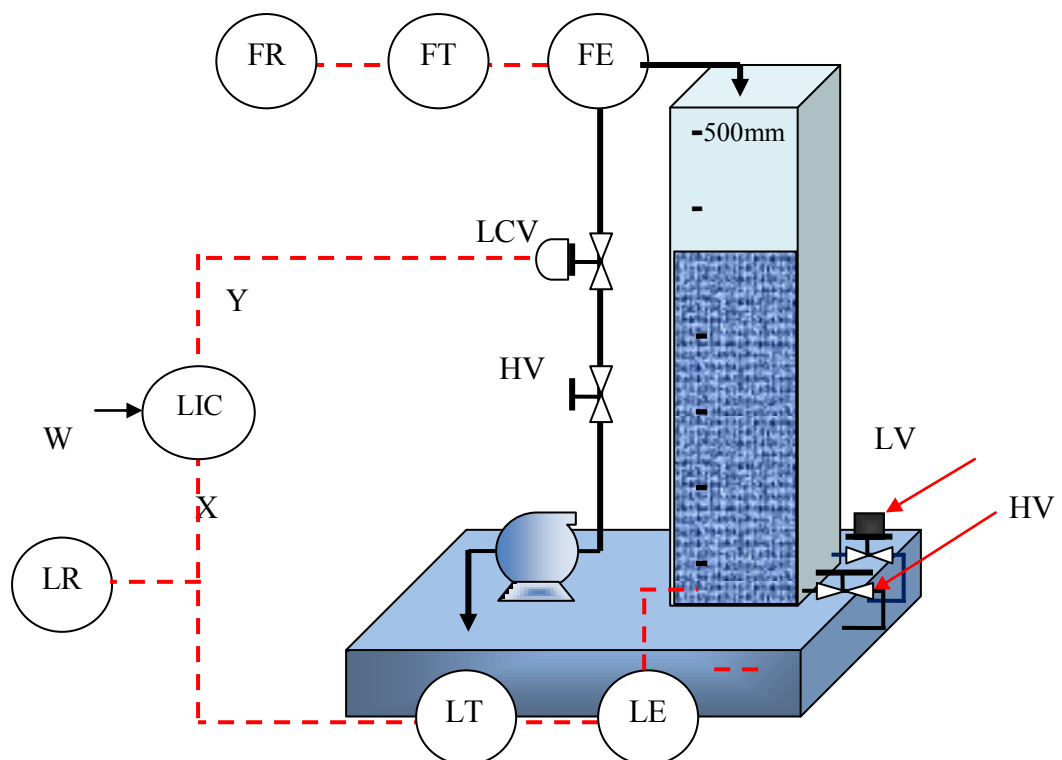


Figure2.2 : Schéma P&ID de la station de niveau

FE : Capteur de débit (Flow Element).

FT : Transmetteur de débit (Flow transmitter).

FR : Enregistreur de débit (Flow Recorder).

LIC : Régulateur indicateur de niveau (Level Indicator Controller).

LE : Capteur de niveau (Level Element)

LT : Transmetteur de niveau (Level Transmitter)

LR : Enregistreur de débit (Level Recorder)

LCV : Vanne proportionnelle (Level Controller Valve)

HV : Vanne manuelle (Hand Valve)

2.3. Le capteur de niveau

Le contrôle de niveau est utilisé dans de nombreux et divers domaines, afin de connaître l'état de remplissage de récipients, c'est-à-dire le niveau qu'y atteint le produit présent.

La mesure de niveau est une mesure continue, c'est-à-dire que le capteur et son conditionneur délivrent un signal proportionnel au niveau de liquide dans le réservoir. Ce qui permet à l'opérateur d'avoir la valeur exacte de volume du liquide (ou le volume encore disponible dans le réservoir).

2.4. Méthode de mesure de niveau

- **Mesure de niveau par flotteurs** : ces types de capteurs sont utilisés surtout pour la détermination des alarmes des niveaux limites, ou pour l'actionnement de vanne automatique de sécurité.
- **Mesure de niveau par variation de la capacité électrique des matériaux** : ces capteurs basés sur ce principe, sont employés surtout en présence de substances présentant des variations de capacité considérable.
- **Mesure de niveau par faisceau d'ultrasons** : le principe de mesure est basé sur la réflexion sonore de la surface des matériaux.

2.5. Conditionnement du signal du capteur

L'acquisition du signal de sortie du transducteur passe par le conditionneur de mesure à fin de rendre le signal de mesure standard et conforme aux normes de la carte d'acquisition. La plage de conditionnement disponible sur l'unité de régulation de niveau/débit est de (0-10Vcc) linéarisée.

3.6. Amplificateur de puissance de la vanne proportionnelle

L'organe réglant utilisé dans cette unité est une électrovanne proportionnelle qui assure la régulation de niveau de la station. Cette électrovanne est alimentée par une tension continue allant de 0 à 24Vcc. Pour fournir cette tension il faut réaliser un amplificateur de puissance.

L'amplificateur de puissance et le conditionneur de mesure sont rassemblés sous un seul boîtier comme le montre la figure (2.3)

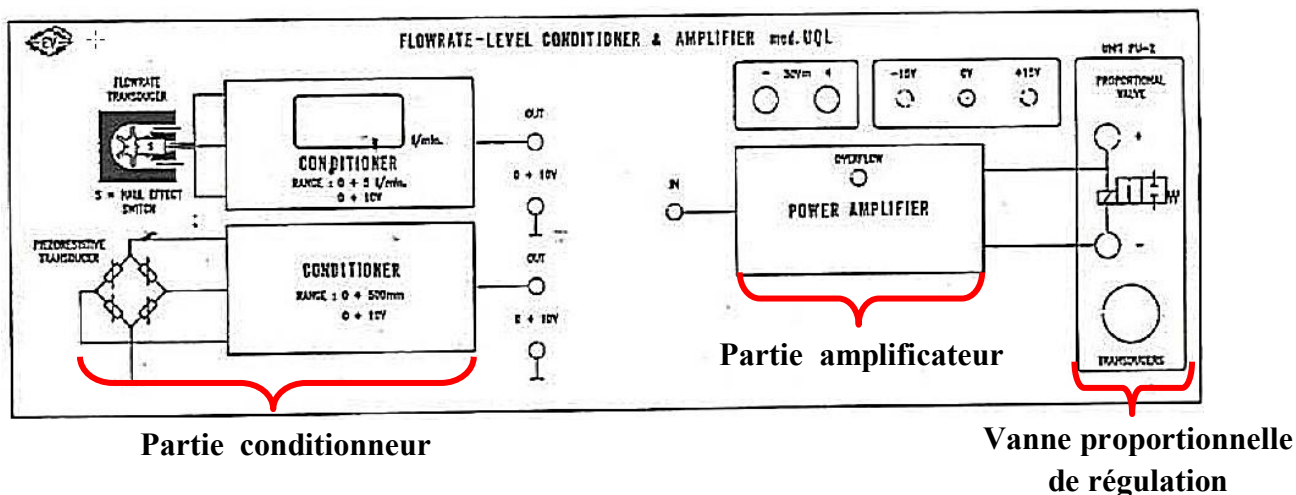


Figure 2.3 : Bloc conditionneur et amplificateur de puissance

2.7. Transducteur de niveau de la station PUL-2 [7]

Le capteur utilisé dans la station PUL-2 est fabriqué par la société MICRO SWITCH HONEYWELL DIVISION, il est référencé sous le sigle 170PC28HG1 (pour de plus informations, voir datasheet).

Les caractéristiques les plus importantes de ce transducteur sont les suivantes :

Sensibilité	0.0696 bars
Variation max de tension	45mV
Plage de température	0-50°C

Tableau 2.1 : Caractéristique du transducteur de niveau

2.7.1. Etalonnage du capteur de niveau

Le capteur-transmetteur utilisé a été étalonné sur l'intervalle [0-500mm] et le signal de mesure ainsi délivré est une tension dans l'intervalle [0-10V].

Nous avons vérifié cet étalonnage et nous avons obtenu la courbe suivante de la figure (2.4).

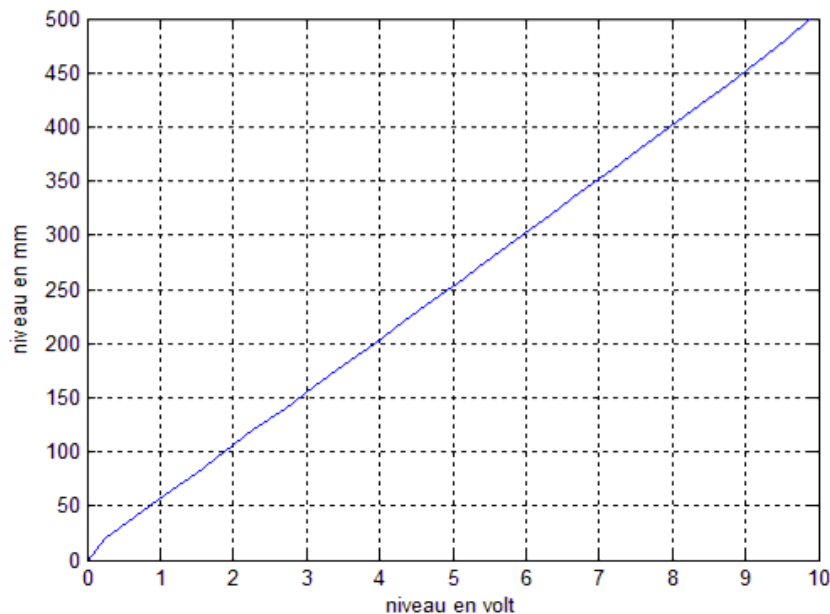


Figure 2.4 : Vérification d'état du capteur-transmetteur de niveau

La courbe d'étalonnage du capteur –transmetteur de niveau donne une tension de sortie du transmetteur proportionnelle et linéaire par rapport au niveau de la cuve.

2.8. Débitmètre à aubes

Le débitmètre à aubes avec sortie à impulsion et le transducteur de débit employés dans l'équipement utilisent le fluide en mouvement pour faire tourner une roue à aubes à une vitesse de rotation proportionnelle au débit.

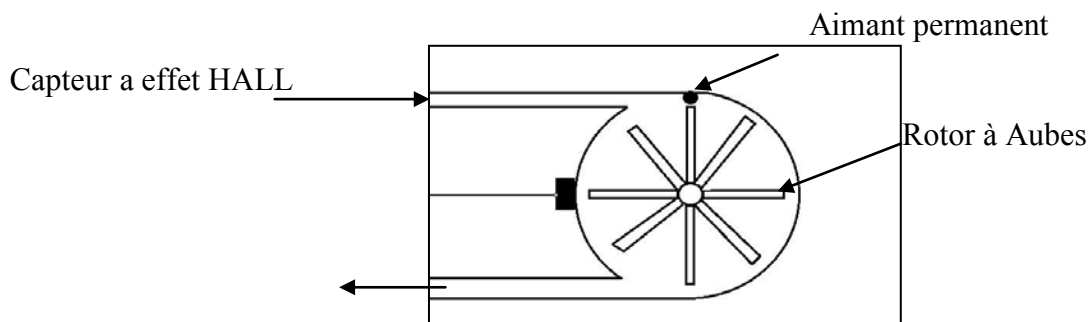


Figure 2.5 : Structure interne du débitmètre à aubes

On mesure la vitesse de rotation de la roue en tours/minute et à chaque tour correspond un certain volume bien déterminé de liquide déplacé.

Le rapport est linéaire à l'intérieur de certaines limites de débit et de viscosité du fluide.

On doit choisir les débitmètres à aubes pour un débit maximale supérieur de 30-50% au débit maximale prévu, car le fonctionnement en dessous de ce débit maximale est beaucoup plus sûr et fiable. Le modèle choisi, GICAR type A, est pourvu de deux aimants permanents placés

sur deux palettes opposées, lesquels passent devant un transducteur magnétique à effet Hall et fournissent deux impulsions électriques par tout.

2.8.1 Etalonnage du capteur de débit

Le capteur-transmetteur de débit utilisé a été étalonné sur l'intervalle [0-5L/min] et le signal de mesure ainsi délivré est une tension comprise dans l'intervalle [0-10Vcc].

Nous avons vérifié cet étalonnage et nous avons obtenu la courbe suivante de la figure (2.6).

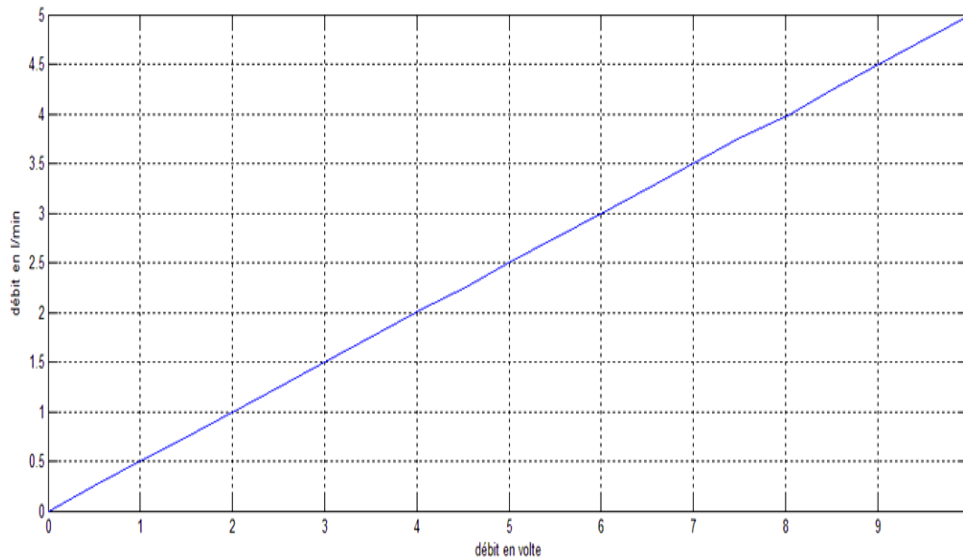


Figure 2.6: Vérification d'état du capteur-transmetteur de débit

2.9. Vanne proportionnelle

La vanne proportionnelle est une vanne dont le niveau du clapet est réglable à souhait. Elle est active pour une plage de tension spécifiée par le constructeur (exemple : 0-10V). Le niveau d'ouverture du clapet correspond donc à une certaine tension d'entrée appliquée à la vanne plus le signal s'approche de la tension maximale de la vanne plus le clapet de la vanne s'ouvre. Les vannes proportionnelles manuelles ont le même principe de fonctionnement que les vannes proportionnelles électriques, ces dernières doivent être ouvertes manuellement grâce à un système mécanique.

2.10. Contrôle automatique de la vanne proportionnelle

Le contrôle automatique de niveau est très répandue en industrie, sa structure de base est identique à celle des autres automatismes, sauf que la vanne proportionnelle qui est l'actionneur à contrôler nécessite un amplificateur de puissance.

On peut appliquer les différents régulateurs qu'on connaît déjà :

- On/Off, régulation Tout Ou Rien.
- Proportionnelle.
- Proportionnelle et intégrale.
- Proportionnelle, intégrale et dérivée.

- Régulation en cascade.
- Régulation flou.

2.11. Comportement de la vanne proportionnelle

On trace la variation de débit en fonction de la tension exercé sur la vanne.

D'après la figures (2.7), on constate que :

Quelque soit le sens de variation de débit de « 0,0 à 5,5 l/min » ou de « 5,5 l/min à 0,0 l/min » l'évolution est presque linéaire dans les deux cas. On peut observer aussi que pour les mêmes tensions exercées sur la vanne, le débit est légèrement différent selon le sens croissant ou décroissant d'ouverture ou de fermeture de la vanne modulante.

Tension d'entrée (volt)	0	1	1.53	2.15	2.39	2.63	2.85	3.07	3.32	3.53	4.78	5.32	5.90	6.6	7.12	8.01
Débit (l/min)	0	0	0.1	0.26	0.39	0.54	0.67	0.83	1.01	1.2	2.48	2.96	3.43	4	4.33	4.9

9.21	10	9.54	8.88	8.17	7.54	6.73	6	5.29	4.71	3.89	3.28	2.2	1.24	0
5.45	5.52	5.5	5.32	5.11	4.81	4.36	3.84	3.36	2.81	2.07	1.57	0.72	0.2	0

Tableau 2.2: Valeurs expérimentales du comportement de la vanne proportionnelle de régulation

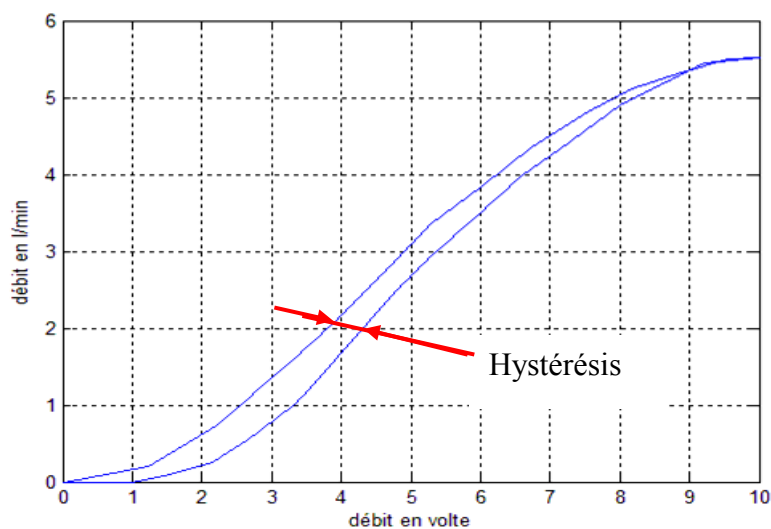


Figure 2.7: Hystérésis de la vanne proportionnelle de régulation

Il est a noté que le tracé du comportement de la vanne proportionnelle en fermeture ne suit pas exactement la même trajectoire que celui du comportement de la vanne proportionnelle en ouverture ce qui caractérise l'effet d'hystérésis.

2.12. Caractéristique statique du processus

Pour identifier le système autour d'un point de fonctionnement, il faut tracer la caractéristique statique (entrée /sortie) du processus.

Dans notre cas l'entrée (E) est une tension continue réglable de (0V-10Vcc) appliquée à une vanne régulatrice, la sortie (S) est la tension délivrée par le capteur-conditionneur de niveau. Les résultats expérimentaux obtenus sont présentés dans le (tableau 2.3).

tension d'entrée(v)	0	4.06	4.55	4.99	5.48	6.03	6.36	6.77	7.14	7.53
Niveau (mm)	0	50	90	150	215	295	335	390	435	490

Tableau2.3 : Relevé de la caractéristique statique

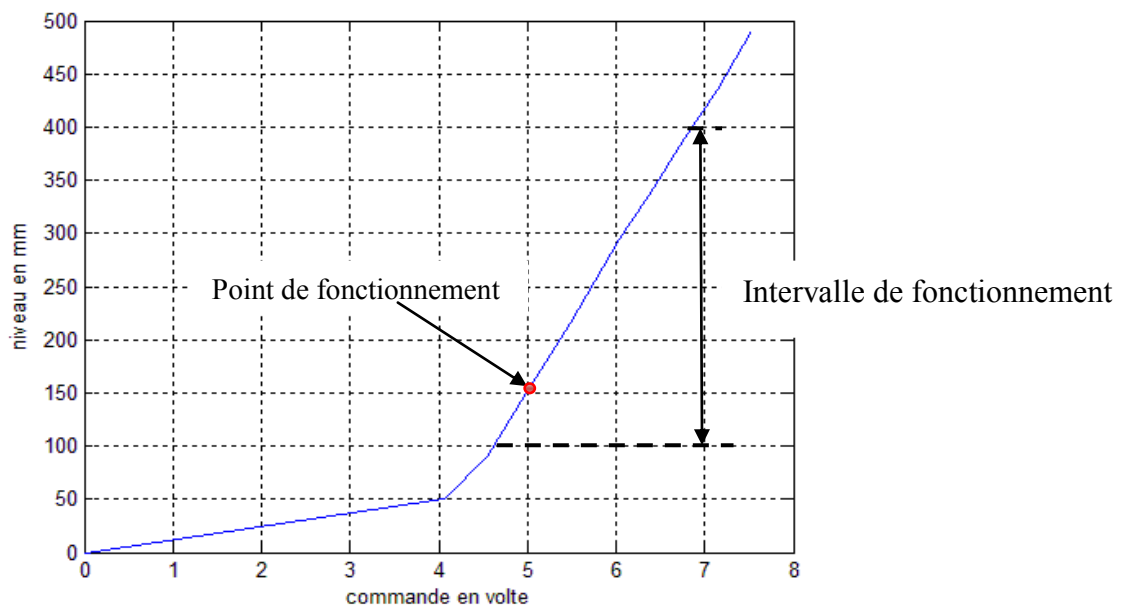


Figure 2.8: Tracé de la caractéristique statique du processus

La figure (2.8) ci-dessus présente un système qui n'est pas linéaire. Sur toute la plage de travail. Dans la suite de notre travail nous allons choisir une zone linéaire comprise entre l'intervalle 4.7 et 7V.

2.13. Identification du système

Après avoir choisis la zone linéaire, on sélectionne un point de fonctionnement dans cette zone, qui est 5.052V. On applique un échelon de tension de 6.070V comme le montre la (figure 2.9).

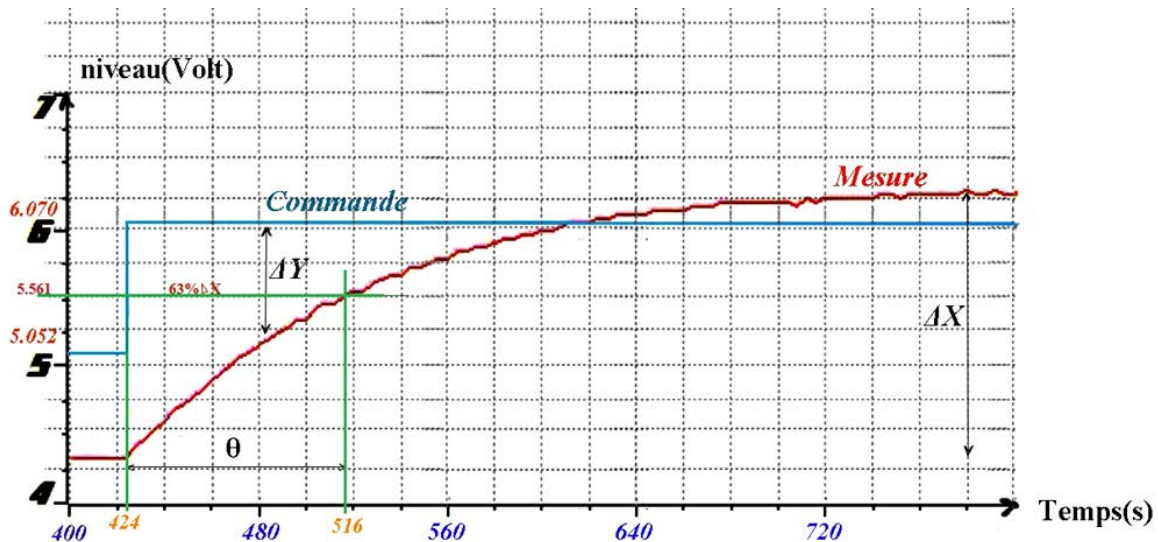


figure2.9 : Identification des paramètres du modèle

La réponse indicielle du système identifié, présente la réponse d'un élément du 1^{er} ordre sans retard, dont la fonction de transfert est sous la forme suivante :

$$G(p) = \frac{G_s}{1 + \theta P}$$

Avec un gain statique $G_s = \frac{\Delta X}{\Delta Y} = \frac{6.28 - 4.35}{6.070 - 5.052} = \frac{1.95}{1.01} = 1.91$

Pour déterminer la constante de temps : on trace conjointement la droite d'ordonnée (63% X) parallèle à l'axe des abscisses ce qui donne : $\theta = 92$ s

La fonction de transfert du système globale est la suivante:

$$G(p) = \frac{1.91}{1 + 92P}$$

2.14. Identification de système de débit

Pour cette identification nous allons garder toujours la même zone linéaire. Le point de fonctionnement ici est 5.008V nous allons soumettre le système de débit a un échelon de tension 6.04V comme le montre la (figure 2.10).

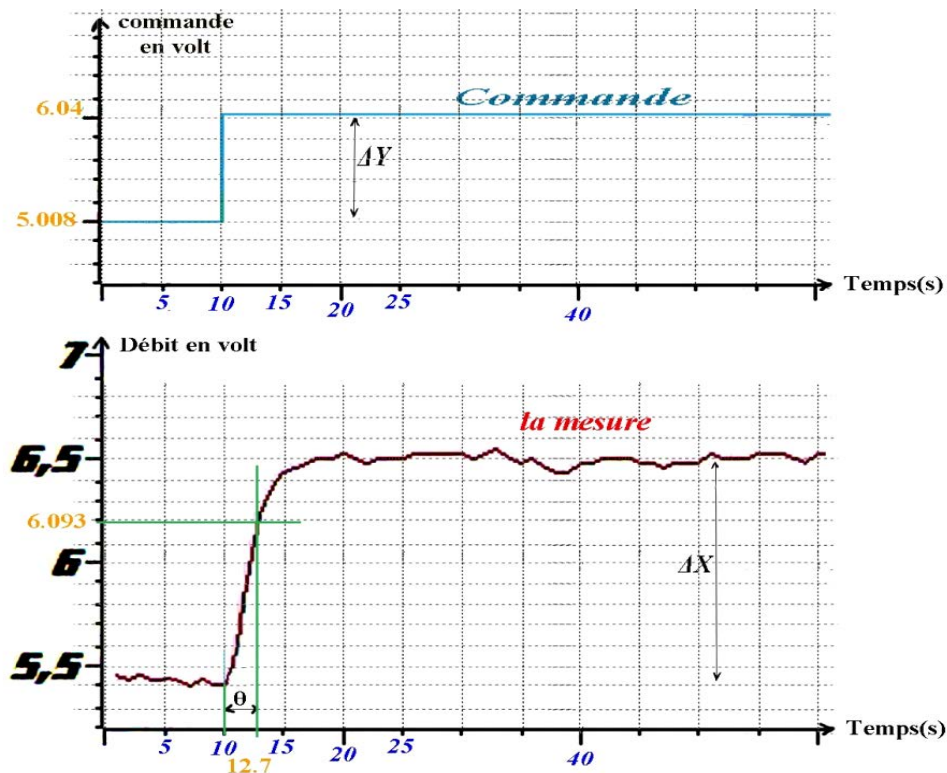


figure2.10 : Identification des paramètres du modèle

La réponse indicielle présente aussi un élément du 1^{er} ordre sans retard. L'identification des paramètres du modèle en utilisant la même méthode d'identification que précédemment

donne un gain statique : $G_s = \frac{\Delta X}{\Delta Y} = \frac{6.5 - 5.4}{6.04 - 5.008} = \frac{1.1}{1.03} = 1.06$

Avec $\tau = 2.7$ s

La fonction de transfert globale est :

$$G(p) = \frac{1.06}{1 + 2.7P}$$

Avec P étant l'opérateur de Laplace (d/dt)

2.15. Conclusion

Dans ce chapitre nous avons présenté la station de niveau PUL-2 et avec vérification des différents éléments de cette station (capteur –conditionneur de niveau, capteur –conditionneur de débit et la vanne proportionnelle). Après avoir fini de présenter la station nous avons réalisé une identification pour le système afin de trouver son modèle mathématique (fonction de transfert) qui servira à l'élaboration de la loi de commande et la conception des correcteurs à adapter.

Chapitre 3

Présentation de la commande par Logique Floue

3.1. Introduction

La commande floue a pour but de traiter des problèmes de commande de processus à partir uniquement des connaissances du comportement de système que les spécialistes du procédé doivent formuler sous forme linguistique (la connaissance des diverses fonctions de transferts n'est pas nécessaire). Ces connaissances du comportement permettent d'élaborer les règles de commande implantées dans le contrôleur flou.

3.2. La logique floue [8]

Contrairement à la logique classique (binaire), les variables de la logique floue peuvent prendre plusieurs valeurs entre 0 et 1. La logique floue est fondée sur la théorie des sous ensemble qui permet de traiter des problèmes vagues, imprécis et flous. La figure (3.1) représente respectivement l'ensemble classique (A) et l'ensemble flou (B)

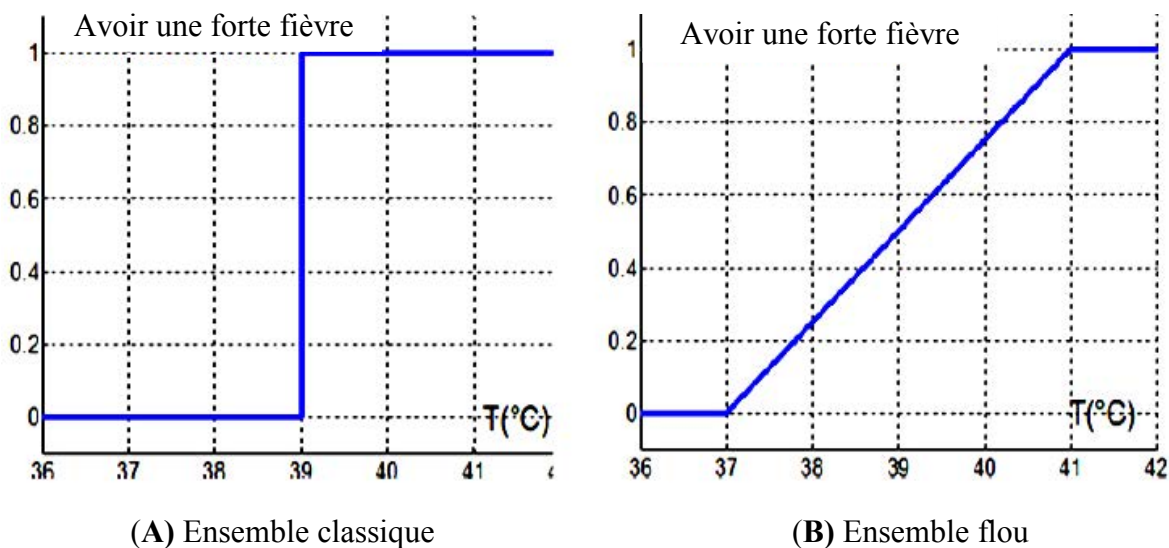


Figure 3.1: Représentation des ensembles

3.3. Régulation par la logique floue

Contrairement à la régulation classique, la régulation par la logique floue ne traite pas une information comme une relation mathématique bien définie, mais elle utilise des inférences avec plusieurs règles.

3.4. Principe de la commande floue [6]

La structure de la commande floue, peut être décomposée en trois grands modules. Comme présenté en figure (3.2).

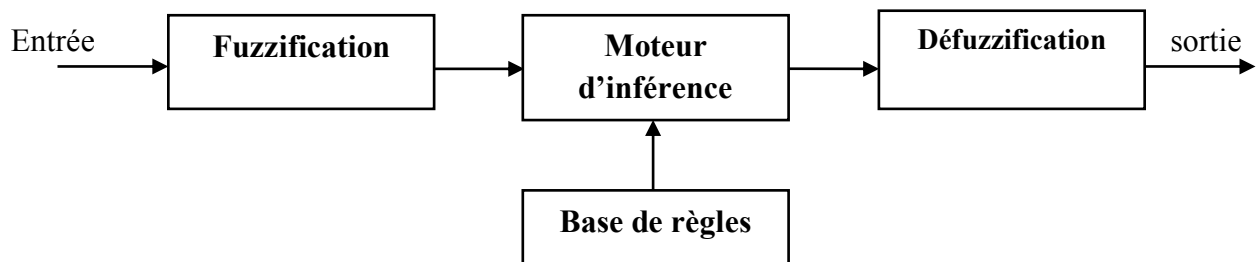


Figure 3.2 Structure générale d'une commande floue

Le bloc d'entrée de fuzzification permet d'associer à chacune des entrées réelles, en utilisant les fonctions d'appartenance, avec un degré d'appartenance pour chacun des sous ensemble flous définis sur l'univers de discours.

Le bloc représentant le moteur d'inférence constitue la base de règles, de type « Si.... Alors » permettant de calculer les degrés d'appartenance aux sous ensemble flous correspondant à la commande du système. Le bloc de sortie de défuzzification, permet de transformer les degrés d'appartenance des sous ensemble de commande en grandeur numérique applicable à l'actionneur de contrôle.

3.4.1. Fuzzification [6]

C'est la transformation des grandeurs physiques (entrées) en valeurs linguistiques (sous ensemble flou) représentés par leurs fonctions d'appartenances qui peuvent être traitées, ensuite, par le mécanisme d'inférence.

Pour réaliser cette phase de fuzzification on doit définir la plage de variations possibles de l'entrée considéré (univers du discours), une partition en classe floue de cet univers et les fonctions d'appartenances de chacune de ces classes.

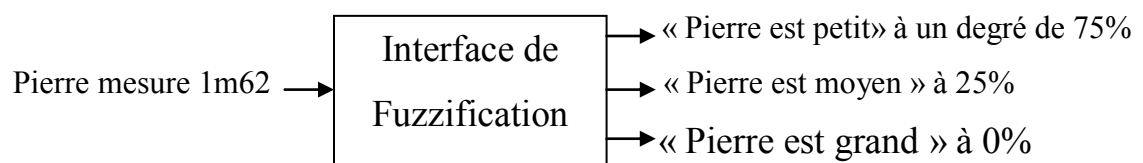


Figure. 3.3 : exemple de fuzzification

34.1.1. L’univers de discours

L’univers de discours d’une variable donnée qui représente son domaine de variation, cet univers de discours peut alors être divisé en plusieurs sous-ensembles au moyen de fonctions d’appartenance.

3.4.1.2. Ensemble flou

L’ensemble flou est l’élément de base de la logique floue introduit pour la première fois par L.Zadeh en 1965. L’ensemble flou est une extension de l’ensemble classique. Chaque ensemble peut prendre toutes les valeurs comprises entre 0 et 1.

3.4.1.2.1. Opérateurs entre ensembles flous

Les variables linguistiques sont liées entre elles au niveau des règles d’inférence par des opérateurs ET ou OU. Il s’agit d’opérateurs de la logique floue qui interviennent sur les fonctions d’appartenance représentant les variables linguistiques. De plus, il existe l’opérateur NON qui s’appelle (complément, négation, inverse).

Le tableau 3.1 présente les différents opérateurs proposés par L. Zadeh

	Opérateur de ZADEH	Opération logique	
Intersection	$\mu_{A \cap B} = \min(\mu_A, \mu_B)$	ET	
Union	$\mu_{A \cup B} = \max(\mu_A, \mu_B)$	OU	
Négation	$\mu_{\bar{A}} = 1 - \mu_A$	NON	

Tableau 3.1 : Opérateurs entre ensembles flous

3.4.1.3. Fonction d'appartenance

La fonction d'appartenance permet de mesurer le degré d'appartenance d'un élément à un ensemble flou. Elle est noté $\mu_a(x)$, les fonctions d'appartenance peuvent théoriquement prendre n'importe quelle forme mais en général les formes les plus utilisées sont définies par des formes géométrique.

3.4.1.4. Différentes formes pour les fonctions d'appartenance

Les fonctions d'appartenance les plus utilisées sont représenté sur les figures (3.4) suivantes :

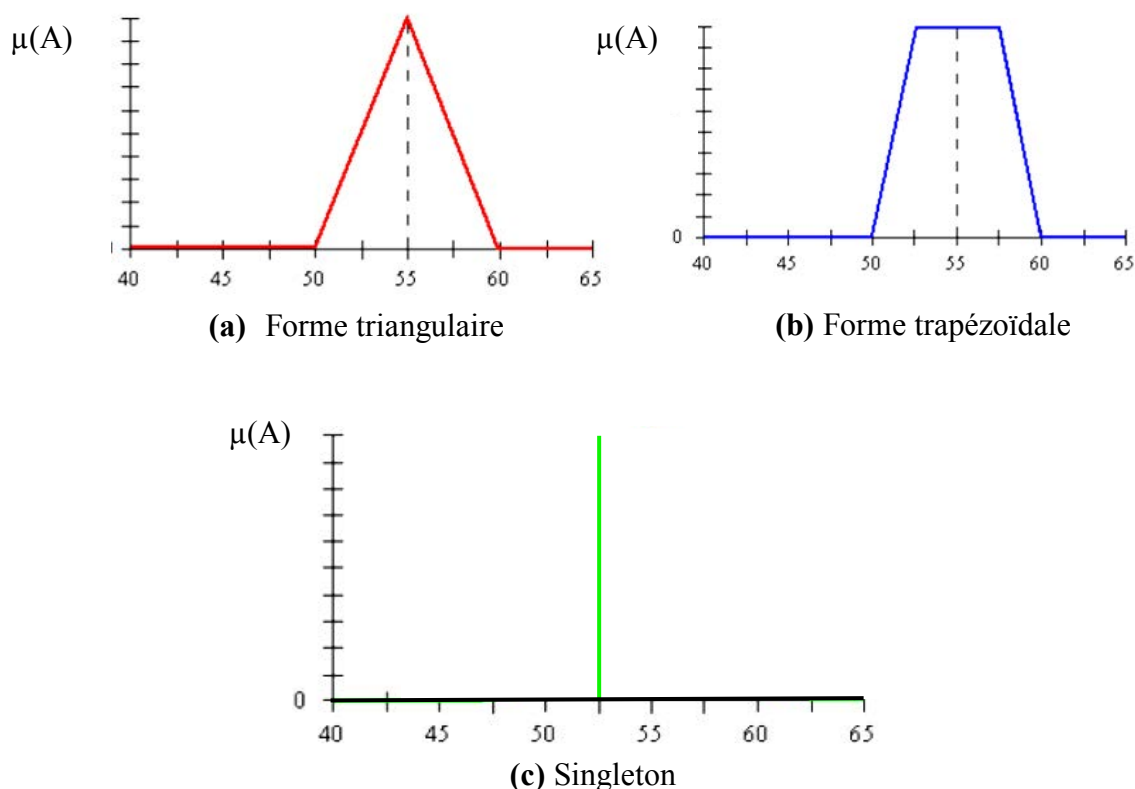


Figure 3.4: Fonctions d'appartenance

Dans le cas du réglage par logique floue, on utilise en général des formes trapézoïdales et triangulaires pour les fonctions d'appartenance. Bien qu'il n'existe pas de règles précises pour la définition des fonctions d'appartenance.

3.4.2. Moteur d'inférence floue [6]

Le moteur d'inférence floue est un mécanisme de décision qui permet à partir d'un fait observé de la base des règles floues, de prendre une décision en exploitant le raisonnement approximatif. Dans les inférences de régulateur par logique floue interviennent les opérateurs (et, ou) qui

s'appliquent aux variables à l'intérieur d'une règle tandis que l'opérateur OU lie les différentes règles.

Il existe plusieurs méthodes pour réaliser ces opérateurs dans une inférence et qui s'appliquent aux fonctions d'appartenance

- Méthode d'inférence max-min (Mamdani)
- Méthode d'inférence max-prod (Larsen)
- Méthode d'inférence somme-prod (Sugeno)

3.4.2.a. Méthode d'inférence Max-produit

La méthode max/produit réalise, en général au niveau de la condition, l'opérateur OU par la formation maximum et l'opérateur ET par la formation d'un minimum. La conclusion dans chaque règle, introduite par l'opérateur ALORS, lie le facteur d'appartenances de la condition avec la fonction d'appartenance de la variable de sortie X_r par l'opérateur ET, est réalisé dans cette méthode par la formation du produit. (voir l'exemple figure (3.5))

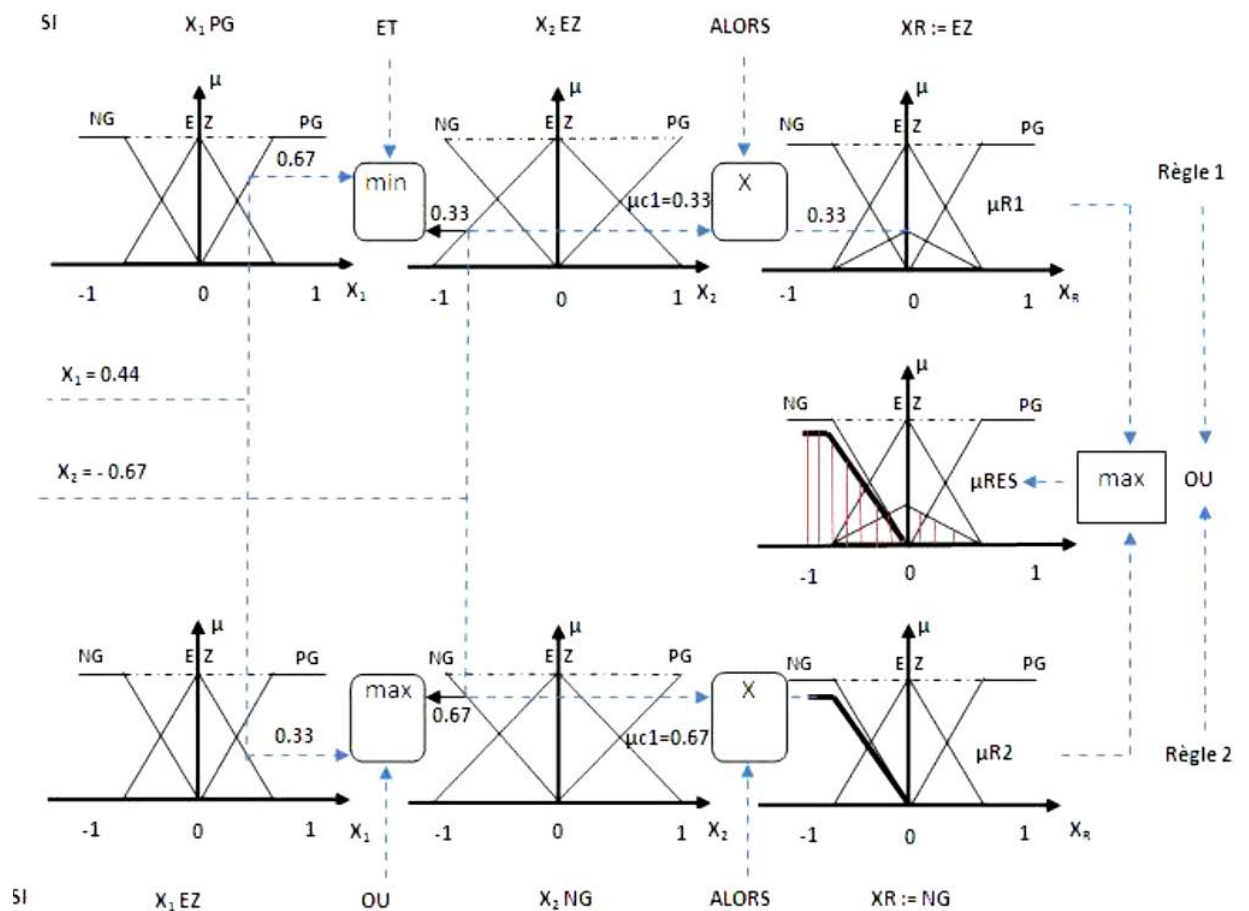


Figure 3.5 : Exemple d'inférence Max-produit

3.4.2.b. Méthode d'inférence Max-min

La méthode MIN/MAX (mamdani) est la méthode la plus universelle mais qui n'est guère applicable en raison du temps du calcul très long. Dans cette méthode l'opérateur OU réalise la formation d'un maximum et l'opérateur ET la formation du minimum.

La conclusion dans chaque règle, introduit par ALORS, lie le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie X_r par l'opérateur ET, réalisé dans le cas présent par la formation du minimum. la figure (3.6) donne un exemple d'application de cette méthode.

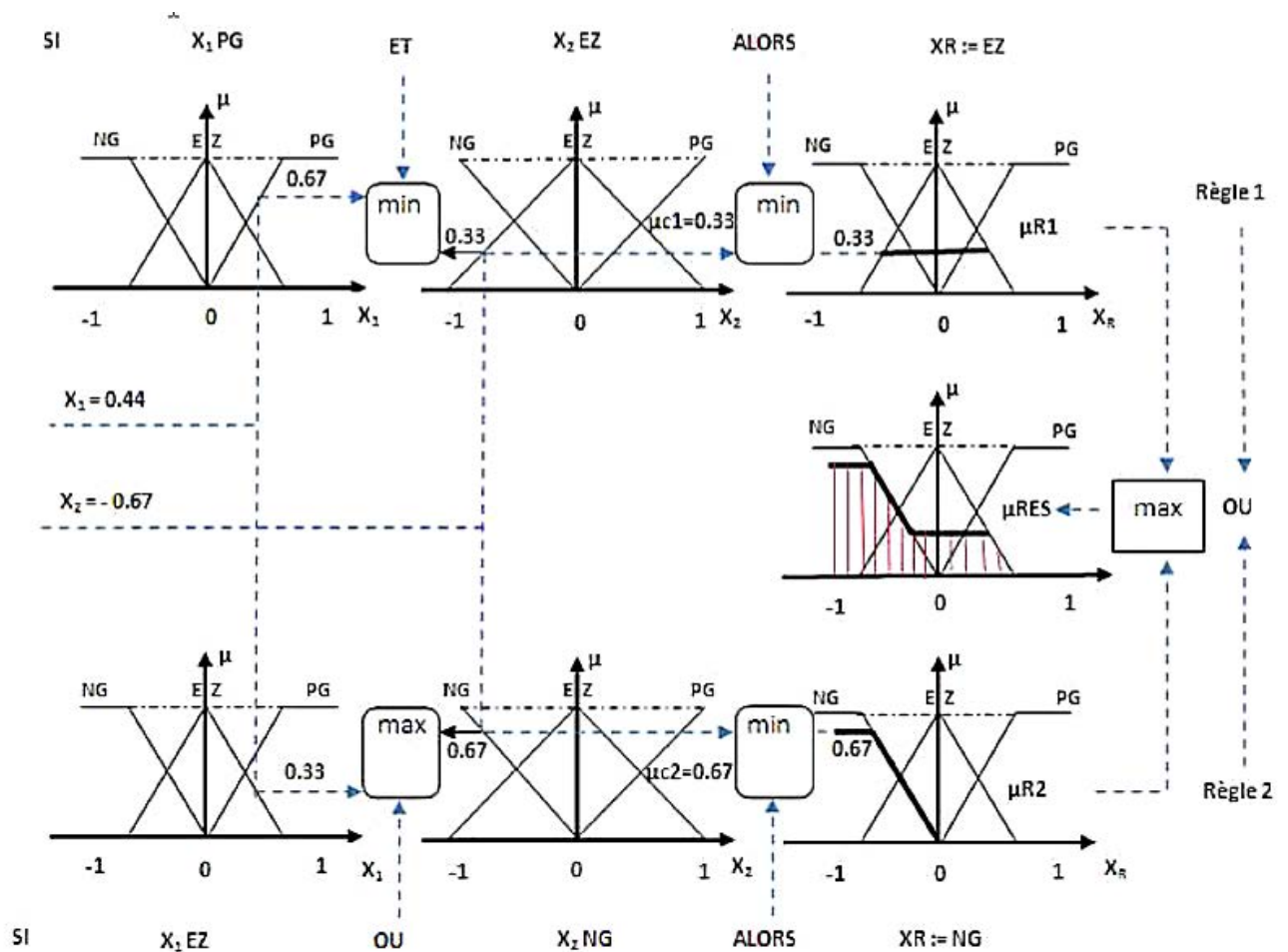


Figure 3.6 : Exemple d'inférence Max-min

3.4.2.c. Méthode d'inférence Somme-Produit :

Cette méthode est la plus utilisé car son temps de calcul est court.

Par opposition aux méthodes d'inférences précédentes, cette méthode réalise au niveau de la condition, l'opérateur OU par formation de la somme, plus précisément par la valeur moyenne, tandis que l'opérateur ET est réalisé par la formation du produit.

La conclusion de chaque règle introduite par ALORS, liant le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie X_r par l'opérateur ET, est réalisée la formation du produit, un exemple illustratif est donné dans la figure (3.7)

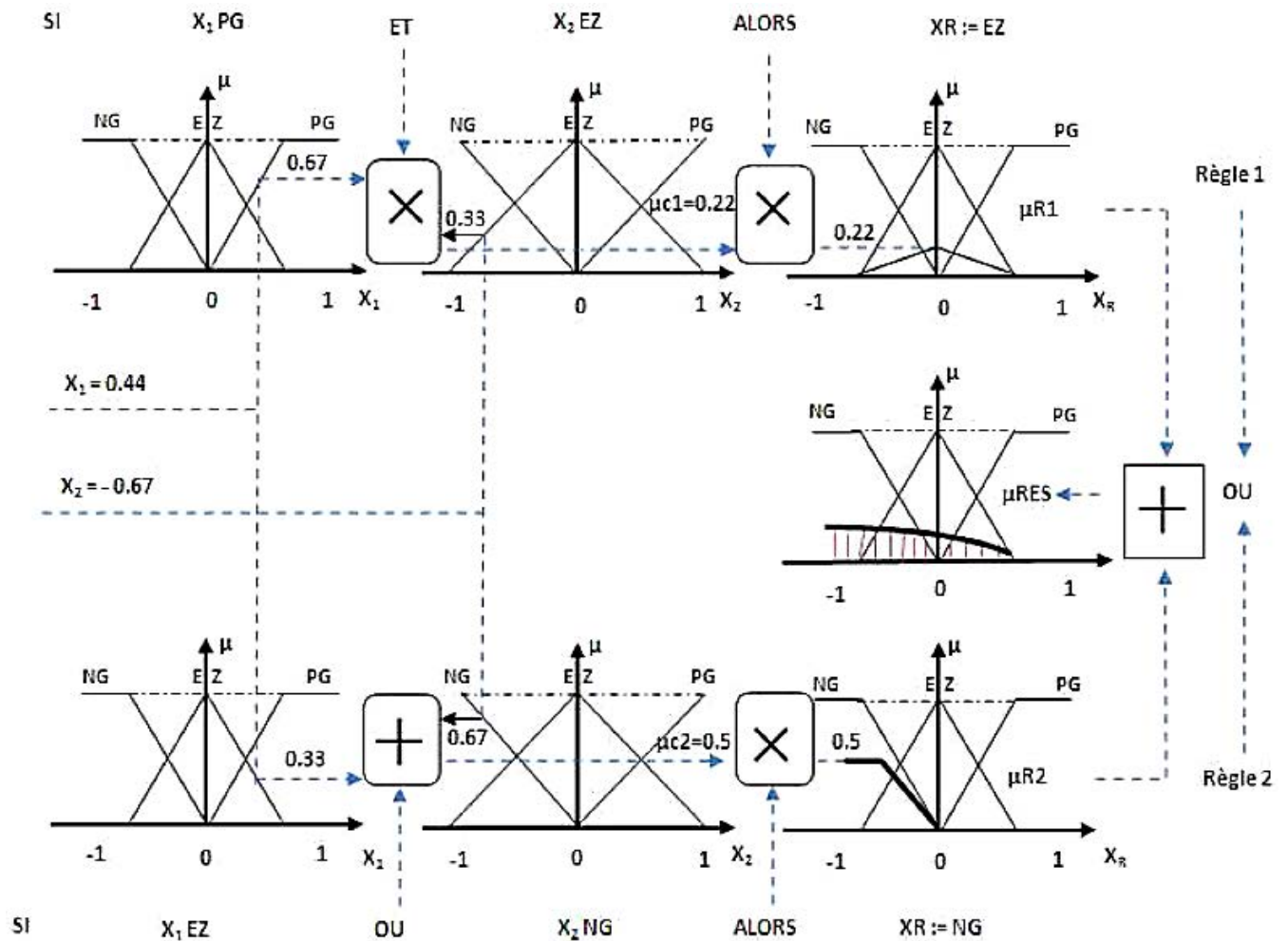


Figure 3.7 : Exemple d'inférence Somme-Produit

3.4.3. Base de connaissances

La base de connaissance contient la définition des ensembles flous, les facteurs d'échelle pour la normalisation des ensembles de référence et la partition de l'espace flou d'entrée et de sortie.

La base des règles floues rassemble l'ensemble des règles floues de type « Si-Alors » décrivant en termes linguistiques basés sur la connaissance d'un expert du comportement dynamique du système. L'ensemble des règles se présente sous la forme d'une énumération de type:

- Si condition 1 et/ou condition 2 (et/ou) alors action sur les sorties
- Si condition 3 et/ou condition 4 (et/ou) alors action sur les sorties
- Si condition 5 et/ou condition 6 (et/ou) alors action sur les sorties

3.4.4. Défuzzification

A la fin de l'inférence, l'ensemble flou de sortie est déterminé mais il n'est pas directement utilisable pour donner une information précise à l'opérateur ou commander un actionneur. Il est nécessaire de passer du « monde flou » au « monde réel » en utilisant le bloc de défuzzification. Il existe plusieurs méthodes de défuzzification, on rencontre souvent dans la littérature spécialisée les méthodes basées sur le calcul du « centre de gravité » ainsi que la méthode de la moyenne des maximums de l'ensemble flou.

3.4.4.1. Méthode moyenne des maximums (MM)

Cette méthode consiste à choisir comme valeur de sortie celle correspondant à l'abscisse de la moyenne des maximums de la fonction d'appartenance.

$$MM = \frac{\int x dx}{\int dx} \quad (3.1)$$

Ce qui représente la moyenne des valeurs de sorties les plus vrai semblables comme montrée en figure (3.9)

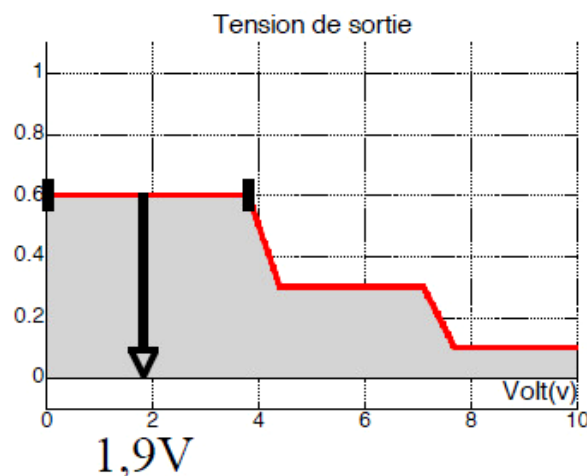


Figure 3.8 : Défuzzification par la méthode moyenne des maximums

3.4.4.2. Méthode du centre de gravité (CG)

Le calcul du centre de gravité permet bien d'obtenir une seule valeur pour la grandeur de sortie. Son calcul est cependant relativement complexe puisqu'il nécessite le calcul d'une intégrale, ou dans le cas simple de fonctions d'appartenance en raies, d'une somme pondérée.

$$CG = \frac{\int_a^b \mu_A(x) x dx}{\int_a^b \mu_A(x) dx} \quad (3.2)$$

Avec μ_A étant la fonction d'appartenance

La figure (3.9) donne une illustration de la défuzzification par la méthode du calcul du centre de gravité.

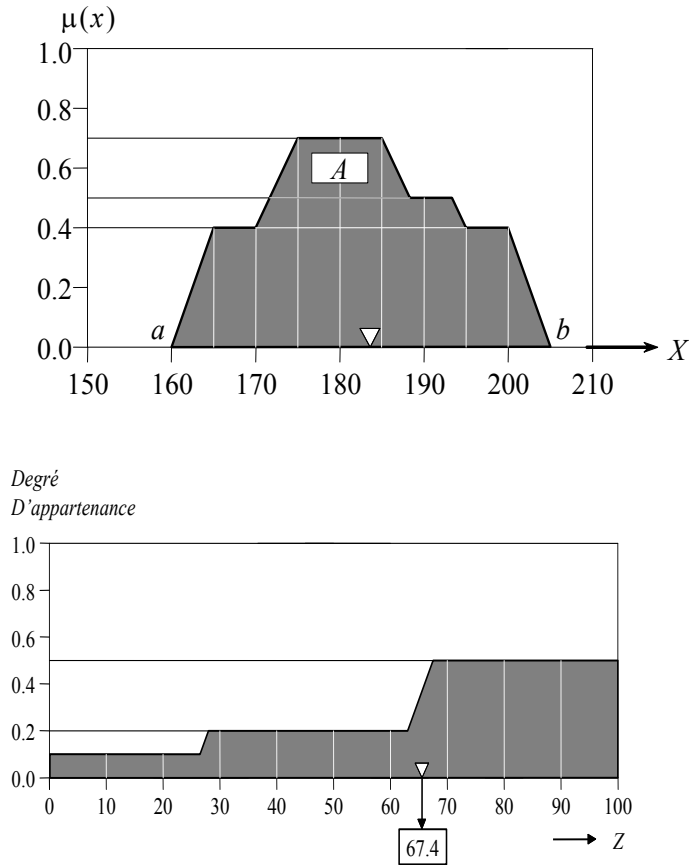


Figure 3.9 : Défuzzification par centre de gravité

Une estimation raisonnable du centre de gravité (CG) consiste à évaluation à partir d'un ensemble fini de point ce qui donne :

$$CG = \frac{\sum_{\min}^{\max} \mu_a(x)x}{\sum_{\min}^{\max} \mu_a(x)} \quad (3.3)$$

L'application pratique du calcul du CG par la formule (3.3) issue de l'illustration de la figure (3.9), donne le résultat suivant :

$$CG = \frac{(0+10+20) \times 0.1 + (30+40+50+60) \times 0.2 + (70+80+90+100) \times 0.5}{0.1+0.1+0.1+0.2+0.2+0.2+0.2+0.5+0.5+0.5+0.5} = 67.4$$

3.5. Différents types de régulateurs flous

3.5.1. Le régulateur de type Mamdani [10]

Mamdani avait présenté pour la première fois la technique de réglage par logique floue, qui consiste à déterminer un ensemble de comportement dynamique du système à commander.

Un contrôleur de ce type est conçu pour réguler, asservir une variable de sortie d'un système, soit uniquement à partir de l'erreur, ou à partir de l'erreur et de sa variation.

La forme générale des règles de Mamdani est :

$$\text{Si } x_1 = A1 \text{ ET } x_2 = A2 \text{ ALORS } y = B1$$

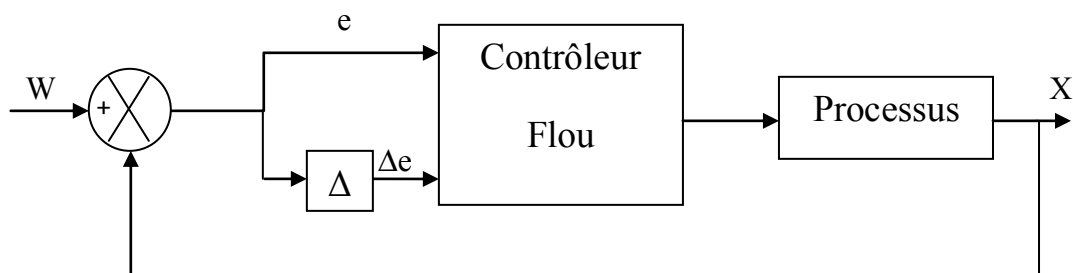


Figure 3.10. : Structure régulateur flou de type Mamdani

3.5.2. Régulateur de type Sugeno

Le modèle de Sugeno est connu sous le nom TSK, car il a été proposé par Talkagi, Sugeno et Kang en 1988. Dans ce type de régulateurs les sorties sont des fonctions linéaires. La méthode de Sugeno est bien adaptée à l'automatique et aux problèmes d'ingénierie en général. Le système flou de type Sugeno, utilise des règles écrites de la manière suivante :

$$\text{Si } x_1 = A1 \text{ ET } x_2 = A2 \text{ ALORS } y = z_1x_1 + z_2x_2 + z_0$$

3.6. Transcription du principe de la base floue en langage Ladder [11]

Le langage Ladder, est très utilisé par les automaticiens pour écrire des combinaisons logiques. Il permet en effet de les représenter graphiquement.

En voici un exemple, traitant le confort de l'air ambiant: l'air chaud et humide est inconfortable. Les situations les plus confortables thermiquement sont celles pour lequel l'air est chaud et sec, ou froid et humide. Cette constatation physiologique peut être transcrite par le Ladder flou de la figure (3.11) correspondant à la combinaison suivante :

Confort bon = (Température faible ET Humidité forte) OU (Température forte ET Humidité faible) Elle représente une définition possible de la sensation de confort ressentie par une personne dans une ambiance thermique pour laquelle l'air est immobile.

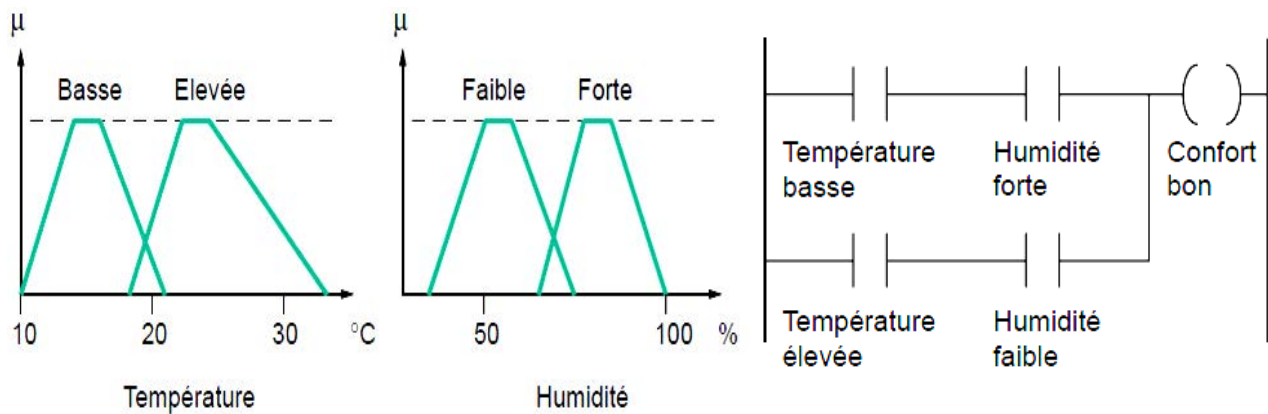


Figure 3.11 : Transcription du principe de base floue en Ladder

3.7. Avantages et inconvénients du régulateur flou

La commande par logique floue comporte évidemment un certain nombre d'avantages et d'inconvénients.

a) Avantages :

- La non-nécessité d'une modélisation du processus à commander, ainsi que d'une analyse mathématique profonde.
- La possibilité de bénéficier et d'implémenter des connaissances et des expertises humaines sur le système à commander.
- La maîtrise de systèmes à régler avec un comportement complexe (fortement non linéaire et difficile à modéliser).

Inconvénients :

- Le manque de méthodes systématiques précises pour la conception et la synthèse d'une commande par logique floue
- L'impossibilité de la démonstration de la stabilité du circuit de commande en toute généralité, (en absence d'un modèle valable)
- La cohérence des inférences est non garantie a priori (apparition possible des règles d'inférence contradictoires).

3.8. Conclusion

Dans ce chapitre, nous avons présenté les concepts de la logique floue ainsi que description générale des constituants d'un système flou. La logique floue ne nécessite pas de modèle du système à commander, il est uniquement nécessaire d'avoir une vue externe du comportement de ce système. Une analyse de ce comportement permet d'élaborer les règles de commande implantées dans le contrôleur flou.

Dans la suite des travaux, nous allons passer à l'implémentation d'un correcteur flou sous Automate Programmable Industriel de type SIMATIC en se basant uniquement sur les outils utilisés dans la programmation en langage Ladder.

Chapitre 4

Développement de la solution programmable et de supervision sous environnement SIMATIC-S7 / WINCC flexible

4.1. Introduction

L'objectif de ce chapitre est de présenter une description du matériel (API) et logiciels (step7/WinCC) utilisés pour l'implémentation des lois de commande pour réaliser une régulation du Processus de niveau PUL-2, par le correcteur PID intégré et d'un correcteur flou développé à base d'un API SIMATIC S7- 300 qui est programmé sous le logiciel Step7 ainsi que la conception d'une interface IHM de supervision en utilisant l'outil WinCC flexible.

4.2. Historique sur l'automate programmable industriel (API) [9]

Les automates programmables sont apparus aux USA vers les années 1969, à l'époque ils étaient destinés essentiellement à automatiser les chaînes de montages automobiles. Développé par la Bedford Associates du Massachusetts aux USA. Le MODICON 084 (MODular DIGital CONTroller) a été le 1^{er} API au monde à avoir intégré l'industrie productive.

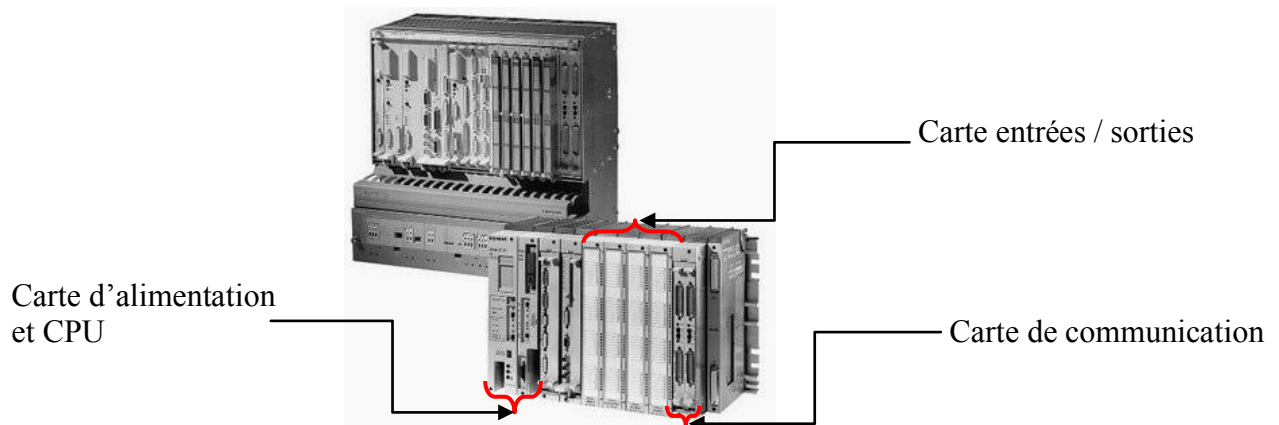


Figure 4.1 : API (MODICON 084)

4.3. Définition

API (Automate Programmable Industriel) est un appareil électronique destiné à la commande de processus industriels par un traitement séquentiel. Qui comporte une mémoire programmable par un utilisateur automaticien à l'aide d'un langage adapté (Le langage List, Le langage Ladder...etc) pour le stockage interne des instructions donnée afin de satisfaire un objectif donné. L'automate permet de contrôler, coordonner et d'agir sur l'actionneur, ce qui permet de simplifier la tâche d'automatisation des processus industriels.

4.4. Structure de l'automate programmable industriel

4.4.1. Aspect extérieur

Les automates peuvent être de type compact ou modulaire

a) Automate de type compact :

Il intègre le processeur, l'alimentation, les entrées et les sorties. Selon les modèles et les fabricants, il pourra réaliser certaines fonctions supplémentaires (comptage rapide, E/S analogiques ...) et recevoir des extensions en nombre limité. Ces automates sont de fonctionnement simple et ils sont généralement destinés à la commande des petits automatismes (micro automate).

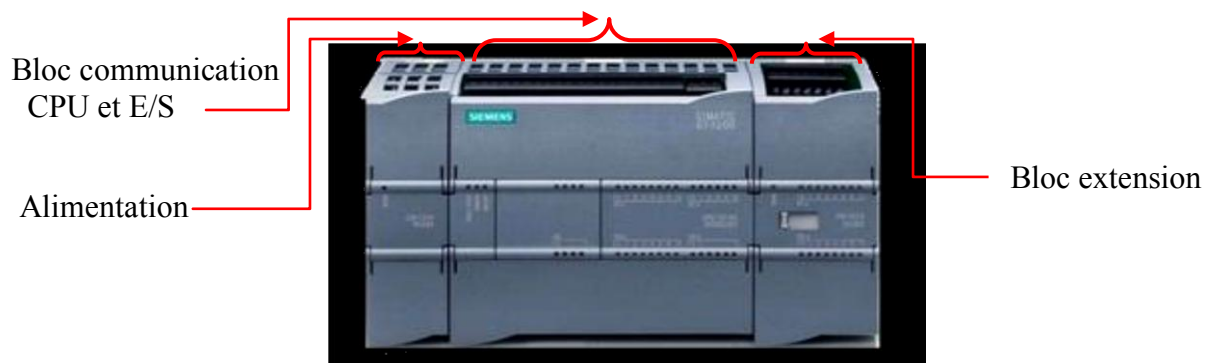


Figure 4.2: API d'un type compact (siemens-S200)

b) Automate De type modulaire

Le processeur, l'alimentation et les interfaces d'entrées / sorties résident dans des unités séparées (modules) et sont fixées sur un ou plusieurs racks contenant le "fond de panier" (bus plus connecteurs). Ces automates sont intégrés dans les automatismes complexes où de puissance, ils ont une grande capacité de traitement et une flexibilité.

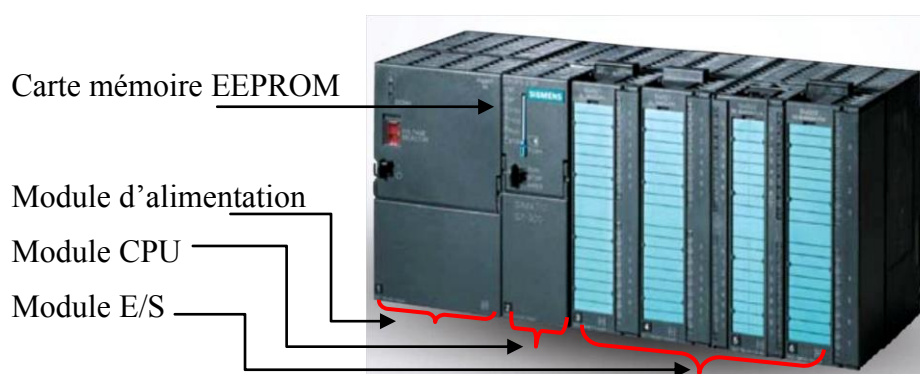


Figure 4.3: API d'un type modulaire (siemens S7)

Remarque

Les automates compacts permettent de commander des sorties en tout ou rien et gèrent parfois des fonctions de comptage et de traitement analogique. Par contre Les automates modulaires permettent de réaliser de nombreuses autres fonctions grâce à des modules intelligents que l'on dispose sur un ou plusieurs racks. Ces modules ont l'avantage de ne pas surcharger le travail de la CPU car ils disposent bien souvent de leur propre processeur.

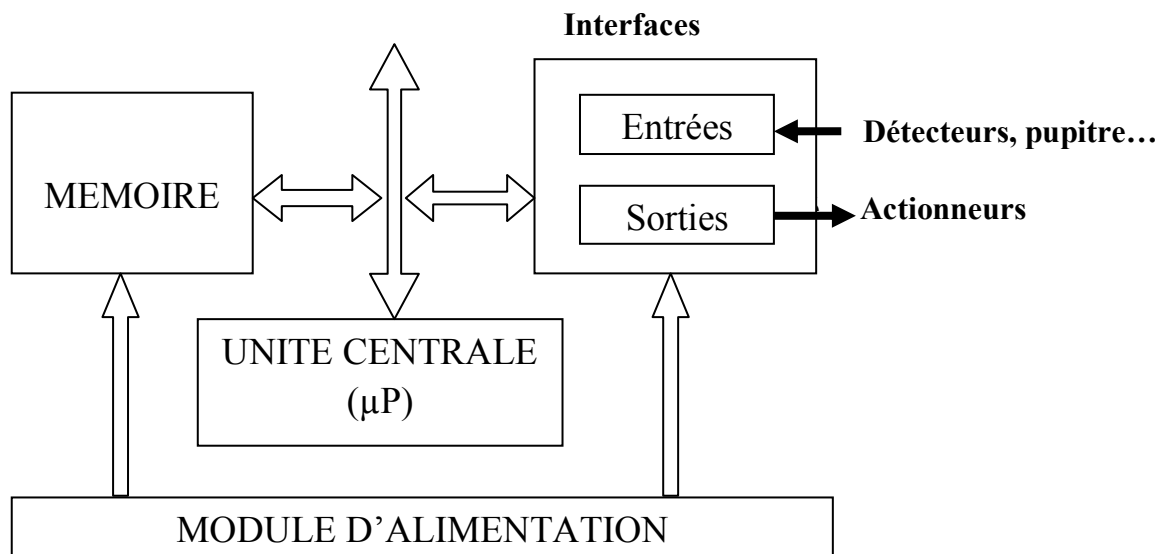
4.4.2. Structure interne de l'automate programmable industriel

Figure 4.4 : Structure interne des automates.

La robustesse et la facilité d'emploi des automates programmables industriels font qu'ils sont très utilisés dans la partie commande des systèmes industriels automatisés. L'automate programmable reçoit et envoie les informations par modules d'entrées et de sorties (logiques, numériques ou analogiques) et puis commandé par CPU ("Central Processing Unit) suivant le programme inscrit dans sa mémoire. Un API se compose donc de quatre grandes parties (voir la **figure 4.4**) :

- ✓ Le processeur.
- ✓ La zone mémoire.
- ✓ Le module Entrée/Sortie.
- ✓ Module d'alimentation.

a) Module d'alimentation : ce module doit fournir l'énergie nécessaire au fonctionnement correct de l'ensemble de l'automate. Il sera dimensionné en fonction des consommations des différentes parties ;

b) Unité centrale : elle traite les données et réalise toutes les fonctions logiques, arithmétiques et de traitement numérique (transfert, comptage, temporisation ...). Elle contient en mémoire le

programme et élabore donc les ordres de commande. Son cœur est composé d'un microcontrôleur alimenté en 5 volts ;

c) Les bus interne : ce sont des circuits chargés d'adapter en tension et en courant les signaux entre l'unité centrale et les entrées-sorties. Ils assurent en outre un isolement entre les entrées-sorties et l'unité centrale ;

d) Mémoires : Elles permettent de stocker le système d'exploitation (ROM ou PROM), le programme (EEPROM) et les données système lors du fonctionnement (RAM). Cette dernière est généralement secourue par pile ou batterie. On peut, en règle générale, augmenter la capacité mémoire par adjonction de barrettes mémoires type PCMCIA ;

e) Interfaces d'entrées / sorties

- **Interfaces d'entrées** : ce sont des circuits spécialisés capables de recevoir en toute sécurité pour l'automate les signaux issus des capteurs. Elles peuvent être logiques (T.O.R.), analogiques, ou numériques.
- **Interfaces de sorties** : ce sont des circuits spécialisés capables de contrôler en toute sécurité pour l'automate les circuits extérieurs. Elles peuvent être logiques (T.O.R.), analogiques, ou numériques.

4.5. Principe de fonctionnement d'un automate programmable industriel

L'automate programmable fonctionne par déroulement cyclique du programme. Le cycle comporte trois opérations successives qui se répètent comme suit :

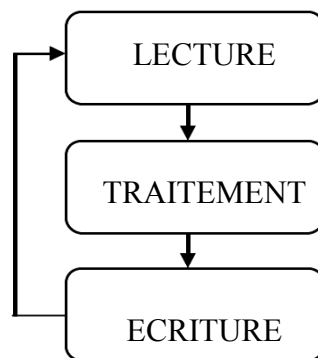


Figure 4.5: Cycle de fonctionnement d'un API

4.6. Les types des automates programmables industriels

4.6.1. Automates de petite gamme : ces automates sont destinées pour de petite application. Le nombre d'entrées sorties ne dépasse pas 48. Ils se présentent dans les boîtiers compacts où tous les modules (CPU, Alimentation, Module d'E/S, interface de communication) sont intégrés dans une mémoire boîtier. Il ne dispose d'aucune possibilité d'extension.

4.6.2. Automates de moyenne gamme : dans cette gamme le nombre d'E/S peut atteindre 400, ces automates ont une structure modulaire extensible.

4.6.3. Automates de haute gamme : ce sont des automates super puissant dont les performances permettant de gérer jusqu'à 2024 E/S et plus. Il dispose d'une structure modulaire

4.7. Nature des informations traitées par l'automate [14]

Les informations peuvent être de type :

4.7.1. Tout ou rien (T.O.R.)

L'information ne peut prendre que deux états (vrai/faux, 0 ou 1). C'est le type d'information délivrée par un détecteur, un bouton poussoir...etc.

4.7.2. Analogique

L'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température...).

4.7.3. Numérique

L'information est contenue dans des mots codés sous forme binaire ou bien hexadécimale.

C'est le type d'information délivrée par un ordinateur ou un module intelligent.

4.8. Choix d'un automate programmable industriel

Le choix d'un automate programmable est en premier lieu le choix d'une société ou d'un groupe et les contacts commerciaux et expériences vécues sont déjà un point de départ.

Voici quelques critères essentiels du choix d'un automate programmable industriel :

- Les capacités de traitement du processeur (vitesse, données, opérations, temps réel...).
- Le type des entrées/sorties nécessaire.
- Le nombre d'entrées/sorties nécessaire.

4.9. Avantages et Inconvénients des APIs [10]

4.9.1 Les Avantages

- Simplification du câblage.
- Modifications du programme faciles à effectuer par rapport à une logique câblée.
- Enormes possibilités d'exploitation.
- Fiabilité professionnelle.
- Possibilité de gérer des tâches parallèles.

Simplicité de la programmation : Pour des applications très simples, il existe des langages ne nécessitant quasiment aucune connaissance en programmation comme le langage Ladder.

4.9.2 Les inconvénients

- Le coût élevé du matériel, principalement avec les systèmes hydrauliques.
- La maintenance doit être structurée.

4.10. Automate S7-300 CPU314

L'automate s7-300 est l'un des automates les plus utilisés dans l'industrie grâce à son coût, ses avantages et ses caractéristiques. C'est un mini-automate modulaire pour les applications d'entrée et de milieu de gamme, avec possibilité d'extensions jusqu'à 32 modules, et une mise en réseau par l'interface Multipoint (MPI), PROFIBUS et Industrial Ethernet (PROFINET).

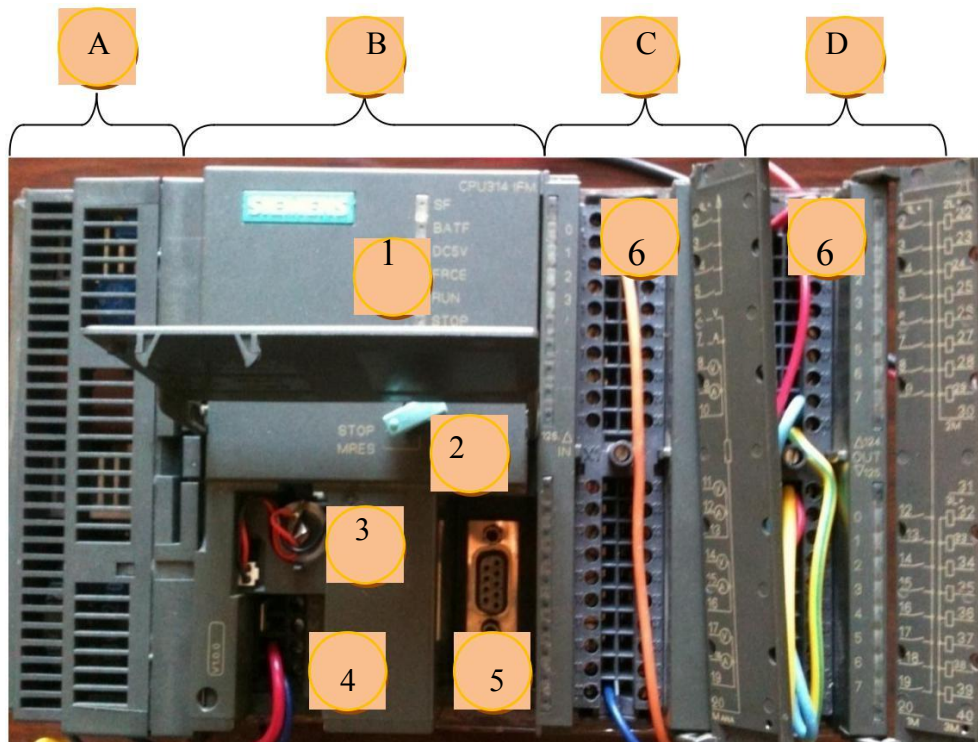


Figure 4.6: SIMATIC S7-300

1	LED de visualisation d'état défaut	4	Bornes pour la tension d'alimentation
2	Commutateur de fonctionnement	5	interface multi point (MPI)
3	Logement pour la pile de sauvegarde	6	Entrées /sorties intégrées
A	Module d'alimentation 24v	C	Module d'entrée intégré
B	La CPU 314	D	Module de sortie intégré

Tableau 4.1: Description des différents éléments de l'API S7 300.

4.11. Langages de programmation [9]

Les langages de programmation CONT, LIST et LOG pour S7-300/400 font partie intégrante du logiciel de base.

4.11.1. La liste d'instructions (LIST)

Est un langage de programmation textuel proche de la machine. Dans un programme LIST, les différentes instructions correspondent, dans une large mesure, aux étapes par lesquelles la CPU traite le programme. Pour faciliter la programmation, LIST a été complété par quelques structures de langage évolué (comme, par exemple, des paramètres de blocs et accès structurés aux données).

4.11.2. Le logigramme (LOG) :

C'est un langage de programmation graphique qui utilise les boîtes de l'algèbre de Boole pour représenter les opérations logiques. Les fonctions complexes, comme par exemple les fonctions mathématiques, peuvent être représentées directement combinées avec les boîtes logiques.

4.11.3 Le schéma à contacts (CONT, langage LADDER)

est un langage de programmation graphique. La syntaxe des instructions fait penser aux schémas de circuits. CONT permet de suivre facilement le trajet du courant entre les barres d'alimentation en passant par les contacts, les éléments complexes et les bobines.

4.12. Programmation de la régulation PID en utilisant le bloc FB 41 intégré [12]

Le logiciel de programmation Step7 offre des blocs fonctionnels (FB) de régulation PID comprennent les blocs pour :

- La régulation continue
- La régulation pas à pas
- La modulation de largeur d'impulsion

Type de régulation	Bloc de régulation	Type de la sortie de régulation
Continue	FB41 (DB41)	Analogique
pas à pas	FB42(DB42)	Impulsion
largeur d'impulsions	FB43(DB43)	Analogique

Tableau 4.2: Les différents types de régulation PID sous step7.

Les FB de régulation proposent une régulation purement logicielle, c'est-à-dire qu'un bloc contient toutes les fonctions du régulateur. Les données nécessaires au calcul cyclique sont stockées dans des blocs de données associés OB, les blocs de données d'instance, ce qui permet aux FB de les appeler plusieurs fois.

Le bloc FB41 sert à réguler des processus industriels à grandeurs d'entrée et de sortie continues sur les automates programmables SIMATIC S7. Le paramétrage du bloc FB 41 nous permet

d'activer ou de désactiver des fonctions partielles du régulateur PID et donc d'adapter ce dernier au système régulé.

Les fonctions les plus importantes sont: la consigne et la mesure, le bloc FB41 réalise un PID prêt à l'emploi avec une sortie continue et possibilité d'ajuster manuellement la valeur de sortie.

4.11.1. Fonctionnement de bloc PID (FB41) [12]

Dans la figure suivante représente le schéma du principe représentant le fonctionnement général d'un bloc PID (FB41) (figure 7.4) :

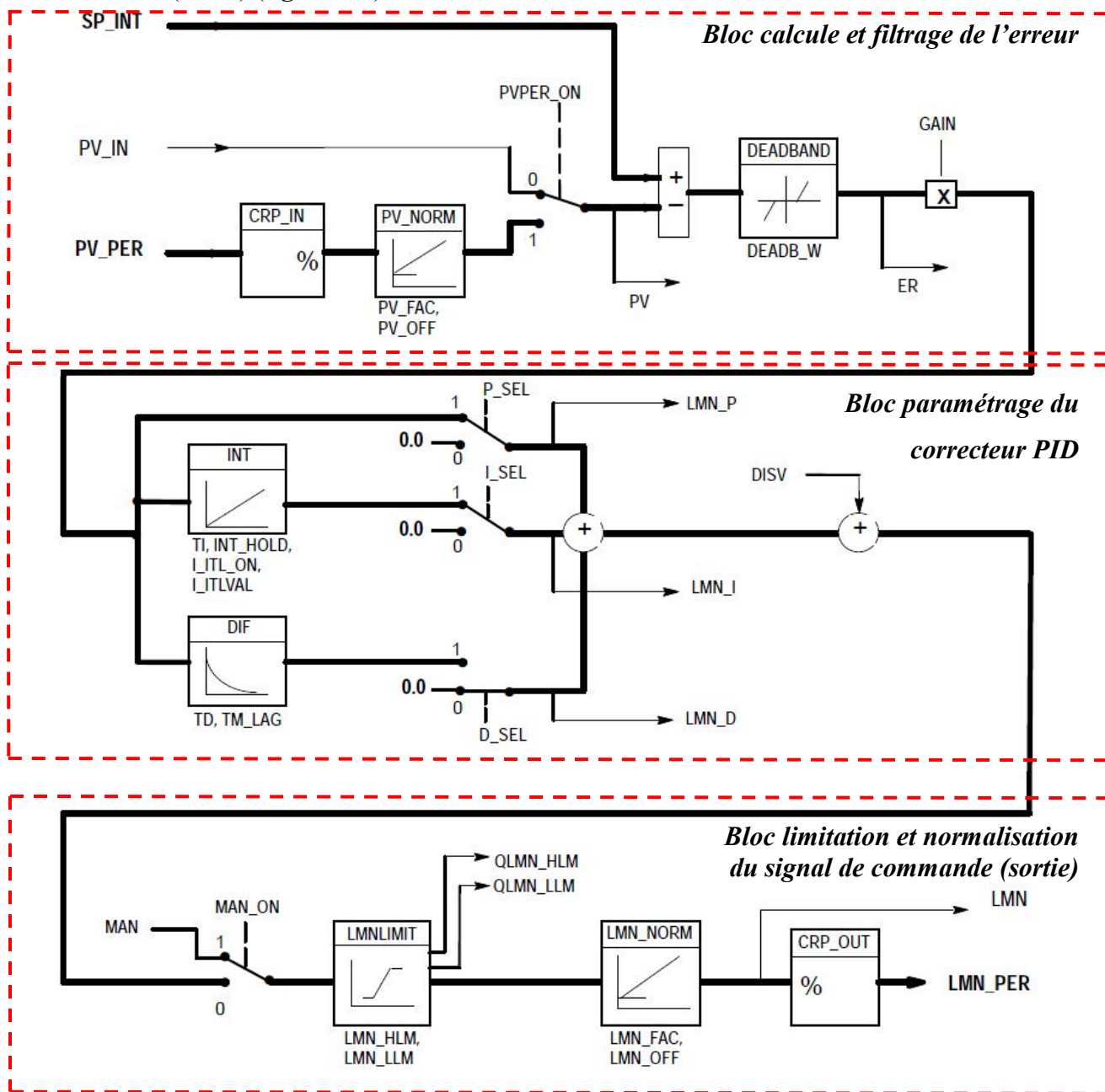


Figure. 4.7: Schéma fonctionnel de bloc PID (FB41)

Le bloc de la figure (4.7) est constitué de trois sous blocs, un bloc de calcul et de conditionnement du signal d'erreur conditionné passe par le bloc du paramétrage du régulateur PID (Ki, Td, Ti), le signal de sortie du bloc sera normalisé et limité dans le bloc de sortie afin d'être appliqué à l'actionneur proportionnel.

III.11.2 Paramétrage du régulateur PID

Les principaux paramètres du régulateur PID dont on aura besoin sont cités dans les deux tableaux (4.3) et (4.4) suivants :

Le tableau (4.3) suivant représente les paramètres d'entrée et types de données de réglage du régulateur PID :

Paramètres	Type de données	Description
COM_RST	BOOL	COMPLETE RESTART / Démarrage Le bloc renferme un sous-programme de démarrage qui est exécuté quand cette entrée est à 1.
P_SEL	BOOL	PROPORTIONAL ACTION ON / Activation de l'action proportionnelle Dans l'algorithme PID, il est possible d'activer et de désactiver séparément chacune des actions. L'action P est active quand cette entrée est à 1.
I_SEL	BOOL	INTEGRAL ACTION ON / Activation de l'action par intégration Dans l'algorithme PID, il est possible d'activer et de désactiver séparément chacune des actions. L'action I est active quand cette entrée est à 1.
D_SEL	BOOL	DERIVATIVE ACTION ON / Activation de l'action par dérivation Dans l'algorithme PID, il est possible d'activer et de désactiver séparément chacune des actions. L'action D est active quand cette entrée est à 1.
CYCLE	TIME	SAMPLE TIME / Période d'échantillonnage Le temps s'écoulant entre les appels de bloc doit être constant. Il est indiqué au niveau de cette entrée.
SP_INT	REAL	INTERNAL SETPOINT / Consigne interne Cette entrée sert à introduire une valeur de consigne.
PV_PER	WORD	PROCESS VARIABLE PERIPHERIE / Mesure de périphérie La mesure en format de périphérie est appliquée au régulateur par cette entrée.
GAIN	REAL	PROPORTIONAL GAIN / Coefficient d'action proportionnelle Cette entrée indique le gain du régulateur.
TI	TIME	RESET TIME / Temps d'intégration Cette entrée détermine la réponse temporelle de l'intégrateur.
TD	TIME	DERIVATIVE TIME / Temps de dérivation Cette entrée détermine la réponse temporelle de l'unité de dérivation.

PV_FAC	REAL	PROCESS VARIABLE FACTOR / Facteur de mesure Cette entrée est multipliée par la mesure. Elle sert à adapter l'étendue de valeur de mesure.
PV_OFF	REAL	PROCESS VARIABLE OFFSET / Décalage de mesure Cette entrée est ajoutée à la mesure. Elle sert à adapter l'étendue de valeur de mesure.

Tableau 4.3: Paramètres d'entrée

Le tableau_2 montre les paramètres de sortie ainsi que leurs types de données :

Paramètre	Type de données	Description
LMN_PER	WORD	MANIPULATED VALUE PERIPHERY / Valeur de réglage de périphérie Cette sortie fournit la valeur de réglage en format de périphérie.
PV	REAL	PROCESS VARIABLE / Mesure Cette sortie donne la mesure effective.
ER	REAL	ERROR SIGNAL / Signal d'erreur Cette sortie donne le signal d'erreur effectif.
LMN_P	REAL	PROPORTIONALITY COMPONENT / Composante P Cette sortie contient la composante proportionnelle de la grandeur de réglage.
LMN_I	REAL	INTEGRAL COMPONENT / Composante I Cette sortie contient la composante intégrale de la grandeur de réglage.
LMN_D	REAL	DERIVATIVE COMPONENT / Composante D Cette sortie contient la composante différentielle de la grandeur de réglage.
QLMN_HLM	BOOL	HIGH LIMIT OF MANIPULATED VALUE REACHED / Grandeur de réglage à la limite supérieure La valeur de réglage est toujours bornée à une limite supérieure et à une limite inférieure. Cette sortie signale le dépassement de la limite supérieure.
QLMN_LLM	BOOL	LOW LIMIT OF MANIPULATED VALUE REACHED / Grandeur de réglage à la limite inférieure La valeur de réglage est toujours bornée à une limite supérieure et à une limite inférieure. Cette sortie signale le dépassement de la limite inférieure.

Tableau 4.4: Paramètres de sorties

4.11.3. Mise en œuvre d'un régulateur PID sous step7

La programmation d'un automate industriel de type SIMATIC S7-300 en tant que régulateur PID se fait avec le logiciel STEP 7, les entrées sorties choisis de régulateur PID (bloc FB41) seront sauvegardés dans le bloc de donné (DB1) d'instance associé à l'appel du bloc régulateur FB 41.

Pour concevoir un régulateur PID nous devons suivre les deux étapes essentielles :

- Configuration matériels
- Elaboration d'un programme en langage Contact sous STEP7 avec affectation des blocs fonctionnelle et données.

4.13. Interface MPI et PC Adapter USB [13]

L'interface MPI (Multi Point Interface) est la liaison utilisée comme support de communication entre le PC et la CPU, et aussi qu'au sein d'un sous-réseau MPI. Cette interface est caractérisée par une vitesse de transmission (par défaut) de 187,5 Mbit/s [9].



Figure 4.8: PC Adapter USB-MPI

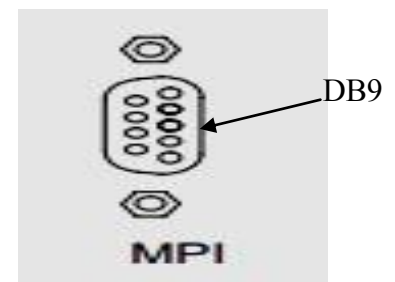


Figure 4.9: Interface (port) MPI

4.14. Superviseur HMI SIMATIC WinCC [15]

SIMATIC WinCC (Windows Control center) est un logiciel de Siemens qui permet de concevoir des IHM (Interface Homme Machine) dans le domaine des automates et de la conduite des processus industriels. Il permet de créer des interfaces graphiques, des vues des procédés, et de contrôler aussi le processus où l'opérateur peut intervenir, modifier ou d'introduire de nouvelles valeurs.

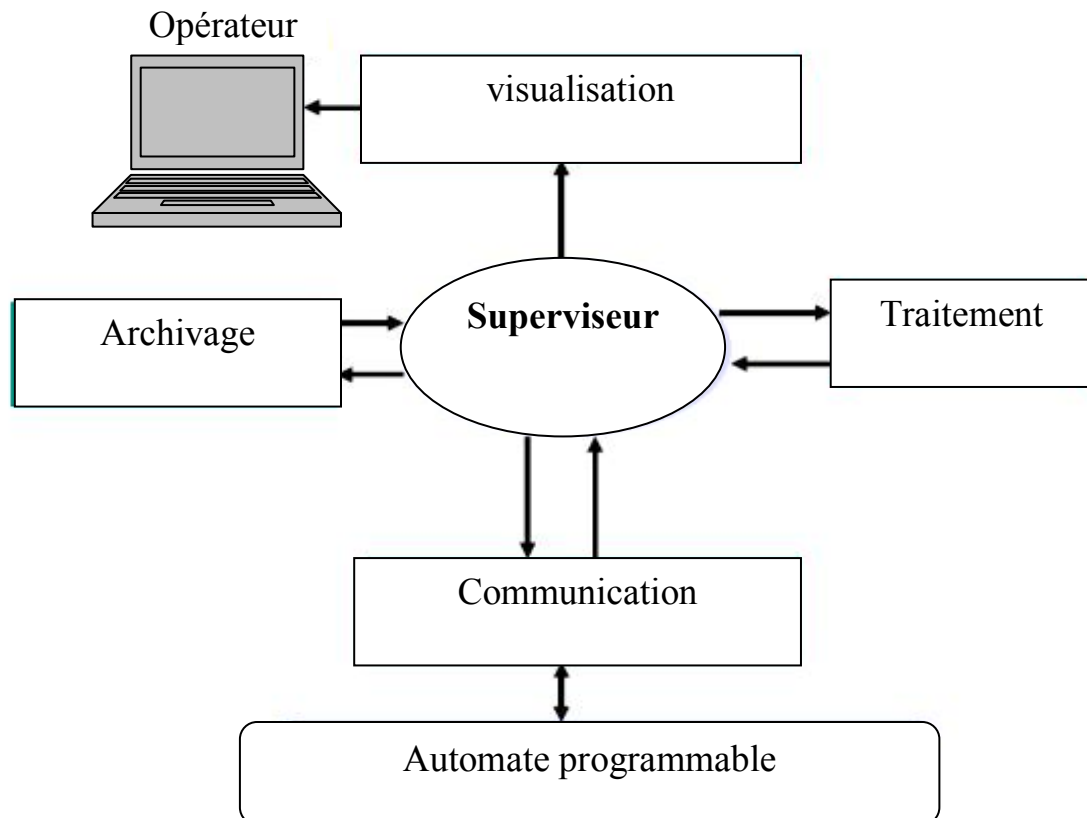


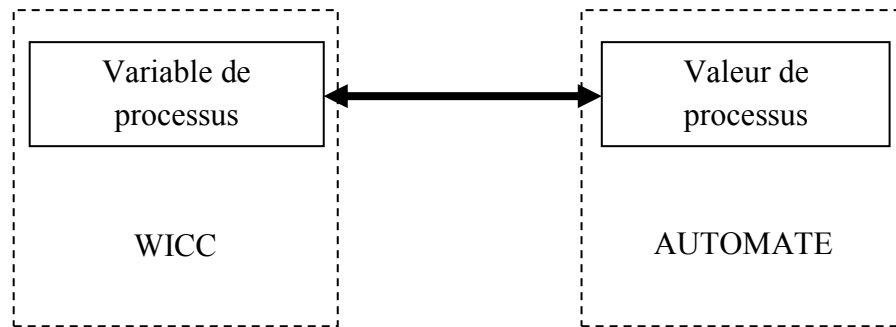
Figure 4.10: Modules fonctionnels d'un système de supervision

4.14.1. Les principales étapes suivies pour créer notre application sous WINCC :

- Créer un projet
- Sélectionner et installer l'API
- Définir les variables dans l'éditeur de variables
- Créer et éditer les vues dans l'éditeur graphics designer
- Paramétrer les propriétés de WINCC runtime
- Activer les vues dans le WINCC runtime
- Utiliser le simulateur pour tester les vues du processus

4.14.2. Communication entre WINCC et automate programmable industriel (API)

Dans le cadre des échanges de données, les variables de processus constituent le lien entre WINCC et les automates programmables. Chaque variable de processus de WINCC correspond à une valeur de processus déterminée dans la mémoire de l'un des automates programmables connectés. Au runtime, WINCC lit sur l'automate programmable la zone de données dans laquelle cette valeur de processus est enregistrée et détermine ainsi la valeur des variables de processus.



4.14.3. Caractéristiques techniques de WINCC

Le logiciel de supervision WinCC est constitué de plusieurs outils utilisés pour simplifier le développement des HMI qui sont:

- Manipulation du projet
- Bibliothèques d'objets de configuration prédéfinis ou confectionnés par l'utilisateur
- Prise en charge linguistique
- Runtime
- Test et assistance à la mise en service
- Communication ouverte entre systèmes HMI et systèmes de niveau supérieur.

4.14.4. WinCC flexible Runtime

L'application runtime de WinCC flexible, permet à l'opérateur de réaliser le contrôle-commande du processus. Les tâches suivantes sont alors exécutées:

- Communication avec les automates du processus.
- Affichage des vues à l'écran de supervision.
- Commande du processus, par exemple. Spécification de consignes ou ouverture et fermeture de vannes, démarrage et arrêt des moteurs et pompes, etc.
- Archivage des données de runtime actuelles, des valeurs de processus et événements d'alarme pour un diagnostic de défaillances.

4.14.5. Création de l'interface WinCC

Dans WinCC flexible, nous allons créer des vues pour le contrôle-commande de machines et d'installations. Pour créer des vues, nous disposons d'objets prédéfinis permettant de représenter notre installation, d'afficher des procédures et de définir des valeurs de processus.

4.14.6. Les Alarmes

Les alarmes montrent les événements ou les états de fonctionnement qui se produisent dans les installations industrielles. Les alarmes peuvent servir à diagnostiquer les erreurs.

On distingue deux types d'alarmes :

a) **Alarmes TOR** : elles indiquent les modifications d'état (vrai ou faux) et elles sont déclenchées par les E/S. une alarme indique si une vanne est ouverte ou fermée par exemple.

b) **Alarmes analogiques** : elles indiquent des dépassements de limites.

Une alarme analogique est déclenchée si le régime d'un moteur est inférieur à une valeur donnée par exemple.

4.15. Conclusion

Dans ce chapitre nous avons présenté les automates programmables industriels APIs et leurs techniques de programmation, on a présenté aussi le logiciel de supervision et de développement des HMI qui est le WinCC flexible.

L'implémentation d'une boucle de régulation classique (PID classique) sur un automate programmable industriel (API) consiste à configurer et paramétrer un bloc régulateur PID (FB 41). Le développement du correcteur flou dans notre cas passe par l'utilisation des différentes opérations possibles à effectuer (addition, soustraction, multiplication,...) et les différents blocs disponibles (transfert, temporisation, comparaison, ...) dans le langage Ladder qui sont les outils de base pour une implémentation d'un régulateur flou sur un Automate standard industriel (s7-300).

Les résultats obtenus sont récupérés grâce au logiciel WINCC flexible et seront présentés dans le chapitre suivant.

Chapitre 5

Test de validation et résultats expérimentaux

5.1. Introduction

Dans ce chapitre nous allons présenter la partie expérimentale de notre travail, où nous allons exploiter les connaissances acquises dans les chapitres précédents pour implémenter des régulateurs (PID, FLOU) dans le but de contrôler la chaîne de régulation de niveau liquide de l'unité opérationnelle PUL-2. Nous allons présenter aussi les interfaces utilisés pour les tests de commande et de supervision que nous avons développées avec WINCC flexible.

Notre objectif est de proposer une méthodologie d'implémentation pratique des lois de commande classique PID et avancé basée sur la logique floue toute en respectant autant que possible les contraintes imposées sur la réponse dynamique du système, de plus notre implémentation doit respecter les limites du langage Ladder des automates programmables industriels standard.

5.2. Synthèse du régulateur PID

Nous avons vu au chapitre (1) plusieurs méthodes de paramétrage du régulateur PID, dans la suite du travail nous allons utiliser la méthode du critère idéal pour la synthèse du correcteur.

5.2.1. Synthèse du correcteur de niveau par la méthode du critère idéal [5]

Le modèle du système obtenu après l'identification est :
$$G_p = \frac{1.91}{1+92p}$$

Nous voulons obtenir une régulation idéale, et cela en déterminant les paramètres du régulateur à l'aide de la formule :

$$G(p) = \frac{\lambda}{T_p} \cdot \frac{\theta p + 1}{G_s}$$

Posons $T = \theta$ et $\frac{\lambda}{G_s} = Gr$

On prend $\lambda = 19.1$

Donc $T_i = \theta = 92\text{sec}$ et $Gr = \frac{\lambda}{G_s} = 10$

Sachant que la fonction de transfert d'un régulateur PI est :

$$C(p) = Gr \cdot \frac{T_i p + 1}{T_i p} \quad \text{Alors} \quad C(p) = 10 \cdot \frac{92p + 1}{92p}$$

5.3. Régulation de niveau par automate SIMATIC S7-300

5.3.1. Programme et paramétrage du régulateur PID sous step7

La commande classique utilise le bloc de régulation PID continu (FB41) qui est pré programmé sur l'API S7-300, l'objectif est de calculer les paramètres adéquats des actions du régulateur à partir du système identifié, ensuite finaliser la configuration du bloc pré programmé (FB41).

Pour la configuration du régulateur PI, l'action dérivée doit être éliminée et cela en donnant la valeur 0 à Td.

La figure (5.1) montre le réglage minimal des paramètres du PI sur l'automate de gamme S7-300 avec CPU 314-2DP.

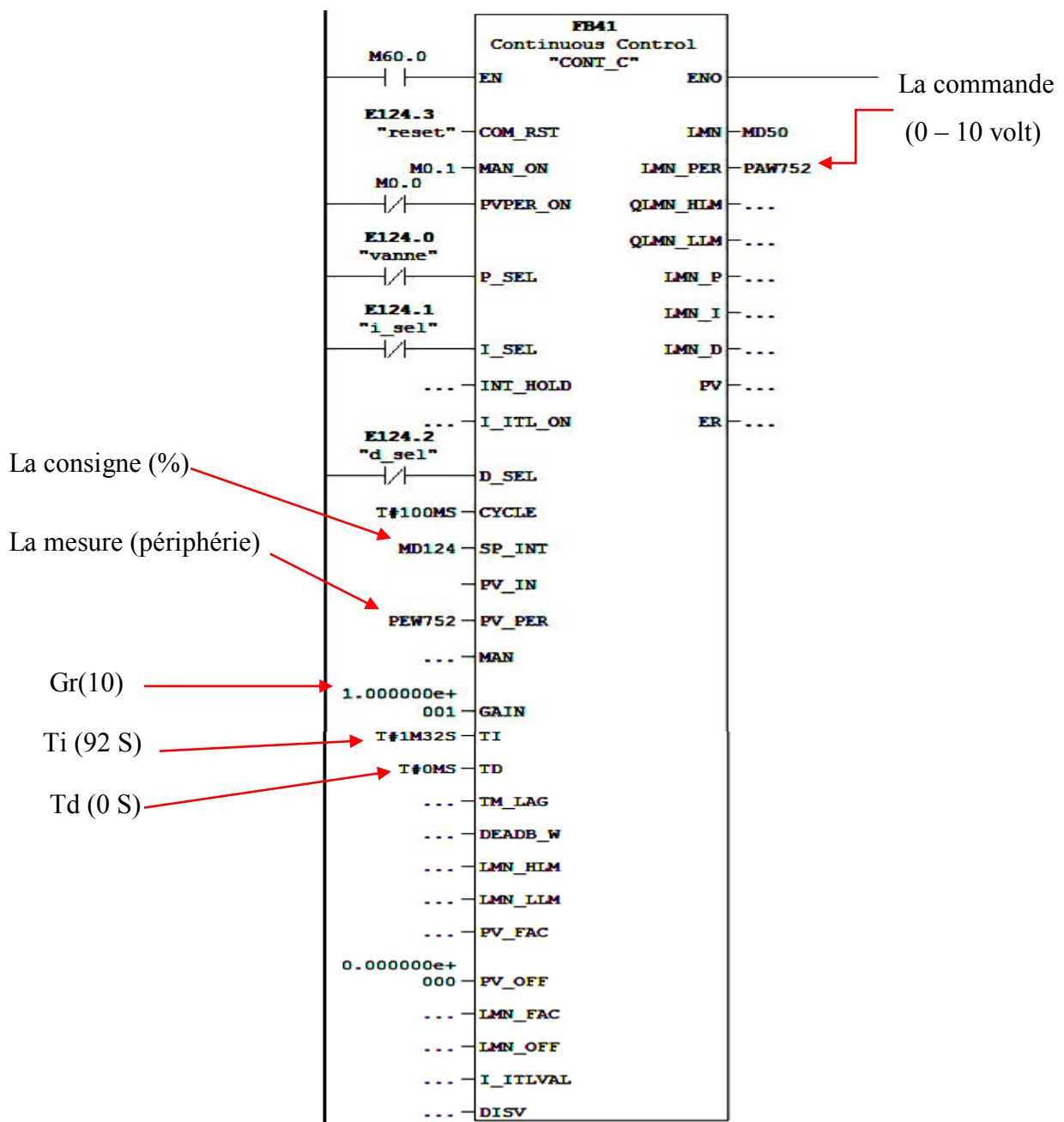


Figure 5.1 : Paramétrage du régulateur sous step7 (FB41)

5.3.2. Résultats expérimentaux de l'application du correcteur PI

Les résultats expérimentaux de l'application du correcteur PI configuré sous CPU 314 -2DP sont présentés en figure (5.2), pour l'acquisition de ces résultats nous avons développé une plat forme sous environnement WINCC flexible.

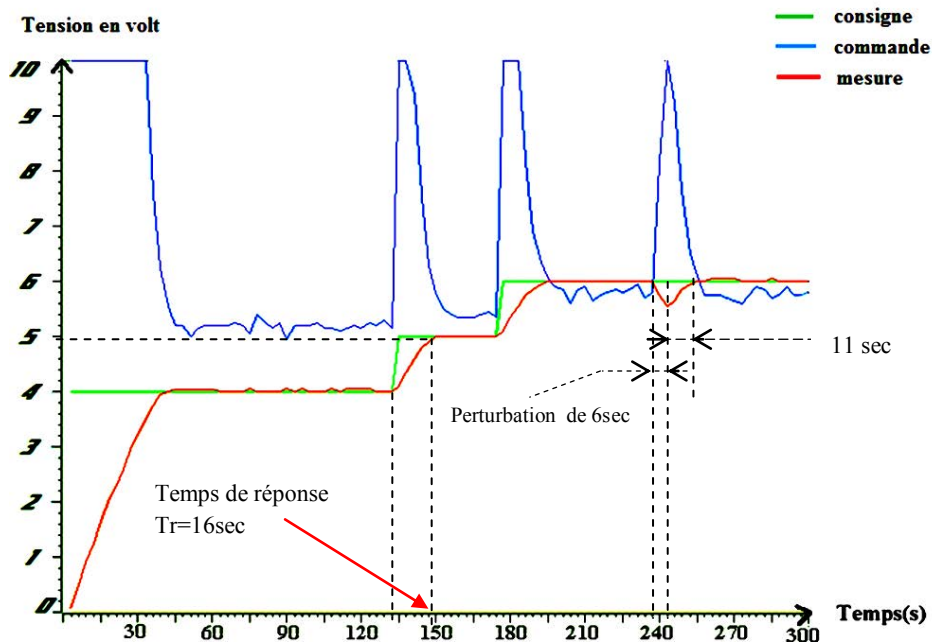


Figure 5.2 : Réponse du système régulé par automate

Au début du test la consigne est fixée à une valeur égale à 4volt après stabilisation du niveau à la valeur de consigne et à l'instant $t=132$ sec on applique un échelon de consigne de 1V pour passer à 5V, ce même test est repris à l'instant $t=174$ sec avec un échelon de 1Volt pour passer à 6Volt.

La réponse expérimentale montre que le régulateur PI choisi et paramétré, suit convenablement les échelons de consigne en mode poursuite avec erreur statique nulle et en temps de réponse de $t_r=16$ sec Pour les échelons de tests.

A l'instant $t=237$ sec un test de rejet de perturbation est effectué, et cela en appliquant une perturbation impulsionnelle d'une durée de 6sec. Le régulateur PI configuré permet un rejet de la perturbation après 11sec, avec erreur statique nulle. Le régulateur PI configuré nous permet des performances dynamiques très acceptables avec élimination des écarts statique.

5.3.3. Test en régulation en Cascade :

Dans la régulation cascade nous avons deux régulateurs, un régulateur maître configuré pour régler le niveau est un régulateur esclave pour la régulation du débit.

Nous allons garder les mêmes paramètres de régulateur de niveau, et avec la même méthode de synthèse nous calculons les paramètres de régulateur du débit.

Le modèle obtenu après l'identification est : $G_p = \frac{1.06}{1+2.7p}$

L'application de critère idéal donne les paramètres du régulateur PI esclave, avec

$T_i = \Theta = 2.7\text{sec}$ et $G_r = \frac{\lambda}{G_s} = 1.8$

5.3.4. Paramétrage et intégration du régulateur PI en cascade sous step7

La figure (5.3) montre les paramètres minimaux des deux régulateurs en cascade à programmer dans le bloc organisationnel OB1

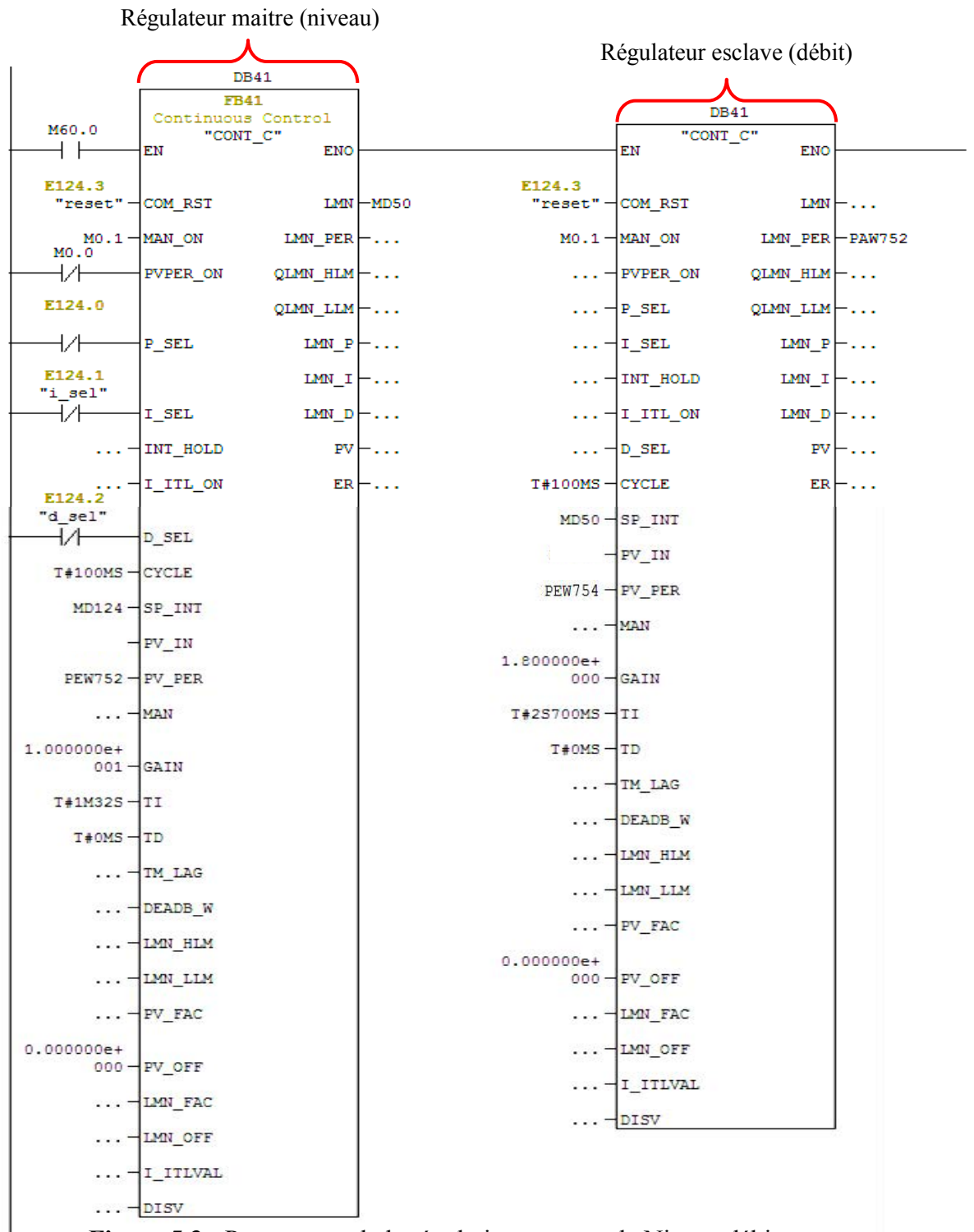


Figure 5.3 : Programme de la régulation en cascade Niveau débit

5.3.5. Résultats expérimentaux PI en cascade (Niveau- Débit)

Le relevé expérimental de la régulation en cascade est présenté en figure 5.4

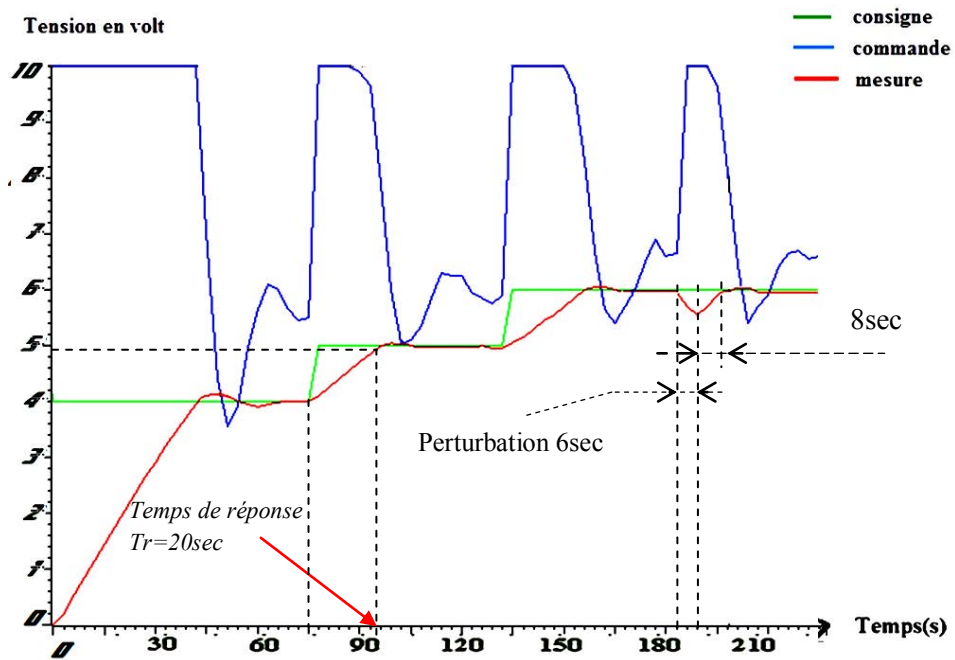


Figure 5.4 : Repense de système en régulation cascade

Nous avons réalisé un test de poursuite en changeant la consigne de 4volt à 5volt puis à 6volt. Le résultat expérimental montre que la mesure suit convenablement le changement de consigne, et on remarque aussi que la commande change en même temps que la consigne pour ramener la mesure à suivre la consigne.

Par la suite nous avons appliqué une perturbation de 6sec à l'instant $t=183sec$. Nous avons observé que le régulateur rejette cette perturbation 8 sec après son apparition.

La régulation cascade est utilisée pour permettre aux procédés qui ont une dynamique lente d'avoir une réponse rapide face aux perturbations extérieures ainsi qu'aux changements de consigne.

D'après les résultats expérimentaux obtenus en régulation en cascade, nous avons constaté l'existence d'un petit dépassement lors du test de poursuite, ce dépassement étant le résultat des contraintes imposées sur la rapidité de la réponse dynamique.

5.4. Programmation du régulateur flou

5.4.1. Entrées / Sorties du régulateur flou

La première étape pour la programmation de régulateur flou est de choisir ses variables d'entrées et de sorties, déterminer avec quels paramètres on peut obtenir la loi commande, et enfin de définir les univers de discours de ses variables. Les entrées du correcteur flou que nous allons développer seront constituées de deux entrées qui sont l'erreur (**e**) et la dérivée de cette erreur (**de**).

- **Programmation de l'entrée d'erreur** : si la sortie est inférieur à la consigne, l'erreur prendra des valeurs positives et dans le cas contraire des valeurs négative. On définit le degré du signe (négative grand **NG**, négative petit **NP**, environ zéro **EZ**, positive petit **PP**, positive grand **PG**) et on élabore cinq fonctions d'appartenance selon les intervalles suivants : [-3 ; -1.3 ; 0 ; 1.3 ; 3]

Les fonctions d'appartenance que nous avons choisie, sont de forme triangulaire.

- **Programmation de la dérivée de l'erreur** : si la variation de niveau évolue dans le sens croissant alors on dit que la dérivée de l'erreur est négative, par contre dans le cas où la variation est dans le sens décroissant, on dit que la dérivée de l'erreur est positive.
- **Programmation de la sortie**: c'est des singletons issus du passage par le bloc de défuzzification.

5.4.2. Fonctions d'appartenance

La figure (5.5) représente les fonctions d'appartenance de l'erreur (e) obtenue du programme de la figure (5.8) et qui est récupérée en utilisant le WINCC flexible.

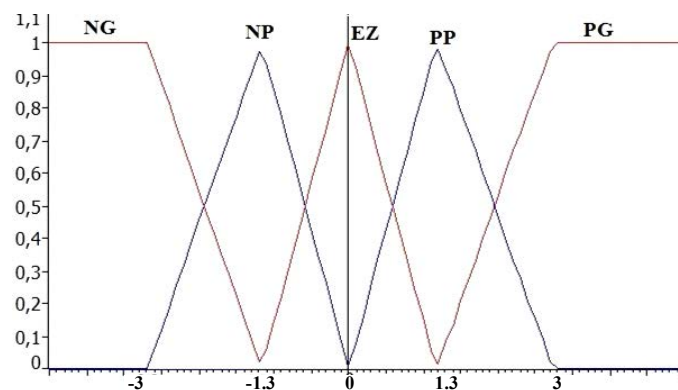


Figure 5.5: Fuzzification de l'erreur

La figure (5.6) montre la fonction d'appartenance de la dérivée de l'erreur (**de**)

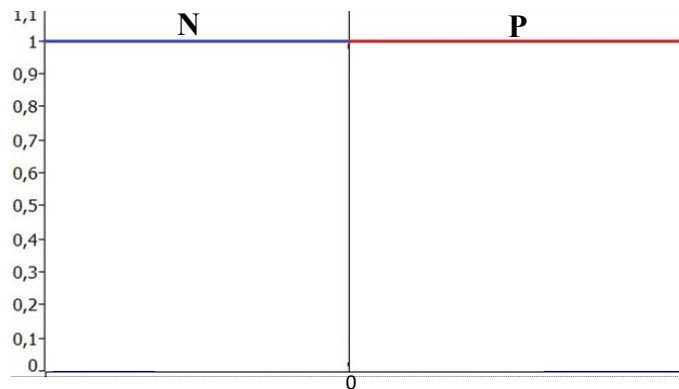


Figure 5.6 : Fonction d'appartenance de la dérivée de l'erreur

La sortie a été partitionnée en 5 valeurs linguistiques; Elle est représentée sur la figure (5.7)

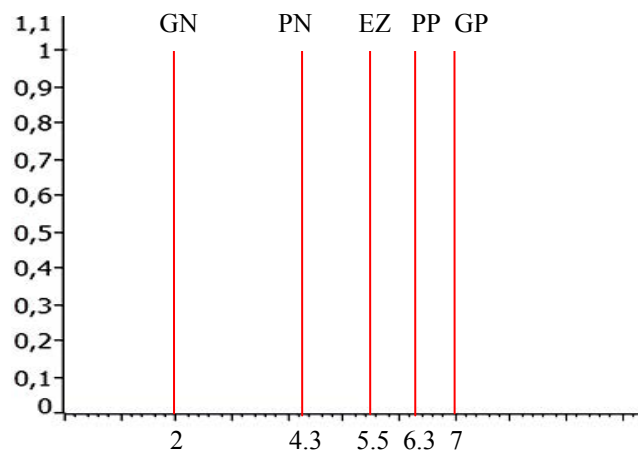


Figure 5.7: Fuzzification de la sortie

Le jeu de règles du régulateur flou à cinq ensembles est représenté dans le tableau (5.1)

de \ e	PG	PP	EZ	NG	NP
P	PG	PP	PP	NP	EZ
N	PP	EZ	NP	NG	NG

Tableau 5.1: Table de règles

5.4.3. L'implémentation du correcteur flou sous automate S7-300

L'implémentation de cette table de règles floues données en figure (5.1) sur l'automate sous langage Ladder, s'effectue grâce à l'utilisation des différents blocs et outils de base de la bibliothèque du logiciel step7 (soustraction, addition,...).

5.4.3.1. Programmation des fonctions d'appartenance

La programmation des fonctions d'appartenance de l'erreur, passe par l'utilisation de deux comparateurs, composant entre l'erreur stockée dans le double mot interne (**MD15**) et les différentes limites des intervalles, si la comparaison est vraie alors le programme de la figure (5.8) calcule l'appartenance de l'erreur qui sera chargée dans les doubles mots **MD90** et **MD165** en utilisant les blocs de soustraction et de division.

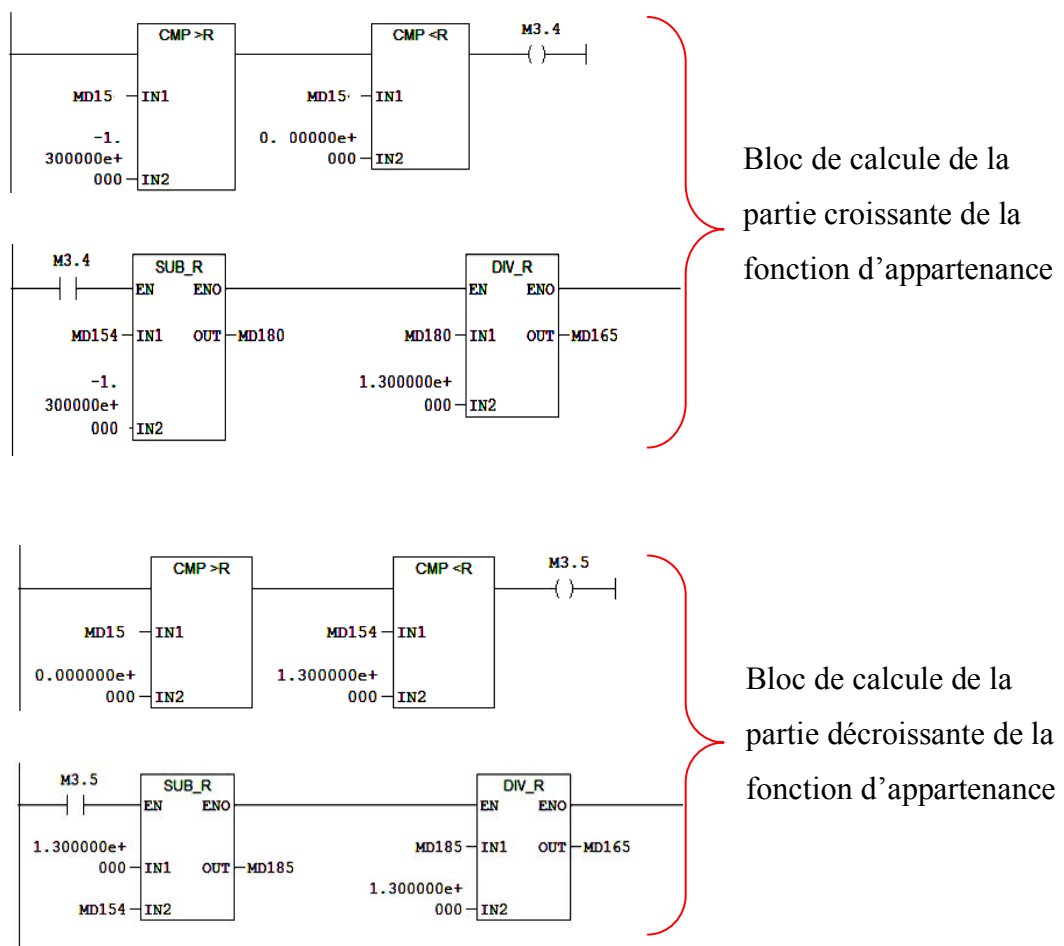
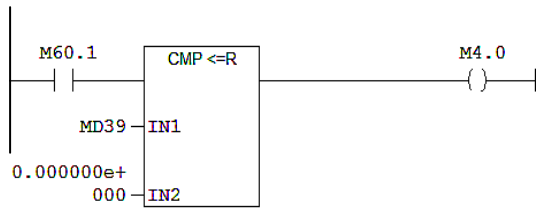


Figure 5.8: programmation de la fonction d'appartenance

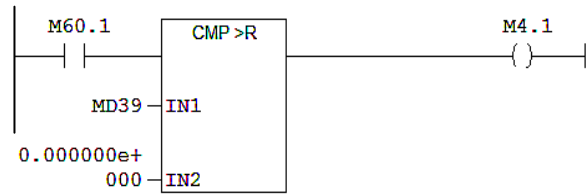
L'implémentation des fonctions d'appartenance de la dérivée de l'erreur (**de**) passe par l'utilisation d'un soustracteur de valeur entre l'erreur **e(t)** et l'erreur **e(t-T)**. Le résultat de la soustraction est comparé à zéro pour **T= 1** seconde ce qui donne :

Si la **de** est **positive** alors le memento interne affecté à ce test **M4.1** est mis à **1**

Si la **de** est **négative** alors le memento interne affecté à cette opération **M4.0** est mis à **1**



Dérivée de l'erreur négative



Dérivée de l'erreur positive

Pour mémoriser la valeur de l'erreur $e(t-T)$, nous avons développé un petit programme où on a utilisé un compteur, un bit de cadence (**M100.1**), des comparateurs et un bloc de transferts. Les valeurs de l'erreur $e(t-T)$ seront récupérées sur le memento **MD44**.

5.4.3.2. Programmation de la commande floue par la méthode du centre de gravité

Les signaux commandes seront calculés et transférés vers une sortie physique de l'automate de la façon suivante :

Si **M1.2** est activé alors charger **MD165** dans **MD221** et **MD90** dans **MD225**, puis une procédure de calcul de la sortie en utilisant le bloc de multiplication et le bloc d'addition est exécutée.

La valeur numérique de la sortie (**MD237**) correspondante au centre de gravité des singletons, sera transformée en nombre entier (0-27648) et charger dans la périphérique de sortie d'adresse **PAW752**.

Une partie du programme calculateur des signaux de sortie est présentée en figure (5.9).

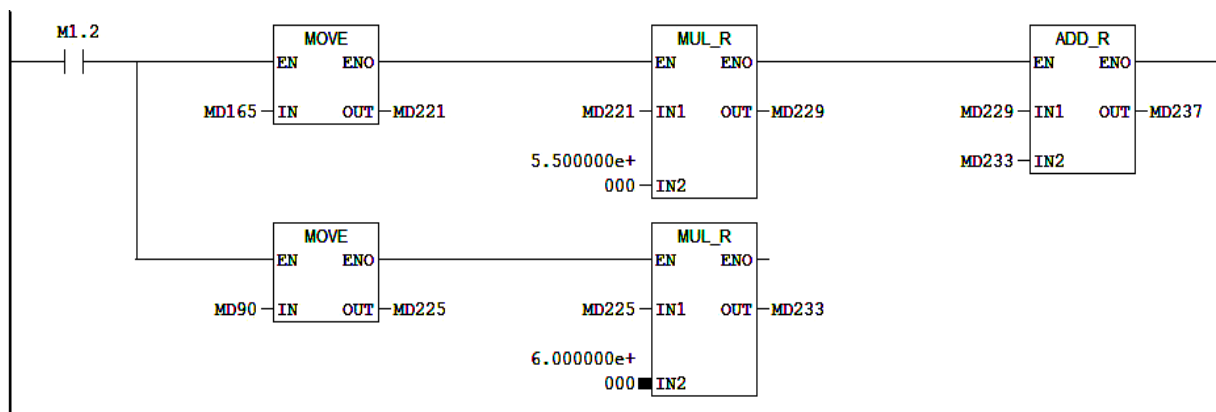


Figure 5.9: Exemple de programme calculateur de signal de commande

La figure (5.10) montre la procédure de conversion et de transfert de la valeur de commande vers l'interface de sortie d'adresse PAW752

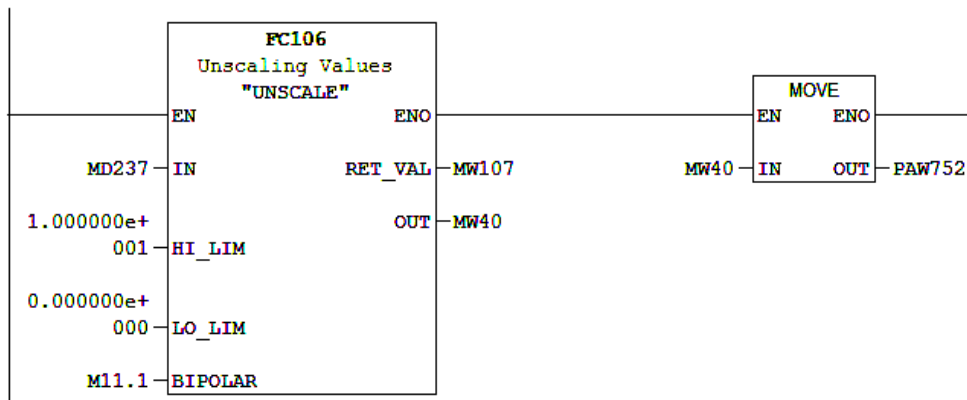


Figure 5.10: Exemple de programme de conversion et de chargement

5.4.4. Résultats expérimentaux de régulateur fluo

La figure (5.11) représente le relevé expérimental de l'application du contrôleur fluo que nous avons développé

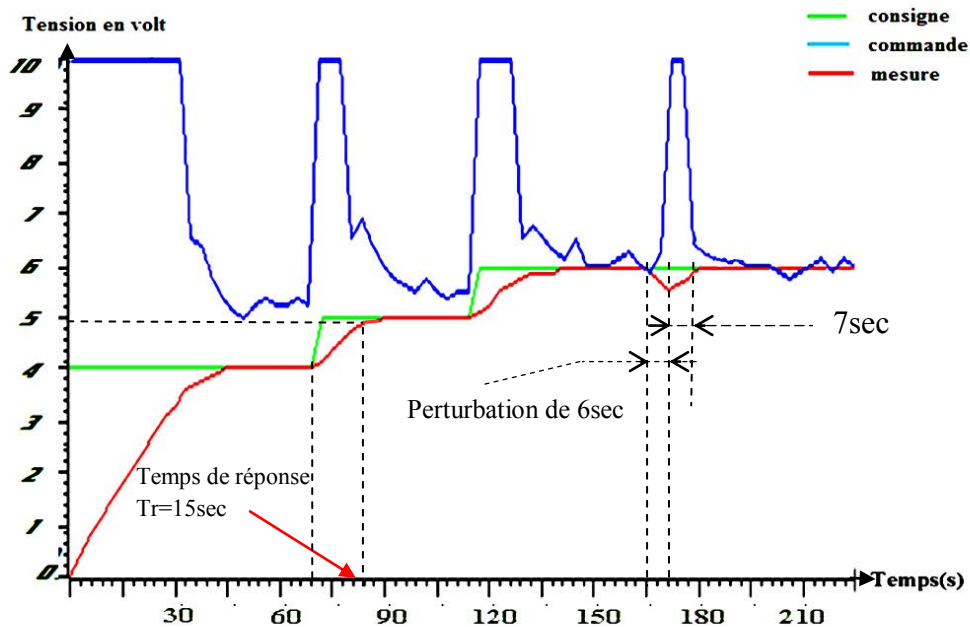


Figure 5.11 : Réponse expérimentale contrôleur fluo

Le relevé expérimental obtenue après l'implémentation de la commande fluo basée sur la table des règles du tableau (5.1) dans l'automate, montre que le contrôleur fluo implémenté a permis de répondre à l'objectif visé. Les résultats obtenus sont satisfaisant puisque l'écart statique est nul et les performances dynamiques sont satisfaisantes.

Le résultat expérimental de l'application du régulateur fluo que nous avons proposé présente

Des performances dynamiques acceptables sans dépassement de valeur en mode poursuite et permet le rejet de perturbation 7sec après sont apparition en mode régulation.

Les résultats donnés par le régulateur flou à cinq ensembles sont nettement meilleurs que ceux donnés par les autres stratégies de régulation à savoir la régulation classique PID et celle en mode PID cascade (Niveau/débit).

Le régulateur que nous avons développé a montré une bonne fiabilité lors des tests de poursuites où le signal de commande suit parfaitement la trajectoire de la consigne.

5.5. Interface de commande PID et FLOU de la station PUL-2/EV

la figure (5.10) représente la vue de commande et de supervision que nous avons développée sous environnement WINCC flexible pour notre système de régulation de niveau PUL-2/EV.

La plateforme de commande et de supervision développée permet, la sélection des deux stratégies de commande (classique et floue), ainsi que la récupération des différentes informations nécessaires à l'exploitation de cette application.

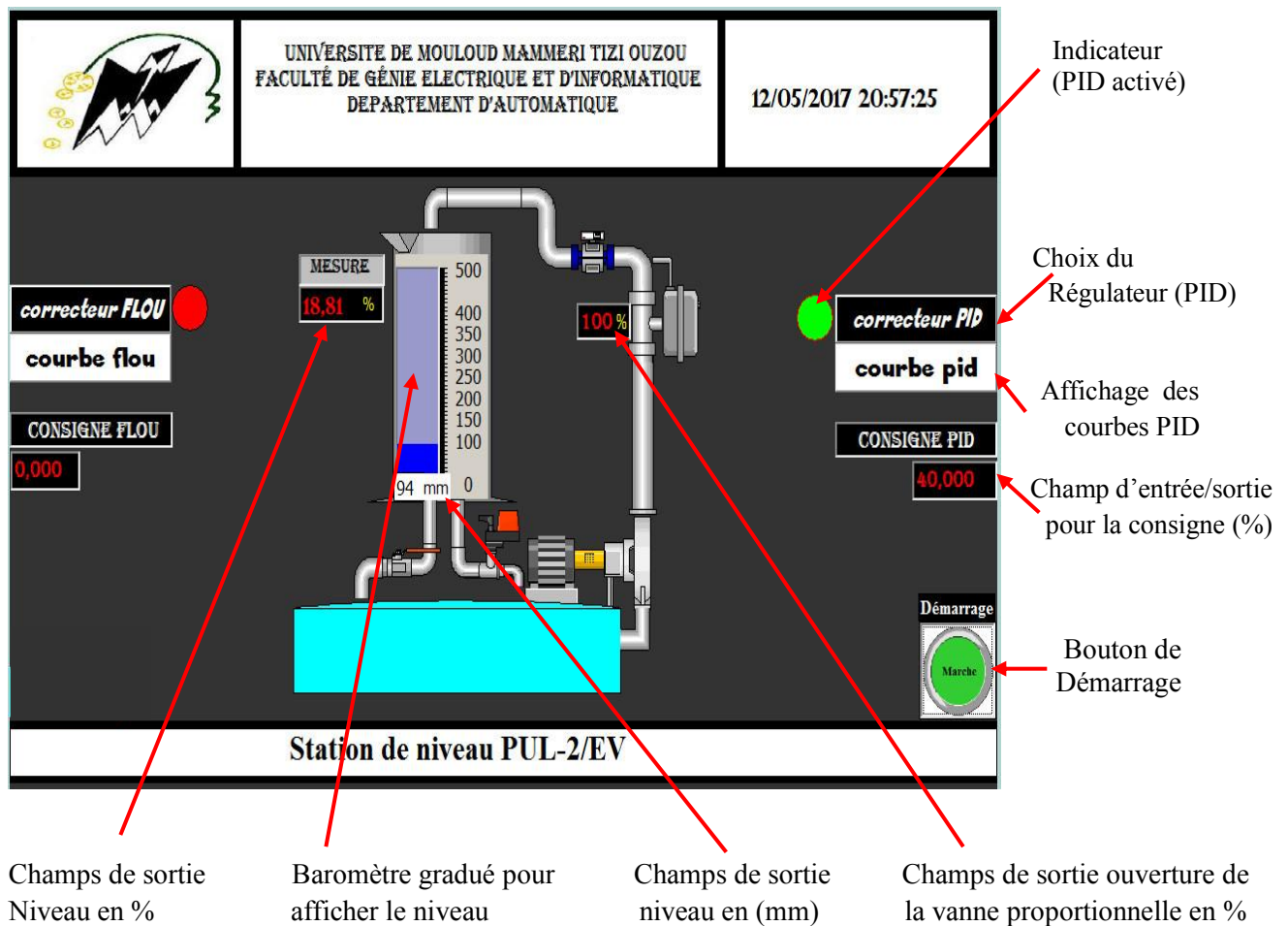


Figure 5.12: Vue d'accueil du système de contrôle de processus PUL-2

5.5.1. Constitution de la vue

Pour les impératif de simplification nous avons inséré dans notre vue les objets dont on a besoin pour représenter le processus de régulation de niveau.

Ainsi nous avons configuré les objets en fonction des exigences de notre système qui sont :

- Champs E/S, boutons, barre gradué, champs de texte, affichage des courbes et objets graphique pour représenter la station PUL-2.
- Les valeurs de processus sont échangées entre l'automate programmable et le logiciel de supervision WinCC flexible au moyen de variables réels.

5.6. Interface d’affichage des courbes de régulateur PID et FLOU

Cette interface est constituée de deux zones pour l’affichage des réponses expérimentales de la mesure et de la consigne à gauche, et de la variable de commande à droite; de plus cette interface est équipée de champs E/S pour l’affichage instantané des valeurs (mesure, erreur...) Ainsi que de boutons poussoir virtuel pour interagir avec le système.

La figure (5.13) présente la vue d’interfaçage du régulateur PID associe au bloc FB41 que nous avons développé

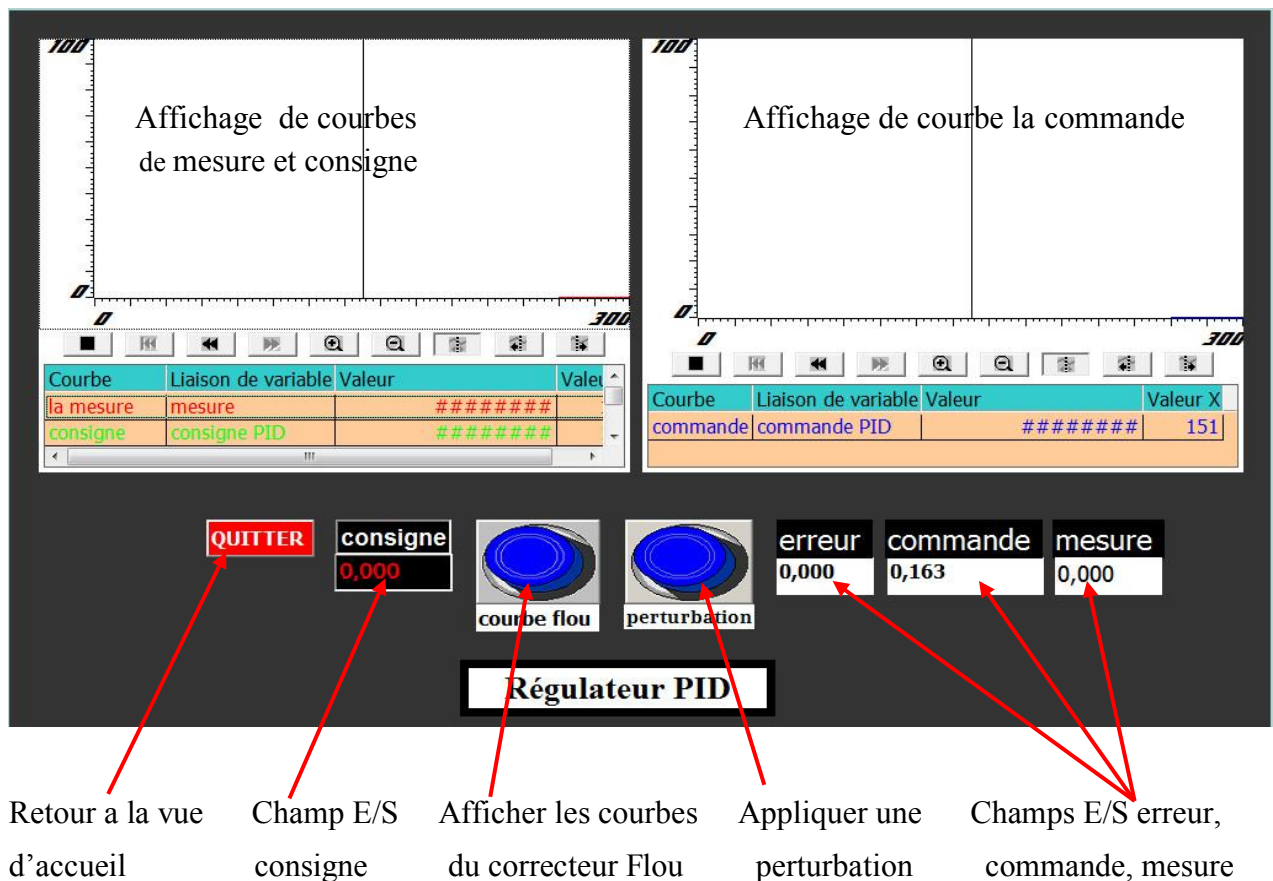


Figure 5.13: Interface de régulation PID avec le bloc FB41

L'interface d'affichage et de dialogue pour la régulation floue est présentée en figure (5.14), elle respecte la même structure et interfaçage que celui présenté en figure (5.13)

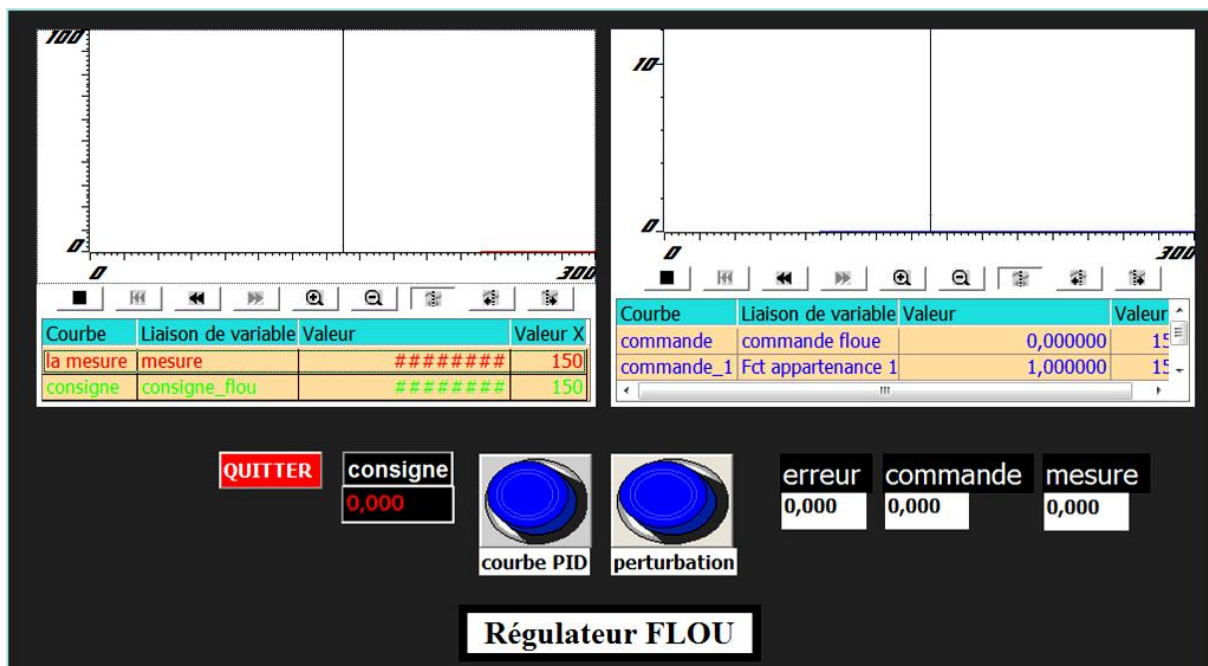


Figure 5. 14: Interface de régulation avec le régulateur flou

5.7. Conclusion

Ce chapitre était consacré à la présentation des différents techniques d'implémentation de régulateurs (PI, FLOU et PI en cascade), et aux résultats expérimentaux obtenus de l'application de ces techniques sur l'automate S7 300, nous avons présenté une interprétation des concepts de la logique floue en langage programmable (LADDER) en utilisant les outils de programmation fourni dans la bibliothèque du logiciel de programmation des APIs (S7-300).

D'après les résultats expérimentaux, on peut déduire que la Commande floue donne des résultats meilleurs, surtout lorsqu'on effectue des perturbations sur le système. Ce qui n'est pas le cas pour la commande classique PI qui perd ses performances dynamiques (stabilité, précision...). Le régulateur flou que nous avons implémenté sous S7-300 étant plus recommandé pour notre application de régulation de niveau vu sa stabilité et ses performances dynamiques aussi bien en mode poursuite qu'en mode rejet de perturbation.

Conclusion Générale

La commande floue est une méthode nouvelle de commande des procédés industriels. Elle utilise des techniques s'appuyant sur le savoir-faire humain, exprimées en mots et en phrases ordinaires, plutôt que sur des équations.

Le but de notre projet était d'élaborer une technique d'implémentation et d'application de la logique floue dans un contexte d'automate programmable S7-300. Afin d'atteindre cet objectif, nous avons choisi le système de régulation de niveau PUL-2 EV. Ce choix est justifié par le fait que ce matériel existe au sein du laboratoire d'automatique industrielle.

Au début nous avons déterminé expérimentalement les fonctions de transfert du système (niveau, débit), pour calculer les paramètres du régulateur PI. Ensuite, nous avons réalisé une commande avec ce régulateur classique ainsi qu'une régulation en cascade. Après les tests pratiques sur la station PUL2/EV, nous avons constaté que la régulation classique PID que nous avons choisie ne nous offre pas des performances dynamiques suffisantes.

Pour palier à ce problème, nous nous sommes intéressés à la commande par la logique floue, afin de profiter de ses avantages et d'étudier ses possibilités d'implémentation pratique.

L'implémentation du régulateur flou sur un API S7 300, avec CPU 314-2DP, nous a permis de développer une procédure de programmation structurée. L'application pratique de ce programme développé sur la station de niveau PUL2/EV, nous a permis d'améliorer les performances dynamiques de la régulation, notamment les performances par rapport au mode rejet de la perturbation. Il est à noter que la procédure de programmation que nous avons choisie lors de l'implémentation du correcteur flou, peut être adaptée à tous types d'automates se programmant en langage Ladder.

Ce travail nous a permis d'acquérir une expérience enrichissante dans le domaine de la régulation et de la programmation des APIs, ainsi que la manipulation des logiciels (Step7, WinCC) de SIEMENS. De plus, ce travail nous a permis également de mettre en pratique les connaissances acquises durant notre cursus.

Enfin, nous espérons que nos efforts puissent servir comme support technique et que ce mémoire soit un bon guide pour les promotions futures, et que notre travail contribuera pour enrichir notre équipement et la qualité des travaux pratiques au sein du laboratoire d'automates programmables et de la régulation industrielle.

Références Bibliographiques

- [1] **DR.M.Rabi**, « régulation analogique industriel », ESTT.G.thermique.
- [2] **Eric Magarotto**, « cours de régulation », support cours, 2004, université de Caen.
- [3] **A. BESANÇON-VODA, S.GENTIL**, « régulateur PID analogique et numérique », technique de l'ingénieur, [R 7416], 1999.
- [4] **P.Prouvost**, «*Instrumentation et régulation en 30 fiche*».Dunod, Paris, 2010.
- [5] **Daniel Dindeleux**, « technique de régulation industriel », Paris, 1989
- [6] **Hansruedi Buhler**, « Réglage par logique floue », CH-1015 Lausanne, 1994
- [7] **Elettronica Veneta**, « Manuel professeur/étudiant », Copyright INEL SPA; PP189
- [8] **C. ROSENTAL** ; «Histoire de la logique floue. Une approche sociologique des pratiques de démonstration», octobre-décembre 1998.
- [9] **Alain Gonzaga, A**, « les automates programmable industriels cours »2004
- [10] **M.LESCIEUX** ; « introduction à la logique floue, application à la commande floue »
- [11] **F.CHEVRIE et F. Guély** ; « cahier technique n 191 la logique floue » édition mars 1998
- [12] SIMATIC Logiciel de base pour S7-300/400 Régulation PID, manuel Siemens AG, 1996.
- [13] SIMATIC PC Adapter USB, manuel Siemens AG 2007, p.6-1.
- [14] **MAIDI, A.**, « cours d'automatismes industriels » ; UMMTO, 2005,2006
- [15] WinCC flexible 2008/Compact / Standard / Advanced ; Manuel d'utilisation référencé 6AV6691-1AB01-3AC0; 2008

Support internet

- [16] www.cira83.com; « boucle de régulation ».

Logiciel utilisés

SIMATIC Manager Step7

SIMATIC WinCC flexible 2008