

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 ACADEMIQUE**

Filière : **Automatique et Informatique Industrielle**

Présenté par

SLIMANI Sofiane

HACHOUR Ahcen

Mémoire dirigé par M. CHARIF Moussa

Thème

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

Mémoire soutenu publiquement le 02 Juillet 2018 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

Remerciements

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.

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
D	précision dynamique	%
Kp	Gain proportionnel	
Ti	Temps d'intégration	seconde
Td	Temps dérivatif	seconde
p	Opérateur de laplace	
Z(t)	Perturbation	
P	Régulateur proportionnel	
PI	Régulateur proportionnel Intégral	
T	Retard	seconde
G(p)	fonction de transfert	
Krc	gain critique	
Tosc	période d'oscillation	seconde
Tr	Temps de réponse	Seconde
PID	Régulateur proportionnel Intégral Dérivé	
μ_A	Fonction d'appartenance	
Q	Débit d'écoulement	m ³ /s
$\Delta u(t)$	Variation de la commande à l'instant t	
$e(t)$	Erreur à l'instant t	

LISTE DES FIGURES

Chapitre 1

Figure 1.1 : Schéma de principe d'une régulation.....	4
Figure 1.2 : Schéma d'une régulation en boucle ouverte.....	5
Figure 1.3 : Schéma d'une régulation en boucle ouverte.....	6
Figure 1.4 : Schéma d'une régulation en cascade.....	8
Figure 1.5 : Réponse indicielle d'un processus.....	11
Figure 1.6 : Système asservi par un correcteur.....	13
Figure 1.7 : Schéma d'un procédé mono variable.....	14
Figure 1.8 : Schéma d'un procédé multi variable.....	14

Chapitre 2

Figure 2.1 : La station de niveau PUL-2.....	17
Figure 2.2 : Schéma Pi&D de la station.....	17
Figure 2.3 : Bloc conditionneur et amplificateur de puissance.....	19
Figure 2.4 : Vérification d'état du capteur-transmetteur de niveau.....	19
Figure 2.5 : Structure interne du débitmètre à aubes.....	20
Figure 2.6 : Hystérésis de la vanne proportionnelle.....	21
Figure 2.7 : Caractéristique statique du processus.....	22
Figure 2.8 : Réponse indicielle du système de niveau.....	22
Figure 2.9 : Réponse indicielle du système de débit.....	23
Figure 2.10 : Couplage des deux stations de régulation de niveau PUL-2.....	24
Figure 2.11 : Système à deux réservoirs couplés.....	25

Chapitre 3

Figure 3.1 : Structure générale d'une commande floue.....	30
Figure 3.2 : Représentation des ensembles.....	30
Figure 3.3 : Fonction caractéristique et fonction d'appartenance.....	31
Figure 3.4 : Exemples de fonctions d'appartenances triangulaire et trapézoïdal	31
Figure 3.5 : Exemple d'inférence Max-produit.....	34
Figure 3.6 : Exemple d'inférence Max-min.....	35
Figure 3.7 : Exemple d'inférence Somme-Produit.....	36

Figure 3.8 : Défuzzification par la méthode moyenne des maximums.....	37
Figure 3.9 : Exemple de défuzzification par centre de gravité.....	38
Figure 3.10 : Structure régulateur flou de type Mamdani.....	39

Chapitre 4

Figure 4.1 : Schéma bloc d'un régulateur TITO PI/PD flou typique.....	42
Figure 4.2 : Fonction d'appartenance des entrés.....	43
Figure 4.3 : Fonction d'appartenance des sorties.....	44
Figure 4.4 : Fonctions d'appartenance résultantes du produit de Larsen.....	47
Figure 4.5 : Partition de l'espace des entrés.....	47

Chapitre 5

Figure 5.1 : Exemples d'automate compacte et modulaire (successivement).....	57
Figure 5.2 : Configuration matériel sous Step7.....	57
Figure 5.3 : Schéma Bloc du régulateur flou sous Simulink.....	62
Figure 5.4 : Activation du moment de cadences de la CPU.....	63
Figure 5.5 : Transfert du niveau d'une seule station sous Simulink.....	67
Figure 5.6 : Shéma de la régulation en mode SISO sous Simulink.....	68
Figure 5.7 : Réponse du système simulé sous Simulink en mode SISO.....	68
Figure 5.8 : Réponse du système régulé par automate en mode SISO.....	69
Figure 5.9 : Shéma de la régulation en mode SITO sous Simulink.....	70
Figure 5.10 : Réponse du système simulé sous Simulink en mode SITO.....	70
Figure 5.11 : Réponse du système régulé par automate en mode SITO.....	71
Figure 5.12 : Réponse du système régulé par le correcteur PI sur automate.....	72
Figure 5.13 : Shéma de la régulation en mode TITO sous Simulink.....	73
Figure 5.14 : Réponse du système régulé sous Simulink en mode TITO.....	74
Figure 5.15 : Réponse du système à deux réservoir couplés régulé par le régulateur TITO...75	75
Figure 5.16 : pupitre de contrôle du système sous WINCC flexible.....	77

LISTE DES TABLEAUX

Chapitre 1

Tableau 1.1 : Paramètres PID obtenus à partir d'une réponse indicielle (ZNt).....12

Tableau 1.2 : Paramètres PID obtenus à partir du point critique (ZNf).....12

Chapitre 3

Tableau 3.1 : Opérateurs entre ensembles flous.....32

Chapitre 4

Tableau 4.1 : La base de règles correspondante au 1^{er} cas.....45

Tableau 4.2 : La base de règles correspondante au 2^{ème} cas.....46

Tableau 4.3 : Valeurs de $e_{1s}(k)$, $\Delta e_{1s}(k)$, $e_{2s}(k)$ et $\Delta e_{2s}(k)$ selon les cellules.....54

Chapitre 5

Tableau 5.1 : Les différents type de donné prédéfinis dans S7-SCL.....59

Tableau 5.2 : Les différentes sections de déclaration de variable dans S7-SCL.....61

Tableau 5.3 : Performances expérimentales des différentes stratégies de commande.....73

SOMMAIRE

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

Chapitre 1 : Présentation des notions de régulation de systèmes

1.1. Introduction	3
1.2. Définition de la régulation.....	3
1.3. Intérêt de la régulation automatique	3
1.4. Principe général de la régulation	3
1.4.1. Comportement en mode régulation	5
1.4.2. Comportement en mode asservissement	5
1.5. Régulation en boucle ouverte	5
1.6. Régulation en boucle fermée	6
1.7. Qualités attendues d'une régulation	7
1.7.1. La stabilité	7
1.7.2. La précision	7
1.7.3. La rapidité	8
1.8. Régulation en cascade	8
1.8.1. Cascade sur une grandeur intermédiaire	9
1.8.2. Cascade sur la grandeur réglante	9
1.9. Régulateurs PID	9
1.10.1 Les différentes actions du régulateur PID.....	10
1.10. Synthèse des paramètres des actions PID.....	11
1.10.1 Méthodes de Ziegler et Nichols (ZN)	11
1.10.2 Méthode de la réponse indicielle	12
1.10.3 Méthode du point critique	12
1.10.4 Méthode de placement de pôles	13
1.11. Les types de procédés	14
1.11.1. Procédé mono variable.....	14
1.11.2. Procédé multi variable	15
1.12. Types de système selon le procédé.....	15
1.12.1. Système SISO (single IN single OUT)	15
1.12.2. Système TISO (two IN single OUT)	15
1.12.3. Système TITO (two IN two OUT).....	15
1.13. Conclusion	15

Chapitre 2 : Présentation de la station de régulation de niveau

2.1. Introduction	16
2.2. Description de l'unité.....	16
2.3. Le capteur de niveau	18
2.4. Conditionnement du signal du capteur	18
2.5. Amplificateur de puissance de la vanne proportionnelle	18

2.6. Transducteur de niveau de la station PUL-2	19
2.6.1. Caractérisation du capteur de niveau	19
2.7. Débitmètre à aubes.....	20
2.7. Vanne proportionnelle	20
2.8. Caractérisation de la vanne proportionnelle	21
2.9. Identification du system	21
2.9.1. Caractéristique statique du système.....	21
2.9.2. Réponse indicielle du système de niveau.....	22
2.9.3. Réponse indicielle du système de débit :	23
2.10. Couplage des deux stations de régulation de niveau PUL-2 :	24
2.10.1. Modèle d'état du système en couplage des deux stations :	25
2.11. Conclusion.....	28

Chapitre 3 : Présentation des éléments de la logique floue

3.1. Introduction	29
3.2. Concept de la logique floue.....	29
3.3. Structure générale d'une commande par logique floue	30
3.3.1. Notions de bases de la logique Flou	30
3.3.2. Opérateurs entre ensembles flous	32
3.3.4. Fuzzification.....	33
3.3.5. Moteur d'inférence floue	33
3.3.6. Base de connaissances	36
3.3.7. Défuzzification	36
3.4. Contrôle par la logique floue.....	38
3.4.1. Différents types de régulateurs flous	39
3.5. Conclusion.....	40

Chapitre 4 : Modélisation du contrôleur flou PI/PD simplifié pour système TITO

4.1. Introduction	41
4.2. Les composants de base du régulateur TITO PI ou PD flou	41
4.2.1. Fuzzification.....	43
4.2.2. La base de règles	44
4.2.3 .Moteur d'inférence	46
4.2.4 Deffuzzification.....	48
4.3. Modèle mathématique du régulateur TITO PI/PD Flou Simplifié	49
4.4. Conclusion.....	55

Chapitre 5 : Teste de validation et résultats expérimentaux

5.1. Introduction	56
5.2. Présentation matériel et implémentation.....	56
5.2.1. Configuration matériel.....	57

5.2.2. Les langages de programmation.....	58
5.2.3. Présentation du langage S7-SCL.....	58
5.3. Implémentation du modèle mathématique du régulateur TITO PI flou simplifié.....	62
5.3.1. Implémentation du bloc régulateur sous Matlab/Simulink.....	62
5.3.2. Implémentation du régulateur sous Step7.....	62
5.4. Test de validation des résultats du modèle mathématique du régulateur TITO flou.....	67
5.4.1. Simulation du régulateur en mode SISO sous Matlab/Simulink :.....	68
5.4.2. Résultats expérimentaux du régulateur SISO.....	69
5.4.3. Simulation du régulateur SITO sous Simulink.....	69
5.4.4. Résultats expérimentaux du régulateur SITO.....	71
5.4.5. Etude comparative avec le régulateur classique PI.....	71
5.4.6. Résultats de simulation du régulateur en mode TITO.....	73
5.4.7. Résultats expérimentaux du régulateur en mode TITO.....	75
5.5. Interface de commande et de supervision de la station PUL-2/EV.....	75
5.5.1. Les courbes.....	75
5.5.2. L'interface d'enregistrement.....	76
5.5.3. La partie commande et mesure sur le système.....	76
5.6. Conclusion.....	78
Conclusion générale.....	79

Introduction générale

De nos jours la régulation occupe une place primordiale dans le domaine de l'automatique, elle consiste à agir de façon à ce qu'une mesure soit égale à une consigne. Si l'on cherche à atteindre une consigne, on parlera de poursuite ou d'asservissement ; si l'on cherche à éliminer des perturbations pour qu'une valeur reste constante, on parlera de régulation. L'industrie utilise à foison des systèmes d'asservissement et régulation : que ce soit pour gérer le débit d'un fluide dans une conduite, la température d'un produit ou le niveau dans une cuve etc. Ainsi 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.

Un système d'inférence flou qui est l'élément de base de la commande par logique floue possède deux points forts ; le premier est qu'il est généralement construit à partir de la connaissance humaine, et le deuxième réside dans sa capacité descriptive due à l'utilisation des variables linguistique, néanmoins les méthodes de calcul utilisées par ces derniers présentent certaines limites d'implémentation pour la plupart des supports industriels tels que les automates programmables industriels.

Notre objectif en ce projet de fin d'étude est de valider le modèle mathématique du régulateur TITO (Two In Two Out) PI flou simplifié proposé par N.K. Arun et B.M. Mohan [13]. En ce travail, les différentes étapes de la commande par logique floue sont transformées et décrit parfaitement par un modèle mathématique.

Pour atteindre notre objectif, nous avons envisagé de tester ce modèle mathématique sur Matlab/Simulink, avant de trouver un moyen de matérialiser ce régulateur basé sur ce modèle mathématique et de l'intégrer sur un API S7-300 afin de réguler le niveau d'eau de la station PUL2/EV.

Notre étude suivra le plan suivant :

Nous nous intéresserons 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 ainsi qu'une identification paramétrique de la station de régulation de niveau PUL-2 et une modélisation mathématique du système des deux stations PUL-2 en couplage (mise en interaction des deux réservoirs).

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'interprétation du travail de N.K. Arun et B.M. Mohan sur le modèle mathématique du régulateur TITO PI flou simplifié à base d'inférence de Larsen.

Dans le dernier chapitre nous allons présenter brièvement l'automate programmable SIEMSENS, leurs différents langages de programmation ainsi les techniques d'implémentation du régulateur en simulation sur Matlab/Simulink et en essai expérimental sur automate et une analyse des différents résultats expérimentaux.

Enfin nous terminons notre travail avec une conclusion générale.

CHAPITRE 01 :

Présentation des notions de Régulation des systèmes

1.1. Introduction

Pour garantir les qualités et les quantités du produit fabriqué en industrie, il est nécessaire de déployer des moyens d'observation, de réflexion et d'action sur le procédé ; c'est le rôle de l'automatique englobant le domaine de la régulation qui constitue l'étape, délicate, de la réflexion du contrôle industriel puisqu'elle doit garantir un fonctionnement du processus conforme à l'objectif fixé.

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 désirée appelée consigne cette grandeur physique est appelée grandeur réglée.

Une boucle de régulation est constitué essentiellement d'un capteur transmetteur mesurant et conditionnant la grandeur a réglé d'un régulateur commandant un actionneur qui agit sur la grandeur à régler.

1.3. Intérêt de la régulation automatique

L'objectif de réguler une grandeur, est d'obtenir d'elle un comportement donné, dans un environnement susceptible de présenter des variations sur la grandeur à régler.

Maintenir la grandeur commandée, ou grandeur réglée, à une valeur de référence malgré les variations des conditions extérieures ; on parle de la régulation en sens strict, Répondre à des changements d'objectif, ou à un objectif variable tel-que la poursuite de cible, on parle d'un fonctionnement d'asservissement.

1.4. Principe général de la régulation

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

d'eau). Dans la majorité des cas ce type de réglage n'est pas suffisant, parce qu'il est trop grossier ou instable. Il faut alors comparer, en permanence, la valeur mesurée de la grandeur réglée à celle que l'on souhaite obtenir et agir en conséquence sur la grandeur d'action, dite grandeur réglant. On a, dans ce cas, constitué une boucle de régulation et plus généralement une boucle d'asservissement.

Cette boucle nécessite la mise en œuvre d'un ensemble de moyens de mesure, de traitement de signal ou de calcul, d'amplification et de commande d'actionneur, constituant une chaîne d'éléments associés : la chaîne de régulation (ou d'asservissement).

Toute chaîne de régulation (ou d'asservissement) comprend trois maillons indispensables : l'organe de mesure, l'organe de régulation et l'organe de contrôle. Il faut donc commencer par mesurer les principales grandeurs servant à contrôler le processus. L'organe de régulation récupère ces mesures et les compare aux valeurs souhaitées, plus communément appelées valeurs de consigne. En cas de non-concordance des valeurs de mesure et des valeurs de consigne, l'organe de régulation envoie un signal de commande à l'organe de contrôle (vanne, moteur, etc.), afin que celui-ci agisse sur le processus. Les paramètres qui régissent le processus sont ainsi stabilisés en permanence à des niveaux souhaités. La figure 1.1 présente le schéma principale d'une chaîne de régulation.

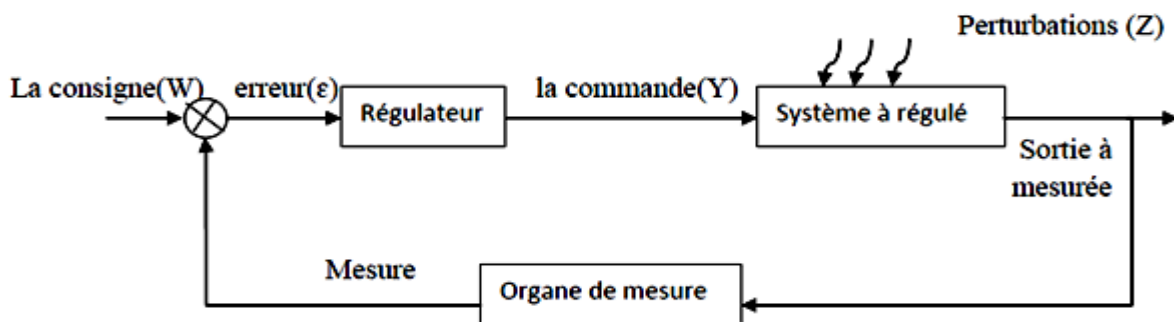


Figure 1.1 : Schéma de principe d'une régulation

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

1.4.1. Comportement en mode régulation

En mode régulation la consigne est maintenue constante sous l'effet d'une modification (ou une variation) des entrées perturbatrices.

L'aspect régulation est l'élément le plus important dans le milieu industriel, car les valeurs des consignes sont souvent fixes. Néanmoins, pour tester les performances et la qualité d'une boucle de régulation, on s'intéresse à l'aspect asservissement.

1.4.2. Comportement en mode asservissement

En mode asservissement, un changement de la valeur de la consigne correspond à une modification du point de fonctionnement du processus.

Si le comportement en asservissement est correct, les performances dynamiques de la boucle de régulation sont conservé, même lorsqu'une perturbation se produit.

1.5. Régulation en boucle ouverte

Dans un contrôle en boucle ouverte, l'organe de contrôle n'agit pas sur le processus à partir de la grandeur mesurée (celle-ci n'est pas contrôlée). Une 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).

Contrairement à un asservissement en boucle fermée, un asservissement en boucle ouverte permet d'anticiper les phénomènes et d'obtenir des temps de réponse très courts. De plus, il n'y a pas d'oscillation à craindre (car il s'agit d'un système dynamiquement stable). Enfin, l'asservissement en boucle ouverte est la seule solution envisageable lorsqu'il n'y a pas de contrôle final possible.

Au niveau des inconvénients, la régulation en boucle ouverte impose de connaître la loi régissant le fonctionnement du processus, et il est très fréquent que l'on ne connaisse pas la loi en question. Autre inconvénient sérieux, il n'y a aucun moyen de contrôler, à plus forte raison de compenser, les erreurs, les dérives, les accidents qui peuvent intervenir à l'intérieur de la boucle ; autrement dit, il n'y a pas de précision ni surtout de fidélité qui dépendent de la qualité intrinsèque des composants. Enfin, la régulation en boucle ouverte ne compense pas les facteurs perturbateurs. La figure 1.2 illustre le schéma de principe de la régulation en boucle ouverte.

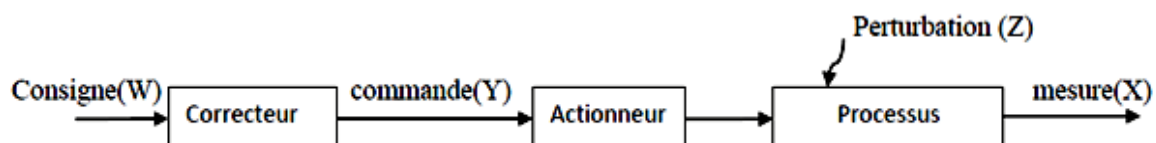


Figure 1.2 : Schéma d'une régulation en boucle ouverte

1.6. Régulation en boucle fermée

Dans une régulation en boucle fermée, la variable de sortie (de la chaîne de régulation), ou grandeur réglante, exerce une influence sur la valeur de la variable d'entrée du système ou variable contrôlée, pour la maintenir dans des limites définies. L'action de la grandeur réglante sur la variable contrôlée s'opère à travers le "processus" qui boucle la chaîne. Voir figure 1.3.

Dans une régulation en boucle fermée, une bonne partie des facteurs perturbateurs 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. Autre inconvénient, sans doute plus important et relatif au comportement dynamique de la boucle qui dépend des caractéristiques des différents composants de la boucle, et notamment du processus. Dans ce cas de figure un mauvais choix de certains composants peut amener la boucle à entrer en oscillation.

Enfin, la régulation en boucle fermée n'anticipe pas. Pour que la régulation envoie une commande à l'organe de contrôle, il faut que les perturbations ou les éventuelles variations de la valeur de consigne se manifestent sur la sortie du processus ce qui peut exiger un délai parfois important.

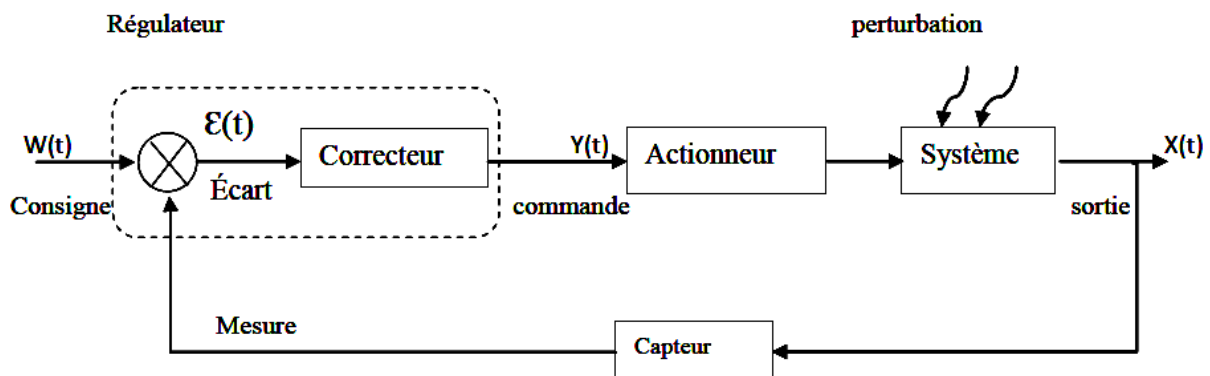


Figure 1.3 : Schéma d'une régulation en boucle fermée

1.7. Qualités attendues d'une régulation

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

1.7.1. 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 le signal de sortie 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].

1.7.2. La précision

La précision d'un procédé ou système régulé se mesure par l'écart entre la consigne demandée et la mesure en régime permanent de la grandeur réglée, on parle alors de précision statique. Plus l'écart statique est petit, plus le système est précis. Compté à la précision dynamique, elle se définit par d'autres caractéristiques tel que : le premier dépassement.

La précision d'un système se résume donc généralement par la qualité des paramètres de celui-ci [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 [1]. La précision d'un système régulé se mesure donc à l'écart entre la consigne demandée et la mesure en régime permanent ; ce qui représente la 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 paramétrages des régulateurs. [1]

1.7.3. 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 Le système régulé est d'autant plus rapide que le temps de réponse à 5% est court. [1]

1.8. Régulation en 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 se trouvant en amont de la variable à régler. Ce type de régulation est intéressant lorsque l'on a affaire à des processus à longs temps de réponse. 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. Si les temps de réponse sont longs, la correction n'intervient donc que tardivement, parfois avec la cause qui l'a produite et dont le sens s'est inversé, provoquant oscillations, instabilité, etc.

Bien évidemment, la régulation en cascade 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. Voir figure 1.4.

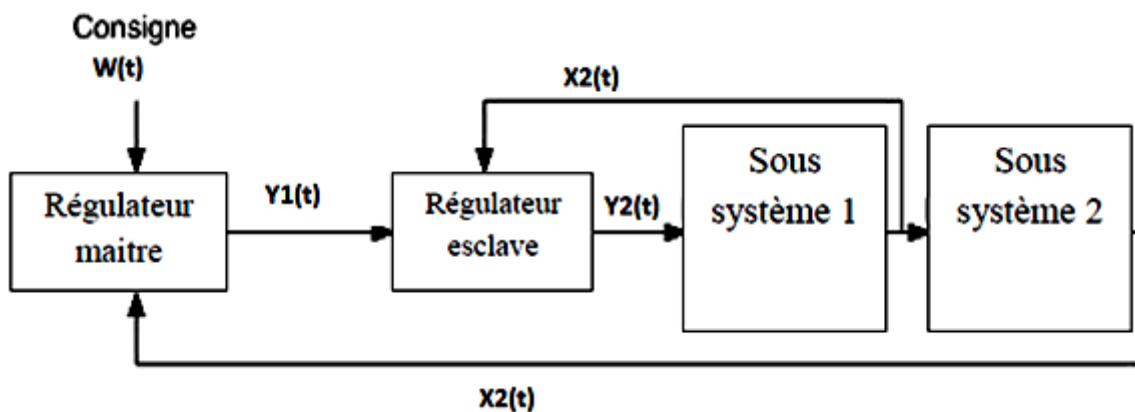


Figure 1.4 : Schéma d'une régulation en cascade

1.8.1. 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 soumis 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.

1.8.2. Cascade sur la grandeur réglante

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

1.9. Régulateurs PID

Le régulateur PID, appelé aussi correcteur PID (proportionnel, intégral, dérivé) est un système de contrôle permettant d'améliorer les performances d'un asservissement, c'est-à-dire un système ou procédé en boucle fermée. Bien que ça soit le régulateur le plus utilisé dans l'industrie où ses qualités de correction s'appliquent à de multiples grandeurs physiques, les valeurs choisies pour les paramètres P, I et D ne sont pas toujours satisfaisante, ni adaptées au processus à régler [4].

Un régulateur PID remplit essentiellement trois fonctions :

- Il fournit un signal de commande $Y(t)$ en tenant compte de l'évolution du signal de sortie $X(t)$ par rapport à la consigne $W(t)$.
- Il élimine l'erreur statique grâce au terme intégrateur.
- Il compense les variations de la sortie grâce au terme dérivateur.

Le régulateur PID classique relie directement le signal de commande $Y(t)$ au signal d'écart

$\mathcal{E}(t)$. Sa description temporelle est la suivante :

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

Avec l'écart défini comme suit :

$$\varepsilon(t) = W(t) - X(t) \quad (1.2)$$

Sa fonction de transfert s'écrit :

$$G_c(s) = \frac{Y(s)}{\varepsilon(s)} = K_p \left(1 + \frac{1}{sT_i} + sT_d \right) \quad (1.3)$$

1.10.1 Les différentes actions du régulateur PID

- **L'action Proportionnelle P**

L'action proportionnelle corrige de manière instantanée, donc rapide, tout écart de la grandeur à régler, elle permet de compenser les grandes inerties du système. Afin de diminuer l'écart de réglage et rendre le système plus rapide, on augmente le gain (on diminue la bande proportionnelle) mais, on est limité par la stabilité du système. Le régulateur P est utilisé lorsqu'on désire régler un paramètre dont la précision n'est pas importante. [5]

$$P = K_p \varepsilon(t) \quad (1.4)$$

- **L'action Intégrale I**

L'action intégrale complète l'action proportionnelle. Elle permet d'éliminer l'erreur résiduelle en régime permanent. Afin de rendre le système plus dynamique (diminuer le temps de réponse), on diminue l'action intégrale ce qui provoque l'augmentation du déphasage ce qui génère l'instabilité en état fermé.

L'action intégrale est utilisée lorsqu'on désire avoir en régime permanent, une précision parfaite, en outre, elle permet de filtrer la variable à régler d'où l'utilité pour le réglage des variables bruitées.

$$I = \frac{1}{T_i} \int_0^t \varepsilon(t) dt \quad (1.5)$$

- **L'action Dérivée D**

L'action dérivée, en compensant les inerties dues au temps mort, accélère la réponse du système et améliore la stabilité de la boucle, en permettant notamment un amortissement rapide des oscillations dues à l'apparition d'une perturbation ou à une variation subite de la consigne.

Dans la pratique, l'action dérivée est appliquée aux variations de la grandeur à régler seule et non de l'écart mesure-consigne afin d'éviter les à-coups dus à une variation subite de la consigne.

L'action D est utilisée dans l'industrie pour le réglage des variables lentes telles que la température, elle n'est pas recommandée pour le réglage d'une variable bruitée ou trop

dynamique. En dérivant un bruit, son amplitude risque de devenir plus importante que celle du signal utile.

$$D = T_d \frac{d\varepsilon(t)}{dt} \quad (1.6)$$

1.10. Synthèse des paramètres des actions PID

1.10.1 Méthodes de Ziegler et Nichols (ZN)

En 1942, Ziegler et Nichols ont proposé deux approches heuristiques basées sur leur expérience et quelques simulations pour ajuster rapidement les paramètres des régulateurs P, PI et PID. La première méthode nécessite l'enregistrement de la réponse indicielle, figure 1.5, en boucle ouverte, alors que la deuxième demande d'amener le système bouclé à sa limite de stabilité. [4]

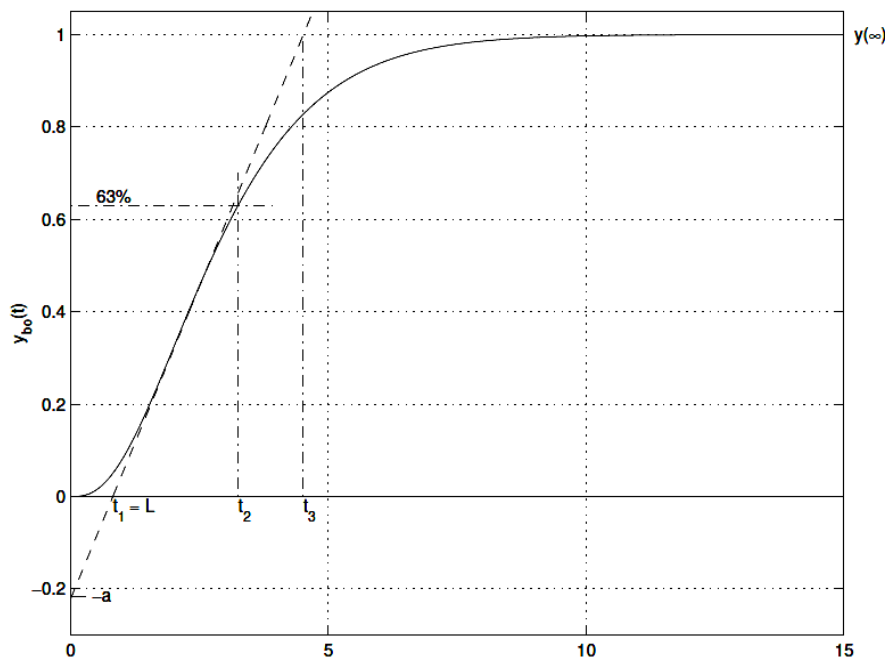


Figure 1.5 : Réponse indicielle d'un processus

1.10.2 Méthode de la réponse indicielle

Pour obtenir les paramètres du régulateur PID, il suffit d'enregistrer la réponse indicielle du processus seul (c'est-à-dire sans le régulateur), puis de tracer la tangente au point d'inflexion de la courbe. On mesure ensuite sa pente p et le retard apparent L correspondant au point d'intersection de la tangente avec l'abscisse (figure 1.8). On peut alors calculer les coefficients du régulateur choisi à l'aide du tableau 1.1.

Type	K_p	T_i	T_d
P	$1/(pL) = 1/a$		
PI	$0.9/(pL)=0.9/a$	$3L$	
PID	$1.2/(pL)=1.2/a$	$2L$	$0.5L$

Tableau 1.1 : Paramètres PID obtenus à partir d'une réponse indicielle (ZNt)

Généralement les gains K_p proposés par Ziegler-Nichols sont trop élevés et conduisent à un dépassement supérieur à 20%. Il ne faut donc pas craindre de réduire K_p d'un facteur 2 pour obtenir une réponse satisfaisante.

1.10.3 Méthode du point critique

Cette méthode est basée sur la connaissance du point critique du processus.

Expérimentalement, on boucle le processus sur un simple régulateur proportionnel dont on augmente le gain jusqu'à amener le système à osciller de manière permanente ; on se trouve ainsi à la limite de stabilité. Après avoir relevé le gain critique K_{cr} du régulateur et la période d'oscillation T_{cr} de la réponse, on peut calculer les paramètres du régulateur choisi à l'aide du tableau 1.2.

Type	K_p	T_i	T_d
P	$0.5K_{cr}$		
PI	$0.4K_{cr}$	$0.8T_{cr}$	
PID	$0.6K_{cr}$	$0.5T_{cr}$	$0.125T_{cr}$

Tableau 1.2 : Paramètres PID obtenus à partir du point critique (ZNf)

Ici également, les valeurs proposées conduisent à un temps de montée relativement court malheureusement assorti d'un dépassement élevé. Cette situation n'étant pas toujours satisfaisante, on peut être amené à corriger les coefficients proposés et, en particulier, à diminuer le gain K_p .

On notera que les paramètres T_i et T_d proposés par les deux méthodes de Ziegler-Nichols sont dans un rapport constant égal à 4. Le régulateur possède donc deux zéros confondus valant $-1/(2T_d) = -2/T_i$.

1.10.4. Méthode de placement de pôles

Cette méthode est analytique et consiste sur la compensation des pôles du système.

On peut utiliser cette méthode pour le calcul des paramètres du contrôleur PI pour un système du 1^{er} ou 2^{ème} ordre comme suit :

a) Fonction de transfert d'un système 1^{er} ordre bouclé par PI [6] :

La figure 1.6 montre un système asservi un correcteur, $G(p)$ est la FT du correcteur et $H(p)$ celle du système.

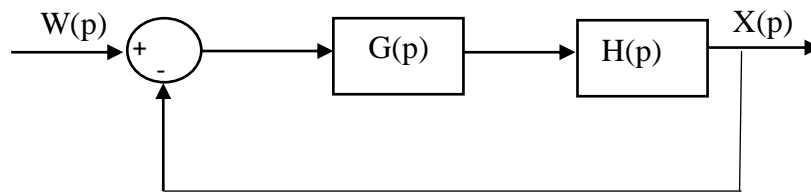


Figure 1.6 : Système asservi par un correcteur

Pour effectuer un placement de pôle on doit calculer la FT globale $F(p)$ en boucle fermée (1.7).

Avec $G(p) = k_p \left(1 + \frac{1}{\tau_i p} \right)$ et $H(p) = \frac{K}{1 + \tau p}$

On a :

$$F_{BF} = \frac{F_{BO}}{1 + F_{BO}} = \frac{G(p)H(p)}{1 + G(p)H(p)}$$

$$F_{BF} = \frac{Kk_p (1 + \tau_i p)}{\tau\tau_i p^2 + p(\tau_i + Kk_p \tau_i) + Kk_p} \tag{1.7}$$

b) calcul des paramètres du PI :

- F_{BF} désirée : premier ordre

On simplifie le pole $\tau_i = \tau \Rightarrow F_{BF} = \frac{1}{1 + \tau' p}$

On a alors :

$$\tau' \text{ Désiré} = \frac{\tau_i}{Kk_p} \Rightarrow k_p = \frac{\tau_i}{K\tau'}$$

- F_{BF} désirée : deuxième ordre

$$F_{BF} = \frac{1 + \tau_2 \tau_n p}{1 + 2\zeta \tau_n p + \tau_n^2 p^2} \tag{1.8}$$

On simplifie les pôles puis on calcule les paramètres comme suit :

$$\left. \begin{aligned} 2\zeta\tau_n &= \frac{1 + Kk_p}{Kk_p} \tau_i \\ \tau_n^2 &= \frac{\tau\tau_i}{Kk_p} \end{aligned} \right\} \rightarrow \begin{aligned} \tau_i &= \tau_n \left(2\zeta - \frac{\tau_n}{\zeta} \right) \\ k_p &= \frac{2\zeta\tau / \tau_n - 1}{K} \end{aligned}$$

$$\tau_2\tau_n = \tau_i \Rightarrow \tau_2 = \frac{\tau_i}{\tau_n}$$

1.11. Les types de procédés

1.11.1. Procédé mono variable

Un système mono variable est un système présentant un bouclage entre une seule commande et une mesure [3]. (Voir figure 1.7)

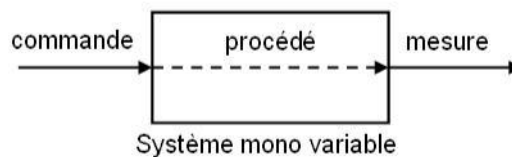


Figure 1.7 : Schéma d'un procédé mono variable.

Exemple : Pour réguler le niveau dans un réservoir, l'arrivé du liquide est commandé avec une vanne proportionnelle qui agit sur le débit. Rien d'autre ne fait varier le niveau.

1.11.2. Procédé multi variable

C'est le cas d'un procédé où une mesure peut être affectée par plusieurs commandes [3].(Voir figure 1.8)

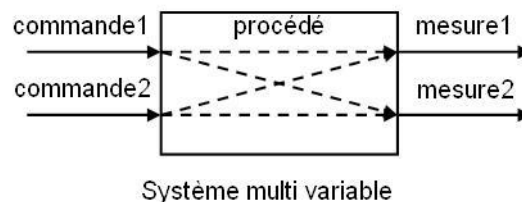


Figure 1.8 : Schéma d'un procédé multi variable.

On peut illustrer comme exemple le contrôle du niveau du liquide dans un réservoir se règle grâce à une vanne régulatrice d'arrivée et une autre de refoulement ; il s'agit d'une configuration multi variable simple(le même niveau pouvant être atteint par des choix différents sur le débit d'entré et de sortie). En revanche, le problème est plus complexe si l'on

désire réguler le niveau de liquide dans deux réservoirs qui sont interagissent par une conduite. Dans ce cas, on ne peut plus régler un niveau sans tenir compte de l'autre. Toute variation sur un actionneur fait varier les deux niveaux.

1.12. Types de système selon le procédé

1.12.1. Système SISO (single IN single OUT)

Un système SISO est un système à procédé mono-variable ayant une seule entrée et une seule sortie avec un bouclage entre l'unique consigne et l'unique commande, Voir figure 1.7.

1.12.2. Système TISO (two IN single OUT)

Ce système est à procédé multi-variable qui possède deux entrées et une seule sortie, les deux consignes sont généralement bouclées en cascade avec la commande.

1.12.3. Système TITO (two IN two OUT)

Ce système est à procédé multi-variable ayant deux entrées et deux sorties, le bouclage des consignes avec les commandes peut se faire indépendamment ou bien en interaction, ce qui fait intervenir un découpleur pour régir le fonctionnement de l'interaction.

Si nous reprenons l'exemple précédent, un système à deux réservoirs liés par une conduite, nous aurons à commander les deux vannes proportionnelle en même temps pour asservir ou réguler les sorties du système tout en essayant d'anticiper l'effet d'interaction des deux niveaux, alors notre procédé est multi-variable et le système est TITO, voir figure 1.8.

1.13. 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 en termes de dépassement, temps de réponse et en précision statique sous l'effet d'un changement de consigne ou de présence de perturbation pour atteindre l'objectif fixé.

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

CHAPITRE 02 :

Présentation de la station de régulation de niveau

2.1. Introduction

La régulation de niveau est une application très répandue en industrie qui fait appel à divers instruments pour assurer l'acquisition, la transmission et le traitement des signaux.

Afin de préparer les deux stations de régulation de niveau PUL-2 présentes au niveau du laboratoire API aux différents tests expérimentaux et de bien maîtriser le procédé, nous allons procéder à l'étalonnage des différents instruments de mesure, l'identification et modélisation mathématique du système.

Dans ce chapitre nous allons nous contenter de présenter une seule unité de régulation de niveau ainsi que les différentes procédures d'étalonnage et d'identification effectuées sur cette station.

2.2. Description de l'unité

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 le liquide (eau) nécessaire pour atteindre le niveau souhaité 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, permettant 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. [15]

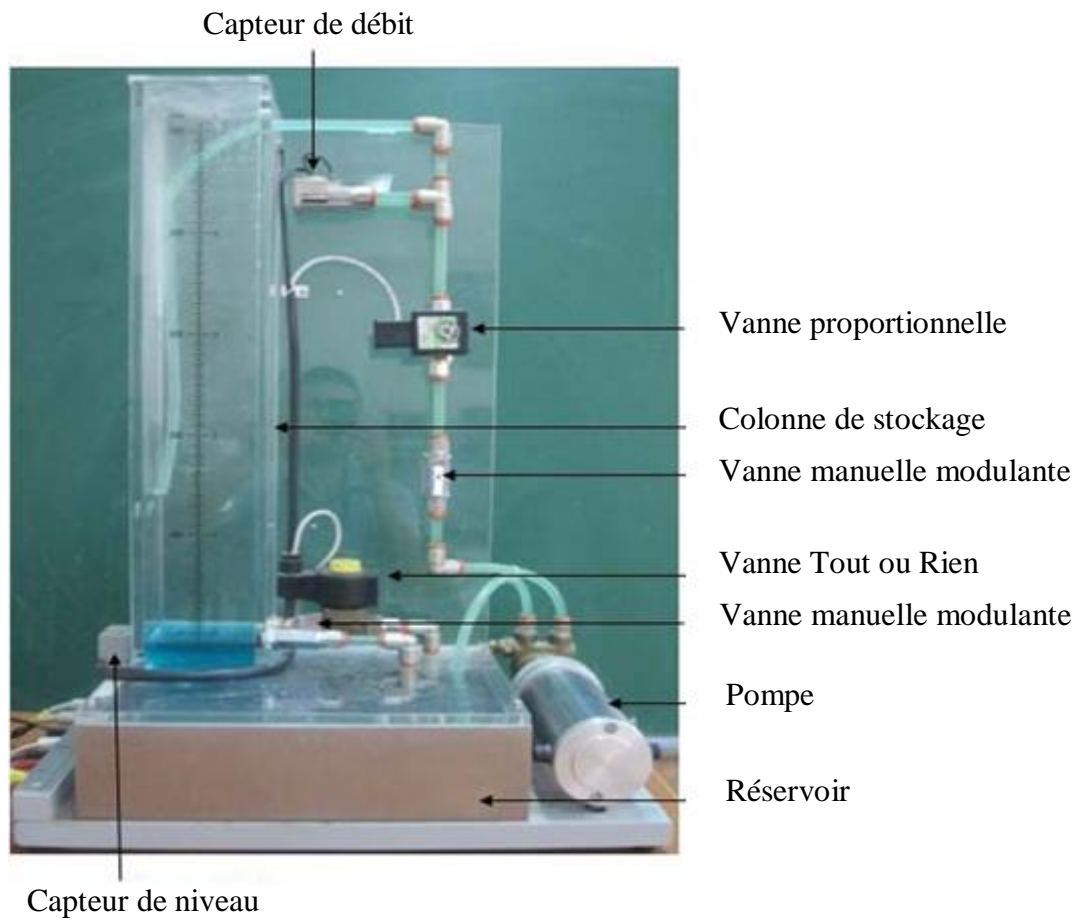


Figure 2.1 : La station de niveau PUL-2

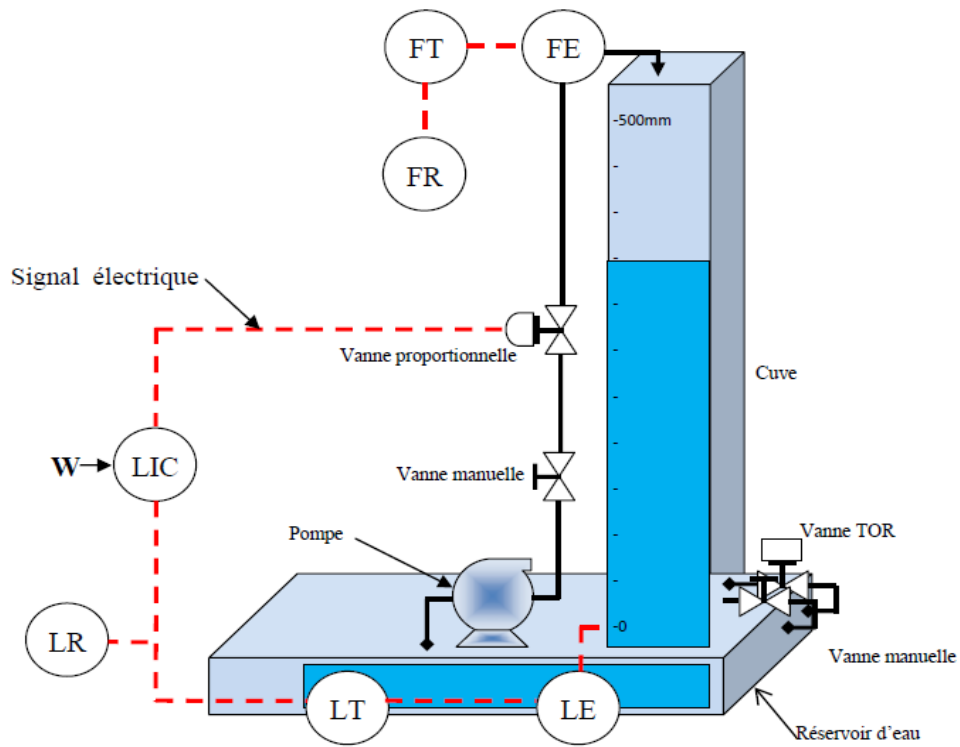


Figure 2.2 : Schéma Pi&D de la station

FE : Capteur de débit (Flow Emitter).

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 Emitter)

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. La mesure de niveau est une mesure continue, c'est-à-dire que le capteur et son transmetteur 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. 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.

2.5. 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)

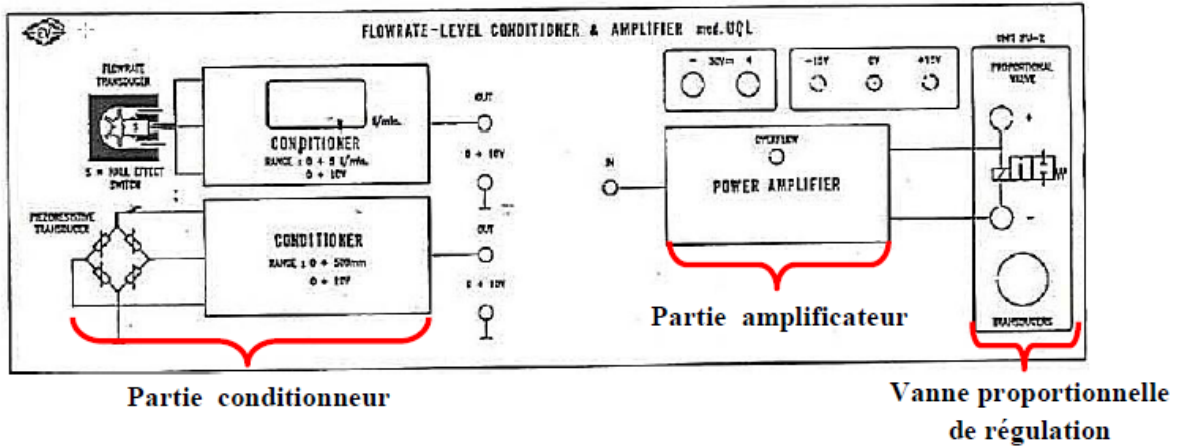


Figure 2.3 : Bloc conditionneur et amplificateur de puissance

2.6. Transducteur de niveau de la station PUL-2

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). [15]

2.6.1. Caractérisation du capteur de niveau

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

Nous avons testé ce capteur de niveau et nous avons obtenu la courbe suivante de la figure (2.4).

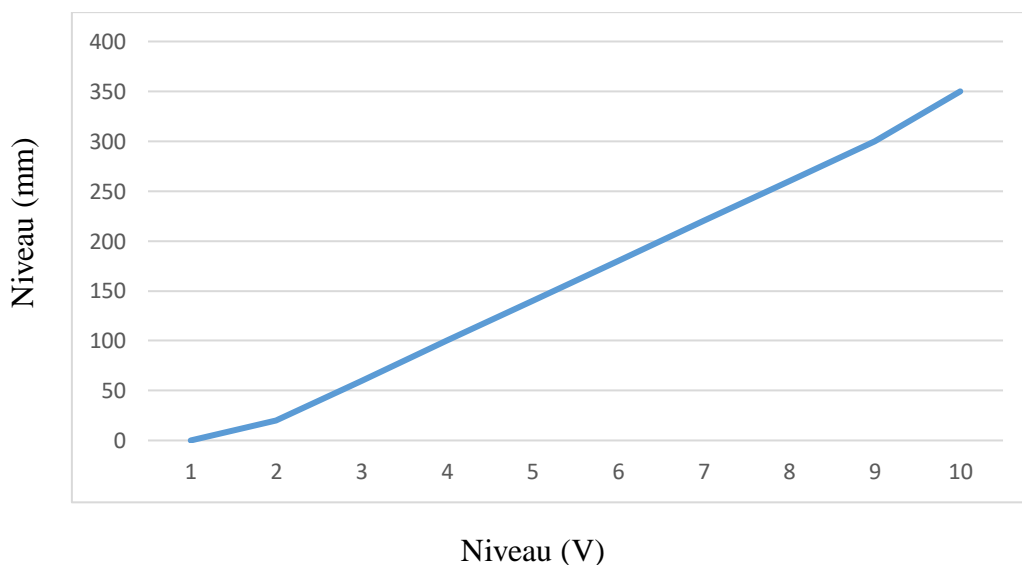


Figure 2.4 : Vérification d'état du capteur-transmetteur de niveau

La courbe caractéristique du capteur transmetteur de niveau donne une tension de sortie du transmetteur proportionnelle et linéaire par rapport au niveau du liquide dans la cuve.

2.7. 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.

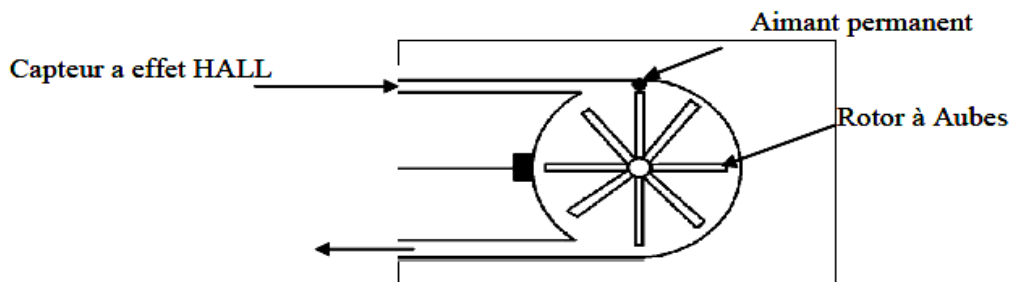


Figure 2.5. Structure interne du débitmètre à aubes

La mesure de débit passe par la mesure de la vitesse de rotation de la roue en tours/minute et à qui correspond un certain volume bien déterminé de liquide déplacé pour chaque tour.

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

Les débitmètres à aubes sont choisis pour un débit maximal supérieur de 30-50% au débit maximal prévu, car le fonctionnement en dessous de ce débit maximal 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.

2.7. Vanne proportionnelle

La vanne proportionnelle est une vanne dont le niveau du clapet est réglable à souhait. Elle est activée 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.

2.8. Caractérisation de la vanne proportionnelle

La caractérisation de la vanne proportionnelle passe par un relevé de la variation du débit en fonction de la tension exercé cette dernière.

D'après la figures (2.6), on constate que :

Quel que 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.

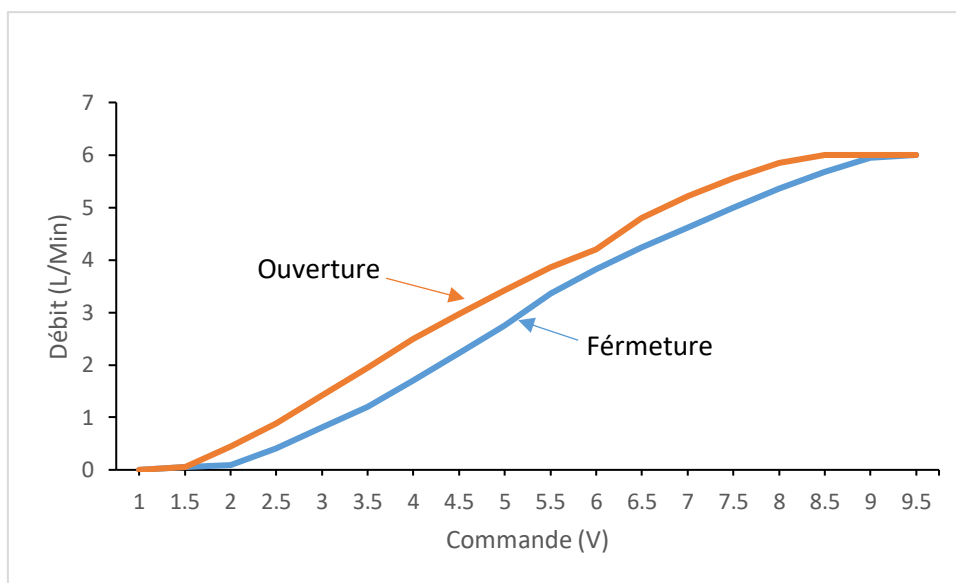


Figure 2.6. Hystérésis de la vanne proportionnelle.

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 montre l'effet d'hystérésis.

2.9. Identification du system

2.9.1. Caractéristique statique du système

Pour identifier le système autour d'un point de fonctionnement, nous allons 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 le niveau d'eau dans le réservoir. Les résultats expérimentaux obtenus sont présentés sur la figure 2.7.

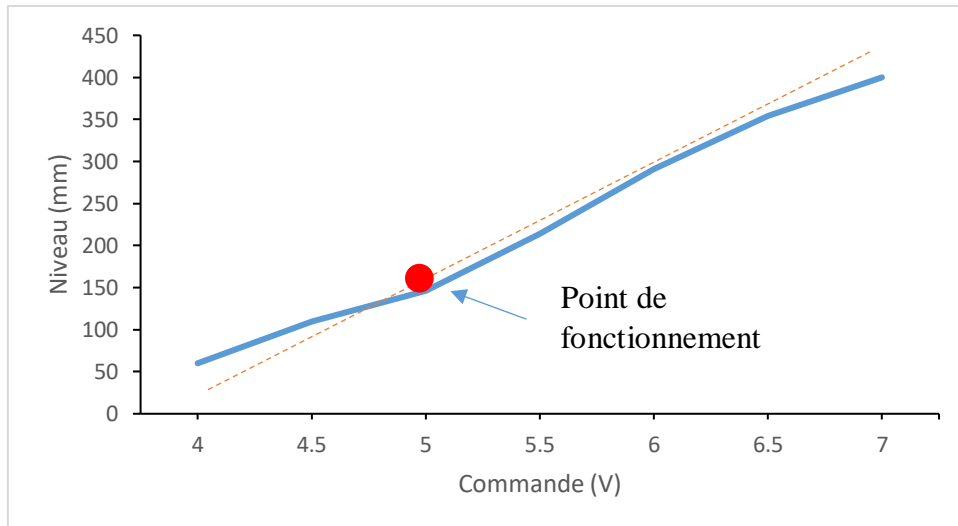


Figure 2.7. Caractéristique statique du processus

La figure (2.7) ci-dessus présente un système linéaire sur toute la plage de travail (de 4V à 7V). Dans la suite de notre travail nous allons identifier notre system autour de cette plage d’exploitation.

2.9.2. Réponse indicielle du système de niveau

Après avoir choisis la zone d’exploitation linéaire, nous avons sélectionné un point de fonctionnement dans cette zone, qui est de 4.5V équivalent à 230 cm en hauteur de niveau. L’application d’un échelon de tension de 0.2V, donne le relevé indiciel présenté en figure 2.8.

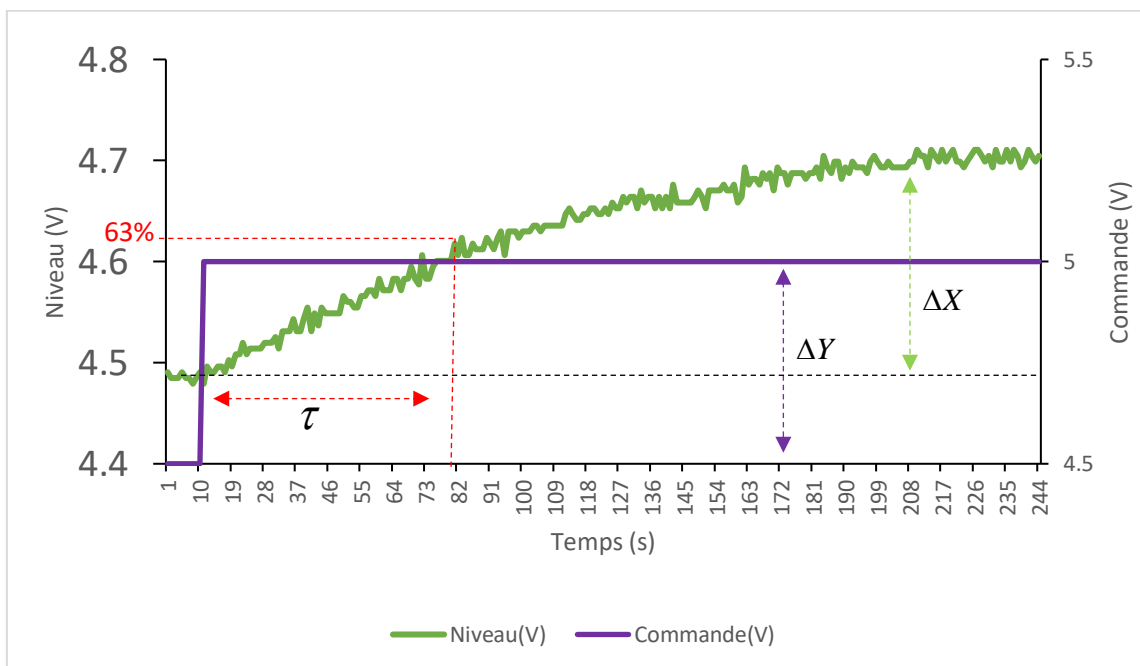


Figure 2.8 : Réponse indicielle du système de niveau.

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

$$G(p) = \frac{G_s}{1 + \tau p} \quad (2.1)$$

Avec un gain statique

$$G_s = \frac{\Delta X}{\Delta Y} = \frac{4.7 - 4.48}{4.5 - 4} = \frac{0.22}{0.5} = 0.44 \quad (2.2)$$

Pour déterminer la constante de temps : on trace conjointement la droite d'ordonnée $63\% \Delta X$ parallèle à l'axe des abscisses ce qui donne : $\tau = 70 \text{ sec}$. La fonction de transfert du système globale est la suivante :

$$G(p) = \frac{0.44}{1 + 70p} \quad (2.3)$$

2.9.3. Réponse indicielle du système de débit :

Le système de débit est un système linéaire. Pour l'identifier nous allons soumettre le système de débit à un échelon d'amplitude 1V comme le montre la (figure 2.10).

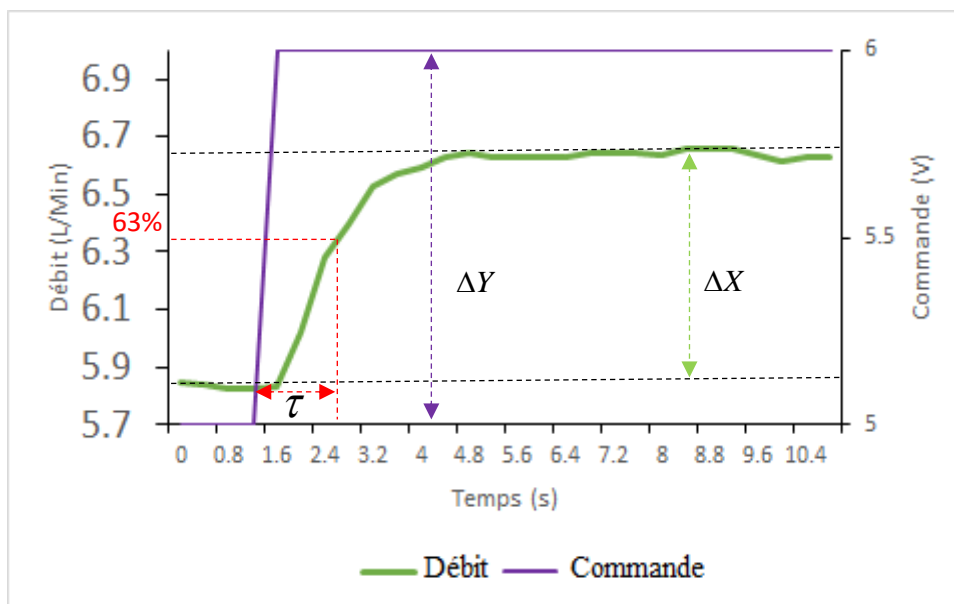


Figure 2.9 : Réponse indicielle du système de débit.

La réponse indicielle de la figure 2.9 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 :

$$G_s = \frac{\Delta X}{\Delta Y} = \frac{6.63 - 5.84}{6 - 5} = \frac{0.79}{1} = 0.79 \quad (2.4)$$

Avec :

$$\tau = 2.8 - 1.6 = 1.2s$$

La fonction de transfert du système de débit est : $G(p) = \frac{0.79}{1 + 1.2p}$

2.10. Couplage des deux stations de régulation de niveau PUL-2 :

Afin d'avoir un système de régulation de niveau multi variable, nous avons assemblé deux station du type PUL-2 identique présente au niveau du laboratoire d'automate programmable et de la régulation industrielle du département d'automatique.

Dans cet assemblage technique nous avons mis en interaction le niveau d'eau des deux réservoirs des stations. La figure 2.10 présente une image du couplage des deux stations PUL-2.

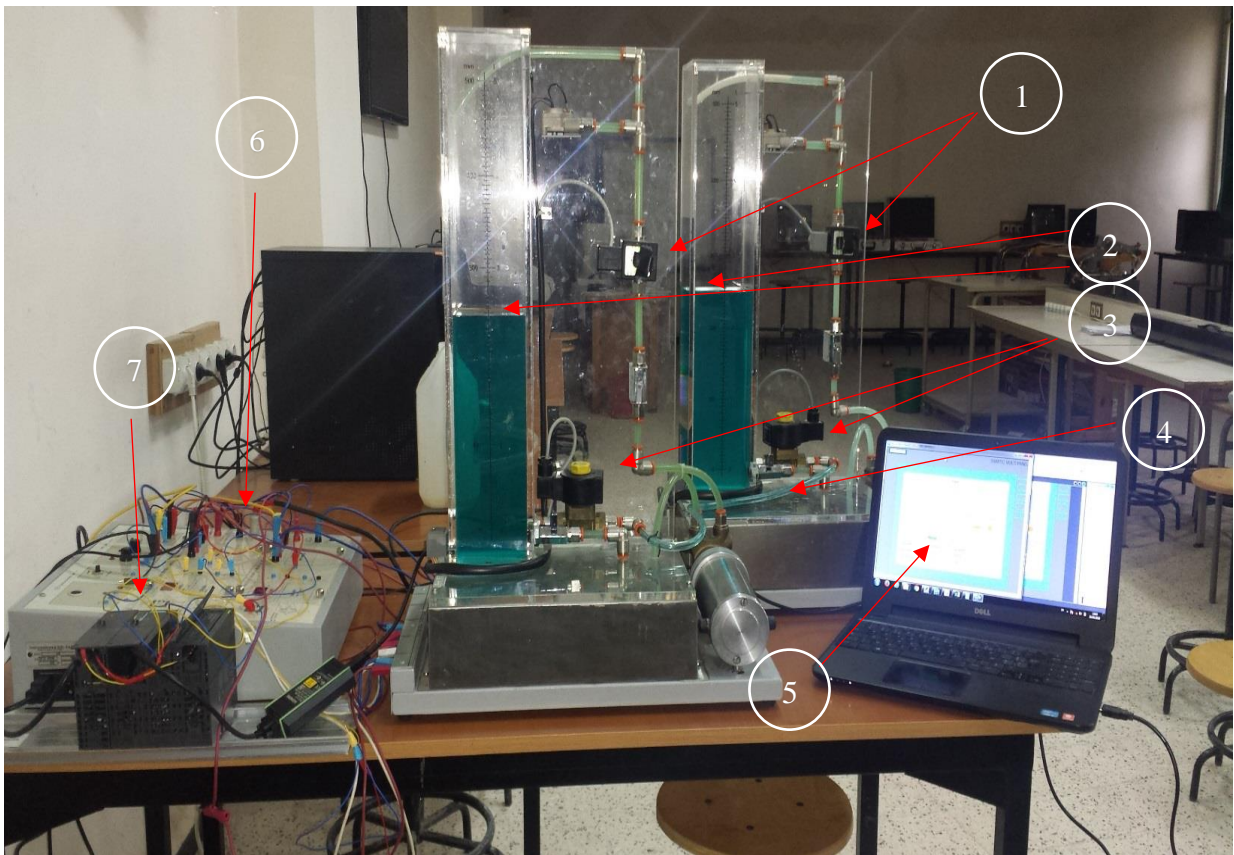


Figure 2.10 : Couplage des deux stations de régulation de niveau PUL-2.

La figure 2.10 est constituée des éléments suivants :

- 1) Vannes proportionnelles.
- 2) Les niveaux d'eau dans les deux colonnes.
- 3) Vannes TOR (pour activer/désactiver l'interaction).
- 4) Conduite d'interaction.
- 5) Interface de supervision (IHM).
- 6) Bloc d'alimentation des deux stations.
- 7) Automate Siemens S7-300.

2.10.1. Modèle d'état du système en couplage des deux stations

Avant d'entamer l'étape d'implémentation réelle du régulateur, il est essentiel de comprendre les équations mathématiques qui régissent le comportement du système à deux réservoirs couplés [16], voir figure 2.11

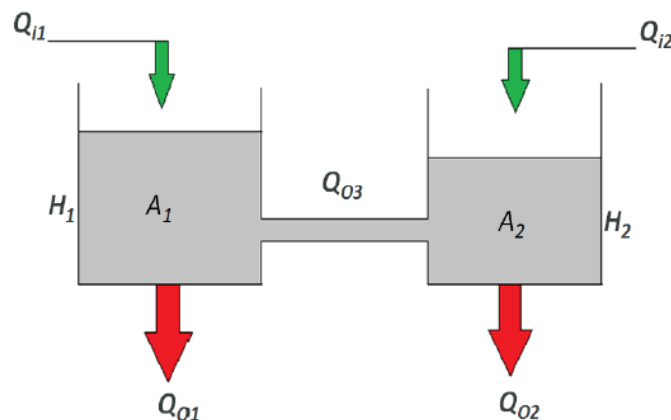


Figure 2.11 : Système à deux réservoirs couplés.

Soit H_1 et H_2 le niveau de liquide dans chaque réservoir et qui sont les deux variables à réguler de notre système. Considérant un simple bilan de masse, le taux de changement du volume d'eau dans chaque réservoir est égal au débit net d'entrée (Q_{i1} et Q_{i2}). Ainsi pour chaque réservoir 1 et 2, l'équation dynamique est développée comme suit :

$$A_1 \left(\frac{dH_1}{dt} \right) = Q_{i1} - Q_{o1} - Q_{o3} \quad (2.5)$$

$$A_2 \left(\frac{dH_2}{dt} \right) = Q_{i2} - Q_{o2} - Q_{o3} \quad (2.6)$$

Où :

H_1, H_2 = La hauteur du liquide dans les réservoirs 1 et 2 respectivement.

A_1, A_2 = Surface de la section des deux réservoirs 1 et 2 respectivement.

Q_{o3} = Débit d'interaction entre les deux réservoirs.

Q_{i1}, Q_{i2} = Débits d'entrées.

Q_{o1}, Q_{o2} = Débits de sorties.

Chaque drain de sortie peut être modélisé sous la forme d'un orifice simple. L'équation de Bernoulli donne un flux de sortie proportionnelle à la racine carrée de la hauteur d'eau et cela dans chaque réservoir. De même, l'écoulement entre les deux réservoirs est proportionnelle à la racine carrée de l'écart entre les deux niveaux H_1 et H_2 .

$$Q_{o1} = \alpha_1 \sqrt{H_1} \quad (2.7)$$

$$Q_{o2} = \alpha_2 \sqrt{H_2} \quad (2.8)$$

$$Q_{o3} = \alpha_3 \sqrt{(H_1 - H_2)} \quad (2.9)$$

Où $\alpha_1, \alpha_2, \alpha_3$ sont des constantes proportionnelles qui sont en fonction des coefficients de décharge du débit, de l'aire de section transversale de chaque orifice et de la constante gravitationnelle.

En combinant l'équation (2.6), (2.7) et (2.8) dans les équations (2.5) et (2.6), nous aurons un ensemble d'équations d'état non linéaires qui décrivent la dynamique du système des réservoirs couplés.

$$A_1 \left(\frac{dH_1}{dt} \right) = Q_{i1} - \alpha_1 \sqrt{H_1} - \alpha_3 \sqrt{(H_1 - H_2)} \quad (2.10)$$

$$A_2 \left(\frac{dH_2}{dt} \right) = Q_{i2} - \alpha_2 \sqrt{H_2} - \alpha_3 \sqrt{(H_1 - H_2)} \quad (2.11)$$

Pour des débits d'entrée donnés Q_{i1}, Q_{i2} les niveaux de liquide dans les réservoirs est à un certain niveau d'équilibre H_1 et H_2 et considérons une faible variation de chaque entrée q_1 pour Q_{i1} et q_2 pour Q_{i2} . Notons le résultat de perturbation du niveau respectivement h_1 et h_2 . À partir des équations (2.10) et (2.11), l'équation devient :

- Pour le réservoir 1 :

$$A_1 \left(\frac{d(H_1 + h_1)}{dt} \right) = (Q_{i1} + q_1) - \alpha_1 \sqrt{H_1 + h_1} - \alpha_3 \sqrt{(H_1 - H_2)} + (h_1 - h_2) \quad (2.12)$$

- Pour le réservoir 2 :

$$A_2 \left(\frac{d(H_2 + h_2)}{dt} \right) = (Q_{i2} + q_2) - \alpha_2 \sqrt{H_2 + h_2} - \alpha_3 \sqrt{(H_1 - H_2)} + (h_1 - h_2) \quad (2.13)$$

La soustraction des équations (2.10) et (2.11) à partir de l'équation (2.12) et (2.13), donnent les équations suivantes sont :

$$\sqrt{H_1 + h_1} = \sqrt{H_1 \left(1 + \frac{H_1}{2H_1} \right)} \quad (2.14)$$

D'où :

$$\sqrt{H_1 + h_1} - \sqrt{H_1} \approx \frac{h_1}{2\sqrt{H_1}} \quad (2.15)$$

De même :

$$\sqrt{H_2 + h_2} - \sqrt{H_2} \approx \frac{h_2}{2\sqrt{H_2}} \quad (2.16)$$

Et :

$$\sqrt{(H_2 - H_1 + h_2 - h_1)} - \sqrt{H_2 - H_1} \approx \frac{h_2 - h_1}{2\sqrt{(H_2 - H_1)}} \quad (2.17)$$

La simplification des équations (2.10) et (2.11) avec ces approximations donnent :

$$A_1 \left(\frac{dh_1}{dt} \right) = q_1 - \frac{\alpha_1}{2\sqrt{H_1}} h_1 - \frac{\alpha_3}{2\sqrt{H_1 - H_2}} (h_1 - h_2) \quad (2.18)$$

$$A_2 \left(\frac{dh_2}{dt} \right) = q_2 - \frac{\alpha_2}{2\sqrt{H_2}} h_2 - \frac{\alpha_3}{2\sqrt{H_1 - H_2}} (h_1 - h_2) \quad (2.19)$$

Ainsi le modèle d'état du système qui décrit le fonctionnement du système à deux réservoirs couplés est le suivant :

$$\dot{x}(t) = A \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} + B \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}$$

$$y(t) = C \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} + D \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}$$

Avec :

$$A = \begin{pmatrix} \frac{-\alpha_1}{2A_1\sqrt{H_1}} + \frac{\alpha_3}{2A_1\sqrt{H_1-H_2}} & \frac{\alpha_3}{2A_1\sqrt{H_1-H_2}} \\ \frac{-\alpha_3}{2A_2\sqrt{H_1-H_2}} & \frac{\alpha_2}{2A_2} + \frac{\alpha_3}{2A_2\sqrt{H_1+H_2}} \end{pmatrix} \quad B = \begin{pmatrix} \frac{1}{A_1} & 0 \\ 0 & \frac{1}{A_2} \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad D = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Ce modèle servira de support pour les différents tests de simulation sous Matlab/Simulink.

2.11. Conclusion

Dans ce chapitre nous avons présenté la station de niveau PUL-2 et avec caractérisation des différents éléments constituant cette station (capteur –conditionneur de niveau, capteur –conditionneur de débit et la vanne proportionnelle). Après présentation de la station nous avons réalisé une identification paramétrique de l'unité PUL-2 sans interaction ainsi qu'une modélisation mathématique pour le système en couplage des deux stations qui serviront aux simulations sous Matlab/Simulink.

CHAPITRE 03 :

Présentation des éléments de la Logique Floue

3.1. Introduction

La logique floue est une extension de la logique classique qui permet la modélisation des imperfections des données et se rapproche dans une certaine mesure de la flexibilité du raisonnement humain.

3.2. Concept de la logique floue

La logique floue trouve ses origines dans un certain nombre de constatations [10] :

- La connaissance que l'être humain a d'une situation quelconque est généralement imparfaite, elle peut être incertaine (il doute de sa validité), ou imprécise (il a du mal à l'exprimer clairement).
- L'être humain résout souvent des problèmes complexes à l'aide de données approximatives : la précision des données est souvent inutile ; par exemple pour choisir un appartement, il pourra prendre en compte la surface, la proximité de commerces, la distance du lieu de travail, le loyer, sans pour autant avoir besoin d'une valeur très précise de chacune de ces données.
- Dans l'industrie ou les techniques, les opérateurs résolvent souvent des problèmes complexes de manière relativement simple et sans avoir besoin de modéliser le système. De même, tout le monde sait qu'un modèle mathématique n'est pas nécessaire pour conduire une voiture et pourtant une voiture est un système très complexe.
- Plus la complexité d'un système augmente, moins il est possible de faire des affirmations précises sur son comportement.

De ces constatations viennent naturellement les déductions suivantes :

- plutôt que de modéliser le système, il est souvent intéressant de modéliser le comportement d'un opérateur humain face au système ; plutôt que par des valeurs numériques précises, le fonctionnement doit être décrit par des qualificatifs globaux traduisant l'état approximatif des variables.

3.3. Structure générale d'une commande par logique floue

Dans la commande par logique floue les entrées peuvent être différentes informations récupérées sur le system à commander tels que les mesures, les erreurs entre les mesures et les sorties, la vitesse dont ces erreurs évoluent.....etc. La figure 3.1 donne la structure générale d'une commande floue.

Après l'étape de défuzzification, les sorties issues peuvent être utilisés pour attaquer les organes de commande du system(les actionneurs).

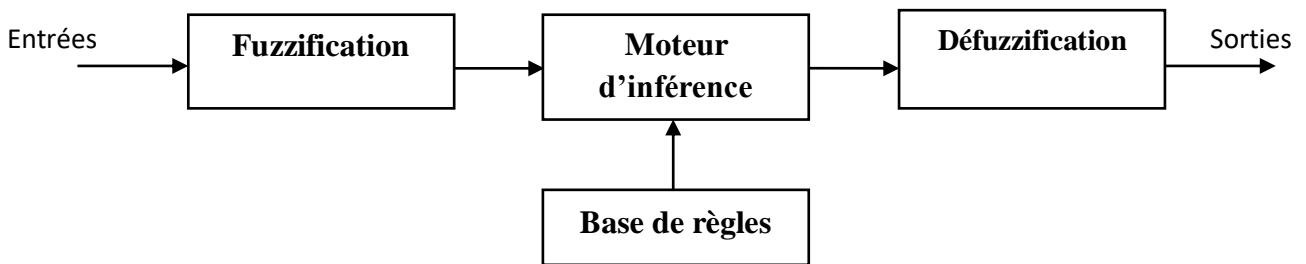
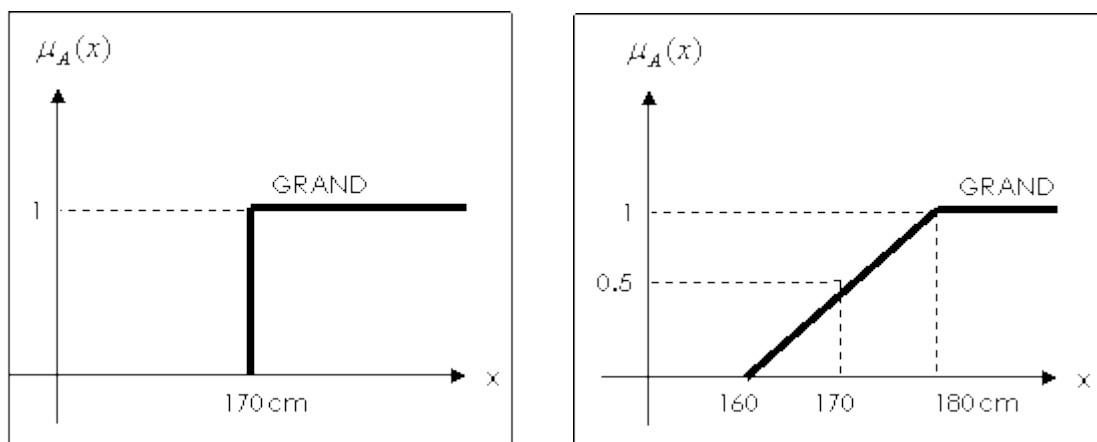


Figure 3.1 : Structure générale d'une commande floue.

3.3.1. Notions de bases de la logique Flou

- **Ensemble flou :**

Dans la théorie classique des ensembles, c'est la fonction caractéristique qui définit l'ensemble, cette fonction ne prend que les deux valeurs discrètes 0 (l'élément n'appartient pas...) ou 1 (...appartient à l'ensemble). Un ensemble flou est défini par une fonction d'appartenance qui peut prendre toutes les valeurs réelles comprises entre 0 et 1 [10]. Un exemple est illustrée en figure 3.2.



(A) Ensemble classique

(B) Ensemble flou

Figure 3.2 : Représentation des ensembles.

- **Fonction d'appartenance :**

La fonction $\mu_A(x)$ qui à toute valeur d'entrée x fait correspondre son degré d'appartenance à l'ensemble A . Cette valeur graduelle est comprise entre 0 et 1. Voir la figure 3.3

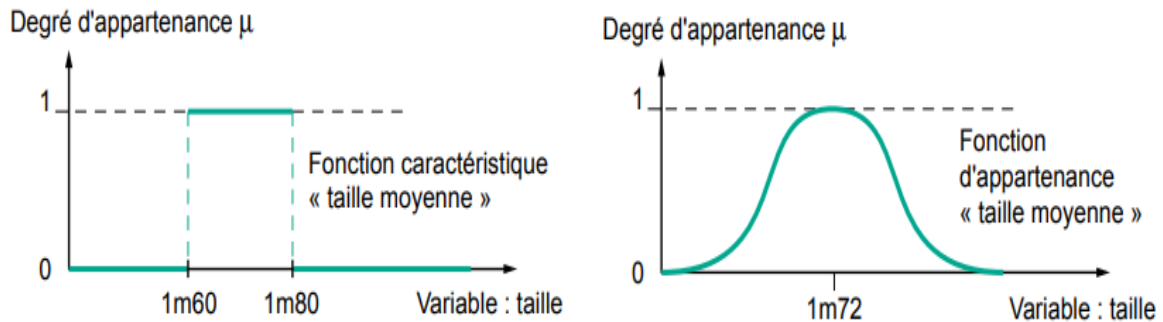


Figure 3.3 : Fonction caractéristique (gauche), fonction d'appartenance (droite).

Les fonctions d'appartenance peuvent théoriquement prendre n'importe quelle forme mais en général les formes les plus utilisées sont définies par des formes géométriques.

- **Différentes formes pour les 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.

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

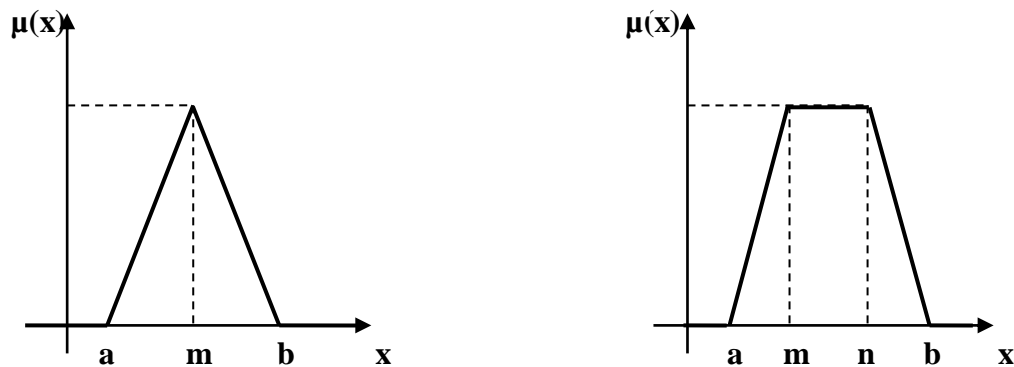


Figure 3.4 : Exemples de fonctions d'appartenances triangulaire et trapézoïdal

- **Degré d'appartenance**

Un élément x appartient à un ensemble flou A avec un degré d'appartenance compris entre 0 et 1, donné par la fonction d'appartenance $\mu_A(x)$.

- **Variable linguistique :**

La variable linguistique est une variable numérique appliquée en entrée, pour fuzzification, ou en sortie, après défuzzification, d'un module de logique floue. On lui attribue l'adjectif linguistique car elle est utilisée, dans les fonctions d'appartenance et les règles, par son nom et non pas par une valeur numérique. La description d'une certaine situation, d'un phénomène ou d'un procédé contient en général des qualificatifs flous tels que :peu, beaucoup, énormément, rarement, fréquemment, souvent, froid, tiède, chaud, petit, moyen, grand... etc.

3.3.2. 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.

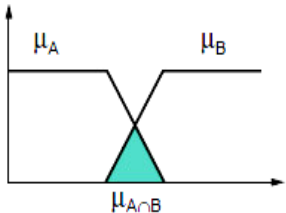
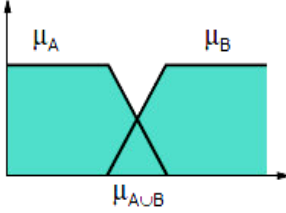
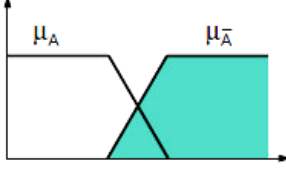
	Opérateur de ZADEH	Opération logique	
Intersection	$\mu_{A \cap B} = \min(\mu_A, \mu_B)$	ET	
Union	$\mu_{A \cup B} = \max(\mu_A, \mu_B)$	OU	
Négation	$\mu_{\bar{A}} = 1 - \mu_A$	NON	

Tableau 3.1 : Opérateurs entre ensembles flous

3.3.4. Fuzzification

La partie fuzzification est la partie qui traite 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 [7].

La réalisation de cette phase de fuzzification passe par une définition de la plage de variations possibles de l'entrée considérée (univers du discours), une partition en classe floue de cet univers et les fonctions d'appartenances de chacune de ces classes.

3.3.5. Moteur d'inférence floue

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 [7] [12].

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)

On utilisera en général la méthode max-min, celle-ci provoque une commande avec des sauts mais ce n'est en général pas ou peu contraignant. Il est à noter que la méthode somme-produit permet de développer un calcul analytique de la sortie permettant, si l'on possède un modèle du système, de calculer les caractéristiques temporelles de la sortie.

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. Le résultat 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 de la figure 3.5

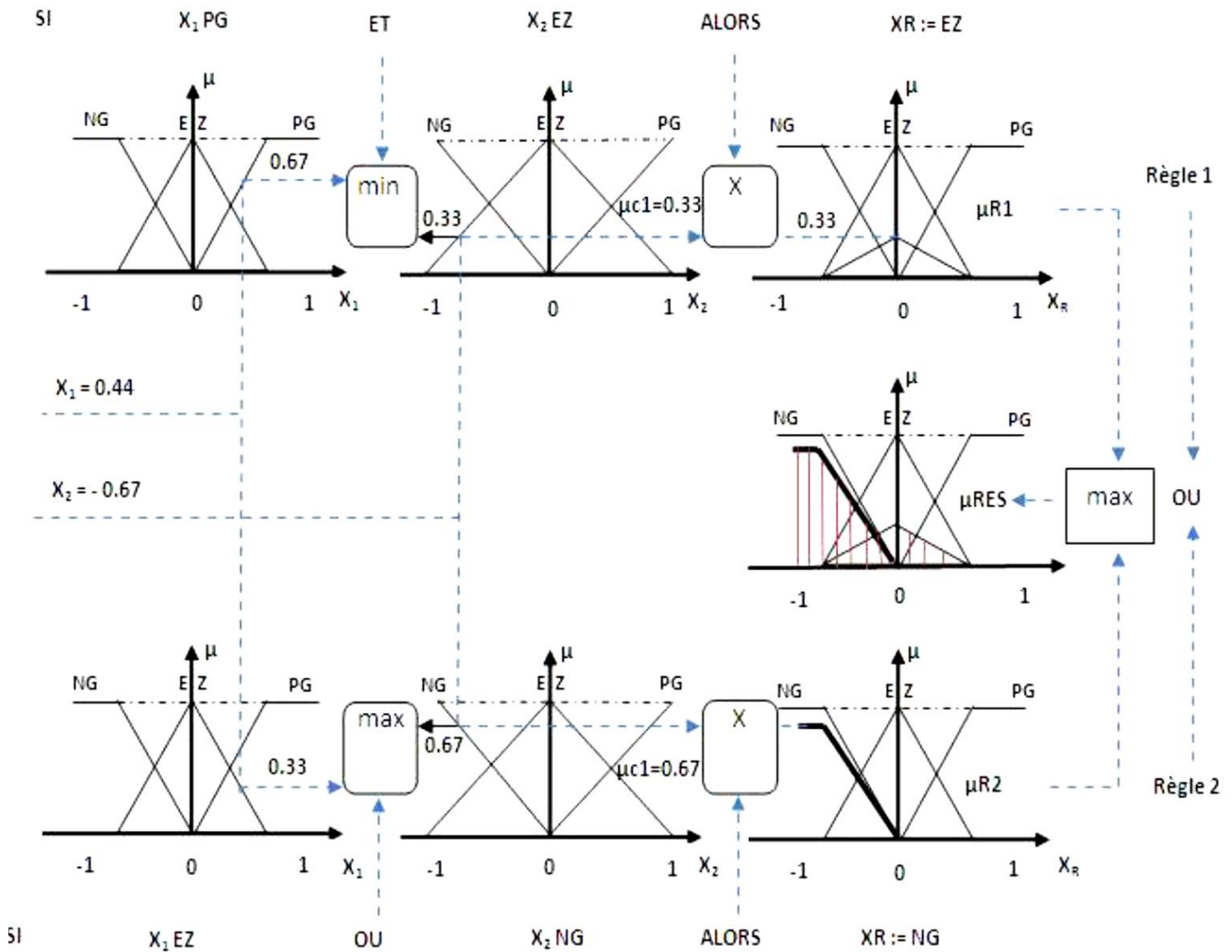


Figure 3.5 : Exemple d'inférence Max-produit

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.

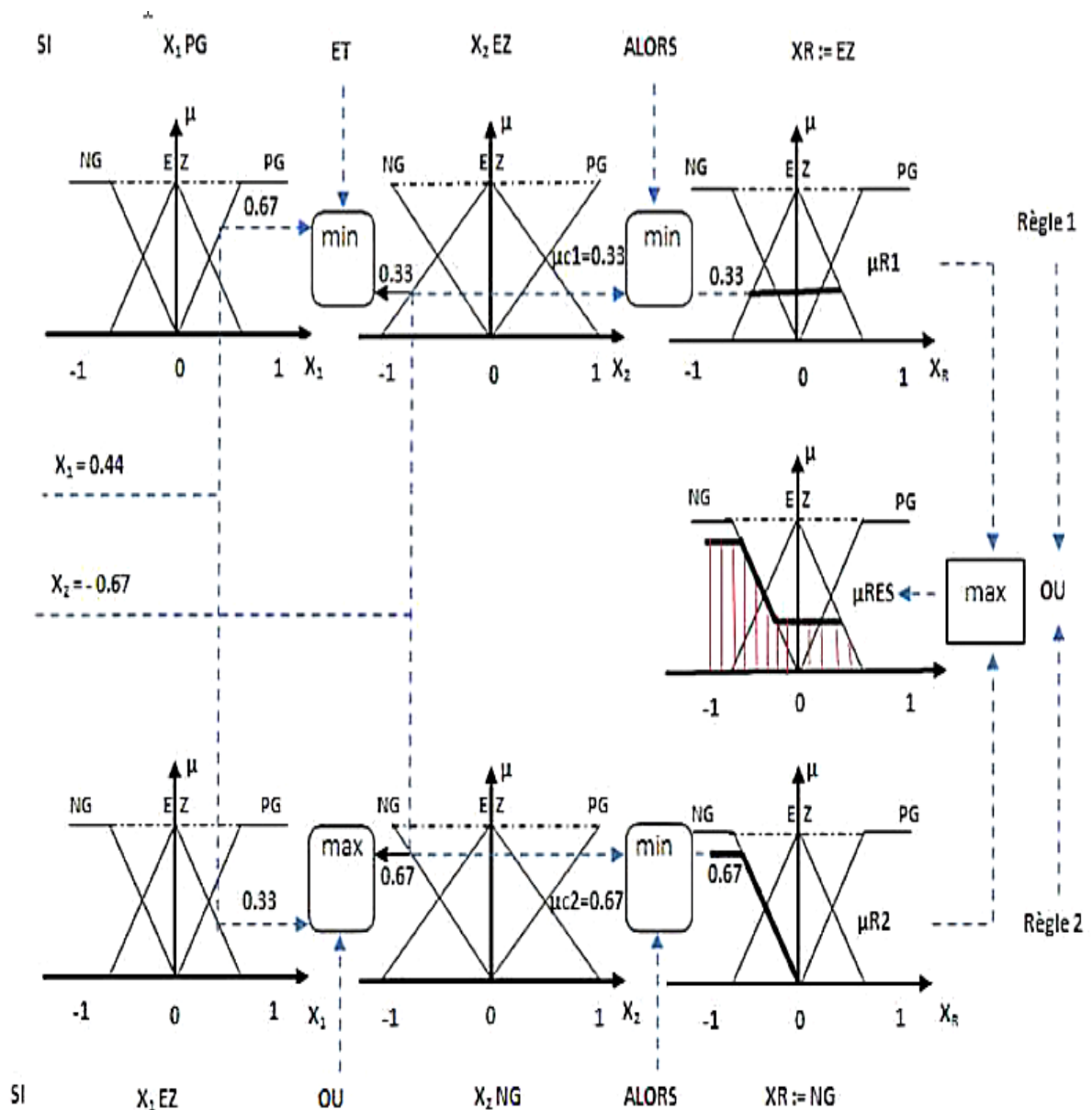


Figure 3.6: Exemple d'inférence Max-min

c) Méthode d'inférence Somme-Produit

Cette méthode est la plus utilisée 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.

Le résultat 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 réalisé la formation du produit, un exemple illustratif est donné dans la figure 3.7.

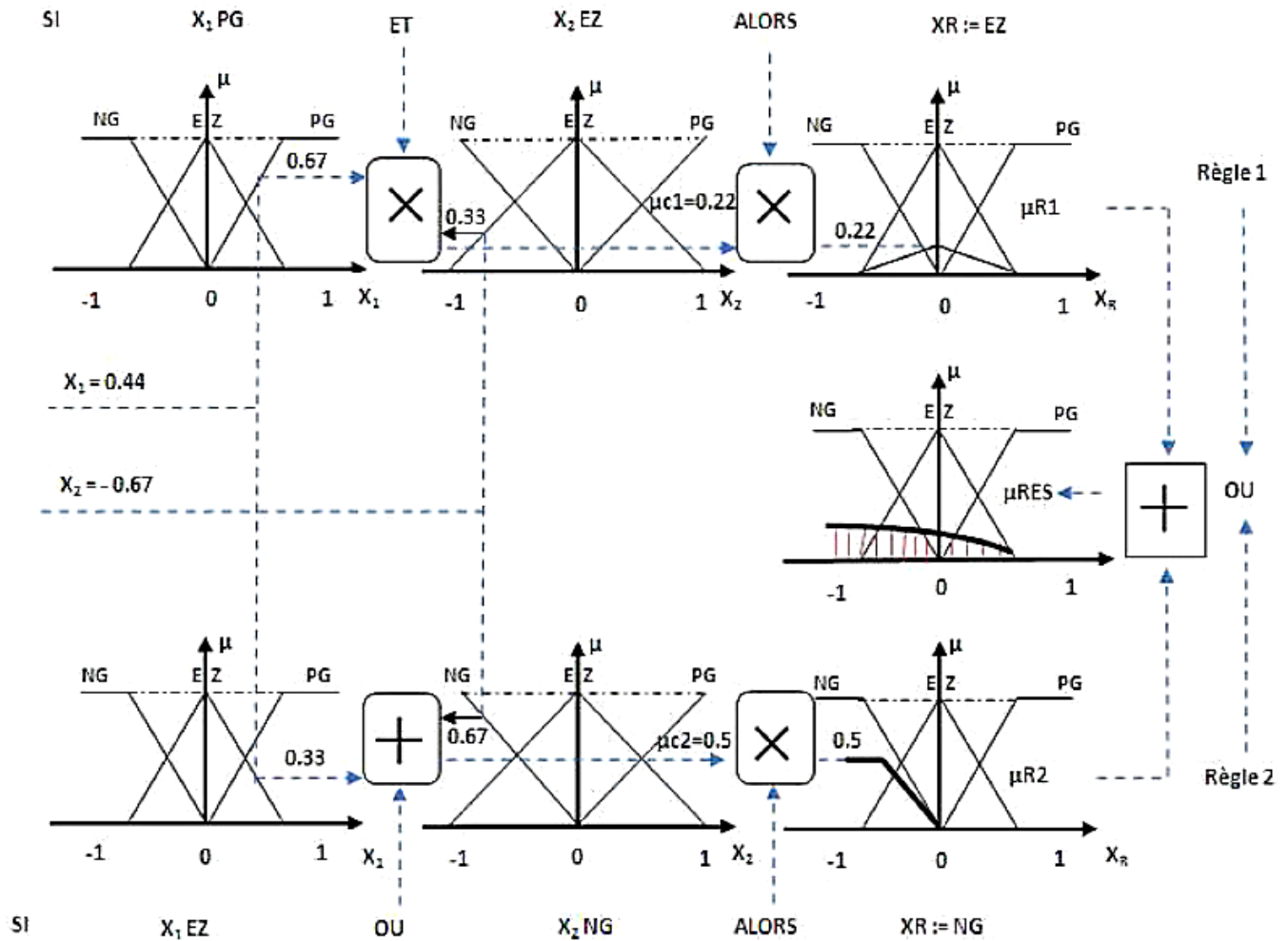


Figure 3.7 : Exemple d'inférence Somme-Produit

3.3.6. Base de connaissances

La base de connaissance contient l'ensemble des fonctions d'appartenance et des règles d'un système flou contenant l'expertise, la connaissance de l'opérateur, de l'expert, etc. Souvent les règles floues sont de la forme « Si-Alors », elles décrivent les 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

3.3.7. Défuzzification

Comme on l'a vu précédemment, les méthodes d'inférence fournissent une fonction d'appartenance résultante $\mu_{res}(x)$ pour la variable de sortie x . Il s'agit donc d'une information

floue. Les actionneurs actuels, utilisés dans les boucles de commande ne s'accompagnent pas de ce genre de décision, il convient de la transformer en une grandeur de commande précise : c'est le but de l'étape de défuzzification. Les méthodes les plus couramment utilisées sont : La méthode du centre de gravité et la méthode de la moyenne des maximums.

a) 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.

$$MM = \frac{\int x dx}{\int dx}$$

Ce qui représente la moyenne des valeurs^s de sorties les plus vraisemblables comme montrée en figure 3.8.

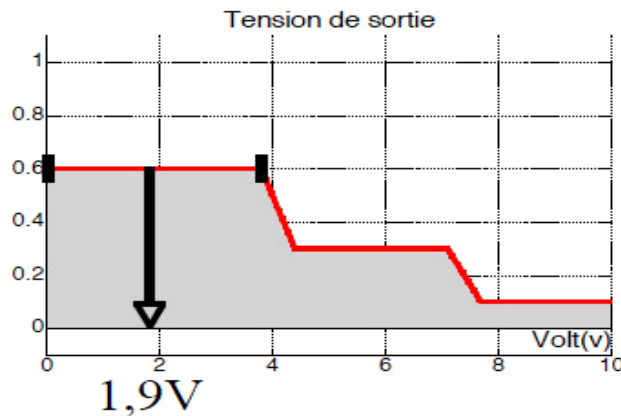


Figure 3.8 : Défuzzification par la méthode moyenne des maximums

b) Méthode du centre de gravité (CG)

La méthode la plus utilisée est de loin la méthode dite du centre de gravité. Il s'agit de calculer la position du centre de gravité de la fonction d'appartenance résultante, l'abscisse de ce centre de gravité devient la sortie du régulateur et donc la commande du système.

La figure 3.9 donne un exemple de défuzzification par la méthode du calcul du centre de gravité.

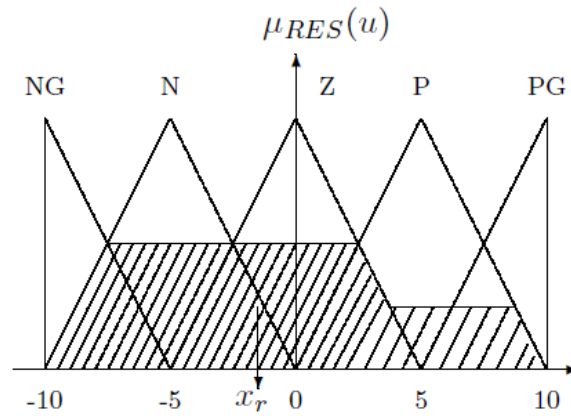


Figure 3.9 : Exemple de défuzzification par centre de gravité

$$x_r = \frac{\int_{-10}^{10} \mu_{RES}(x) x dx}{\int_{-10}^{10} \mu_{RES}(x) dx}$$

Avec μ_{res} étant la fonction d'appartenance résultante.

Ce centre de gravité est relativement facile à calculer lorsque la fonction d'appartenance résultante est constituée de morceaux de droites. Dans les autres cas, ce calcul peut être gourmand en temps de calcul et donc pénalisant lors de l'implantation sur un microprocesseur. C'est l'une des raisons qui font préférer les fonctions d'appartenance de formes trapézoïdales (triangulaires) et les opérateurs logiques de type min-max.

3.4. Contrôle par la logique floue

La logique floue est bien connue des automaticiens pour ses applications dans le contrôle-commande de procédés, appelé alors couramment « contrôle flou ». Tout comme un contrôleur (ou correcteur) classique, le contrôleur flou s'insère dans la boucle de régulation et calcule la commande à appliquer au procédé suivant une ou plusieurs consignes et une ou plusieurs mesures effectuées sur celui-ci. Les bases de règles floues sont intéressantes en commande car elles permettent [10] :

- de prendre en compte une expertise existante de nature qualitative.
- de prendre en compte des variables que l'on sait difficilement intégrer dans la boucle.
- d'améliorer le fonctionnement de contrôleurs classiques, par : autoréglage hors ligne ou en ligne des gains de ces contrôleurs, modification de leur sortie (feedforward) en

fonction d'événements qui ne peuvent pas être pris en compte par une technique classique.

Pour que l'utilisation de règles floues puisse être envisagée, il faut impérativement qu'il existe une expertise, un savoir-faire humain. Les bases de règles floues n'apportent pas de solution quand personne ne sait comment fonctionne le système ou ne peut le piloter manuellement. Lorsque ce savoir-faire existe et est transcribable sous forme de règles floues, la logique floue permet de le mettre facilement en œuvre et le fonctionnement est facilement compréhensible par l'utilisateur. Au-delà de l'automatisation, la logique floue permet une réelle capitalisation du savoir-faire du terrain. Lorsqu'une expertise humaine existe, l'utilisation de règles floues est envisageable, à plus forte raison lorsque des imperfections entachent la connaissance du système, lorsque celui-ci est très complexe et que sa modélisation est difficile ou lorsque la façon de l'aborder passe par une vue globale de certains de ses aspects. Les règles floues ne se substituent pas aux méthodes classiques d'automatique mais les complètent.

3.4.1. Différents types de régulateurs flous

Un domaine d'application de la logique floue qui devient fréquent est celui du réglage et de la commande des régulations industrielles. Cette méthode permet d'obtenir une loi de commande souvent efficace, sans devoir faire appel à des développements théoriques importants. Elle présente l'intérêt de prendre en compte les expériences acquises par les utilisateurs et opérateurs du processus à commander.

3.4.1. a. Le régulateur de type Mamdani

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 [9].

La forme générale des règles de Mamdani est :

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

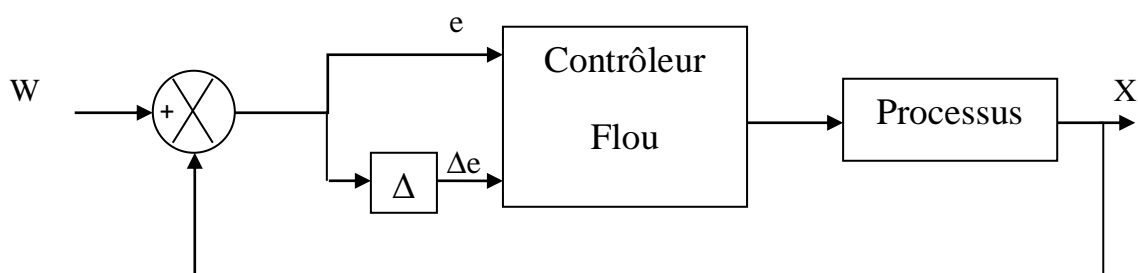


Figure 3.10 : Structure régulateur floue de type Mamdani

3.4.1.b. 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 :

$$\text{Si } x_1 = A1 \text{ ET } x_2 = A2 \text{ ALORS } y = z_1 x_1 + z_2 x_2 + z_0$$

3.5. Conclusion

Dans ce chapitre, nous avons présenté les concepts de la logique floue ainsi que la 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 comportement permet d'élaborer les règles de commande implantées dans le contrôleur floue.

Dans la suite des travaux, nous allons présenter une technique simplifiée de modélisation mathématique d'un correcteur floue ; cette méthode simplifiée sera une base méthodologique d'implémentation en logique programmé sous support API SIMATIC (S7-300).

CHAPITRE 04 : Modélisation du contrôleur flou PI\PD simplifié à base d'inférence larsen pour système TITO

4.1. Introduction

Révéler des modèles mathématiques de régulateurs flous est très important car il peut fournir des informations sur la façon dont un régulateur flou ressemble, comment il fonctionne, et comment il est similaire ou différent d'un régulateur classique. Une fois les modèles sont disponibles, les systèmes et théories de contrôle bien développées peuvent être utilisées pour analyser / synthétiser / mettre en œuvre des régulateurs flous.

Dans ce chapitre, un modèle mathématique d'un régulateur flou simplifié de type Mamdani deux entrées - deux sorties (TITO) proportionnel intégral (PI) ou d'une proportionnelle dérivé (PD) en utilisant l'inférence de Larsen est présenté. Le modèle est publié dans un article scientifique par deux chercheurs indien Arun et Mohan en 2016 qui ont présenté leur travail lors d'une conférence scientifique internationale en Inde [14].

Les composants de la commande floue comprennent deux ensembles flous de chaque variable d'entrée, cinq ensembles flous de chaque variable de sortie, cinq règles de commande linéaires, produit algébrique pour l'opérateur ET, la somme bornée pour l'opérateur OU, l'inférence par le produit Larsen, et le centre des sommes (CoS) pour la défuzzification. Chaque sortie du régulateur flou est indiquée comme étant la somme de deux régulateurs PI non linéaire (ou PD) avec des gains variables qui changent de façon continue avec les sorties du système TITO.

4.2. Les composants de base du régulateur TITO PI ou PD flou

Les relations d'entrée-sortie du régulateur PI TITO discret linéaire sous forme de vitesse sont données par :

$$\Delta u_1(k) = K_{P_1} \Delta e_1(k) + K_{I_1} e_1(k) + s_{21} [K_{P_2} \Delta e_2(k) + K_{I_2} e_2(k)] \quad (4.1)$$

$$\Delta u_1(k) = u_1(k-1) + \Delta u_1(k) \quad (4.2)$$

$$\Delta u_2(k) = s_{12} [K_{P_1} \Delta e_1(k) + K_{I_1} e_1(k)] + K_{P_2} \Delta e_2(k) + K_{I_2} e_2(k) \quad (4.3)$$

$$\Delta u_2(k) = u_2(k-1) + \Delta u_2(k) \quad (4.4)$$

De même, les relations entrée-sortie du régulateur PD discret linéaire sous forme de position sont données par :

$$\Delta u_1(k) = K_{P_1} e_1(k) + K_{D_1} \Delta e_1(k) + s_{21} [K_{P_2} e_2(k) + K_{D_2} \Delta e_2(k)] \quad (4.5)$$

$$\Delta u_2(k) = s_{12} [K_{P_1} e_1(k) + K_{D_1} \Delta e_1(k)] + K_{P_2} e_2(k) + K_{D_2} \Delta e_2(k) \quad (4.6)$$

Où

$$e_1(k) = r_1(k) - y_1(k) \quad (4.7)$$

$$\Delta e_1(k) = e_1(k) - e_1(k-1) \quad (4.8)$$

$$e_2(k) = r_2(k) - y_2(k) \quad (4.9)$$

$$\Delta e_2(k) = e_2(k) - e_2(k-1) \quad (4.10)$$

K_{P_1}, K_{P_2} : Gains proportionnels.

K_{I_1}, K_{I_2} : Gains intégrales.

K_{D_1}, K_{D_2} : Gains dérivés.

r_1, r_2 : Consignes.

u_1, u_2 : Commande du régulateur TITO.

y_1, y_2 : Sorties du system TITO.

s_{12}, s_{21} : Signes (Négative ou Positive).

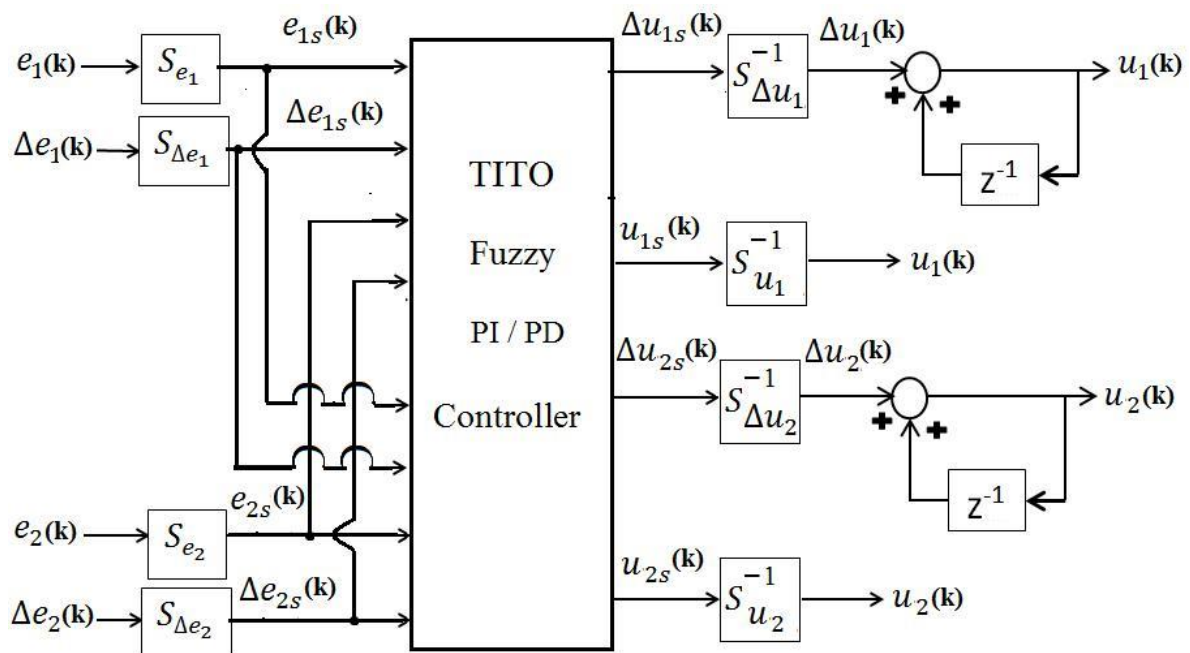


Figure 4.1 : Schéma bloc d'un régulateur TITO PI/PD flou typique

Les signes de s_{12} et s_{21} dépendent des interactions entre les deux boucles du système TITO.

La figure 4.1 montre le schéma bloc d'un régulateur TITO PI/PD flou typique .dans cette forme les paramètres S_{e_1} , $S_{\Delta e_1}$, S_{e_2} et $S_{\Delta e_2}$ sont les facteurs de normalisation des entrées, $S_{\Delta u_1}^{-1}$, $S_{\Delta u_2}^{-1}$, (pour le PI), $S_{u_1}^{-1}$, $S_{u_2}^{-1}$ (pour le PD) sont les facteurs de normalisation des sorties, $e_{1s}(k)$, $\Delta e_{1s}(k)$, $e_{2s}(k)$ et $\Delta e_{2s}(k)$ sont les entrées normalisées et $\Delta u_{1s}(k)$, $\Delta u_{2s}(k)$ (pour le PI) , $u_{1s}(k)$, $u_{2s}(k)$ (pour le PD) représentent les sorties de commande .

La nature des relations d'entrée-sortie du régulateur PI / PD floue dépend des composants du régulateur flou. Les composants de base du régulateur à logique floue sont la fuzzification, la base de règles, le moteur d'inférence, et la defuzzification qui sont décrits dans ce qui suit.

4.2.1. Fuzzification

Les fonctions d'appartenance représentée dans la figure 4.2, sont considérées pour les variables d'entrée normalisée.

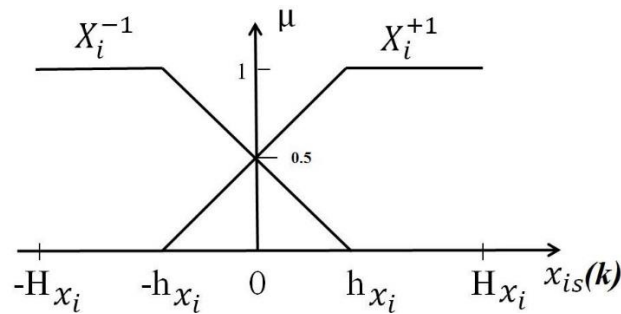


Figure 4.2 : Fonction d'appartenance des entrés

La description mathématique des fonctions d'appartenance est donnée par :

$$\mu_{X_i^{-1}}(x_{is}(k)) = \begin{cases} 1 & \text{Si } -H_{xi} \leq x_{is}(k) \leq -h_{xi} \\ \frac{-x_{is}(k) + h_{xi}}{2h_{xi}} & \text{Si } -h_{xi} \leq x_{is}(k) \leq h_{xi} \\ 0 & \text{Si } h_{xi} \leq x_{is}(k) \leq H_{xi} \end{cases} \quad (4.11)$$

$$\mu_{X_i^{+1}}(x_{is}(k)) = \begin{cases} 0 & \text{Si } -H_{xi} \leq x_{is}(k) \leq -h_{xi} \\ \frac{x_{is}(k) + h_{xi}}{2h_{xi}} & \text{Si } -h_{xi} \leq x_{is}(k) \leq h_{xi} \\ 1 & \text{Si } h_{xi} \leq x_{is}(k) \leq H_{xi} \end{cases} \quad (4.12)$$

Ou $x_{is}(k)$ pour $i=1, 2, 3, 4$ représente les entées normalisées $e_{1s}(k), \Delta e_{1s}(k), e_{2s}(k)$ et $\Delta e_{2s}(k)$ respectivement . On remarque que $\mu_{x_i^{-1}}(x_{is}(k)) + \mu_{x_i^{+1}}(x_{is}(k)) = 1, \forall x_{is}(k)$. les fonctions d'appartenance des sorties normalisées $\Delta u_{js}(k)$ pour $j = 1, 2$ sont montrées en figure 4. 3.

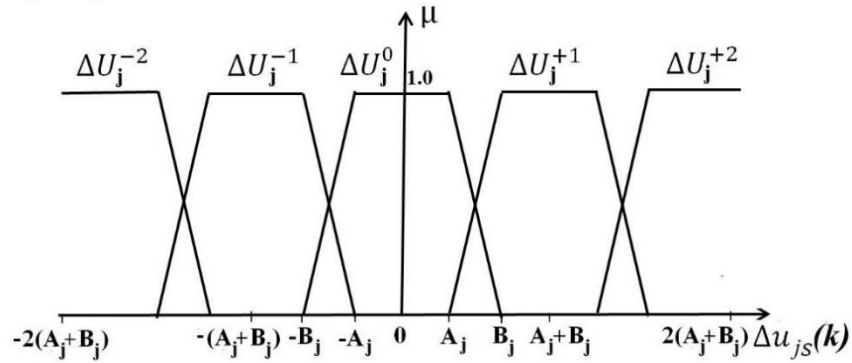


Figure 4.3 : Fonction d'appartenance des sorties

4.2.2. La base de règles

Dans cette approche la base de règles est constitué de 5 ensemble flou pour les sorties, la base de règle est construite autour de ces 5 règles. Pour cette étude, la forme utilisée pour les règles de commande est la suivante :

$$\begin{aligned} \text{SI } e_{1s}(k) = E_1^{I_1} \text{ ET } \Delta e_{1s}(k) = \Delta E_1^{I_2} \text{ ET } e_{2s}(k) = E_2^{I_3} \text{ ET } \Delta e_{2s}(k) = \Delta E_2^{I_4} \\ \text{ALORS } \Delta u_{1s}(k) = \Delta U_1^{J_1}, \Delta u_{2s}(k) = \Delta U_2^{J_2} \end{aligned} \quad (4.13)$$

Ou $J_1 = \frac{I_1 + I_2}{2} + s_{21} \frac{I_3 + I_4}{2}$ et $J_2 = s_{12} \frac{I_1 + I_2}{2} + \frac{I_3 + I_4}{2}$, I_1, I_2, I_3 et I_4 sont les indices (-1 ou +1) des ensembles flou des entrées, J_1 et J_2 sont les indices (-2,-1, 0,+1 ou +2) des ensembles flou des sorties avec les fonctions d'appartenance $\mu_{\Delta U_j^{j_n}}$ pour $j,n=1$ et 2.

Selon les valeurs de s_{21} (-1 ou +1) et s_{12} (-1 ou +1) quatre bases de règles sont possible :

Cas 1 : $s_{21} = -1$ et $s_{12} = -1$

La base de règles dans ce cas est donnée dans le tableau 4.1.

Cas 2 : $s_{21} = +1$ et $s_{12} = +1$

La base de règles dans ce cas est donnée dans le tableau 4.2.

Cas 3 : $s_{21} = -1$ et $s_{12} = +1$

Dans ce cas, la base de règles pour J_1 et J_2 sont respectivement tirées du tableau 4.1 et 4.2.

Cas 4 : $s_{21} = +1$ et $s_{12} = -1$

Dans ce cas, la base de règles pour J_1 et J_2 sont respectivement tirées du tableau 4.2 et 4.1.

Le produit algébrique est utilisé pour évaluer le ET dans la partie antécédente de la règle de commande (4.13). Par exemple, de la règle R_3 du tableau 1, nous avons

$$\mu_{R_3}(\Delta u_{js}) = \mu_{E_1^{-1}}(e_{1s}) \cdot \mu_{\Delta E_1^{-1}}(\Delta e_{1s}) \cdot \mu_{E_2^{+1}}(e_{2s}) \cdot \mu_{\Delta E_1^{+1}}(\Delta e_{2s}) \quad (4.14)$$

Règle	I_1	I_2	I_3	I_4	J_1	J_2
R_1	(-1	-1	-1	-1) OR	0	0
	(-1	1	-1	1) OR		
	(-1	1	1	-1) OR		
	(1	-1	-1	1) OR		
	(1	-1	1	-1) OR		
R_2	(1	1	1	1)	-1	1
	(-1	-1	-1	1) OR		
	(-1	-1	1	-1) OR		
	(-1	1	1	1) OR		
R_3	(1	-1	1	1)	-2	2
	-1	-1	1	1)		
R_4	(-1	1	-1	-1) OR	1	-1
	(1	-1	-1	-1) OR		
	(1	1	-1	1) OR		
	(1	1	1	-1)		
R_5	1	1	-1	-1	2	-2

Tableau 4.1 : La base de règles correspondante au 1^{er} cas

Les parties antécédentes de certaines des règles dans les tableaux 1 et 2 sont combinés avec un opérateur OU. Dans notre cas l'opérateur OU est utilisé comme somme bornée qui est défini comme suit :

$$s(\mu_A(x), \mu_B(y)) = \min \{1, \mu_A(x) + \mu_B(y)\} \quad (4.15)$$

En utilisant la définition ci-dessus et la règle R_2 dans le tableau 1, on peut écrire

$$s(\mu_{R_{21}}(\Delta u_{js}), \mu_{R_{22}}(\Delta u_{js}), \mu_{R_{23}}(\Delta u_{js}), \mu_{R_{24}}(\Delta u_{js})) = \min \{1, \mu_{R_2}(\Delta u_{js})\} \quad (4.16)$$

Où

$$\mu_{R_2}(\Delta u_{js}) = \sum_{i=1}^4 \mu_{R_{2i}}(\Delta u_{js}) \quad (4.17)$$

Et

$$\begin{aligned} \mu_{R_{21}}(\Delta u_{js}) &= \mu_{E_1^{-1}}(e_{1s}) \cdot \mu_{\Delta E_1^{-1}}(\Delta e_{1s}) \cdot \mu_{E_2^{-1}}(e_{2s}) \cdot \mu_{\Delta E_1^{+1}}(\Delta e_{2s}) \\ \mu_{R_{22}}(\Delta u_{js}) &= \mu_{E_1^{-1}}(e_{1s}) \cdot \mu_{\Delta E_1^{-1}}(\Delta e_{1s}) \cdot \mu_{E_2^{+1}}(e_{2s}) \cdot \mu_{\Delta E_1^{-1}}(\Delta e_{2s}) \\ \mu_{R_{23}}(\Delta u_{js}) &= \mu_{E_1^{-1}}(e_{1s}) \cdot \mu_{\Delta E_1^{+1}}(\Delta e_{1s}) \cdot \mu_{E_2^{+1}}(e_{2s}) \cdot \mu_{\Delta E_1^{+1}}(\Delta e_{2s}) \\ \mu_{R_{24}}(\Delta u_{js}) &= \mu_{E_1^{+1}}(e_{1s}) \cdot \mu_{\Delta E_1^{-1}}(\Delta e_{1s}) \cdot \mu_{E_2^{+1}}(e_{2s}) \cdot \mu_{\Delta E_1^{+1}}(\Delta e_{2s}) \end{aligned}$$

Règle	I_1	I_2	I_3	I_4	J_1	J_2
R_1	-1	-1	-1	-1	-2	-2
R_2	(1	-1	-1	-1) OR	-1	-1
	(-1	1	-1	-1) OR		
	(-1	-1	1	-1) OR		
	(-1	-1	-1	1)		
R_3	(-1	-1	1	1) OR	0	0
	(-1	1	-1	1) OR		
	(-1	1	1	-1) OR		
	(1	-1	-1	1) OR		
	(1	-1	1	-1) OR		
R_4	(1	1	-1	-1)	1	1
	(-1	1	1	1) OR		
	(1	-1	1	1) OR		
	(1	1	-1	1) OR		
R_5	1	1	1	1	2	2

Tableau 4.2 : La base de règles correspondante au 2^{ème} cas.

4.2.3 .Moteur d'inférence

Les expression de μ_{R_l} pour $l=1,2,\dots,5$ sont utilisées pour modifier les fonctions d'appartenance des ensembles flous ΔU_j^{-2} , ΔU_j^{-1} , ΔU_j^0 , ΔU_j^{+1} , et ΔU_j^{+2} . Les fonctions d'appartenance des ensembles flous modifié (inférée) $\Delta U_j^{\sigma^{-2}}$, $\Delta U_j^{\sigma^{-1}}$, $\Delta U_j^{\sigma^0}$, $\Delta U_j^{\sigma^{+1}}$ et $\Delta U_j^{\sigma^{+2}}$, obtenus par le produit de Larsen sont illustrés par la surface hachuré sur la figure 4.4 .

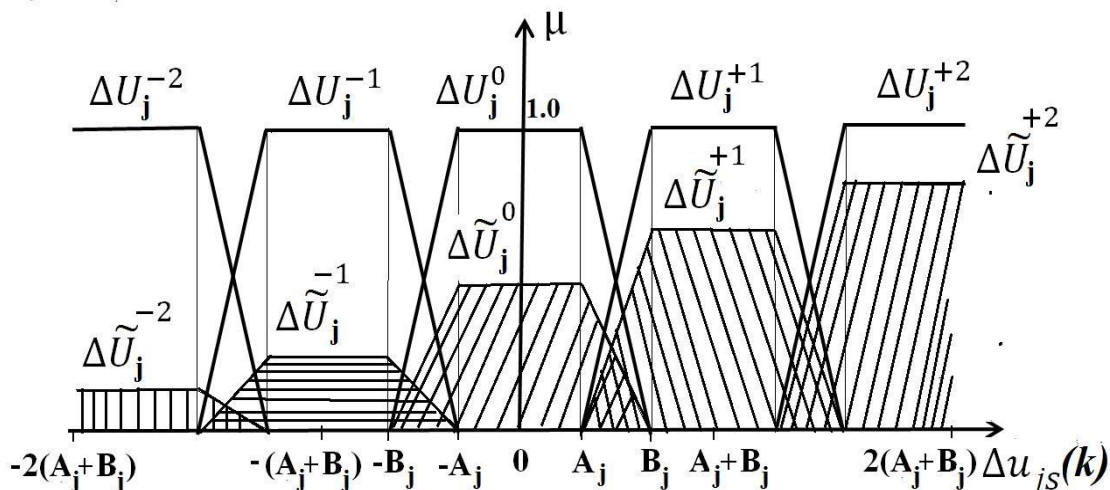


Figure 4.4 : Fonctions d'appartenance résultantes du produit de Larsen

Afin de localiser un point dans l'espace à quatre dimensions des entrée normalisée définie par $-H_{e1} \leq e_{1s}(k) \leq H_{e1}$, $-H_{\Delta e1} \leq \Delta e_{1s}(k) \leq H_{\Delta e1}$, $-H_{e2} \leq e_{2s}(k) \leq H_{e2}$ et $-H_{\Delta e2} \leq \Delta e_{2s}(k) \leq H_{\Delta e2}$, l'espace complet est partitionné en 81 cellules et chaque cellule et notée par la paire (p, q) où p et q viennent de (a) et (b) de la figure 4.5 respectivement .

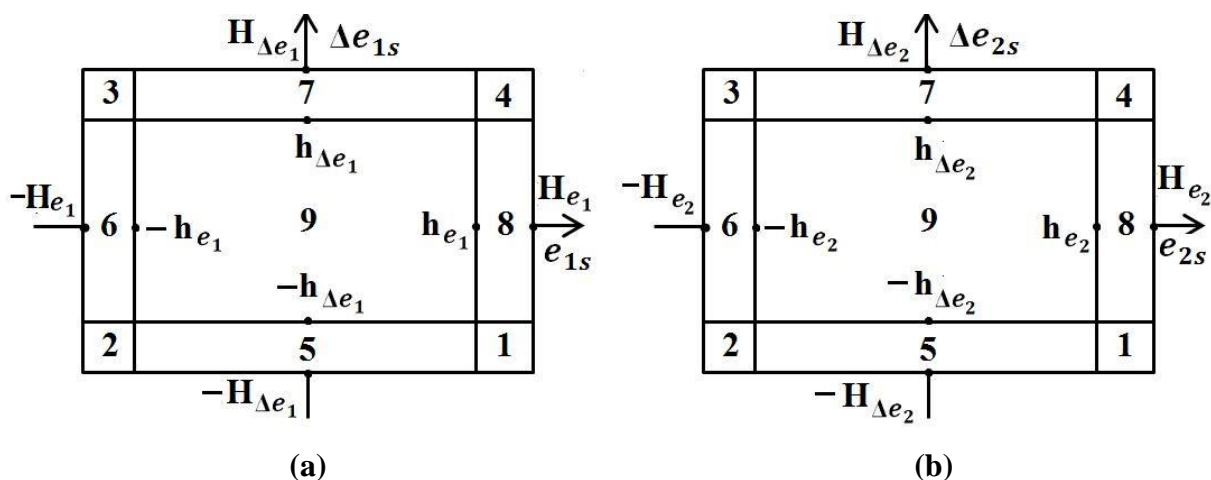


Figure 4.5 : Partition de l'espace des entrés

Lemme 1 :

$$\sum_{l=-2}^2 \mu_l = 1 \tag{4.18}$$

Démonstration : On prend en considération le tableau 4.1 et $\Delta u_{1s}(k)$ pour prouver (4.18).

Depuis le tableau 4.1 nous avons :

$$\mu_{-2} = \mu_{R_3}$$

$$\mu_{-1} = \mu_{R_2} = \mu_{R_{21}} + \mu_{R_{22}} + \mu_{R_{23}} + \mu_{R_{24}}$$

$$\mu_0 = \mu_{R_1} = \mu_{R_{11}} + \mu_{R_{12}} + \mu_{R_{13}} + \mu_{R_{14}} + \mu_{R_{15}} + \mu_{R_{16}}$$

$$\mu_{+1} = \mu_{R_4} = \mu_{R_{41}} + \mu_{R_{42}} + \mu_{R_{43}} + \mu_{R_{44}}$$

$$\mu_{+2} = \mu_{R_5}$$

Remplaçons les expressions de $\mu_{R_{11}}, \dots, \mu_{R_{16}}, \mu_{R_{21}}, \dots, \mu_{R_{24}}, \mu_{R_3}, \mu_{R_{41}}, \dots, \mu_{R_{44}}$ et μ_{R_5} dans la partie gauche de l'équation (18), après simplification du résultat la démonstration est réalisée.

En vue de ce Lemme, l'équation (16) peut s'écrire comme suit

$$\min \{1, \mu_{R_2}(\Delta u_{js})\} = \mu_{R_2}(\Delta u_{js}) \quad (4.19)$$

4.2.4 Defuzzification

Selon la méthode de CoS (centre des sommes) [13], la valeur nette de la sortie normalisé est donnée par :

$$\Delta u_{js}^*(k) = \frac{\sum_{l=1}^5 A(\mu_{R_l})C(\mu_{R_l})}{\sum_{l=1}^5 A(\mu_{R_l})} \quad (4.20)$$

Où $A(\mu_{R_l})$ et $C(\mu_{R_l})$ sont respectivement la surface et le centre de l^éme fonction

d'appartenance inférée de la j^éme sortie. L'expression des surfaces et des centres des fonctions d'appartenance inférée des sorties sont :

$$A(\mu_{-2}) = 0.5(A_j + B_j)\mu_{-2}; A(\mu_{+2}) = 0.5(A_j + B_j)\mu_{+2}$$

$$A(\mu_{-1}) = (A_j + B_j)\mu_{-1}; A(\mu_{+1}) = (A_j + B_j)\mu_{+1}$$

$$A(\mu_0) = (A_j + B_j)\mu_0$$

$$C(\mu_{-2}) = \frac{-5(A_j + B_j)^2 - A_j B_j}{3(A_j + B_j)}; C(\mu_{+2}) = \frac{5(A_j + B_j)^2 + A_j B_j}{3(A_j + B_j)}$$

$$C(\mu_{-1}) = -(A_j + B_j); C(\mu_{+1}) = (A_j + B_j)$$

$$C(\mu_0) = 0$$

Lemme 2 :

$$\sum_{l=-2}^2 A(\mu_l) = (A_j + B_j)[1 - 0.5(\mu_{-2} + \mu_{+2})] \quad (4.21)$$

Démonstration : il est simple de démontrer ceci comme suit

$$\begin{aligned}\sum_{l=-2}^2 A(\mu_l) &= (A_j + B_j)[0.5\mu_{-2} + \mu_{-1} + \mu_0 + \mu_{+1} + 0.5\mu_{+2}] \\ &= (A_j + B_j)\left[\sum_{l=-2}^2 \mu_l - 0.5(\mu_{-2} + \mu_{+2})\right] \\ &= (A_j + B_j)[1 - 0.5(\mu_{-2} + \mu_{+2})]\end{aligned}$$

Ceci complète la démonstration. Ce Lemme joue un rôle important pour démontrer que le modèle mathématique du régulateur TITO PI/PD Flou simplifié est non linéaire.

4.3. Modèle mathématique du régulateur TITO PI/PD Flou Simplifié

Ici, les relations mathématique entre les entrées-sorties du régulateur sont obtenues en utilisant le produit algébrique comme opérateur ET, somme bornée pour le OU, produit de Larsen pour l'inférence et la méthode CoS pour la défuzzification.

4.3.1. Commandes dans la cellule (1,1) et (3,3)

$$\Delta u_{1s}(k) = 0$$

Et

$$\Delta u_{2s}(k) = 0$$

4.3.2. Commandes dans la cellule (1,5)

$$\Delta u_{1s}(k) = \frac{s_{21}(A_1 + B_1)(e_{2s}(k) - h_{e_2})}{2h_{e_2}}$$

Et

$$\Delta u_{2s}(k) = \frac{(A_2 + B_2)(e_{2s}(k) - h_{e_2})}{2h_{e_2}}$$

4.3.3. Commandes dans la cellule (1,9)

$$\Delta u_{1s}(k) = \frac{s_{21}(A_1 + B_1)\{h_{\Delta_{e_2}} e_{2s}(k) + h_{e_2} \Delta_{e_{2s}}(k)\}}{2h_{e_2} h_{\Delta_{e_2}}}$$

Et

$$\Delta u_{2s}(k) = \frac{s_{21}(A_2 + B_2)\{h_{\Delta_{e_2}} e_{2s}(k) + h_{e_2} \Delta_{e_{2s}}(k)\}}{2h_{e_2} h_{\Delta_{e_2}}}$$

4.3.4. Commandes dans la cellule (5,1)

$$\Delta u_{1s}(k) = \frac{(A_1 + B_1)(e_{1s}(k) - h_{e_1})}{2h_{e_1}}$$

Et

$$\Delta u_{2s}(k) = \frac{s_{12}(A_2 + B_2)(e_{1s}(k) - h_{e_1})}{2h_{e_1}}$$

4.3.5. Commandes dans la cellule (5,5)

$$\Delta u_{1s}(k) = \frac{N_1}{D_1}$$

Et

$$\Delta u_{2s}(k) = \frac{N_2}{D_2}$$

Où

$$\begin{aligned} N_1 = & \{17(A_1 + B_1)^2 + A_1 B_1\} \{h_{\Delta e_1} h_{e_2} h_{\Delta e_2} e_{1s}(k) + s_{21} h_{e_1} h_{\Delta e_1} h_{\Delta e_2} e_{2s}(k) \\ & - (1 + s_{21}) h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2}\} \\ & + \{-7(A_1 + B_1)^2 + A_1 B_1\} \{s_{21} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} e_{1s}(k) \\ & - (1 + s_{21}) h_{\Delta e_1} h_{\Delta e_2} e_{1s}(k) + h_{e_1} h_{\Delta e_1} h_{\Delta e_2} e_{2s}(k)\} \end{aligned}$$

$$D_1 = 3(A_1 + B_1) \{(15 - s_{21}) h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} + (1 + s_{21}) h_{\Delta e_1} h_{\Delta e_2} \{h_{e_2} e_{1s}(k) + h_{e_1} e_{2s}(k) - e_{1s}(k) e_{2s}(k)\}\}$$

$$\begin{aligned} N_2 = & \{17(A_2 + B_2)^2 + A_2 B_2\} \{s_{12} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} e_{1s}(k) + h_{e_1} h_{\Delta e_1} h_{\Delta e_2} e_{2s}(k) \\ & - (1 + s_{12}) h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2}\} \\ & + \{-7(A_2 + B_2)^2 + A_2 B_2\} \{s_{21} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} e_{1s}(k) \\ & - (1 + s_{12}) h_{\Delta e_1} h_{\Delta e_2} e_{1s}(k) e_{2s}(k) + s_{12} h_{e_1} h_{\Delta e_1} h_{\Delta e_2} e_{2s}(k)\} \end{aligned}$$

Et

$$D_2 = 3(A_2 + B_2) \{(15 - s_{12}) h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} + (1 + s_{12}) h_{\Delta e_1} h_{\Delta e_2} \{h_{e_2} e_{1s}(k) + h_{e_1} e_{2s}(k) - e_{1s}(k) e_{2s}(k)\}\}$$

4.3.6. Commandes dans la cellule (5,9)

$$\Delta u_{1s}(k) = \frac{N_3}{D_3}$$

Et

$$\Delta u_{2s}(k) = \frac{N_4}{D_4}$$

Où

$$\begin{aligned} N_3 = & \{17(A_1 + B_1)^2 + A_1 B_1\} \{h_{\Delta e_1} h_{e_2} h_{\Delta e_2} e_{1s}(k) - h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} \\ & + s_{21} h_{e_1} h_{\Delta e_1} \{h_{\Delta e_2} e_{2s}(k) + h_{e_2} \Delta e_{2s}(k)\}\} \\ & + \{-7(A_1 + B_1)^2 + A_1 B_1\} \{h_{\Delta e_1} e_{2s}(k) \Delta e_{2s}(k) (e_{1s}(k) - h_{e_1}) \\ & - s_{21} h_{\Delta e_1} e_{1s}(k) \{h_{\Delta e_2} e_{2s}(k) + h_{e_2} \Delta e_{2s}(k)\}\} \end{aligned}$$

$$\begin{aligned}
 D_3 = & 3(A_1 + B_1) \{ 15h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} + h_{\Delta e_1} e_{1s}(k) e_{2s}(k) \Delta e_{2s}(k) \\
 & + h_{\Delta e_1} h_{e_2} h_{\Delta e_2} e_{1s}(k) - h_{e_1} h_{\Delta e_1} e_{2s}(k) \Delta e_{2s}(k) \\
 -s_{21} \{ & h_{\Delta e_1} h_{\Delta e_2} e_{1s}(k) e_{2s}(k) - h_{e_1} h_{\Delta e_1} h_{\Delta e_2} e_{2s}(k) \\
 & + h_{\Delta e_1} h_{e_2} e_{1s}(k) \Delta e_{2s}(k) - h_{e_1} h_{\Delta e_1} h_{e_2} \Delta e_{2s}(k) \} \} \\
 N_4 = & \{ 17(A_2 + B_2)^2 + A_2 B_2 \} \{ s_{12} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} e_{1s}(k) - s_{12} h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} \\
 & + h_{e_1} h_{\Delta e_1} \{ h_{\Delta e_2} e_{2s}(k) + h_{e_2} \Delta e_{2s}(k) \} \} \\
 + \{ & -7(A_2 + B_2)^2 + A_2 B_2 \} \{ s_{12} h_{\Delta e_1} e_{2s}(k) \Delta e_{2s}(k) (e_{1s}(k) - h_{e_1}) \\
 & - h_{\Delta e_1} e_{1s}(k) \{ h_{\Delta e_2} e_{2s}(k) + h_{e_2} \Delta e_{2s}(k) \} \}
 \end{aligned}$$

Et

$$\begin{aligned}
 D_4 = & 3(A_2 + B_2) \{ 15h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} + h_{\Delta e_1} e_{1s}(k) e_{2s}(k) \Delta e_{2s}(k) \\
 & + h_{\Delta e_1} h_{e_2} h_{\Delta e_2} e_{1s}(k) - h_{e_1} h_{\Delta e_1} e_{2s}(k) \Delta e_{2s}(k) \\
 -s_{12} \{ & h_{\Delta e_1} h_{\Delta e_2} e_{1s}(k) e_{2s}(k) - h_{e_1} h_{\Delta e_1} h_{\Delta e_2} e_{2s}(k) \\
 & + h_{\Delta e_1} h_{e_2} e_{1s}(k) \Delta e_{2s}(k) - h_{e_1} h_{\Delta e_1} h_{e_2} \Delta e_{2s}(k) \} \}
 \end{aligned}$$

4.3.7. Commandes dans la cellule (9,1)

$$\Delta u_{1s} = \frac{(A_1 + B_1) \{ h_{\Delta e_1} e_{1s}(k) h_{e_1} \Delta e_{1s}(k) \}}{2h_{e_1} h_{\Delta e_1}}$$

Et

$$\Delta u_{2s} = \frac{s_{12}(A_2 + B_2) \{ h_{\Delta e_1} e_{1s}(k) h_{e_1} \Delta e_{1s}(k) \}}{2h_{e_1} h_{\Delta e_1}}$$

4.3.8. Commandes dans la cellule (9,5)

$$\Delta u_{1s}(k) = \frac{N_5}{D_5}$$

Et

$$\Delta u_{2s}(k) = \frac{N_6}{D_6}$$

Où

$$\begin{aligned}
 N_5 = & \{ 17(A_1 + B_1)^2 + A_1 B_1 \} \{ h_{e_2} h_{\Delta e_2} \{ h_{\Delta e_1} e_{1s}(k) + h_{e_1} \Delta e_{1s}(k) \} \\
 & - s_{21} h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} + s_{21} h_{e_1} h_{\Delta e_1} h_{\Delta e_2} \Delta e_{2s}(k) \} \} \\
 + \{ & -7(A_1 + B_1)^2 + A_1 B_1 \} \{ s_{21} h_{\Delta e_2} e_{1s}(k) \Delta e_{1s}(k) (e_{2s}(k) - h_{e_2}) \\
 & - h_{\Delta e_2} e_{2s}(k) \{ h_{\Delta e_1} e_{1s}(k) + h_{e_1} \Delta e_{1s}(k) \} \}
 \end{aligned}$$

$$\begin{aligned}
 D_5 = & 3(A_1 + B_1) \{ 15h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} + h_{\Delta e_2} e_{1s}(k) \Delta e_{1s}(k) e_{2s}(k) \\
 & - h_{e_2} h_{\Delta e_2} e_{1s}(k) \Delta e_{1s}(k) - h_{e_1} h_{\Delta e_1} h_{\Delta e_2} e_{2s}(k) \\
 -s_{12} \{ & h_{\Delta e_1} h_{\Delta e_2} e_{1s}(k) e_{2s}(k) - h_{e_1} h_{\Delta e_1} h_{\Delta e_2} e_{2s}(k) \\
 & - h_{\Delta e_1} h_{e_2} h_{\Delta e_2} e_{1s}(k) - h_{e_1} h_{e_2} h_{\Delta e_2} \Delta e_{1s}(k) \} \} \\
 N_6 = & \{ 17(A_2 + B_2)^2 + A_2 B_2 \} \{ s_{12} h_{e_2} h_{\Delta e_2} \{ h_{\Delta e_1} e_{1s}(k) + h_{e_1} \Delta e_{1s}(k) \} \\
 & - h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} + h_{e_1} h_{\Delta e_1} h_{\Delta e_2} e_{2s}(k) \} \} \\
 + \{ & -7(A_2 + B_2)^2 + A_2 B_2 \} \{ h_{\Delta e_2} e_{1s}(k) \Delta e_{1s}(k) (e_{2s}(k) - h_{e_2}) \\
 & - s_{12} h_{\Delta e_2} e_{2s}(k) \{ h_{\Delta e_1} e_{1s}(k) + h_{e_1} \Delta e_{1s}(k) \} \}
 \end{aligned}$$

Et

$$\begin{aligned}
 D_6 = & 3(A_2 + B_2) \{ 15h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} + h_{\Delta e_2} e_{1s}(k) \Delta e_{1s}(k) e_{2s}(k) \\
 & - h_{e_2} h_{\Delta e_2} e_{1s}(k) \Delta e_{1s}(k) + h_{e_1} h_{\Delta e_1} h_{\Delta e_2} e_{2s}(k) \\
 -s_{12} \{ & h_{\Delta e_1} h_{\Delta e_2} e_{1s}(k) e_{2s}(k) + h_{e_1} h_{\Delta e_2} h_{\Delta e_1} e_{2s}(k) \\
 & - h_{\Delta e_1} h_{e_2} h_{\Delta e_2} e_{1s}(k) - h_{e_1} h_{e_2} h_{\Delta e_2} \Delta e_{1s}(k) \} \}
 \end{aligned}$$

4.3.9. Commandes dans la cellule (9,9)

$$\Delta u_{1s} = \gamma_{e_{11}} e_{1s}(k) + \gamma_{\Delta e_{11}} \Delta e_{1s}(k) + s_{21} \{ \gamma_{e_{12}} e_{2s} + \gamma_{\Delta e_{12}} \Delta e_{2s}(k) \} \quad (4.22)$$

Où

$$\begin{aligned}
 \gamma_{e_{11}} = & \frac{\{ 17(A_1 + B_1)^2 + A_1 B_1 \} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} + \{ -7(A_1 + B_1)^2 + A_1 B_1 \} h_{\Delta e_1} e_{2s}(k) \Delta e_{2s}}{D(e_{1s}, \Delta e_{1s}, s_{21} e_{2s}, s_{21} \Delta e_{2s})} \\
 \gamma_{\Delta e_{11}} = & \frac{\{ 17(A_1 + B_1)^2 + A_1 B_1 \} h_{e_1} h_{e_2} h_{\Delta e_2} + \{ -7(A_1 + B_1)^2 + A_1 B_1 \} h_{e_1} e_{2s}(k) \Delta e_{2s}}{D(e_{1s}, \Delta e_{1s}, s_{21} e_{2s}, s_{21} \Delta e_{2s})} \\
 \gamma_{e_{12}} = & \frac{\{ 17(A_1 + B_1)^2 + A_1 B_1 \} h_{e_1} h_{\Delta e_1} h_{\Delta e_2} + \{ -7(A_1 + B_1)^2 + A_1 B_1 \} h_{\Delta e_2} e_{1s}(k) \Delta e_{1s}}{D(e_{1s}, \Delta e_{1s}, s_{21} e_{2s}, s_{21} \Delta e_{2s})} \\
 \gamma_{\Delta e_{12}} = & \frac{\{ 17(A_1 + B_1)^2 + A_1 B_1 \} h_{e_1} h_{\Delta e_1} h_{e_2} + \{ -7(A_1 + B_1)^2 + A_1 B_1 \} h_{e_2} e_{1s}(k) \Delta e_{1s}}{D(e_{1s}, \Delta e_{1s}, s_{21} e_{2s}, s_{21} \Delta e_{2s})}
 \end{aligned}$$

Et

$$\begin{aligned}
 D(e_{1s}, \Delta e_{1s}, s_{21} e_{2s}, s_{21} \Delta e_{2s}) = & 3(A_1 + B_1) \{ 15h_{e_1} h_{\Delta e_1} h_{e_2} h_{\Delta e_2} - e_{1s} \Delta e_{1s} e_{2s} \Delta e_{2s} \\
 & - h_{e_2} h_{\Delta e_2} e_{1s} \Delta e_{1s} - h_{e_1} h_{\Delta e_1} e_{2s} \Delta e_{2s} - h_{\Delta e_1} h_{\Delta e_2} e_{1s} s_{21} e_{2s} \\
 & - h_{e_1} h_{\Delta e_2} \Delta e_{1s} s_{21} e_{2s} - h_{\Delta e_1} h_{e_2} e_{1s} s_{21} \Delta e_{2s} - h_{e_1} h_{e_2} \Delta e_{1s} s_{21} \Delta e_{2s} \} \\
 \Delta u_{2s} = & s_{12} \{ \gamma_{e_{21}} e_{1s}(k) + \gamma_{\Delta e_{21}} \Delta e_{1s}(k) + \gamma_{e_{22}} e_{2s} + \gamma_{\Delta e_{22}} \Delta e_{2s}(k) \} \quad (4.23)
 \end{aligned}$$

$$\gamma_{e_{21}} = \frac{\{17(A_2+B_2)^2+A_2B_2\}h_{\Delta e_1}h_{e_2}h_{\Delta e_2} + \{-7(A_2+B_2)^2+A_2B_2\}h_{\Delta e_1}e_{2s}(k)\Delta e_{2s}}{D(s_{12}e_{1s}, s_{12}\Delta e_{1s}, e_{2s}, \Delta e_{2s})}$$

$$\gamma_{\Delta e_{21}} = \frac{\{17(A_2+B_2)^2+A_2B_2\}h_{e_1}h_{e_2}h_{\Delta e_2} + \{-7(A_2+B_2)^2+A_2B_2\}h_{e_1}e_{2s}(k)\Delta e_{2s}}{D(s_{12}e_{1s}, s_{12}\Delta e_{1s}, e_{2s}, \Delta e_{2s})}$$

$$\gamma_{e_{22}} = \frac{\{17(A_2+B_2)^2+A_2B_2\}h_{e_1}h_{\Delta e_1}h_{\Delta e_2} + \{-7(A_2+B_2)^2+A_2B_2\}h_{\Delta e_2}e_{1s}(k)\Delta e_{1s}}{D(s_{12}e_{1s}, s_{12}\Delta e_{1s}, e_{2s}, \Delta e_{2s})}$$

$$\gamma_{\Delta e_{22}} = \frac{\{17(A_2+B_2)^2+A_2B_2\}h_{e_1}h_{\Delta e_1}h_{e_2} + \{-7(A_2+B_2)^2+A_2B_2\}h_{e_2}e_{1s}(k)\Delta e_{1s}}{D(s_{12}e_{1s}, s_{12}\Delta e_{1s}, e_{2s}, \Delta e_{2s})}$$

Et

$$D(s_{12}e_{1s}, s_{12}\Delta e_{1s}, e_{2s}, \Delta e_{2s}) = 3(A_2+B_2)\{15h_{e_1}h_{\Delta e_1}h_{e_2}h_{\Delta e_2} - e_{1s}e_{2s}\Delta e_{2s} - h_{e_2}h_{\Delta e_2}e_{1s}\Delta e_{1s} - h_{e_1}h_{\Delta e_1}e_{2s}\Delta e_{2s} - h_{\Delta e_1}h_{\Delta e_2}e_{1s}s_{21}e_{2s} - h_{e_1}h_{\Delta e_2}\Delta e_{1s}s_{12}e_{2s} - h_{\Delta e_1}h_{e_2}e_{1s}s_{12}\Delta e_{2s} - h_{e_1}h_{e_2}\Delta e_{1s}s_{12}\Delta e_{2s}\}$$

Les expressions des sorties du régulateur dans le reste des cellules peuvent être obtenues en remplaçons $e_{1s}(k), \Delta e_{1s}(k), e_{2s}(k)$ et $\Delta e_{2s}(k)$ par leurs valeurs appropriées montrées au tableau 4.3 dans les équation (4.22) et (4.23).

Cell	e_{1s}	Δe_{1s}	e_{2s}	Δe_{2s}	Cell	e_{1s}	Δe_{1s}	e_{2s}	Δe_{2s}
(3, 8)	$-h_{e1}$	$h_{\Delta e_1}$	h_{e2}	r	(7, 9)	r	$h_{\Delta e_1}$	r	r
(4, 8)	h_{e1}	$h_{\Delta e_1}$	h_{e2}	r	(8, 9)	h_{e1}	r	r	r
(2, 9)	$-h_{e1}$	$-h_{\Delta e_1}$	r	r	(9, 2)	r	r	$-h_{e2}$	$-h_{\Delta e_2}$
(3, 9)	$-h_{e1}$	$h_{\Delta e_1}$	r	r	(9, 3)	r	r	$-h_{e2}$	$h_{\Delta e_2}$
(4, 9)	h_{e1}	$h_{\Delta e_1}$	r	r	(9, 4)	r	r	h_{e2}	$h_{\Delta e_2}$
(5, 2)	r	$-h_{\Delta e_1}$	$-h_{e2}$	$-h_{\Delta e_2}$	(9, 6)	r	r	$-h_{e2}$	r
(5, 3)	r	$-h_{\Delta e_1}$	$-h_{e2}$	$h_{\Delta e_2}$	(9, 7)	r	r	r	$h_{\Delta e_2}$
(5, 4)	r	$-h_{\Delta e_1}$	h_{e2}	$h_{\Delta e_2}$	(9, 8)	r	r	h_{e2}	r

Cell	e_{1s}	Δe_{1s}	e_{2s}	Δe_{2s}	Cell	e_{1s}	Δe_{1s}	e_{2s}	Δe_{2s}
(1, 2)	h_{e1}	$-h_{\Delta e1}$	$-h_{e2}$	$-h_{\Delta e2}$	(6, 1)	$-h_{e1}$	r	h_{e2}	$-h_{\Delta e2}$
(1, 3)	h_{e1}	$-h_{\Delta e1}$	$-h_{e2}$	$h_{\Delta e2}$	(6, 2)	$-h_{e1}$	r	$-h_{e2}$	$-h_{\Delta e2}$
(1, 4)	h_{e1}	$-h_{\Delta e1}$	h_{e2}	$h_{\Delta e2}$	(6, 3)	$-h_{e1}$	r	$-h_{e2}$	$h_{\Delta e2}$
(2, 1)	$-h_{e1}$	$-h_{\Delta e1}$	h_{e2}	$-h_{\Delta e2}$	(6, 4)	$-h_{e1}$	r	h_{e2}	$h_{\Delta e2}$
(2, 2)	$-h_{e1}$	$-h_{\Delta e1}$	$-h_{e2}$	$-h_{\Delta e2}$	(7, 1)	r	$h_{\Delta e1}$	h_{e2}	$-h_{\Delta e2}$
(2, 3)	$-h_{e1}$	$-h_{\Delta e1}$	$-h_{e2}$	$h_{\Delta e2}$	(7, 2)	r	$h_{\Delta e1}$	$-h_{e2}$	$-h_{\Delta e2}$
(2, 4)	$-h_{e1}$	$-h_{\Delta e1}$	h_{e2}	$h_{\Delta e2}$	(7, 3)	r	$h_{\Delta e1}$	$-h_{e2}$	$h_{\Delta e2}$
(3, 1)	$-h_{e1}$	$h_{\Delta e1}$	h_{e2}	$-h_{\Delta e2}$	(7, 4)	r	$h_{\Delta e1}$	h_{e2}	$h_{\Delta e2}$
(3, 2)	$-h_{e1}$	$h_{\Delta e1}$	$-h_{e2}$	$-h_{\Delta e2}$	(8, 1)	h_{e1}	r	h_{e2}	$-h_{\Delta e2}$
(3, 3)	$-h_{e1}$	$h_{\Delta e1}$	$-h_{e2}$	$h_{\Delta e2}$	(8, 2)	h_{e1}	r	$-h_{e2}$	$-h_{\Delta e2}$
(3, 4)	$-h_{e1}$	$h_{\Delta e1}$	h_{e2}	$h_{\Delta e2}$	(8, 3)	h_{e1}	r	$-h_{e2}$	$h_{\Delta e2}$
(4, 1)	h_{e1}	$h_{\Delta e1}$	h_{e2}	$-h_{\Delta e2}$	(8, 4)	h_{e1}	r	h_{e2}	$h_{\Delta e2}$
(4, 2)	h_{e1}	$h_{\Delta e1}$	$-h_{e2}$	$-h_{\Delta e2}$	(5, 6)	r	$-h_{\Delta e1}$	$-h_{e2}$	r
(4, 3)	h_{e1}	$h_{\Delta e1}$	$-h_{e2}$	$h_{\Delta e2}$	(5, 7)	r	$-h_{\Delta e1}$	r	$h_{\Delta e2}$
(4, 4)	h_{e1}	$h_{\Delta e1}$	h_{e2}	$h_{\Delta e2}$	(5, 8)	r	$-h_{\Delta e1}$	h_{e2}	r
(2, 5)	$-h_{e1}$	$-h_{\Delta e1}$	r	$-h_{\Delta e2}$	(6, 5)	$-h_{e1}$	r	r	$-h_{\Delta e2}$
(3, 5)	$-h_{e1}$	$h_{\Delta e1}$	r	$-h_{\Delta e2}$	(6, 6)	$-h_{e1}$	r	$-h_{e2}$	r
(4, 5)	h_{e1}	$h_{\Delta e1}$	r	$-h_{\Delta e2}$	(6, 7)	$-h_{e1}$	r	r	$h_{\Delta e2}$
(1, 6)	h_{e1}	$-h_{\Delta e1}$	$-h_{e2}$	r	(6, 8)	$-h_{e1}$	r	h_{e2}	r
(2, 6)	$-h_{e1}$	$-h_{\Delta e1}$	$-h_{e2}$	r	(7, 5)	r	$h_{\Delta e1}$	r	$-h_{\Delta e2}$
(3, 6)	$-h_{e1}$	$h_{\Delta e1}$	$-h_{e2}$	r	(7, 6)	r	$h_{\Delta e1}$	$-h_{e2}$	r
(4, 6)	h_{e1}	$h_{\Delta e1}$	$-h_{e2}$	r	(7, 7)	r	$h_{\Delta e1}$	r	$h_{\Delta e2}$
(1, 7)	h_{e1}	$-h_{\Delta e1}$	r	$h_{\Delta e2}$	(7, 8)	r	$h_{\Delta e1}$	h_{e2}	r
(2, 7)	$-h_{e1}$	$-h_{\Delta e1}$	r	$h_{\Delta e2}$	(8, 5)	h_{e1}	r	r	$-h_{\Delta e2}$
(3, 7)	$-h_{e1}$	$h_{\Delta e1}$	r	$h_{\Delta e2}$	(8, 6)	h_{e1}	r	$-h_{e2}$	r
(4, 7)	h_{e1}	$h_{\Delta e1}$	r	$h_{\Delta e2}$	(8, 7)	h_{e1}	r	r	$h_{\Delta e2}$
(1, 8)	h_{e1}	$-h_{\Delta e1}$	h_{e2}	r	(8, 8)	h_{e1}	r	h_{e2}	r
(2, 8)	$-h_{e1}$	$-h_{\Delta e1}$	h_{e2}	r	(6, 9)	$-h_{e1}$	r	r	r

Remarque : r veut dire « reste ».

Tableau 4.3 : Valeurs de $e_{1s}(k)$, $\Delta e_{1s}(k)$, $e_{2s}(k)$ et $\Delta e_{2s}(k)$ selon les cellules.

Ce modèle décrit un régulateur TITO Flou simplifié ayant :

- Deux ensembles flou pour chaque entrée et cinq pour chaque sorties du régulateur (fonctions d'appartenance).
- Cinq règles d'inférence.

Et utilisons le produit algébrique pour l'opérateur ET, la somme bornée pour l'opérateur OU, l'inférence par le produit de Larsen et la méthode CoS pour la defuzzification.

Il est à noter que :

- Le régulateur est à structure variable car elle diffère d'une cellule à une autre.
- Le régulateur est à gain non linéaire variable.
- La variation de commande est nulle dans les cellule (1,1) et (3,3), à l'origine (0,0,0,0) et dans les cellules (2,2) et (4,4) quand $s_{21} = -1$ et $s_{12} = -1$.

- Les variations des commandes minimale $\Delta u_{1\min} = \frac{-\{5(A_1 + B_1)^2 + A_1 B_1\}}{3(A_1 + B_1)}$ et

$$\Delta u_{2\min} = \frac{-\{5(A_2 + B_2)^2 + A_2 B_2\}}{3(A_2 + B_2)}$$

apparaissent dans la cellule (2,2) lorsque $s_{21} = 1$ et $s_{12} = 1$.

- Les variations des commandes maximale $\Delta u_{2\max} = \frac{\{5(A_2 + B_2)^2 + A_2 B_2\}}{3(A_2 + B_2)}$ et

$$\Delta u_{1\max} = \frac{\{5(A_1 + B_1)^2 + A_1 B_1\}}{3(A_1 + B_1)}$$

apparaissent dans la cellule (4,4) lorsque $s_{21} = 1$ et $s_{12} = 1$.

4.4. Conclusion

Dans ce chapitre nous avons présenté le modèle mathématique d'un régulateur TITO PI Flou simplifié en interprétant les travaux de recherche publié dans [13].

Dans la suite des travaux, nous allons implémenter ce modèle mathématique du contrôleur PI-flou simplifié, cette implémentation sous API S7-300 doit respecter autant que possible le développement proposé dans les travaux de [13], afin de vérifier sa validité dans un système réel.

CHAPITRE 05 :

Teste de validation et résultats expérimentaux

5.1. Introduction

Dans ce chapitre nous allons présenter l'automate programmable industriel utilisé et sa configuration et le langage de programmation (S7-SCL) qui nous permettrons d'implémenter notre commande.

Nous allons aussi mettre en évidence la partie expérimentale de notre travail, où nous allons exploiter les connaissances acquises dans les chapitres précédents pour implémenter les différents régulateur (**modèle mathématique du régulateur TITO flou simplifié, PI classique**) dans le but de contrôler la chaîne de régulation de niveau du liquide de l'unité opérationnelle PUL-2 et des deux stations en couplage. 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 du modèle mathématique du régulateur flou simplifié exposé précédemment toute en exploitant autant que possible les avantages proposé par le langage S7-SCL pour assurer une optimisation en temps de calcul.

5.2. Présentation matériel et implémentation

Dans la norme IEC 61131-1 on définit l'automate comme étant un système numérique destiné à être utilisé dans un environnement industriel.

Un API (automate programmable industriel) est un appareil électronique utilisé pour la commande de processus industriel par traitement séquentiel, il comporte en générale une alimentation délivrant 24V, une CPU doté d'une mémoire pour le chargement et le stockage du programme, des modules d'E\S (numériques et analogiques) et possède des liaisons (MPI, PROFIBUS) pour la communication avec le PC (pupitre de commande).

Ces automates peuvent être sous forme compacte où tous les composants sont regroupés dans le même block. Ou bien de type modulaire avec des composants séparé sous forme de modules et assemblé sur un RACK.

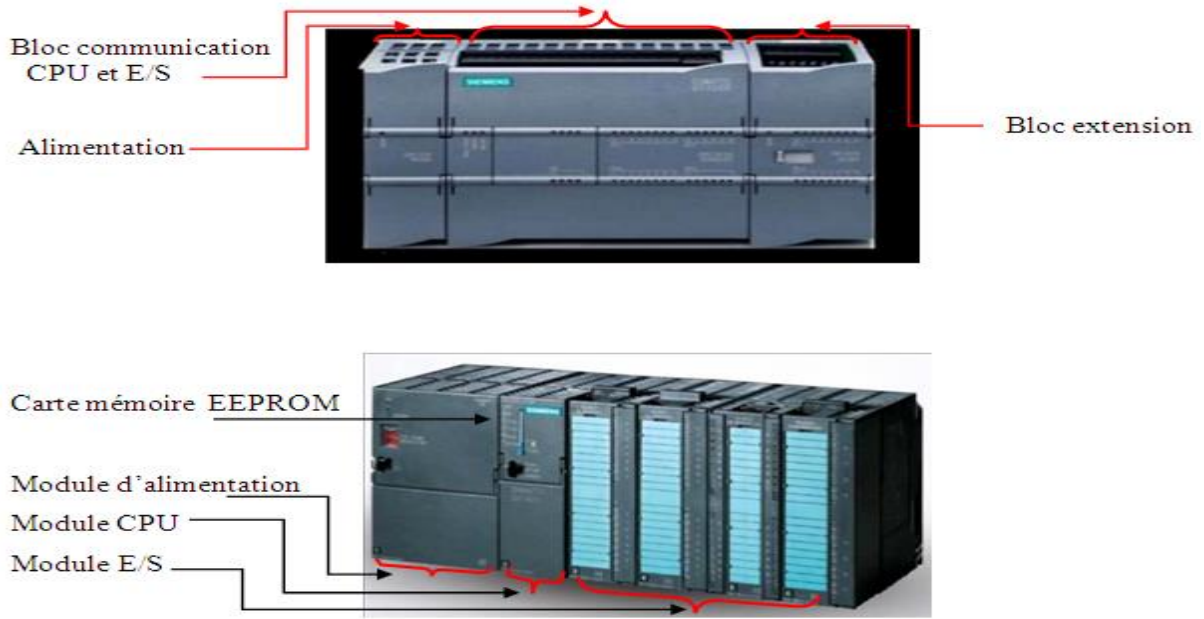


Figure 5.1 : Exemples d'automate compact et modulaire (successivement).

5.2.1. Configuration matériel

Pour la commande du système nous avons utilisé un automate S7 300 alimenté par une PS 307 2A, équipé de la CPU 314 C-2 DP avec interface de communication (DP) qui permet de réaliser des liaisons MPI ou PROFIBUS avec d'autres automates ou d'autres périphéries.

Notre automate possède également 24 entrées numériques (DI 24), 16 sorties numériques (DO 16), 5 entrées analogiques (AI 5), 2 sorties analogiques (AO 2), une unité pour le comptage et une autre pour le positionnement. (Voir figure 5.2)

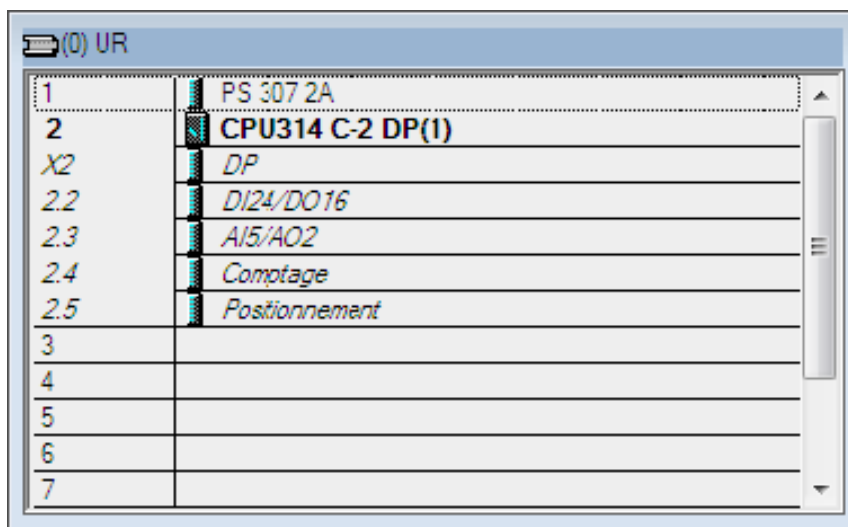


Figure 5.2 : Configuration matériel sous Step7.

5.2.2. Les langages de programmation

Les langages de programmation CONT, LIST et LOG pour S7-300/400 sont une partie intégrante du logiciel de base. Dans notre projet nous allons le langage S7-SCL qui sera présenté avec les autres langages dans la suite.

a) Langage CONT (Ladder)

C'est un langage de programmation graphique. La syntaxe des instructions fait penser aux schémas de circuits. Le langage 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.

b) Langage LIST

Le langage 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, le langage LIST a été complété par quelques structures de langage évolué (comme, par exemple, des paramètres de blocs et accès structurés aux données).

c) Langage 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.

d) Langage S7-SCL

Le langage S7-SCL (Structured Control Language) est un langage de programmation évolué proche du PASCAL, utilisé pour programmer des automates programmables du type SIMATIC S7.

5.2.3. Présentation du langage S7-SCL

a) Environnement de développement

Un environnement de développement répondant à la fois aux exigences de SCL et de STEP 7 permet d'optimiser et de simplifier l'utilisation de SCL[14]. Il comprend :

- Un éditeur, permettant d'écrire des programmes avec des fonctions (FC), des blocs fonctionnels (FB), des blocs d'organisation (OB), des blocs de données (DB) et des

types de données utilisateur (UDT). Des fonctions puissantes assistant le programmeur dans sa tâche.

- un compilateur séquentiel pour compiler le programme préalablement édité en code machine MC7. Celui-ci est exécutable sur toutes les CPU des systèmes d'automatisation S7-300/400 à partir de la CPU 314.
- un débogueur, permettant de rechercher des erreurs de programmation logiques dans un programme correctement compilé. La recherche d'erreurs s'effectue dans le langage source.

b) Avantages par rapport aux autres langages

-Les algorithmes complexes, le traitement de structures complexes ainsi que les calculs se font plus facilement.

-Langage similaire au PASCAL, facile à apprendre pour les techniciens et les informaticiens.

-Le nombre de ligne en langage SCL est généralement plus faible que dans les autres langages.

c) Les différents types de données

Un type de données simple définit la structure de données qui ne peuvent pas être divisées en éléments plus petits. Il répond à la norme EN 1131-3. Un type de données simple décrit une zone de mémoire de longueur fixe et correspond à une grandeur exprimée sous forme binaire, entière, réelle, sous forme de durée, d'heure ou de caractère. Les types de données suivants sont prédéfinis dans S7-SCL[14].

Groupe	Types de données	Signification
Types de données binaires	BOOL BYTE WORD DWORD	Les données de ce type occupent 1 bit , 8 bits, 16 bits ou 32 bits.
Types de données sur caractère	CHAR	Les données de ce type occupent 1 caractère et un seul du jeu de caractères ASCII.
Types de données numériques	INT DINT REAL	Les données de ce type permettent de traiter des valeurs numériques.
Types de données temporels	TIME DATE TIME_OF_DAY S5TIME	Les données de ce type représentent les diverses valeurs de durée et de date dans STEP 7.

Tableau 5.1 : Les différents type de donn e pr ed efinis dans S7-SCL

d) Exemple de programmation d'un Bloc FC :

```

1 FUNCTION FC1 : VOID
2
3 VAR INPUT
4 // Paramètre d'entrées
5 E1:BOOL;
6 E2:BOOL;
7 E3:BOOL;
8 END_VAR
9
10 VAR OUTPUT
11 // Paramètre de sorties
12 A1:BOOL;
13 A2:BOOL;
14 END_VAR
15
16 VAR TEMP
17 // Variables Temporaires
18 END_VAR
19
20 BEGIN
21 // Instructions
22 A1:= E1 AND E2;
23 A2:=A1 AND E3;
24 END_FUNCTION

```

e) Exemple de programmation d'un Bloc FB :

```

1 FUNCTION_BLOCK FB1
2
3 VAR INPUT
4 // Paramètre d'entrées
5 E1:BOOL;
6 E2:BOOL;
7 E3:BOOL;
8 END_VAR
9
10 VAR OUTPUT
11 // Paramètre de sorties
12 A1:BOOL;
13 A2:BOOL;
14 END_VAR
15
16 VAR TEMP
17 // Variables Temporaires
18 END_VAR
19
20 VAR
21 // Variables Statiques
22 END_VAR
23
24 BEGIN
25 // Instructions
26 A1:= E1 AND E2;
27 A2:=A1 AND E3;
28 END_FUNCTION_BLOCK

```

f) Les différentes sections de déclaration :

A chaque catégorie de variable locale ou de paramètre est affectée une section de déclaration propre, caractérisée par un couple de mots-clés. Chaque section contient les déclarations qui y sont autorisées. L'ordre de ces sections est quelconque.

Le tableau suivant indique les différentes sections de déclaration et les blocs où on peut les déclarer [14].

Données	Syntaxe	FB	FC	OB
Variable comme : variable statique	VAR ... END_VAR	X	X *)	
variable temporaire	VAR_TEMP ... END_VAR	X	X	X
Paramètre de bloc comme : paramètre d'entrée	VAR_INPUT ... END_VAR	X	X	
paramètre de sortie	VAR_OUTPUT ... END_VAR	X	X	
paramètre d'entrée/sortie	VAR_IN_OUT ... END_VAR	X	X	

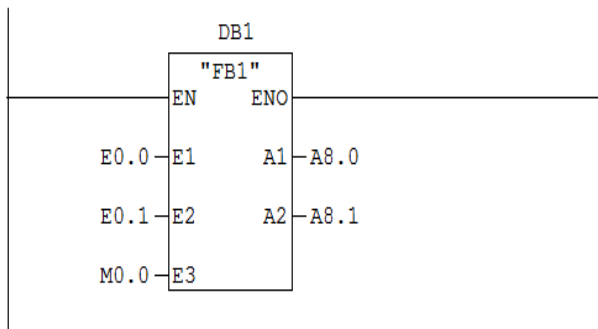
Tableau 5.2 Les différentes sections de déclaration de variable dans S7-SCL

g) Exemples d'appel de Bloc programmé en SCL :

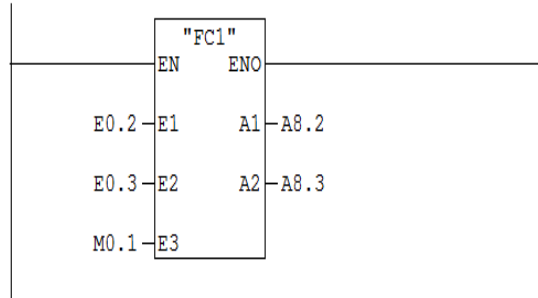
Après la compilation d'un programme SCL le bloc correspondant sera généré.

En reprenons les deux exemples ci-dessus (FC1 et FB1), après compilation, ces dernier peuvent être appelé comme suit :

Réseau 1 :



Réseau 2 :



5.3. Implémentation du modèle mathématique du régulateur TITO PI flou simplifié

Pour l'implémentation du modèle mathématique du régulateur sous Matlab/Simulink nous avons programmé des blocs de fonctions intégrées (Embedded functions) par contre sous l'automate S7-300, nous avons opté au langage S7-SCL (Structured Control List) présenté précédemment pour assurer une certaine flexibilité du programme. Des blocs FB et FC sont programmés afin de réaliser les différents calculs nécessaires.

5.3.1. Implémentation du bloc régulateur sous Matlab/Simulink

Après identification de la cellule globale, celle-ci est utilisé dans le bloc de calcul de commandes pour utiliser la formule adéquate aux calculs des sorties parmi les 81 cas.

Le schéma bloc du régulateur sous Simulink est présenté à la figure 5.3.

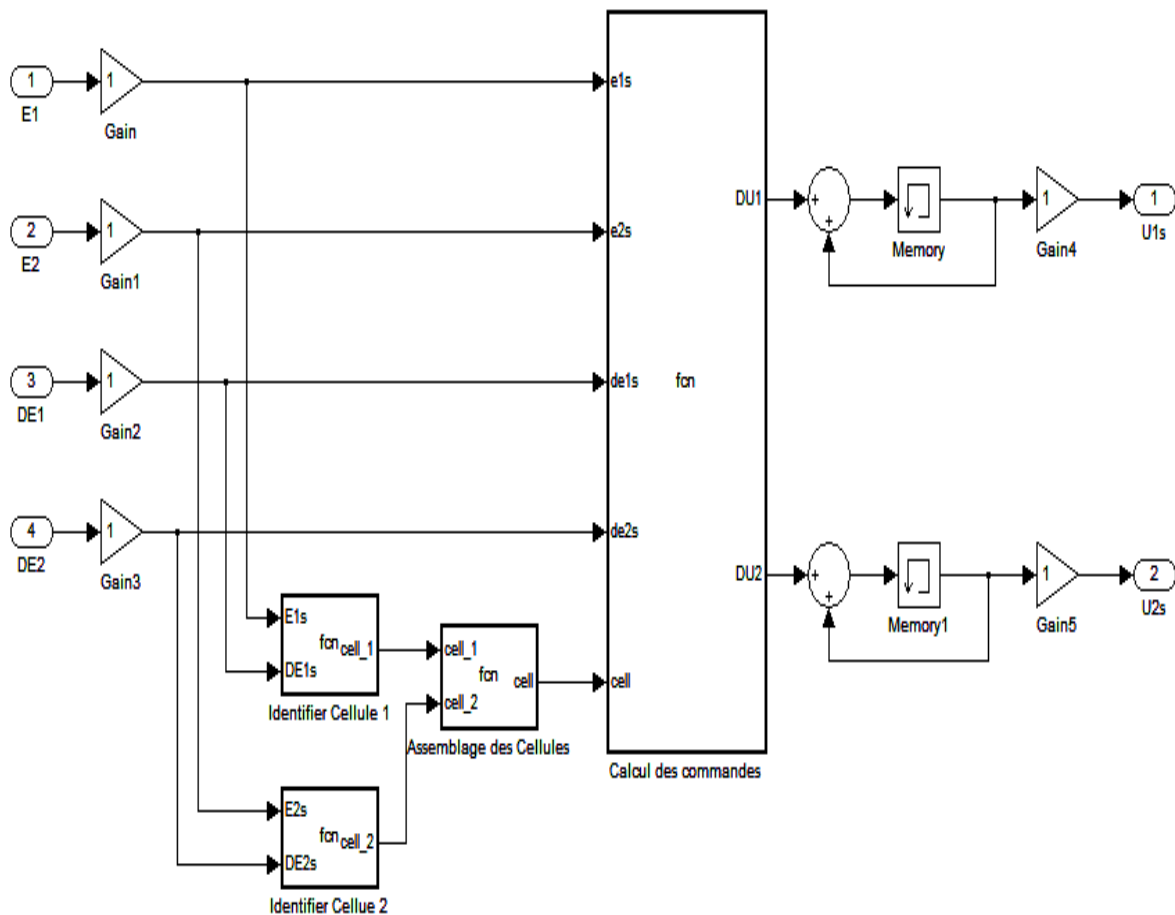


Figure 5.3 : Schéma Bloc du régulateur sous Simulink.

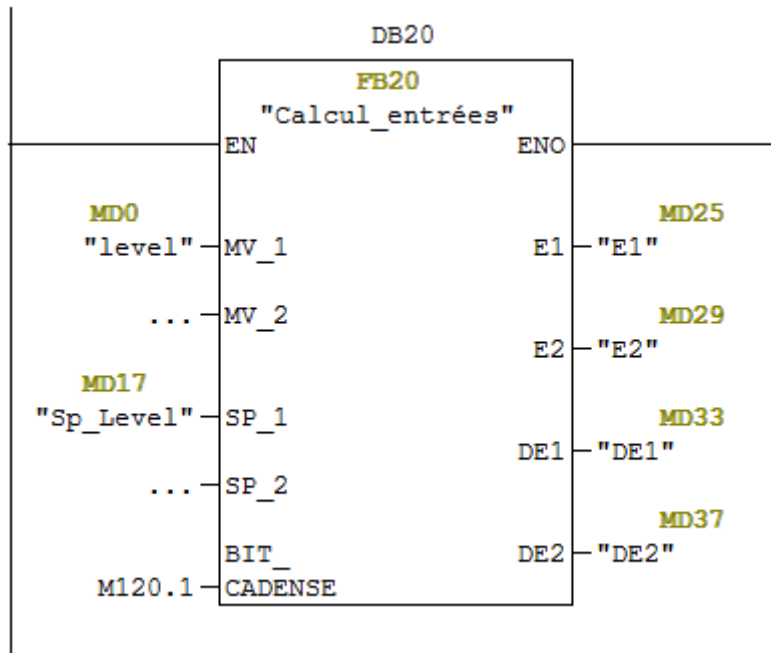
5.3.2. Implémentation du régulateur sous Step7

a) Les entrées du régulateur

La première étape pour la programmation du régulateur est de calculer ses variables d'entrées.

Le correcteur flou que nous allons implémenter sera constitué de quatre entrées qui sont la

première erreur e_1 , dérivée de cette erreur Δe_1 , la deuxième erreur e_2 , la dérivée de cette erreur Δe_2 . Pour effectuer cette tâche nous avons programmé un bloc FB sous Step7 :



Où E désigne l'erreur (Différence entre la consigne SP et la mesure MV) et DE désigne la dérivée de l'erreur. Le bit de cadence est associé à celui de la CPU de l'automate, son rôle et de synchroniser l'exécution des différents calculs du régulateur. Voir figure 5.4.

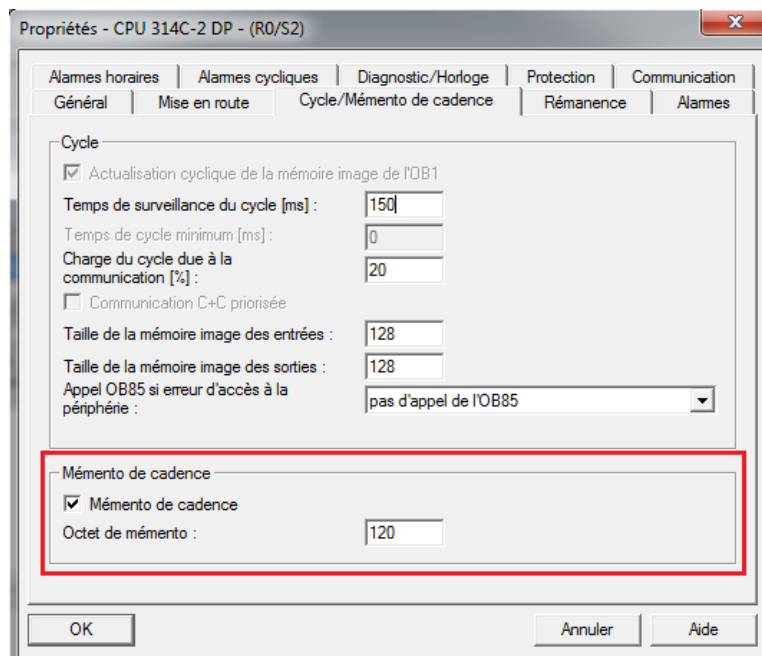
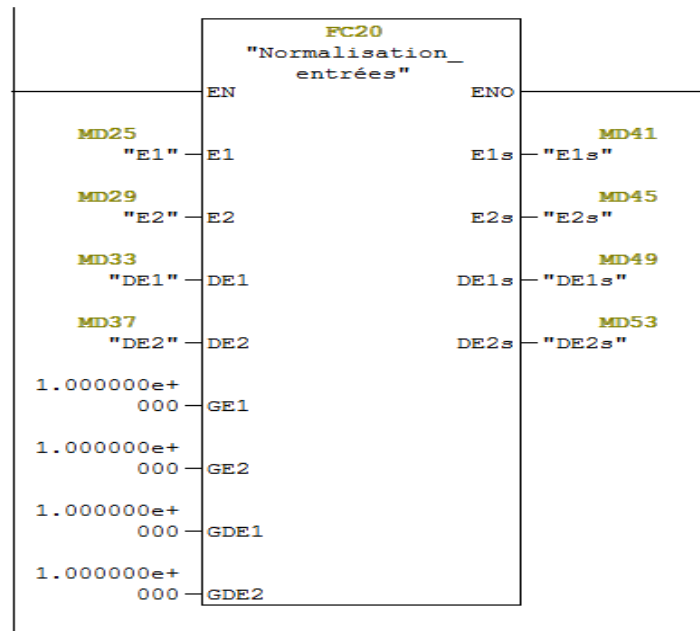


Figure 5.4 : Activation du momento de cadences de la CPU.

b) Normalisation des entrées :

Des gains de normalisation sont utilisés en entrée du contrôleur flou simplifié. Ils permettent de changer la sensibilité du régulateur flou sans en changer sa structure.



GE1, GE2, GDE1, GDE2 représentent respectivement les gains de normalisation des entrées du régulateur respectivement $e_1, e_2, \Delta e_1$ et Δe_2 .

c) Déclaration des différentes constantes

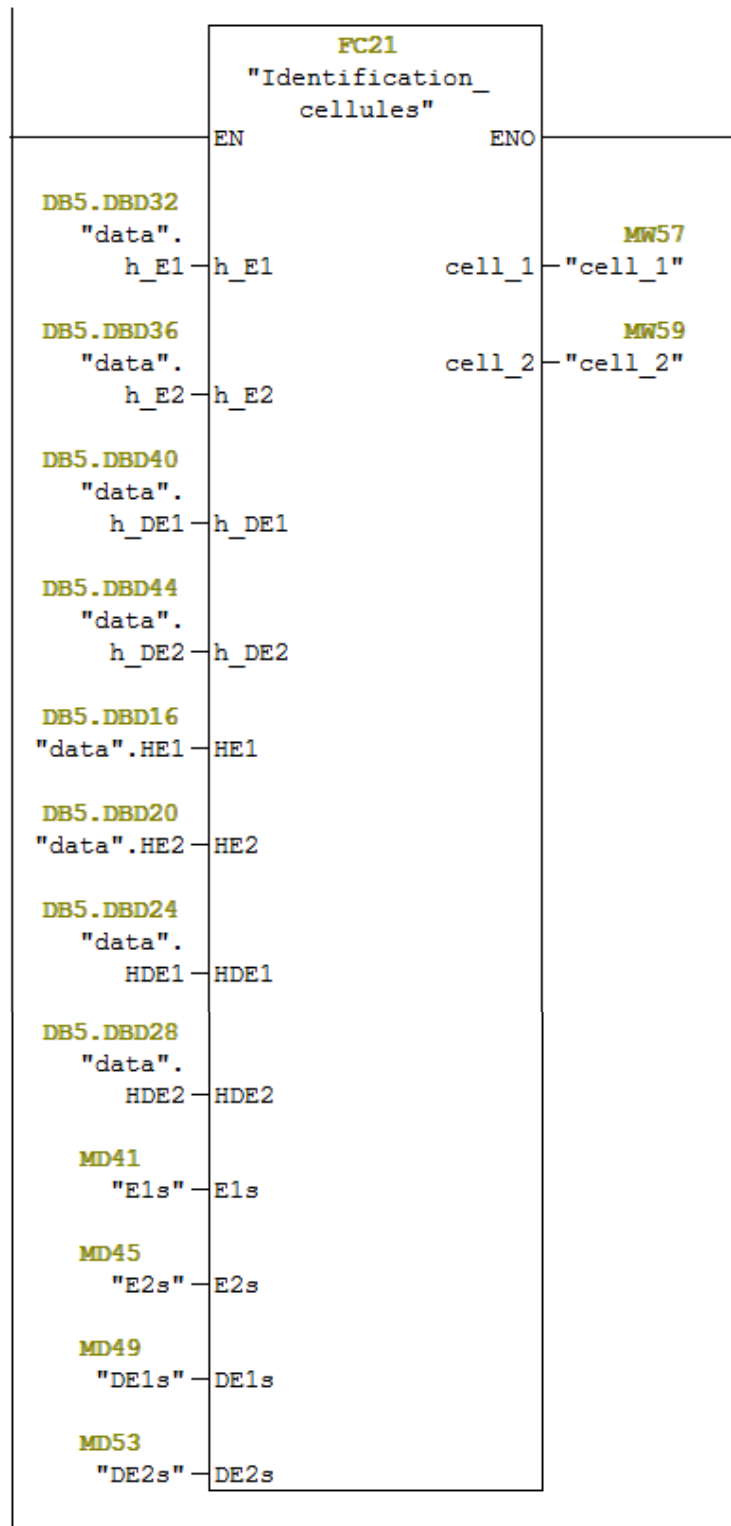
Pour assurer une certaine flexibilité et facilité de paramétrage, les différents paramètres utilisés par le régulateur sont déclaré au niveau d'un bloc de donnée globale comme suit :

Adresse	Nom	Type	Valeur initiale	Commentaire
0.0		STRUCT		
+0.0	GE1	REAL	2.000000e+000	
+4.0	GE2	REAL	3.000000e+000	
+8.0	GDE1	REAL	5.000000e-002	
+12.0	GDE2	REAL	2.000000e+000	
+16.0	HE1	REAL	3.000000e+000	
+20.0	HE2	REAL	3.000000e+000	
+24.0	HDE1	REAL	1.500000e+000	
+28.0	HDE2	REAL	1.500000e+000	
+32.0	h_e1	REAL	1.000000e+000	
+36.0	h_e2	REAL	1.000000e+000	
+40.0	h_DE1	REAL	6.000000e-001	
+44.0	h_DE2	REAL	6.000000e-001	
+48.0	A1	REAL	4.000000e+000	
+52.0	B1	REAL	4.000000e+000	
+56.0	A2	REAL	1.000000e+001	
+60.0	B2	REAL	1.000000e+001	
+64.0	s12	REAL	1.000000e+000	
+68.0	s21	REAL	1.000000e+000	
=72.0		END_STRUCT		

d) Détermination des cellules :

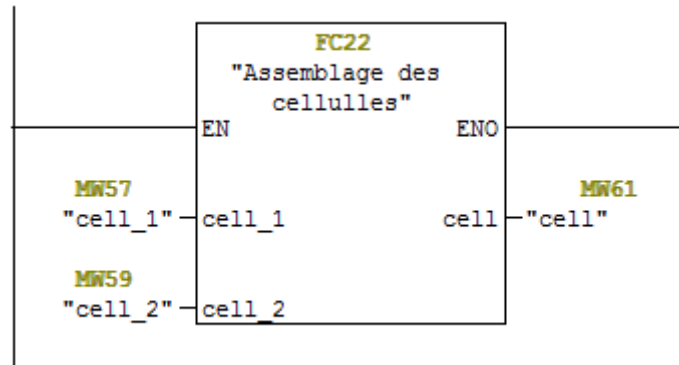
Ce bloc utilise les sous espaces présentées précédement (Chapitre 4) pour déterminer les cellules qui englobe les valeurs des entrées du régulateur(erreurs et dérivées de l'erreur).

Selon la valeur de chaque erreur et sa dérivée, la cellule correspondante peut prendre une valeur en entier entre 1 et 9.



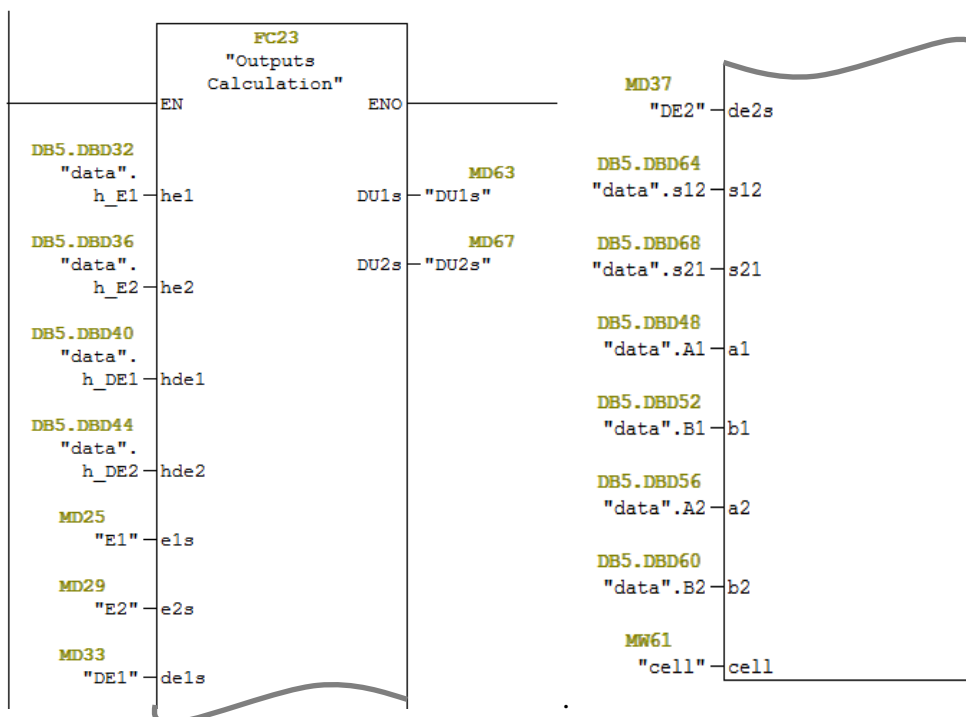
e) Assemblage des cellules

Pour assembler les cellules précédemment déterminées, nous avons utilisé un bloc fonction FC22. Ce bloc fusionne le résultat des deux cellules trouvées à partir du bloc FC21 pour donner un entier exprimant la cellule globale allant de 11 jusqu'à 99 (sachant qu'aucune cellule ne peut prendre la valeur 0), l'équation utilisée pour le calcul de la sortie du régulateur dépend de cette dernière.



f) Calcul des sorties du régulateur ($\Delta U_{1s}, \Delta U_{2s}$)

Le bloc FC23 calcule les sorties du régulateur qui sont les variations de commande en fonctions de la cellule globale (cell). Pour calculer ces sorties, on doit introduire à ce bloc les paramètres $h_{e_1}, h_{e_2}, h_{\Delta e_1}, h_{\Delta e_2}, s_{12}, s_{21}, A_1, B_1, A_2, B_2$ et les entrées $e_{1s}(k), \Delta e_{1s}(k), e_{2s}(k), \Delta e_{2s}(k)$.



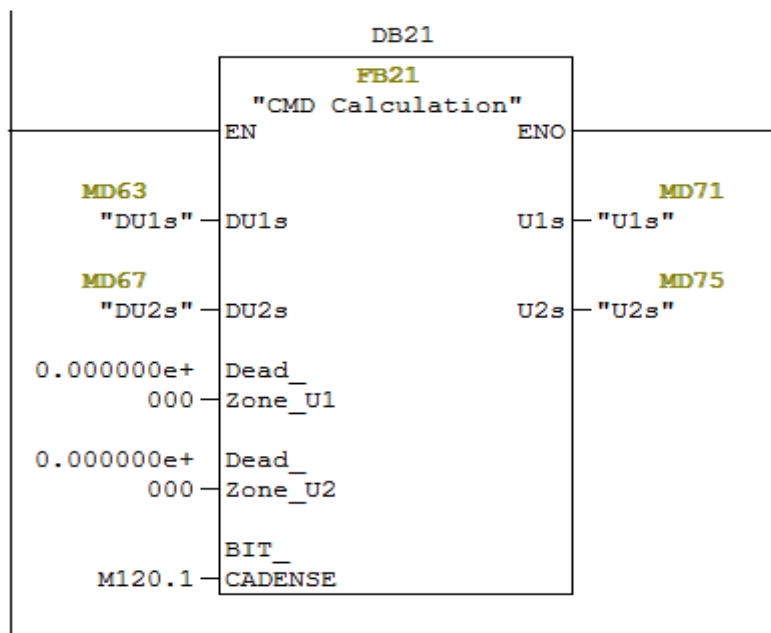
g) Calcul des commandes (U_1, U_2)

Le bloc de calcul des commandes U_{1s} et U_{2s} effectue les opérations suivantes

$$u_{1s}(k) = u_{1s}(k-1) + \Delta u_{1s}(k)$$

$$u_{2s}(k) = u_{2s}(k-1) + \Delta u_{2s}(k)$$

Et les paramètres Dead Zone désigne les zones mortes sur les $\Delta u(k)$ comme présenté ci-contre.



5.4. Test de validation des résultats du modèle mathématique du régulateur TITO flou

Le modèle du procédé est modélisé sous Simulink par la fonction de transfert identifié précédemment sur une seule station (sans interaction) avec initialisation du niveau à la valeur 4 (niveau à l'équilibre).

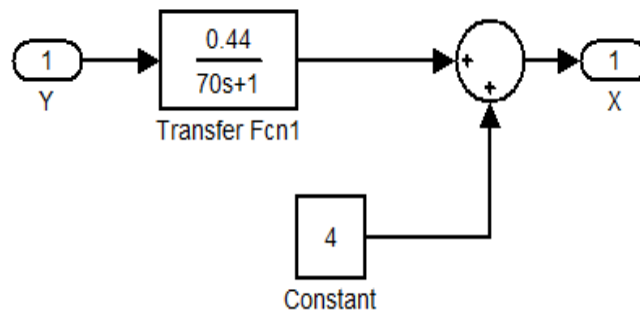


Figure 5.5 : Transfert du niveau d'une seule station sous Simulink.

5.4.1. Simulation du régulateur en mode SISO sous Matlab/Simulink :

Pour réguler le niveau de liquide sur un réservoir découplé (sans interaction entre les niveaux des deux réservoirs), où l’erreur sur la deuxième mesure est annulée.

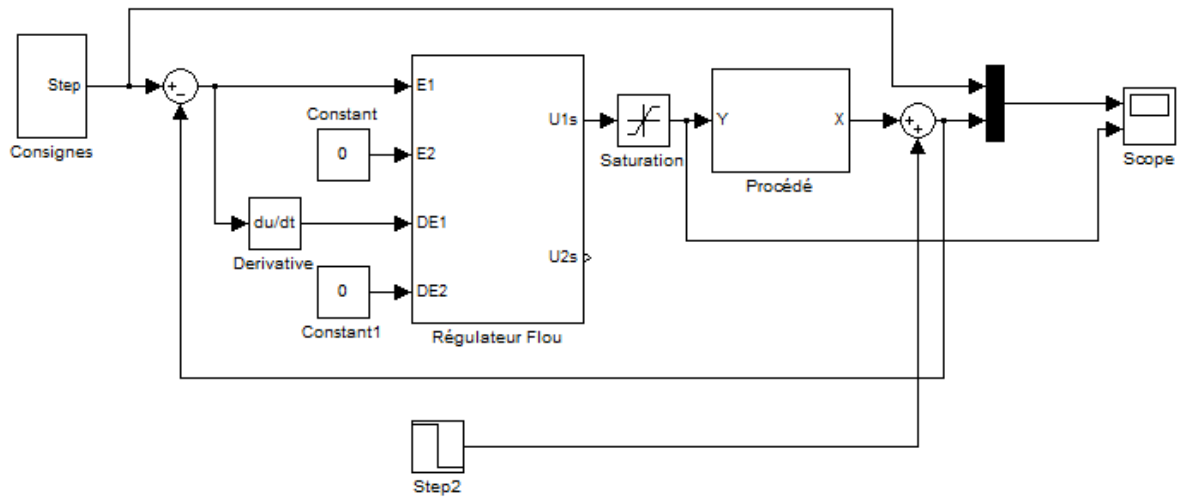


Figure 5.6 : Shéma de la régulation en mode SISO sous Simulink.

Au début de la simulation la sortie X est initialisé à la valeur 4V et à l’instant t=10 sec nous avons appliqué un échelon de consigne de 1V pour passer à 5V, de même à l’instant t=60 sec nous avons appliqué un échelon de consigne de 1V pour passer à 6V. Les résultats de simulation montre que le régulateur programmé et paramétré, suit convenablement les échelons de consigne en mode poursuite avec erreur statique nulle sans dépassement et un temps de réponse de $t_r=10\text{sec}$ pour les échelons de tests et un rejet de perturbation satisfaisant.

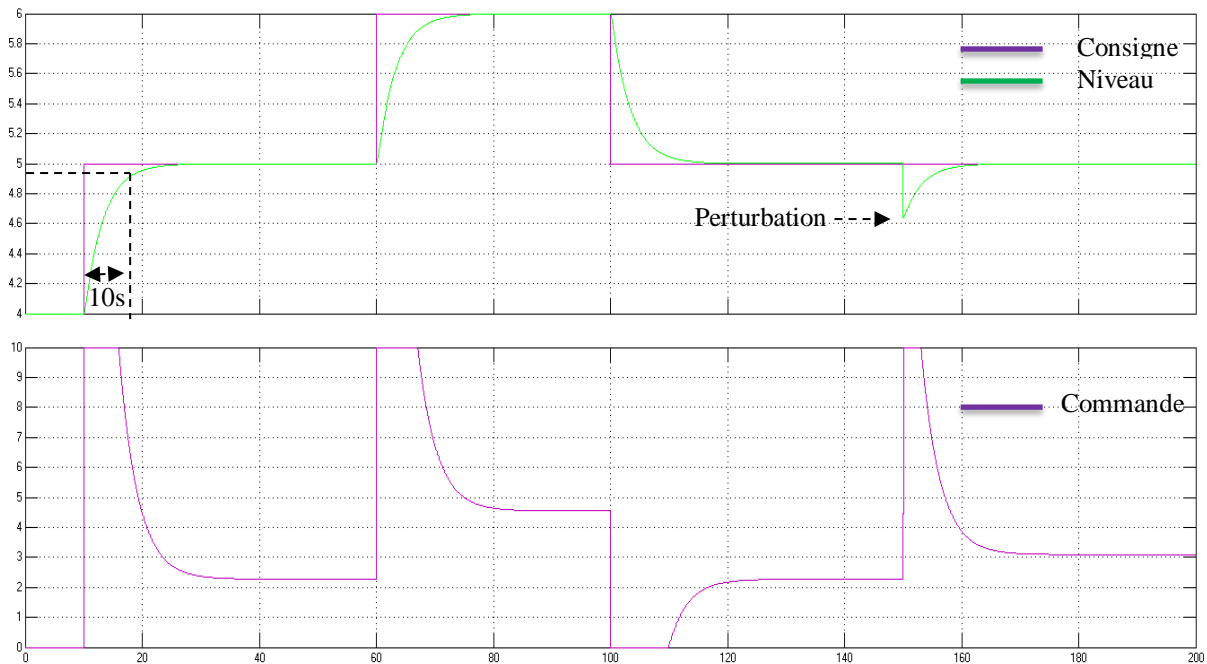


Figure 5.7 : Réponse du système simulé sous Simulink en mode SISO.

5.4.2. Résultats expérimentaux du régulateur SISO

Le relevé expérimental obtenu après l'implémentation du modèle mathématique du régulateur flou simplifié dans l'automate, sont très proche des résultats obtenus sous Simulink. Les résultats obtenus sont satisfaisant puisque l'écart statique est nul avec un temps de réponse de $tr=11\text{sec}$.

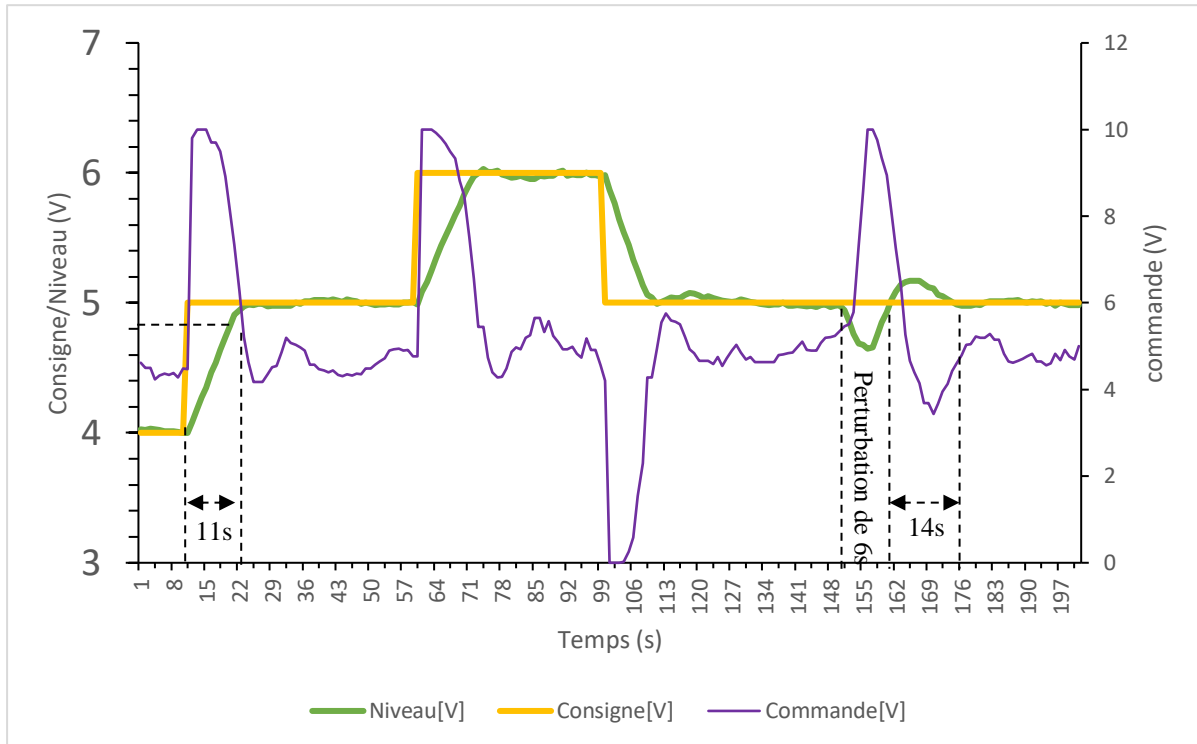


Figure 5.8 : Réponse du système régulé par automate en mode SISO.

Le résultat expérimental obtenu présente des performances dynamiques acceptables sans dépassement de valeur en mode poursuite et permet le rejet de perturbation en 14sec après son apparition en mode régulation avec un dépassement de 16%.

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.

5.4.3. Simulation du régulateur SITO sous Simulink

En ce test, on reprend le principe de la régulation cascade où le transfert du système de niveau présente la boucle principale et le transfert du système de débit présente la boucle secondaire qui est plus rapide.

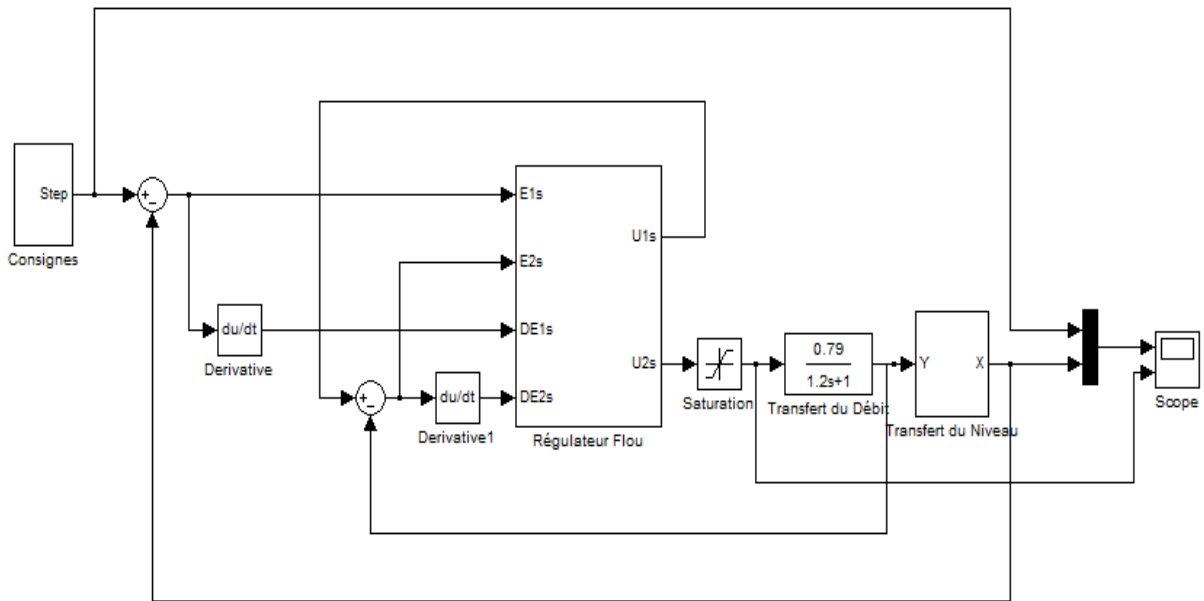


Figure 5.9 : Schéma de la régulation en mode SITO sous Simulink.

Avec les mêmes procédures de tests (consignes), les résultats de simulation montre que le régulateur en fonctionnement cascade, suit convenablement les échelons de consigne en mode poursuite avec erreur statique nulle, sans dépassement et un temps de réponse de $t_r=1$ sec pour les échelons de tests, comme le montre la figure 5.10.

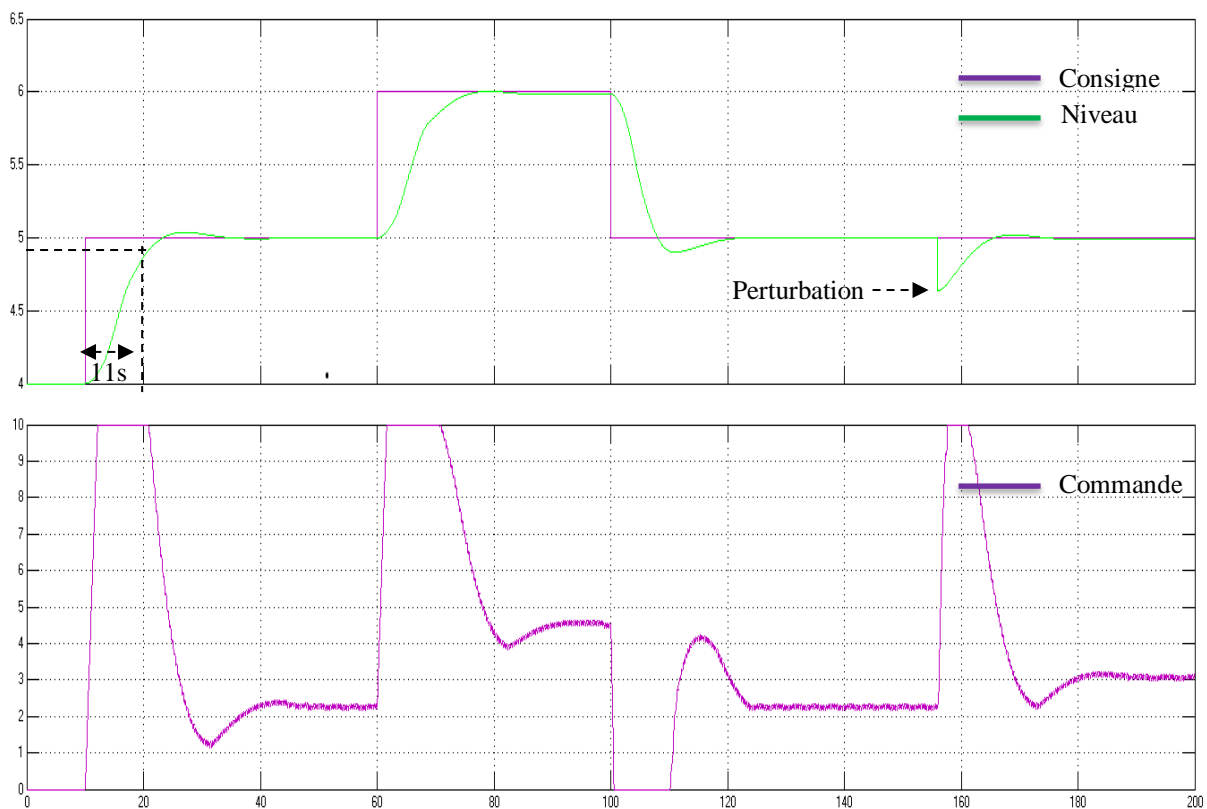


Figure 5.10 : Réponse du système simulé sous Simulink en mode SITO.

5.4.4. Résultats expérimentaux du régulateur SITO

Le relevé expérimental obtenue est très proche des résultats obtenus sous Simulink qui sont satisfaisant puisque l'écart statique est nul avec un temps de réponse de $t_r=12\text{sec}$. (Voir figure 5.11)

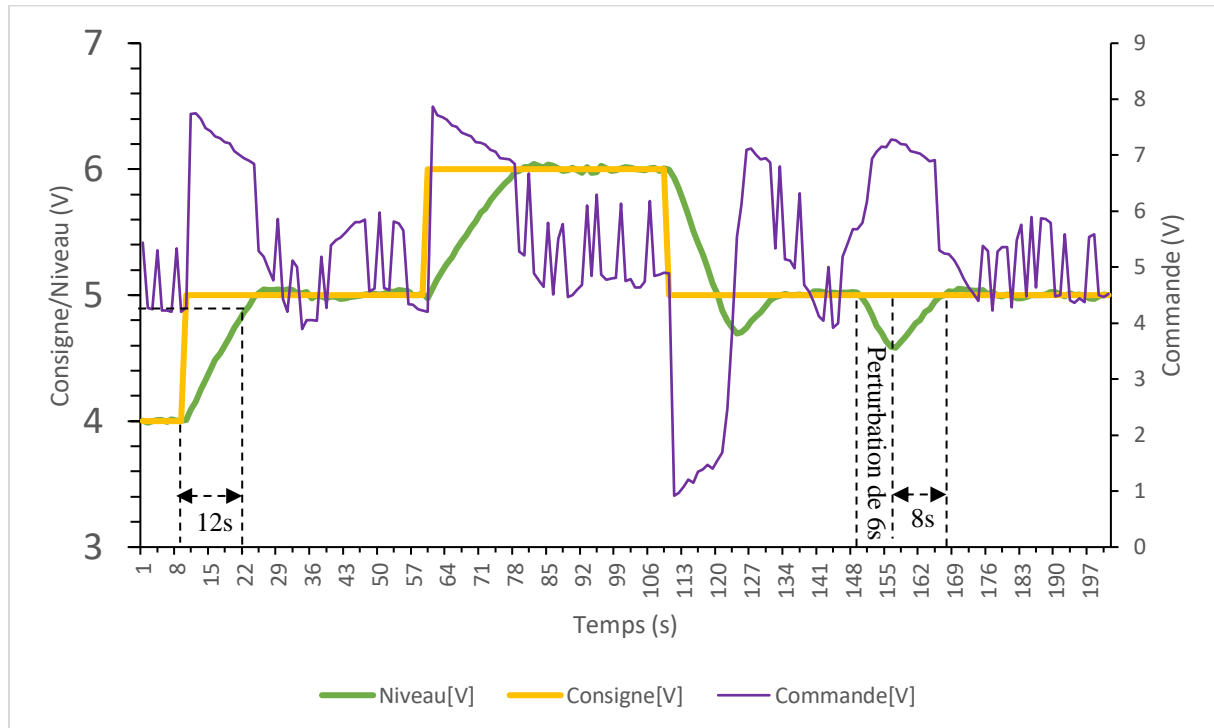


Figure 5.11 : Réponse du système régulé par automate en mode SITO.

Le régulateur en mode SITO présente des performances meilleures en mode régulation où il permet le rejet de perturbation en 8sec après leur apparition et sans dépassement néanmoins l'effet d'hystérésis de la vanne proportionnelle sur la commande est plus apparent en ce mode.

5.4.5. Etude comparative avec le régulateur classique PI

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 déterminer les paramètres du correcteur PI, nous allons utiliser la méthode de placement de pôles exposé précédemment au premier chapitre. Les paramètres obtenus sont **70sec** pour T_i et **10** pour le gain, en fonctionnement PI, l'action dérivée doit être éliminée et cela en donnant la valeur 0 à T_d . Les résultats expérimentaux obtenus sur automate sont présentés à la figure 5.12.

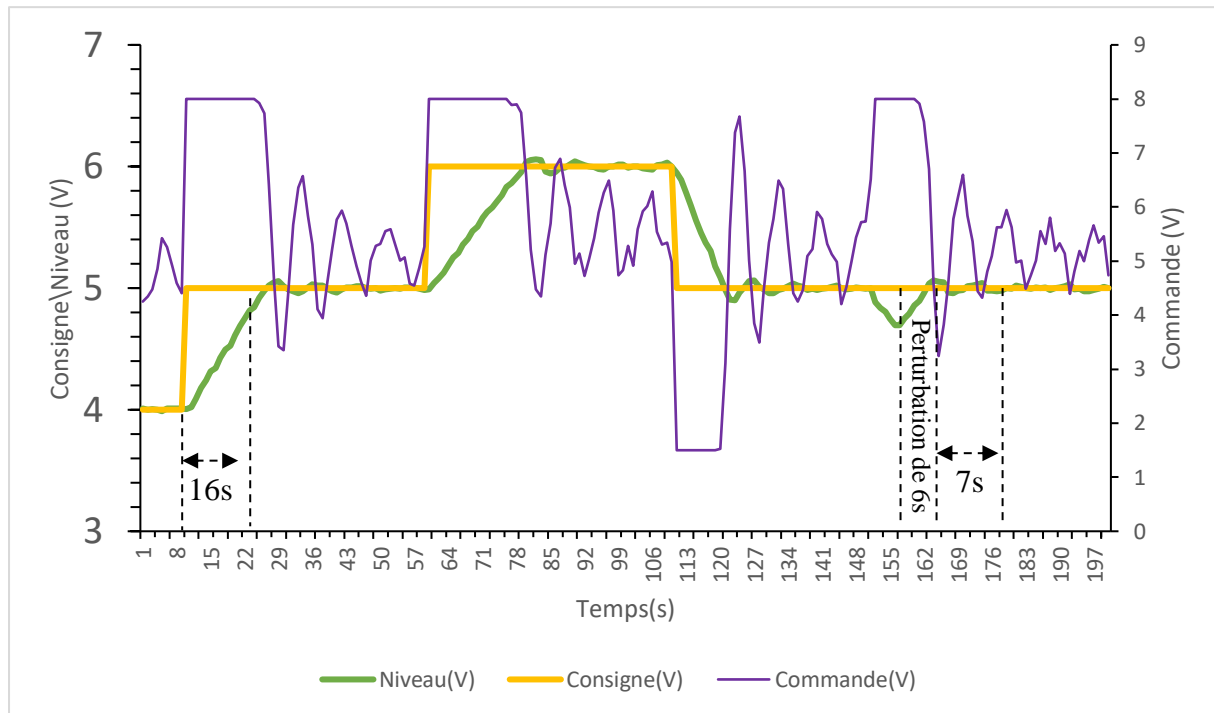


Figure 5.12 Réponse du système régulé par le correcteur PI sur automate.

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 $t_r=16\text{sec}$ Pour les échelons de tests.

A l'instant $t=150\text{sec}$ 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 7sec , avec erreur statique nulle. Le régulateur PI configuré nous permet des performances dynamiques acceptables avec élimination des écarts statique néanmoins les résultats donnés par le modèle mathématique du régulateur TITO flou simplifié en fonctionnement SISO (élimination de l'interaction) sont nettement meilleurs que ceux donnés par les autres stratégies de régulation à savoir la régulation classique PID néanmoins ce dernier l'emporte en terme de rejet de perturbation.

Le tableau 5.3 montre la comparaison entre les performances dynamiques expérimentales données par les différentes stratégies de commande.

	SISO	SITO	PI
Temps de réponse	11sec	12sec	16sec
Rejet de perturbation	14sec	8sec	7sec
Nature de la commande	Douce	Oscillatoire	Oscillatoire

Tableau 5.3 : Performances expérimentales des différentes stratégies de commande.

5.4.6. Résultats de simulation du régulateur en mode TITO

Pour tester le régulateur en mode TITO on reprend le modèle d'état des deux réservoirs en interaction présenté précédemment. Les deux entrées du système présentés sont les deux commandes sur les vannes proportionnelles (débits d'entrées q_1 et q_2) et les sorties sont les deux niveaux des réservoirs h_1 et h_2 .

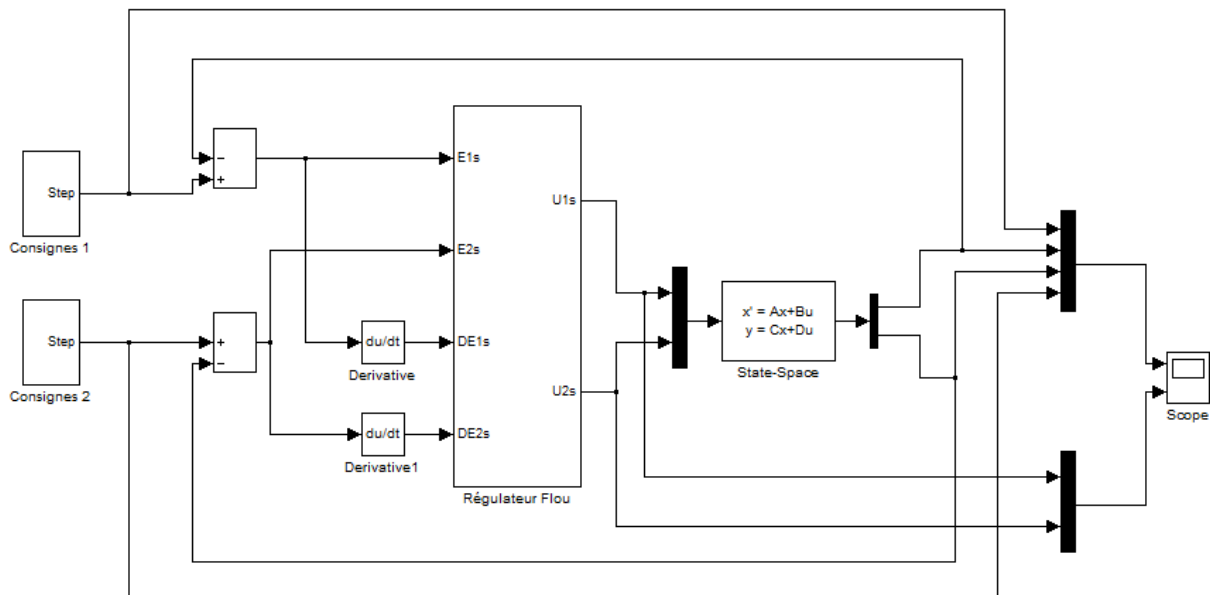
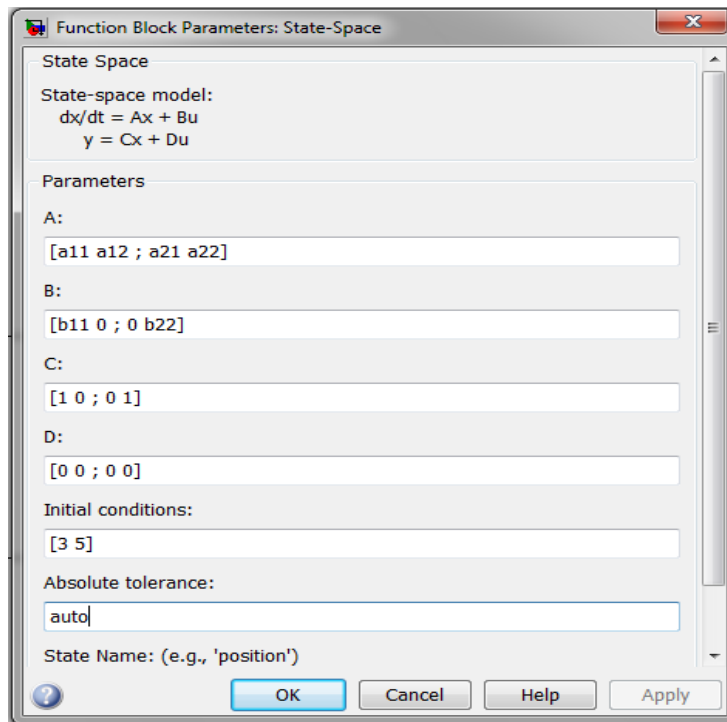


Figure 5.13 : Shéma de la régulation en mode TITO sous Simulink

Les paramètres a_{11} , a_{12} , a_{21} , a_{22} , b_{11} et b_{22} qui caractérisent le modèle d'état du système à deux réservoirs en couplage sont calculés dans l'espace de travail de Matlab par le script jointé en annexe. Ainsi le block State-Space (modèle d'état) est paramétré comme suit :



Les résultats de simulation sont présentés sur la figure 5.14.

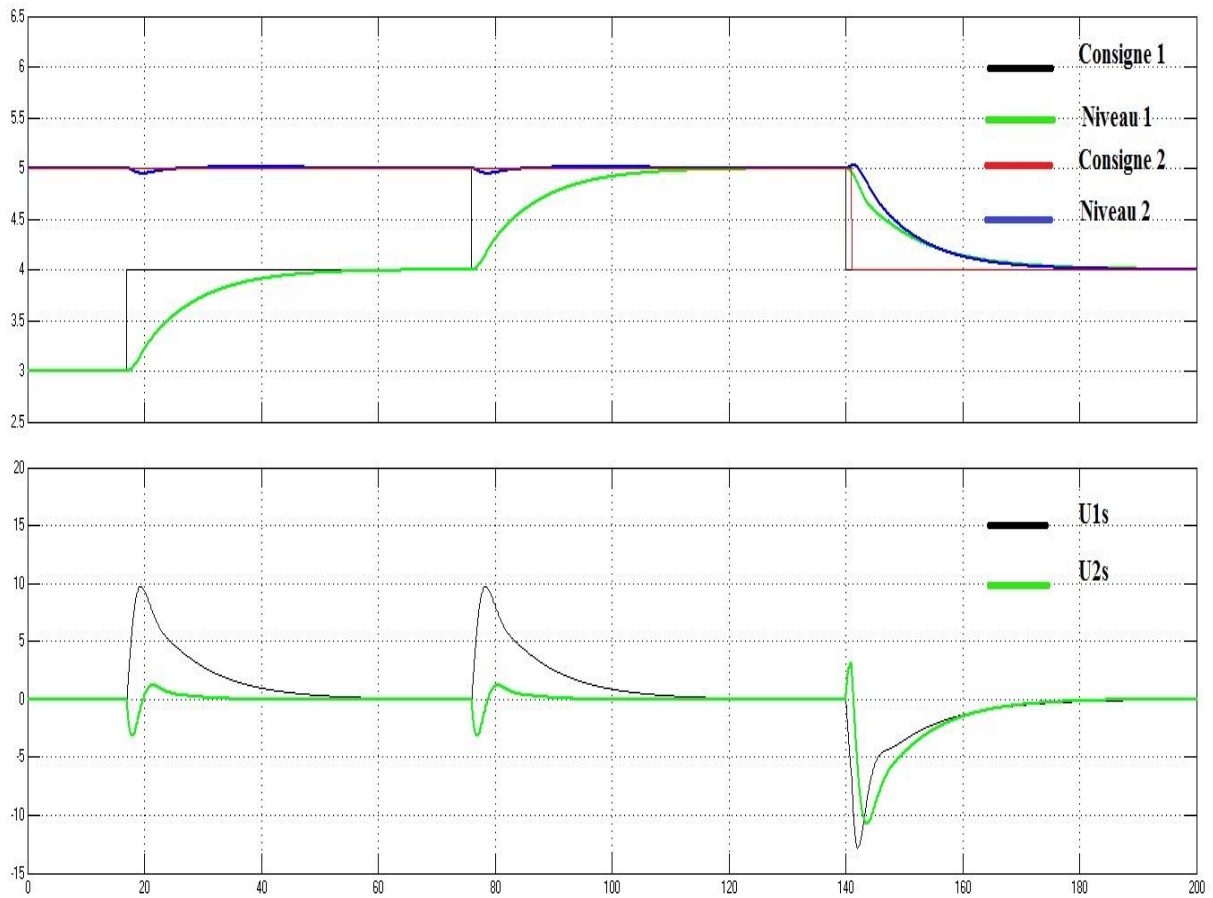


Figure 5.14 : Réponse du système régulé sous Simulink en mode TITO.

Les résultats de simulation obtenus sous Simulink montrent que le modèle mathématique du régulateur TITO flou simplifié répond aux objectifs désirés ou les sorties suivent parfaitement les consignes en assurant :

- La stabilité : les deux sorties du système sont stables.
- La précision : l'écart statique est nul pour les deux sorties du système.
- L'atténuation de l'effet de d'interaction.

5.4.7. Résultats expérimentaux du régulateur en mode TITO

Le résultat des essais expérimentaux présenté en figure 5.15 du régulateur PI flou simplifié que nous avons implémenté basé sur les travaux [13], appliqué au système à réservoirs couplés, présente un profil stable, avec de petites variations sur les profils de test.

Pour évaluer les performances de notre correcteur, nous avons choisi un profil en mode asservissement basé sur des échelons positif et négatif. Les résultats pratiques montrent que le correcteur PI flou simplifié présente des réponses stables avec une bonne poursuite du profil de test ce qui valide l'approche d'implémentation que nous avons adoptée.

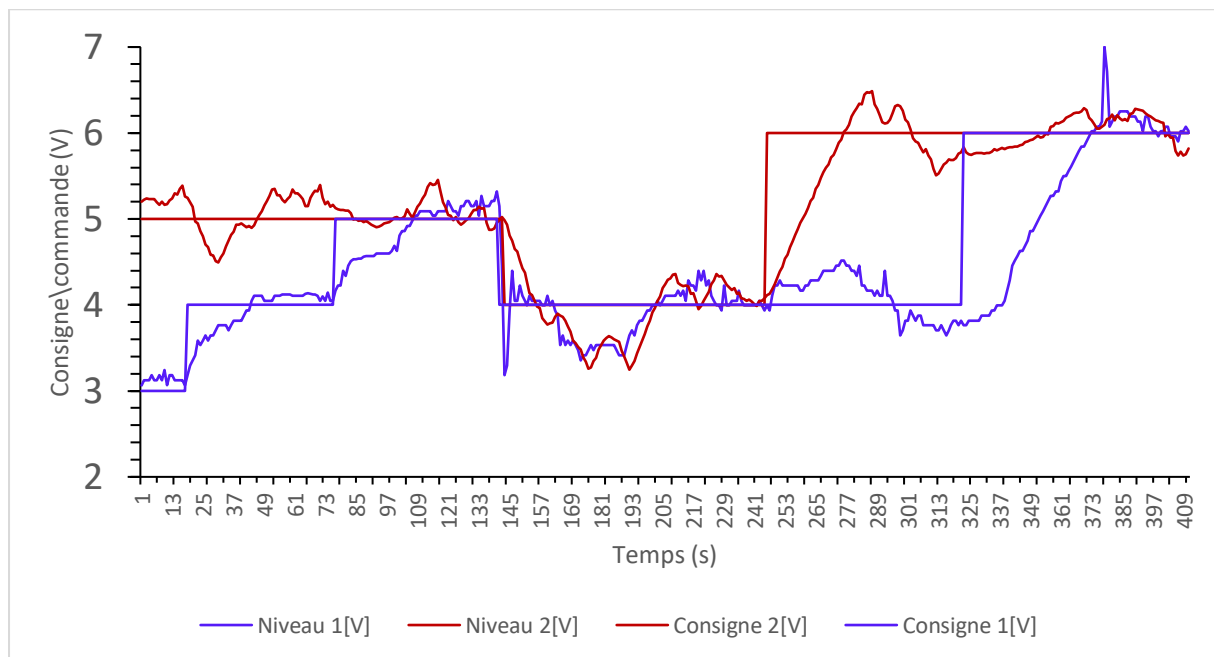


Figure 5.15 : Réponse du système à deux réservoirs couplés régulés par le régulateur TITO.

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

La vue de commande et de supervision que nous avons élaborée sous environnement WINCC flexible est simple et ne contient pas d'icône représentant le système réel car notre but est d'étudier les performances du contrôleur flou simplifié. Notre IHM (interface homme machine)

est constitué de trois parties (les courbes, l'interface d'enregistrement et la parties commandes et mesures sur le système) comme le montre la figure 5.16.

5.5.1. Zone tracé des courbes

La partie « **Courbes** » a été développée dans le but de suivre les variations des mesures par rapport aux consignes données et de voir la forme de la commande, le tout en temps réel sur l'écran de supervision. Cette zone est constituée d'un graphe où nous pouvons voir l'évolution de la mesure et la variation de la commande, l'axe des abscisses (des amplitudes) est normé de 0 à 10 volt, l'axe des ordonnées (de temps) quant à lui, a été normé de 0 à 200 secondes pour avoir une bonne période de simulation.

5.5.2. L'interface d'enregistrement

Cette partie de l'IHM communique les données d'acquisition (niveau, commande, consigne) qu'elle reçoit de l'API S7-300 vers une feuille **EXCEL** pour constituer les courbes présentées précédemment dans ce chapitre grâce au script programmé en langage **Visual Basic** sur **WinCC** et présentée dans l'annexe.

Sur le pupitre de commande l'interface d'enregistrement est constituée de :

- Un bouton (charger le script) pour lancer la récupération des données.
- Un bouton (enregistrer) qui une fois mis à 1, enregistre les données visibles dans le graphe pour les envoyer sur feuille **EXCEL**.
- Un champ de sortie (temps) pour afficher les temps d'enregistrement de la courbe.
- Un voyant représentant la cadence ou la fréquence à laquelle les données sont récupérées qui est relié au bit de cadence de la CPU. Nous l'avons fixé à 1 seconde .

5.5.3. Partie commande et mesure sur le système

Dans cette partie, nous avons placés les différents boutons et champs d'E/S nécessaires à la commande des différents contrôleurs. Celle-ci contient les éléments suivants :

- Un bouton (cmd-flou) pour activer le contrôleur flou.
- Un bouton (cmd-manuel) pour basculer vers un contrôle manuel du procédé en cas de réaction anormale des vannes proportionnelles due à une commande trop oscillante.
- Un champ de Sortie (cellule) qui affiche la cellule en cours de traitement.

- Le bouton (interaction) qui sert à gérer l'interaction des deux réservoirs en mode **TITO** en ouvrant et fermant les deux **vannes TOR**.
- Le bouton (pérturbation) qui génère la perturbation en ouvrant la **vanne TOR** pendant 6s, causant ainsi une fuite du liquide.
- Deux champs d'entrées (consigne 1, consigne 2) qui servent à introduire la consigne de niveau à laquelle on veut asservir ou régulé chaque niveau d'un réservoir.
- Deux champs de sorties (U1s, U2s) qui affichent la valeur en cours des deux commandes.
- pour niveau 1, niveau 2, débit 1 et débit 2 ceux sont des champs de sorties qui affichent les niveaux et les débits relatifs à chaque une des deux stations.

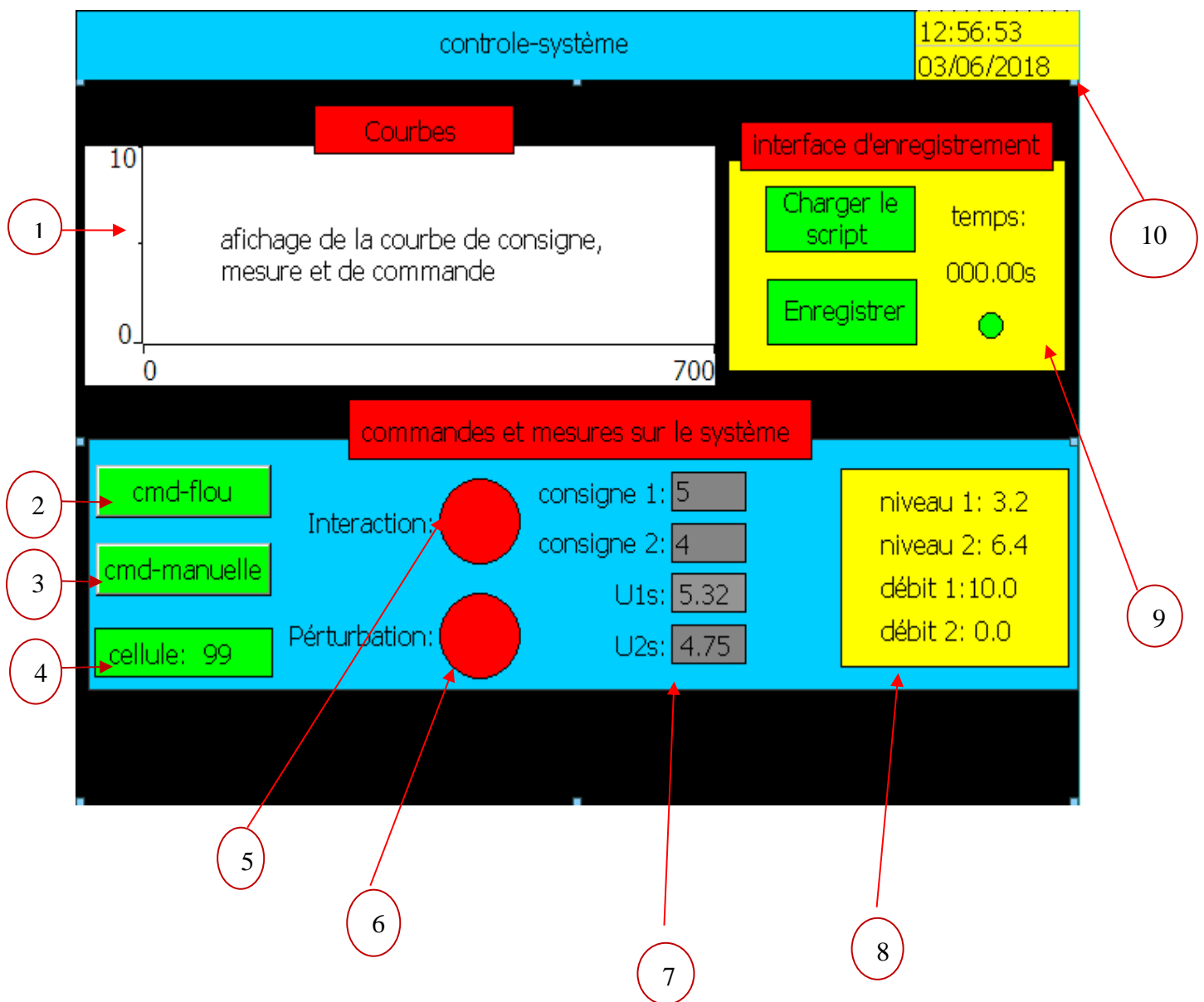


Figure 5.16 : Pupitre de contrôle du système sous WINCC flexible.

- 1) Affichage graphique de la consigne, de la commande et de la mesure.
- 2) Bouton d'activation du contrôleur flou (régulation en mode flou).
- 3) Bouton d'activation de la commande manuelle.
- 4) Champ de Sortie pour l'affichage de la cellule en cours de traitement.
- 5) Bouton d'activation de l'interaction.
- 6) Bouton d'activation de la perturbation.
- 7) Champs d'E/S pour introduire les consignes 1 et 2 ; et afficher les commandes **U1s** et **U2s** en cours.
- 8) Champs de sorties pour afficher les deux débits et les deux niveaux en cours.
- 9) Interface d'enregistrement des courbes en temps réel.
- 10) Affichage de la date et de l'heure de l'exécution de l'application.

5.6. Conclusion

Ce chapitre était consacré à la présentation de la procédure d'implémentation du modèle mathématique du régulateurs TITO PI flou simplifié présenté par **NK.Arun et BM.Mohan** en **2016**, et à la comparaison des différents résultats de simulation obtenus de l'application des différentes stratégies de commande sur Matlab/Simulink et expérimentaux sur l'automate S7 300.

Nous avons présenté notre approche pour implémenter ce régulateur sur automate en langage S7-SCL en utilisant les différents avantages fourni par ce langage évolué de programmation des APIs (S7-300).

D'après les résultats expérimentaux obtenus, nous pouvons valider cette nouvelle approche de conception de régulateur flou. Cette technique a montré ses capacités à simplifier la tâche d'implémentation des régulateurs flous sur des supports destinés à usage industriel tel que les automates programmables industriels.

Conclusion générale

La commande flou est sans doute la méthode la plus efficace et pertinente lorsqu'il s'agit de réguler des procédés industriels. Cette commande doit son efficacité au fait qu'elle utilise des techniques s'appuyant sur le savoir-faire humain. En employant des variables linguistiques les règles floues contrôlent le système, assurent le maximum de précision et la meilleure stabilité. Bien que cette méthode soit déjà assez facile à l'usage dans l'automatisme, elle a été simplifiée d'avantage grâce aux travaux de **N.K. Arun et B.M. Mohan** qui ont su exprimer un contrôleur flou simplifié à deux entrées et deux sorties, sous forme mathématique.

Le but de notre travail a donc été d'implémenter le modèle mathématique du contrôleur flou simplifié à deux entrées et deux sorties en mode **PI** avec produit d'inférence Larsen sur un automate programmable **S7-300** doté de la **CPU 314-2DP**, pour pouvoir commander la machine hydraulique **PUL2**, afin de nous assurer de son fonctionnement sur un système réel. En choisissant l'option de programmer en langage **SCL** structuré (avec blocs fonctionnelles) sous **STEP7**, nous nous sommes vite rendu compte de la flexibilité et l'aisance de l'implémentation de ce modèle, vu que l'ensemble du modèle flou standard (normalisation, fuzzification ...etc.) est exprimé sous forme d'équations mathématiques représentant les dérivées $\Delta U1S$ et $\Delta U2S$ des deux commandes normalisées **U1S** et **U2S** successivement.

Avant d'entamer les tests du contrôleur flou simplifié, nous avons procédé aux tests des capteurs de niveau et de débit, ensuite nous avons créé une interaction entre deux stations **PUL2**, grâce à leurs vannes **TOR**, pour avoir un système **TITO**. Nous avons aussi déterminé les fonctions de transfert régissant le système pour constituer la boucle de régulation où notre contrôleur reçoit les erreurs mesurées par les capteurs et délivre les commandes pour les deux actionneurs (vannes proportionnelles).

Ceci étant fait nous avons effectué le câblage **PC/AUTOMATE/PUL2**, et après le chargement de notre programme sur l'automate **S7-300**, nous avons procédé à une batterie de tests pour un contrôleur **SISO**, c'est-à-dire ; sur une seule **PUL2** (en fermant les vannes **TOR**), pour un **SITO** (cascade), un **TITO** et un contrôleur **PI standard**. La démarche suivie lors des tests était la même pour les quatre contrôleurs. Nous commençons par stabiliser le niveau sur la consigne 4, puis nous le remontons à 5 et 6 pour tester la poursuite en montée, nous le ramenons à 5 pour tester la poursuite en descente et enfin nous testons une perturbation sous forme de fuite du liquide. Les résultats obtenus montrent des performances dynamiques parfaites vu l'absence d'écarts statiques et d'oscillation de la mesure (sauf pour

le cas du **TITO** ou nous avons rencontré des défaillances du matériel), pour ce qui est de la poursuite et du rejet de perturbations les quatre contrôleurs réagissent comme suite :

- Le **SISO** a donné une poursuite parfaite de la consigne en montée et en descente avec un temps de réponse de 11s, et arrive à rejeter la perturbation en 14s (avec un léger dépassement) et la commande qu'il délivre est très stable.
- Le **PI standard** assure aussi une bonne poursuite dans les deux sens, il est plus lent en temps de réponse avec 16s, rejette la perturbation en 7s seulement mais sa commande est un peu oscillatoire.
- Le **SITO (cascade)** a une bonne poursuite en montée mais présente un dépassement en descente qui est due à l'hystérésis de la vanne proportionnelle mais offre un meilleur rejet de la perturbation en 8s et sans dépassement.
- Pour le **TITO** nous avons juste pu prouver sa fonctionnalité grâce à la courbe récupérer avant la défaillance du système, la **figure 5.10** montre que chaque mesure suit sa consigne avec des temps de réponse relativement petits mais nous ne pouvons pas en tirer d'avantage de conclusion.

Afin de comparer les résultats du système réels avec des résultats théoriques nous avons simulé les mêmes régulateurs avec les fonctions de transfert du système sous **MATLAB/SIMULINK**. Le résultat a été quasi identique et nous avons pu récupérer la courbe claire du contrôleur **TITO** qui montre de très bonne performances en profil d'asservissement où les sorties du système suivent les consignes et de précision aussi avec un écart statique nul sur les deux sorties du système.

Au terme de notre travail nous sommes arrivés à conclure que **le modèle mathématique du contrôleur flou simplifié à deux entrées et deux sorties en mode PI avec produit d'inférence larsen** est belle et bien implémentable dans un automate **S7-300** et qu'il assure de très bonnes performances statiques et dynamiques. Grâce à l'approche comparative que nous avons adoptée lors des tests, nous pouvons affirmer que chaque contrôleur possède une performance particulière. Le **SISO** assure une compensation de l'hystérésis de la vannes proportionnelle, le **SITO (cascade)** rejette la perturbation très rapidement, le **PI standard** est très rapide en temps de réponse et le **TITO** montre que le domaine multi variables ne pose pas de problème ; ce qui nous lance naturellement dans la perspective de regrouper ces quatre derniers pour obtenir un seul contrôleur qui répond à toutes les exigences.

En réalisant ce travail nous avons d'un côté approfondie notre expérience dans le domaine de la régulation des procédés automatisés, nous avons également enrichi nos capacités de programmation sous **STEP7**, **WinCC Flexible** et **MATLAB** ; et découvert des techniques d'acquisition de données tel que le **script** utilisé pour la récupération des courbes sur **EXCEL** à partir de **WinCC flexible**. Et d'un autre coté nous avons mis en œuvre les connaissances accumulées durant notre formation.

En fin, nous espérons que notre travail puisse servir de support de recherches pour les promotions à venir et nous les invitons à pousser leurs recherches encore plus loin.

Références Bibliographiques

- [1] **DR.M.Rabi**, « régulation analogique industriel », ESSTT.G.thermique, 2008
- [2] **Eric Magarotto**, « cours de régulation », support cours, 2004, université de Caen.
- [3] **Eric Vitté**, « Le magazine Schneider Electric de l'enseignement technologique et professionnel », juin 2004
- [4] **Freddy Mudry**, « Ajustage des Paramètres des Régulateurs PID », Ecole d'ingénieurs du Canton de Vaud, avril 2002
- [5] **Belkacem Ould-Bouamama**, « cours de régulation », support de cours, Mars 1998
- [6] **Pr I. Zambettakis**, « cours d'automatique », support de cours, université Toulouse 3, 2010
- [7] **Hansruedi Buhler**, « Réglage par logique floue », CH-1015 Lausanne, 1994
- [8] **C. ROSENTAL**, « Histoire de la logique floue. Une approche sociologique des pratiques de démonstration », octobre-décembre 1998.
- [9] **M.LESCIEUX**, « introduction à la logique floue, application à la commande floue », 2007
- [10] **F.Chevrie et F.Guély**, « la logique Floue », Cahier technique Schneider, Edition mars 1998
- [11] **Gonzalo Cabodevila**, « Commande et Identification Floue », École Nationale Supérieure de Mécanique et des Microtechniques de Besançon, 2005
- [12] **Pierre schulz**, « Asservissement par logique floue, le moteur d'inférence » HESGE/laboratoire/ système asservie 19 février 2002.
- [13] **Nk.Arun et Bm.Mohan**, « Mathematical modelling of the simplest fuzzy two-input two-output proportional integral or proportional derivative controller via Larsen product inference », 2016.
- [14] **Hans Berger**, « Automating with Step7 in STL and SCL », 6th revised and enlarged edition, 2012.
- [15] **Electronica Veneta**, « Manuel professeur/étudiant », Copyright INEL SPA; PP189, 2006.
- [16] **Vishal Vasistha**, « PID OUTPUT FUZZIFIED WATER LEVEL CONTROL IN MIMO COUPLED TANK SYSTEM », INTERNATIONAL JOURNAL OF MECHANICAL ENGINEERING AND TECHNOLOGY (IJMET), Aout 2013.

Logiciel utilisés

SIMATIC Manager Step7 V5.5
SIMATIC WinCC flexible 2008, Professionnel
Matlab/Simulink 2010
Microsoft Excel 2013