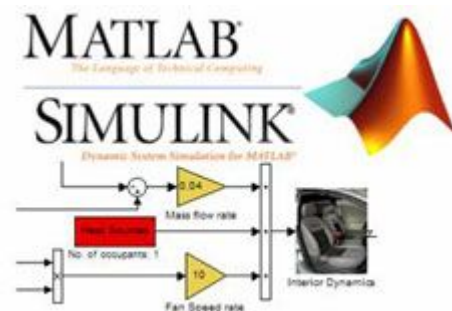
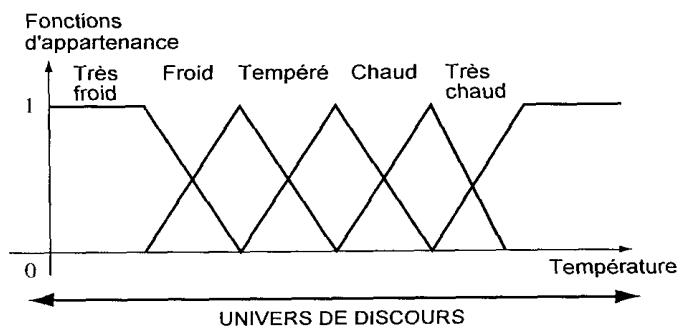




**Université Sidi Mohamed ben Abdellah  
Ecole Supérieure de Technologie - Fès  
Département Génie des procédés**



## ETUDE EN SIMULATION D'UNE RÉGULATION THERMIQUE PAR LOGIQUE FLOUE



Réalisé par :

**BOUISFI Achraf**

**BELRHALI Yasser**

Encadré par :

**Mr RABI**

**Année Universitaire 2009 / 2010**



# *Dédicace*

*Nous dédions notre rapport de fin d'études à :*

## **À NOS CHERS PARENTS :**

*Pour tous les sacrifices qu'ils ont consacrés pour notre éducation et nos études.*

## **À TOUS LES MEMBRES DE NOS FAMILLES :**

*Nous vous souhaitons plein de succès et beaucoup de bonheur dans votre vie.*

## **À NOS ENSEIGNANTS :**

*Veillez trouver dans ce travail l'expression de notre profonde reconnaissance et notre grand estime.*

---

# REMERCIEMENTS

Avant tous, nous remercions Dieu tout puissant de nous avoir aidés à porter ce travail à son terme.

Nos remerciements sont particulièrement destinés à notre cher encadrant « Mr RABI MOHAMED » pour l'aide, la collaboration, les précieux conseils et la compréhension qu'il a montré à notre égard, ainsi que pour toutes les informations qu'il n'a cessé de nous fournir pendant toute la période consacrée à la réalisation de ce modeste travail.

Nous tenons à faire parvenir nos sincères expressions de gratitude et de reconnaissance à nos enseignants, pour l'aide, la compréhension, ainsi que pour la précieuse formation qu'ils n'ont cessé de nous fournir pendant toute la période de notre formation au sein de département Génie des Procédés.

Nous remercions également nos amis et plus particulièrement ceux qui ont contribué de près ou de loin à la réalisation de ce travail.



# SOMMAIRE

Notations et symboles.....	6
Table de Figures.....	7-8
<u>Introduction</u> .....	9-10

## Partie 1: Etude théorique

I. Commande Floue d'un procédé.....	12
1- Introduction.....	12
2- Bref historique.....	12
3- Principe de la logique floue.....	13
3.1- Introduction.....	13
3.2- Variables floue.....	13-14-15
3.3- Règle d'inférence.....	15-16
3.4 Fonctions d'appartenance.....	16
4- Combinaison des règles et défuzzification.....	16
4.1 Introduction.....	16
4.2 Combinaison des règles.....	17
4.3 Défuzzification.....	17-18
4.4 Exemple d'application de Logique Floue.....	18-19-20
4.5 Conclusion.....	20
II. Loi de Commande PID.....	21
1- Schéma fonctionnel d'une boucle de régulation.....	21-22
2- Algorithme P.....	22
3- Algorithme PI.....	22-23
4- Algorithme PID.....	23-24
5- Réglage (calcul des actions PID).....	25
5.1- Méthode de Ziegler Nichols en chaine fermée.....	25
5.2- Méthode de Broïda.....	26
6- Conclusion.....	27
III. Cahier des charges.....	28
1- Stabilité.....	28
2- Précision.....	29
3- Rapidité.....	29-30
IV. Avantages et Inconvénients des régulateurs PID et Floue.....	30

## Partie 2 : Application

Présentation de Matlab / Simulink.....	32
I. Commande de la température d'eau dans une cuve : Procédé 1.....	33
1- Présentation et modélisation du Procédé 1.....	33
1-1- Schéma de principe.....	33-34
1-2- Modélisation du procédé.....	34-35
2- Commande du procédé 1.....	35
2-1- Construction du régulateur floue.....	35-36
2-2- Le régulateur PID.....	36
2-3- schéma simulink du procédé 1.....	37
3- Simulation.....	37
3-1 Commande en poursuite par PID et logique Floue.....	38-39-40-41
II Commande de la température d'une serre agricole : Procédé 2.....	41
1- Présentation du Procédé 2.....	41-42
2- Commande du Procédé 2.....	42
2-1- Construction du régulateur floue.....	42-43
2-2- Le régulateur PID.....	43
2-3- schéma simulink du procédé 1.....	43-44
3- Simulation.....	44
3-1 Commande en poursuite par PID et logique Floue.....	44-45-46-47-48
Conclusion générale.....	49
Références.....	50

# NOTATIONS ET SYMBOLES

Symbole	Désignation
$U(t)$	Commande
$\mathcal{E}$	Erreur statique
$K_R$ ou $K_P$	Gain proportionnel
$1/T_i$	Action Intégral
$Y_c$	Consigne
$Y$	Mesure
$U_0$	Valeur Centrale ( Fonctionnement nominal )
$T_d$	Action dérivée
$K_{RC}$	Gain Critique d'un régulateur
$H(s)$ ou $C(s)$	Fonction de transfert réglante
$L(s)$	Fonction de transfert perturbatrice
$P(s)$	Perturbation principale
$E(s)$	Ecart Statique
$\tau$	Retard
$U_{com}$	Tension de Commande
$U_{eff}$	Tension efficace
$C$	Capacité calorifique
$P_T$	Puissance fournie par le thermoplongeur
$e(t)$	Erreur d'entrée du régulateur Floue
$\Delta e(t)$	Variation d'erreur
$s(t)$	Sortie du régulateur Floue



## TABLE DE FIGURES

Figure 1.1 :	Représentation de la logique binaire pour les trois variables.
Figure 1.2 :	Représentation en logique Flou.
Figure 1.3 :	Exemple d'une fonction d'appartenance.
Figure 1.4 :	Degré de vérité de la variable floue.
Figure 1.5 :	Fuzzification de la température externe
Figure 1.6 :	Fuzzification de la température interne
Figure 1.7 :	Fuzzification de la puissance
Figure 1.8 :	Schéma d'une boucle de régulation.
Figure 1.9 :	Schéma général d'un système de régulation.
Figure 1.10 :	Schéma synoptique d'un régulateur PID.
Figure 1.11 :	Mise en oscillation juste entretenues d'un système
Figure 1.12 :	Choix du type de régulation
Figure 1.12.a :	Procédé stable
Figure 1.12.b :	Procédé Instable
Figure 1.13 :	Schéma d'un procédé en boucle fermée.
Figure 1.14 :	Réponse à un échelon de consigne
Figure 1.15 :	Réponse à un échelon de perturbation.
Figure 1.16 :	Temps de réponse 5% d'un système.
Figure 2.1 :	Schéma fonctionnel du procédé 1.
Figure 2.2 :	La modélisation du fonctionnement « GRADIPACK »
Figure 2.3 :	Fuzzification des deux entrées $e$ et $\Delta e$
Figure 2.4 :	Fuzzification de sortie $s$ .
Figure 2.5 :	Schéma Simulink des composants principaux de PID

<b>Figure 2.6 :</b>	<b>Schéma simulink pour la simulation du procédé 1.</b>
<b>Figure 2.7 :</b>	<b>Commande PID, changement de consigne de 30°C à 35°C.</b>
<b>Figure 2.8 :</b>	<b>Commande Floue, changement de consigne 30°C à 35°C.</b>
<b>Figure 2.9 :</b>	<b>Commande PID , changement de consigne de 30°C-50°C .</b>
<b>Figure 2.10 :</b>	<b>Commande Floue , changement de consigne 30°C-50°C.</b>
<b>Figure 2.11 :</b>	<b>Commande PID , changement de consigne 30°C-100°C</b>
<b>Figure 2.12 :</b>	<b>Commande Floue , changement de consigne 30°C-100°C.</b>
<b>Figure 2.13 :</b>	<b>Une serre agricole.</b>
<b>Figure 2.14 :</b>	<b>Fuzzification des variables d'entrée et de sortie.</b>
<b>Figure 2.15 :</b>	<b>Schéma Simulink PID.</b>
<b>Figure 2.16 :</b>	<b>Schéma Simulink , Régulateur Floue.</b>
<b>Figure 2.17 :</b>	<b>Commande flou pour un retard =1s.</b>
<b>Figure 2.18 :</b>	<b>Commande PID pour un retard =1s.</b>
<b>Figure 2.19 :</b>	<b>Commande flou pour un retard =1,5s.</b>
<b>Figure 2.20 :</b>	<b>Commande PID pour un retard= 1,5s.</b>
<b>Figure 2.21 :</b>	<b>Commande flou pour un retard= 2s.</b>
<b>Figure 2.22 :</b>	<b>Commande PID pour un retard =2s.</b>

<b>Tableau 1.1 :</b>	<b>Différents structures d'un régulateur PID.</b>
<b>Tableau 1.2 :</b>	<b>Réglages préconisés par Ziegler et Nichols.</b>
<b>Tableau 1.3 :</b>	<b>Comparatif des différents critères des régulateurs PID et Floue.</b>
<b>Tableau 2.1 :</b>	<b>Résultat d'une commande en poursuite par PID et logique floue du procédé 1</b>
<b>Tableau 2.2 :</b>	<b>Simulation avec un changement de retard.</b>



# INTRODUCTION

L'esprit humain manipule essentiellement des notions qualitatives lorsqu'il traite de l'information , et n'utilise le quantitatif qu'en deuxième ressort pour améliorer ce traitement d'information. Il manipule si mal par lui-même le quantitatif qu'il a besoin d'un support matériel pour le traiter : papier-crayon, calculateur...

Cette manipulation de notions qualitatives donne d'étonnants résultats : un individu se contente pour conduire un véhicule de notions vagues du genre « aller plus vite », « braquer un peu », etc. De la même façon, on dira qu'il fait très froid, froid, chaud ou très chaud , sans y faire correspondre une température précise. La logique ou la commande floue a pour objet la manipulation par la machine de notions imprécises. Il fournit une méthodologie rigoureuse pour la mise en œuvre de raisonnements qui, en fait, se sentent familiers.

La logique floue a été introduite en 1965 par **L.A. Zadeh**. Ses principes ont été appliqués en 1974 par E.H. Mamdani à la construction d'un premier contrôleur flou. Mais ce n'est que depuis quelques années que la commande floue a connu, essentiellement au Japon, un **essor remarquable** ; elle a été appliquée à des problèmes aussi divers que la purification de l'eau, la fabrication du ciment, la marche automatisée d'une rame de métro, etc.

L'attitude des automaticiens à l'égard de la commande floue a d'abord été réservée : après avoir, pendant des décennies, affirmé la nécessité d'identifier le mieux possible un système pour pouvoir construire une commande ayant des performances satisfaisantes, fallait-il faire confiance à cette méthode nouvelle, qui prétendait remplacer les commandes, plus ou moins sophistiquées, basées sur l'identification, par des techniques s'appuyant sur le **savoir-faire humain** plutôt que sur des équations ?

Actuellement, cette attitude a évolué. On peut penser que la commande floue va peu à peu prendre place dans la panoplie de l'ingénieur contemporain, sans supplanter les méthodes traditionnelles, et qu'elle constituera un complément précieux **dans le cas des systèmes difficilement identifiables ou dont les paramètres subissent des variations brutales.**

Aujourd'hui, plusieurs constructeurs de SNCC et d'automates proposent des algorithmes de commande floue.

Dans le présent travail, nous étudions en simulation l'application d'une commande floue à deux procédés thermiques : le premier est une régulation de la température de l'eau dans une cuve équipée d'un thermoplongeur et le deuxième est une régulation de la température d'une serre agricole. Une étude comparative entre la commande floue et la commande classique PID a été effectuée pour le premier procédé en produisant des grands changements de consigne (afin de tester la robustesse des deux commandes vis-à-vis d'une non linéarité). Et, pour le second procédé en introduisant un retard progressif (afin de tester la robustesse des deux commandes vis-à-vis des retards importants).

**PARTIE I : ETUDE THEORIQUE**



# I - Commande Floue d'un procédé

Le but de ce chapitre est de définir et de présenter ce que l'on entend par « Processeur flou ». Pour ce faire, on présente tout d'abord les principes de la logique floue, et ses applications dans le domaine du réglage, puis on décrit les différentes solutions pour un organe de commande (processeur) basé sur les principes du flou.

## 1. Introduction

Depuis quelques années déjà, on trouve sur le marché des appareils de grande consommation (appareils de photos, vidéo, ...) qui sont présentés comme faisant intervenir un réglage par logique floue ou « fuzzy-logic » ou encore « fuzzy-control ».

Au delà de l'argument publicitaire évident, il est intéressant de comprendre ce nouveau concept de réglage, et de l'appliquer à certains types de problèmes de réglage rencontrés par l'ingénieur.

Le principe du réglage par logique floue part du constat suivant: dans les problèmes de régulation auxquels il est confronté, l'homme ne suit pas, à l'image de ses inventions, un modèle mathématique fait de valeurs numériques et d'équations. Au contraire il utilise des termes tel que « un peu trop chaud, aller beaucoup plus vite, freiner à fond, etc... » ainsi que ses propres connaissances qu'il a dans le domaine. Ces connaissances sont, le plus souvent, acquises de façon empirique. Le principe du réglage par la logique floue s'approche de la démarche humaine dans le sens que les variables traitées ne sont pas des variables logiques (au sens de la logique binaire par exemple) mais des variables linguistiques, proches du langage humain de tous les jours. De plus, ces variables linguistiques sont traitées à l'aide de règles qui font référence à une certaine connaissance du comportement du système à régler. Sur la base de ce principe, différentes réalisations ont vu le jour et, actuellement, on trouve deux types d'approches pour le réglage par logique floue. Dans l'une de ces approches, les règles sont appliquées aux variables à l'aide d'une approche numérique par le biais d'un microprocesseur spécialisé ou non ou d'un ordinateur. Dans l'autre approche, les règles sont appliquées aux variables de façon analogique. Ces deux approches permettent de développer des organes de contrôle par le flou que l'on désigne par processeur digital flou ou par processeur analogique flou.

## 2. Bref historique

Les quelques points de repère historique suivant, permettent de situer dans le temps le développement de la logique floue et ses applications au réglage :

- 1965 : Le Prof. L. A. Zadeh de l'Université de Berkeley (Californie) pose les bases théoriques de la logique floue.
- 1973 : L. A. Zadeh propose d'appliquer la logique floue aux problèmes de réglage.
- 1974 : Première application du réglage par la logique floue appliquée à une turbine à vapeur. Suivie en 1980 par une application sur un four à ciment , et en 1983 sur un épurateur d'eau.
- 1985 : Premiers produits industriels (Japon) utilisant le principe de la logique floue appliqué à des problèmes de réglage et de commande.  
Développement de processeurs dédiés à des applications de réglage par la logique floue.

## 3. Principes de la logique Floue

### 3.1 Introduction

La logique floue est une branche des mathématiques et, à ce titre, toute une série de notions fondamentales sont développées. Ces notions permettent de justifier et de démontrer certains principes de base. Dans ce qui suit, on ne retiendra que les éléments indispensables à la compréhension du principe du réglage par la logique floue. Ces éléments sont :

- les variables floues
- les règles d'inférences

### 3.2 Variables floues

Contrairement aux variables binaires qui sont définies par les deux états « vrai » ou « faux », les variables floues présentent toute une gradation entre la valeur « vrai » et la valeur « faux ».

L'exemple qui suit permet de mieux saisir la distinction qui existe entre les variables binaires et les variables floues :

-Si l'on désire classer un groupe d'individu par leur taille en définissant la catégorie des petits par une taille en dessous de 160 cm, la catégorie des moyens par une taille comprise entre 160 cm et 180 cm et la catégorie des grand par une taille supérieure à 180 cm, la logique binaire donne la représentation de la figure 1.1 pour les trois variables « petit », « moyen » et « grand ».

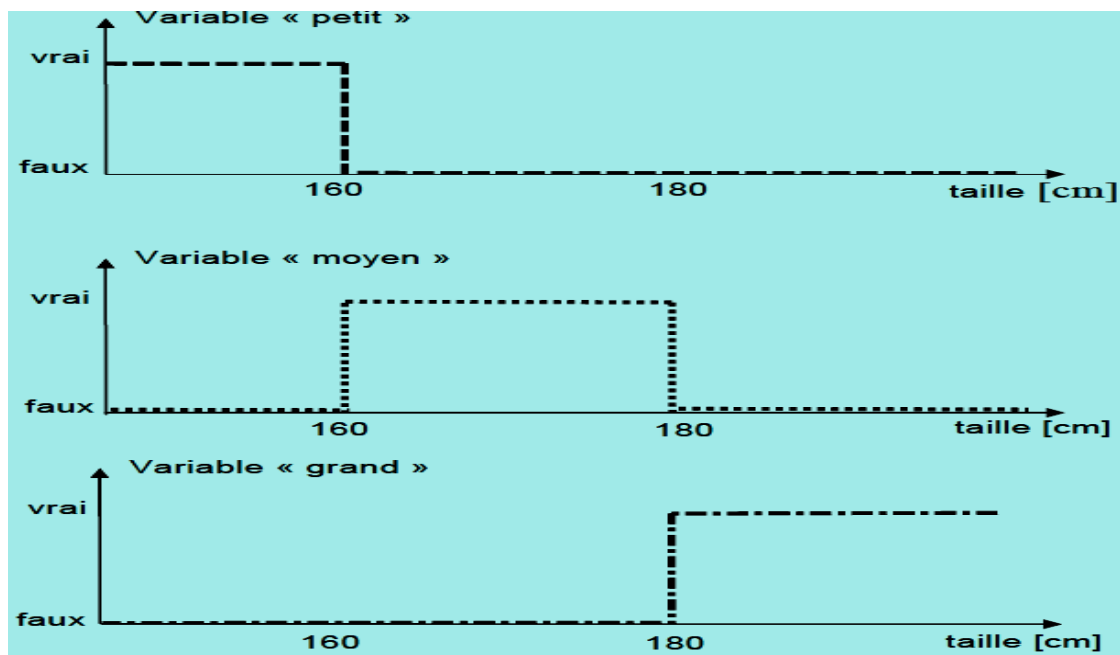


Figure 1.1 : Représentation de la logique binaire pour les trois variables.

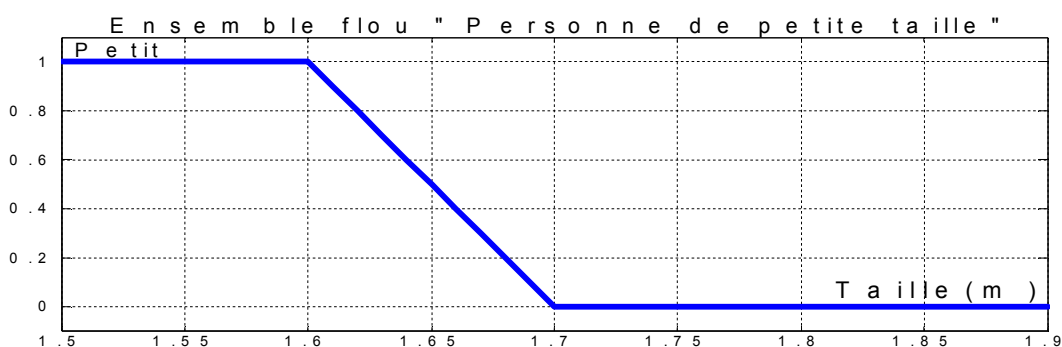
Deux remarques s'imposent au sujet de cette représentation :

D'une part, on préfère représenter l'état de la variable à l'aide de son degré de vérité, en associant la valeur 1 (degré de vérité de 100%) à la valeur « vrai » et le degré de vérité nul à la valeur « faux ».

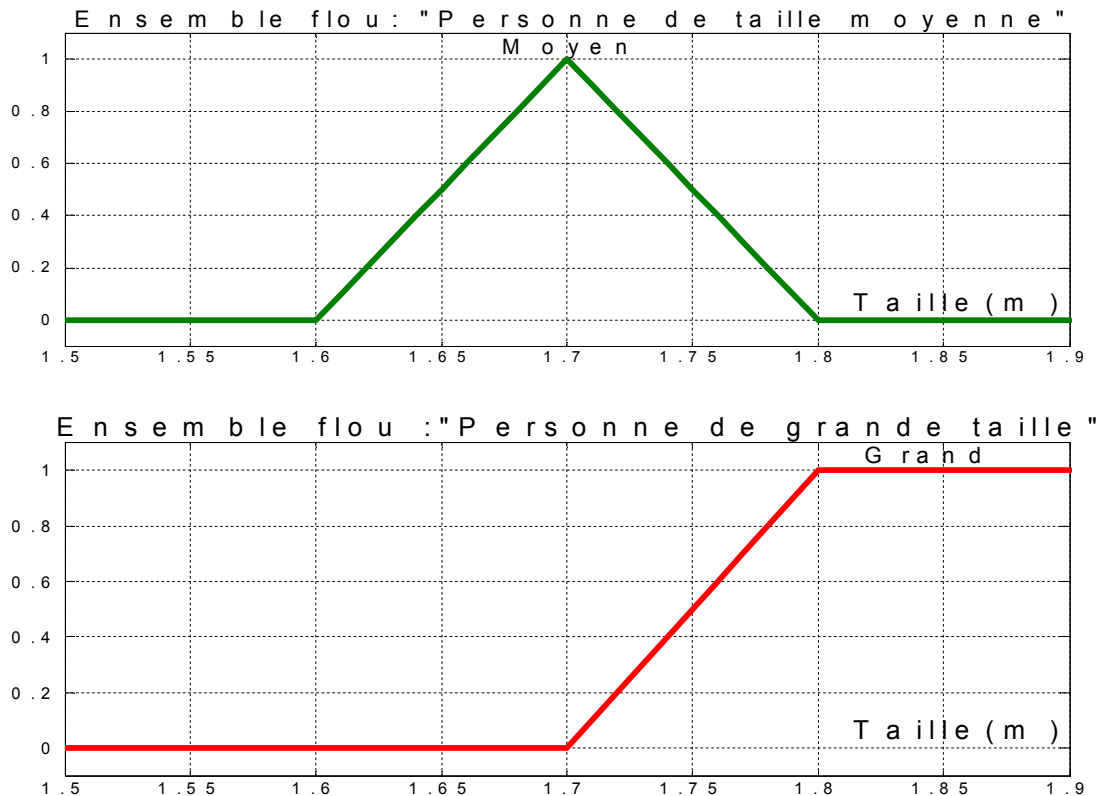
D'autre part, on constate que cette façon de faire, est très éloignée de ce que fait l'être humain lorsqu'il résout ce genre de problème. En effet, l'homme ne fait pas naturellement une distinction franche entre « petit » et « moyen » par exemple. Il utilise des expressions du genre « plutôt petit » pour qualifier une taille légèrement inférieure à 160 cm et « plutôt moyen » pour une taille légèrement supérieure à cette valeur.

En conclusion, la logique binaire présente l'avantage de la simplicité mais elle est assez éloignée de la logique utilisée naturellement par l'être humain.

Si l'on représente le même problème à l'aide de la logique floue, les variables ne sont plus binaires mais présentent une infinité de valeurs possible entre le « vrai » et le « Faux » (figure 1.2).







**Figure 1.2 : Représentation en logique Flou.**

On constate que cette représentation est beaucoup plus proche de la façon dont l'être humain raisonne puisqu'elle permet de faire intervenir des notions telles que « plutôt petit », « assez grand »... Cet avantage se fait, évidemment, au détriment de la simplicité de la représentation.

### 3.3 Règles d'inférence

On appelle règles d'inférence, l'ensemble des différentes règles reliant les variables floues d'entrée d'un système aux variables floues de sortie de ce système. Ces règles se présentent sous la forme :

**Si condition 1 et/ou condition 2 (et/ou...) alors action sur les sorties.**

L'exemple suivant, tiré de la vie quotidienne, permet d'illustrer ceci :

- Lorsque l'on prend une douche, un des problèmes qui se présente est de régler la température de l'eau. La variable d'entrée du système homme-douche est la température de l'eau mesurée à l'aide de nos capteurs de température. Les variables de sorties sont les deux robinets eau chaude et eau froide. Dans la pratique, le réglage de la température se fait en utilisant notre expérience, expérience qui recouvre à la fois nos préférences et notre

connaissance de l'installation sanitaire utilisée. Ce réglage se fait en utilisant des règles du genre :

- ❖ Si la température est très froide **alors** ouvrir à fond l'eau chaude
- ❖ Si la température est un peu trop froide **alors** fermer un peu l'eau froide
- ❖ Si la température est bonne **alors** laisser les deux robinets dans leur état .
- ❖ Si la température est trop chaude **alors** fermer un peu l'eau chaude et ouvrir un peu l'eau froide.

En termes d'intelligence artificielle, ces règles résument en fait l'expérience de l'expert et elles ne sont en général pas définissables de façon unique puisque chaque individu crée ses propres règles.

### 3.4 Fonctions d'appartenance

Il s'agit d'établir une relation entre le degré de vérité de la variable floue et la grandeur d'entrée correspondante (figure 1.3), On parle de fuzzification :

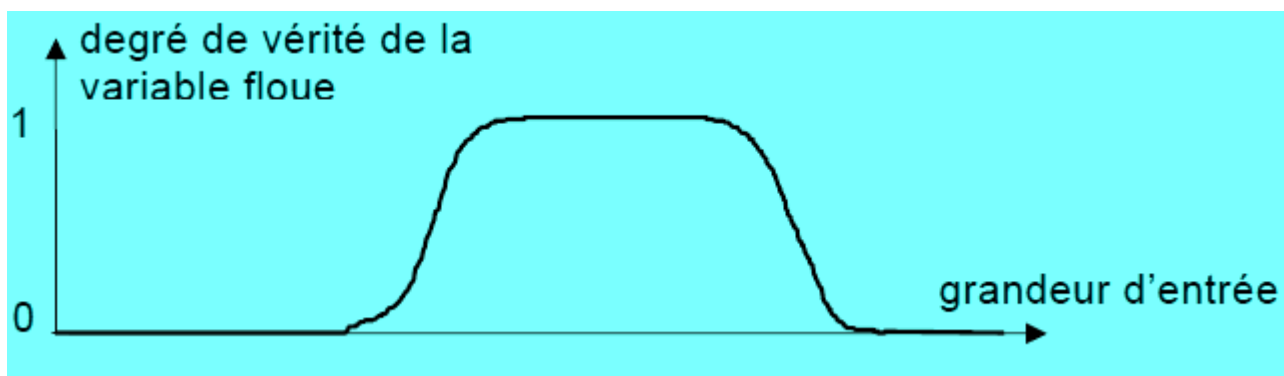


Figure 1.3 : Exemple d'une fonction d'appartenance.

On peut évidemment choisir n'importe quelle forme pour les fonctions d'appartenance. Cependant, en pratique, on utilise les formes trapézoïdales (cas de la figure 1.2).

## 4. Combinaison des règles et défuzzification

### 4.1 Introduction

Les différentes règles d'inférences produisent chacune, une valeur. Ces différentes valeurs doivent être combinées afin d'obtenir la (éventuellement les) variable(s) de sortie. Ensuite la (ou les) variable(s) floue(s) de sortie doivent être converties en une grandeur de

commande (débit, température...) afin d'être appliquée au système à régler. On appelle cette dernière étape la défuzzification.

## 4.2 Combinaison des règles

L'ensemble des règles se présente sous la forme d'une énumération du 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

...

La combinaison de ces différentes règles se fait à l'aide de l'opérateur **ou**. La justification du choix de l'opérateur se fonde sur la pratique du langage courant : en effet, une telle énumération est comprise dans le sens

Si... alors... **ou**

Si... alors...**ou**

...

Bien que l'opérateur **ou** ne soit pas explicitement mentionné.

## 4.3 Défuzzification

Les valeurs obtenues lors de la combinaison des règles appliquées aux intervalles flous de la variable de sortie, définies une fonction d'appartenance.

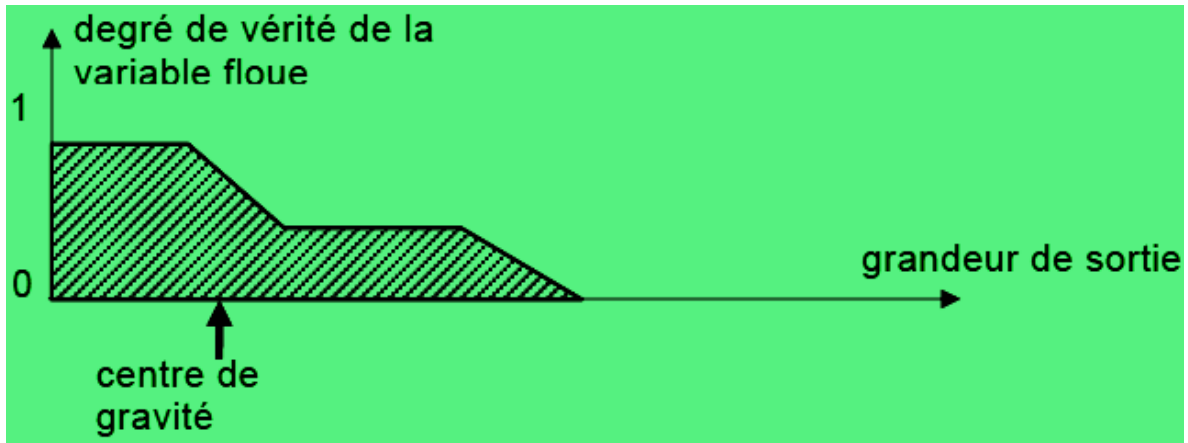
Il s'agit de convertir cette information en une grandeur physique. Plusieurs façons de faire, peuvent être envisagées mais, en pratique, on utilise surtout les deux méthodes suivantes :

- Défuzzification par calcul du centre de gravité.
- Défuzzification par calcul du maximum.

En général, c'est la défuzzification par calcul du centre de gravité qui est la plus utilisée.

### Défuzzification par calcul du centre de gravité

Il s'agit de calculer le centre de gravité de la fonction d'appartenance de la variable de sortie.



**Figure 1.4 :**

#### **Degré de vérité de la variable floue.**

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

#### 4.4. Exemple d'application de Logique Floue

##### Description du problème

On souhaite commander l'installation de chauffage d'un immeuble à l'aide d'un contrôleur flou. On dispose de deux sondes de température : l'une à l'extérieur de l'immeuble ; l'autre à l'intérieur (grandeur interne). Sur la base de ces deux mesures et en faisant appel aux règles d'inférences, le contrôleur flou doit régler la puissance de l'installation de chauffage.

##### Fuzzification de la température externe

On choisit deux intervalles flous et des fonctions d'appartenances de type trapézoïdales en définissant le « froid » comme correspondant à une température inférieure à 5 °C , et le « chaud » comme étant une température supérieure à 20 °C (figure 1.5)

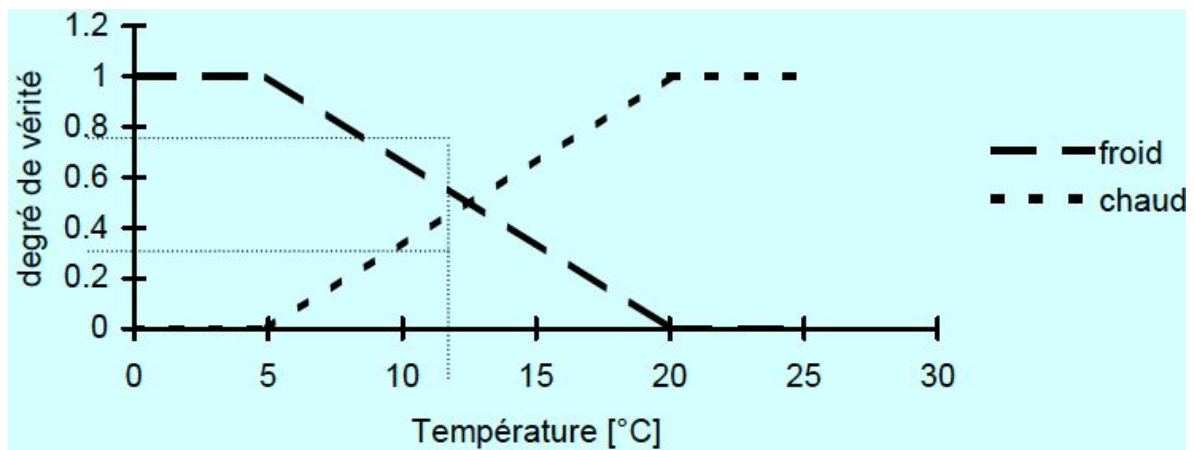


Figure 1.5 : Fuzzification de la température externe

### Fuzzification de la température interne

On choisit trois intervalles flous et des fonctions d'appartenance de type trapézoïdales en définissant le « froid » comme correspondant à une température inférieure à 15 °C, le « bon » comme étant une température comprise entre 19 °C et 21 °C et le « chaud » comme étant une température supérieure à 25 °C (figure 1.6).

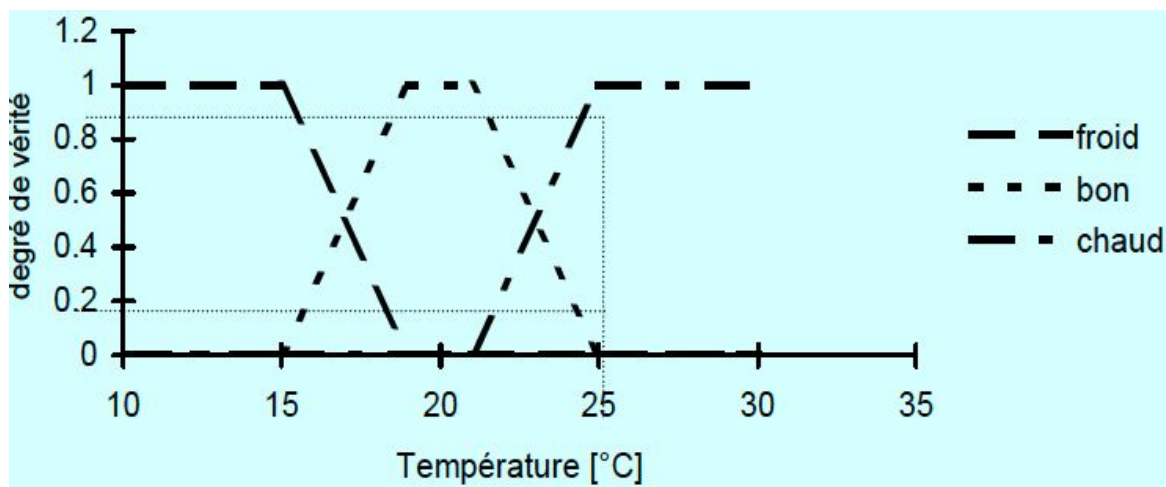


Figure 1.6 : Fuzzification de la température interne

### Fuzzification de la puissance

On choisit quatre intervalles flous pour définir la puissance de l'installation avec des fonctions d'appartenance en forme de raies. On définit les valeurs suivantes :

Valeur de Puissance en %  
Nulle            0

Faible	33
Moyenne	67
Maximale	100

Ce qui définit les fonctions d'appartenances illustrées à la figure 1.7.

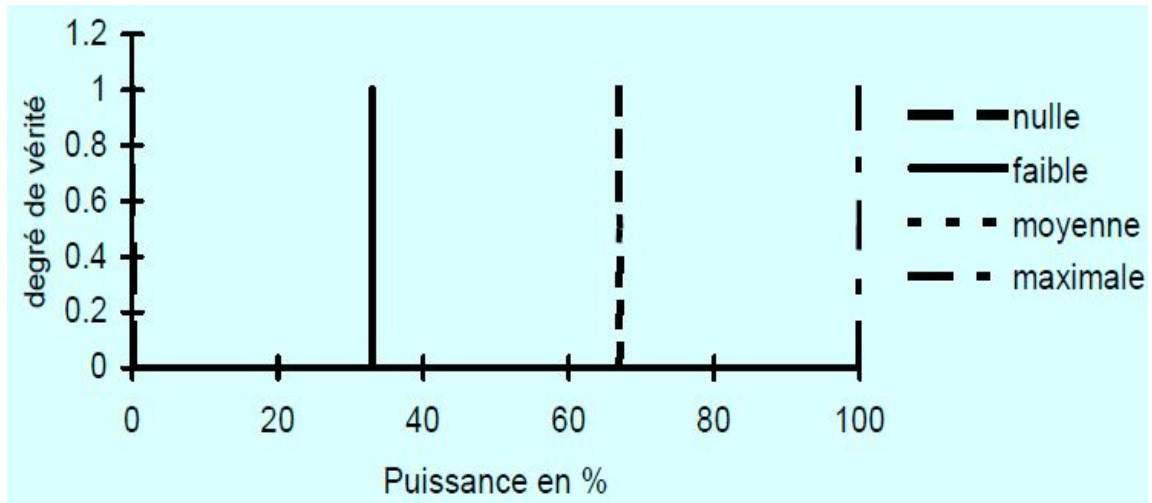


Figure 1.7 : Fuzzification de la puissance

### Règles d'inférences

L'expérience acquise sur l'installation de chauffage a permis de définir les six règles suivantes :

1. **Si** la température extérieure est « froide » **et** la température intérieure est « froide » **alors** mettre la puissance au « maximum »
2. **Si** la température extérieure est « froide » **et** la température intérieure est « bonne » **alors** mettre une puissance « moyenne »
3. **Si** la température extérieure est « froide » **et** la température intérieure est « chaude » **alors** mettre une puissance « faible »
4. **Si** la température extérieure est « chaude » **et** la température intérieure est « froide » **alors** mettre une puissance « moyenne »
5. **Si** la température extérieure est « chaude » **et** la température intérieure est « bonne » **alors** mettre une puissance « faible »
6. **Si** la température extérieure est « chaude » **et** la température intérieure est « chaude » **alors** mettre une puissance « nulle »

### 4.5 Conclusion



Un contrôleur Floue suit un processus qui se présente dans trois étapes successives :

- ✚ **Fuzzification des entrées:** transforme les valeurs d'entrées en quantités floues.
- ✚ **Inférence (avec la base des règles):** applique les règles aux entrées fuzzifiées afin de prendre les décisions.
- ✚ **Défuzzification:** transforme les décisions floues en valeurs de sorties numériques déterminées.

## II - La Loi de commande analogique PID

### 1. Schéma fonctionnel d'une boucle de régulation

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

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

Elle est souvent complétée par :

- un enregistreur.
- des convertisseurs.
- des sécurités.

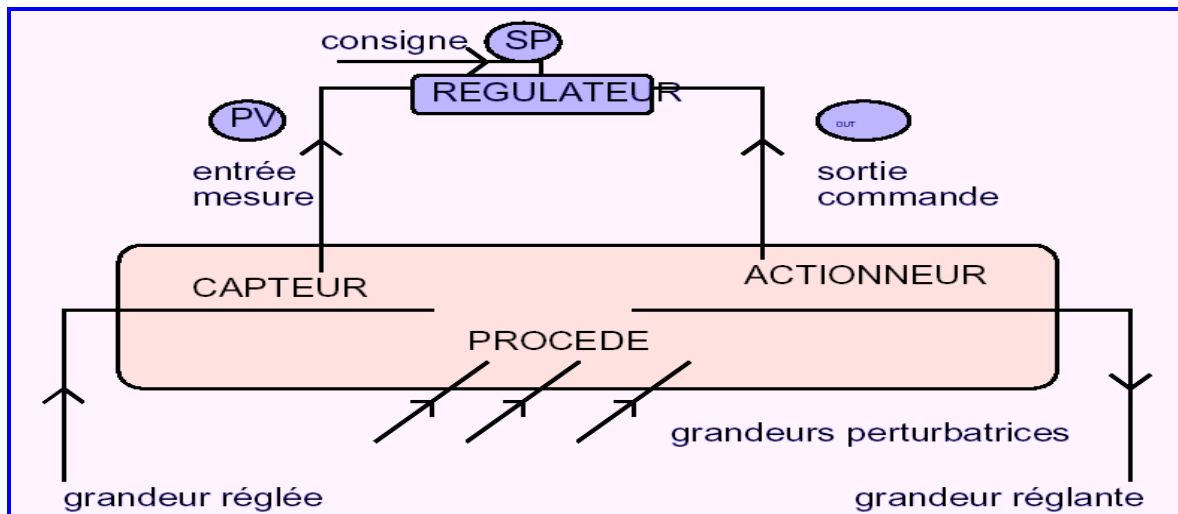


Figure 1.8 : Schéma d'une boucle de régulation.

Le régulateur reçoit deux informations :

- Le signal de mesure (M ou PV) provenant du capteur.
- La consigne (S ou SP) qui peut être locale ou externe.

En fonction de l'écart entre ces deux valeurs et de l'algorithme de calcul pour lequel il a été configuré, il délivre un signal de sortie (S ou OUT) dirigé vers l'actionneur afin d'annuler cette écart et de ramener la mesure vers la valeur de consigne.

## Action de régler :

La plupart des systèmes comportent une réaction naturelle qui s'oppose à l'action et conduit à un nouvel état d'équilibre. Cet état n'est que rarement satisfaisant. On est donc amené à concevoir des ensembles, dans lesquels la grandeur désirée s'aligne rigoureusement sur une grandeur de consigne. On y parvient en appliquant une **rétroaction**.

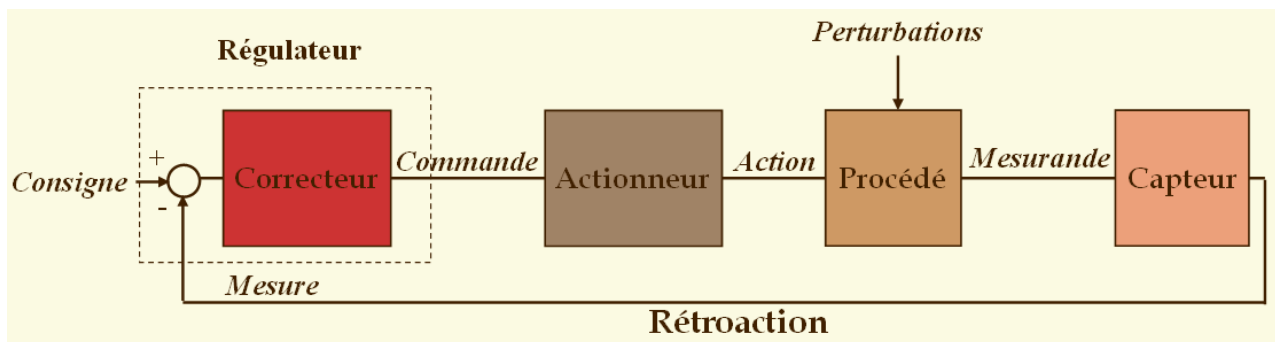


Figure 1.9 : Schéma général d'un système de régulation.

## 2. Algorithme P

Le rôle de l'action proportionnelle est de minimiser l'écart  $\varepsilon$  entre la consigne et la mesure, et elle réduit le temps de monter et le temps de réponse. On constate qu'une augmentation du gain  $K_p$  du régulateur entraîne une diminution de l'erreur statique et permet d'accélérer le comportement global de la boucle fermée.

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

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

En cas d'un régulateur P, et pour définir le sens d'action qui détermine le sens de variation de mesure et de consigne et on écrit :

$$u = u_0 \pm K_R (y_c - y)$$

Avec  $u_0$  est la valeur centrale.

## 3. L'Algorithme PI

Le correcteur intégral est en général associé au correcteur proportionnel, il élabore alors une commande qui peut être donnée par la relation suivante:

La fonction de transfert du correcteur est alors donnée par :

$$u(t) = K_p \left( \varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(\tau) d\tau \right)$$

$$C(p) = K_p \frac{1 + T_i p}{T_i p}$$

Pour un régulateur intégral pur, le régime dynamique est relativement long. D'un autre côté, le régulateur proportionnel réagit immédiatement aux écarts de réglage mais il n'est pas en mesure de supprimer totalement l'erreur statique. La combinaison des actions proportionnelle et intégrale permettent d'associer l'avantage du régulateur P, c'est-à-dire la réaction rapide à un écart de réglage, à l'avantage du régulateur I qui est la compensation exacte de la grandeur pilote.

La transposition de correcteur continu consiste à discrétiser un correcteur continu afin de l'utiliser dans une commande numérique.

## 4. L'Algorithme PID

Le régulateur standard le plus utilisé dans l'industrie, est le régulateur PID (proportionnel intégral dérivé), car il permet de régler à l'aide de ses trois paramètres, les performances (amortissement, temps de réponse, ...) d'un processus modélisé par un deuxième ordre. Nombreux sont les systèmes physiques qui, même en étant complexes, ont un comportement voisin de celui d'un deuxième ordre. Par conséquent, le régulateur PID est bien adapté à la plupart des processus de type industriel, et il est relativement robuste par rapport aux variations des paramètres du procédé, quand on n'est pas trop exigeant sur les performances de la boucle fermée par rapport à celles de la boucle ouverte (par exemple, accélération très importante de la réponse ou augmentation très importante de l'amortissement en boucle fermée).

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



Figure 1.10 : Schéma synoptique d'un régulateur PID.

L'action conjuguée PID permet une régulation optimale en associant les avantages de chaque action : la composante P réagit à l'apparition d'un écart de réglage, la composante D s'oppose aux variations de la grandeur réglée et stabilise la boucle de régulation ; et la composante I élimine l'erreur statique. Et c'est pour cela que ce type de correcteur est le plus utilisé en milieu industriel.

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

1. Il fournit un signal de commande en tenant compte de l'évolution du signal de sortie par rapport à la consigne.
2. Il élimine l'erreur statique grâce au terme intégrateur.
3. Il anticipe les variations de la sortie grâce au terme dérivateur.

La commande  $U(t)$  est donnée par le régulateur PID, dans sa forme Classique, elle est décrite par :

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

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

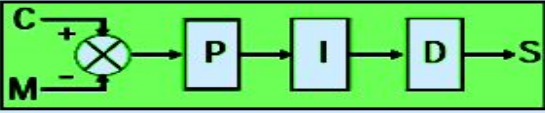
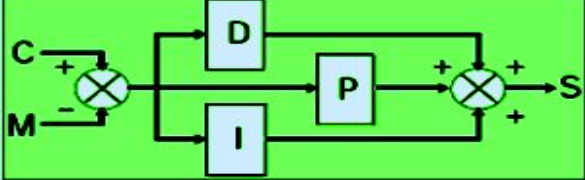
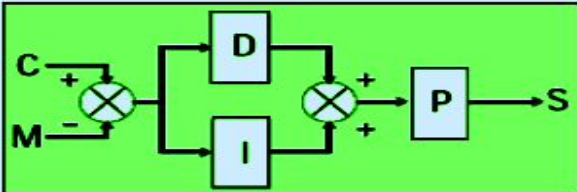
Structure du régulateur PID	Schéma et fonction de transfert
Série	 $K_p \left( \frac{T_i + T_d}{T_i} + \frac{1}{pT_i} + pT_d \right)$
Parallèle	 $K_p + \frac{1}{pT_i} + pT_d$
Mixte	 $K_p \left( 1 + \frac{1}{pT_i} + pT_d \right)$

Tableau 1.1 : Différents structures d'un régulateur PID.

## 5. Réglage (calcul des actions PID)

### 5.1- Méthode de Ziegler Nichols en chaîne fermée

Cette méthode, établie vers 1942, est la plus connue des méthodes pratiques de réglage des boucles de régulation. L'avantage de cette méthode est qu'il n'y a pas besoin de connaître la fonction de transfert du procédé et que le réglage se fait directement sur le site en chaîne fermée.

Après avoir porté la mesure près du point de consigne désiré (manuellement ou automatiquement), le régulateur est réglé en action proportionnelle seule. Le gain  $K_R$ , est alors augmenté progressivement jusqu'à obtention du pompage sans que l'organe réglant ne soit jamais en saturation. Le gain critique  $K_{RC}$  du régulateur est le plus petit gain qui permet l'entretien des oscillations. La période d'oscillation est mesurée sur l'enregistrement (figure 1.11).



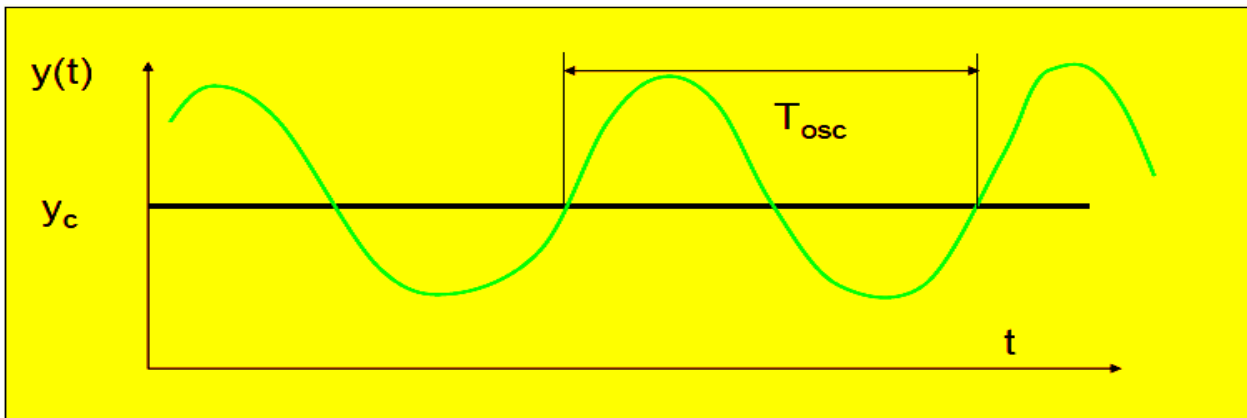


Figure 1.11 : Mise en oscillation juste entretenues d'un système

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

	P	PI	PI parallèle	PID série	PID parallèle	PID mixte
$K_{RC}$	$\frac{K_{RC}}{2}$	$\frac{K_{RC}}{2.2}$	$\frac{K_{RC}}{2.2}$	$\frac{K_{RC}}{3.3}$	$\frac{K_{RC}}{1.7}$	$\frac{K_{RC}}{1.7}$
$T_i$	$T_i$ maxi ou annulée	$\frac{T_{osc}}{1.2}$	$\frac{2 \cdot T_{osc}}{K_{RC}}$	$\frac{T_{osc}}{4}$	$\frac{0.85 T_{osc}}{K_{RC}}$	$\frac{T_{osc}}{2}$
$T_d$	0	0	0	$\frac{T_{osc}}{4}$	$\frac{K_{RC} T_{osc}}{13.3}$	$\frac{T_{osc}}{8}$

Tableau 1.2 : Réglages préconisés par Ziegler et Nichols.

## 5.2- Méthode de Broïda

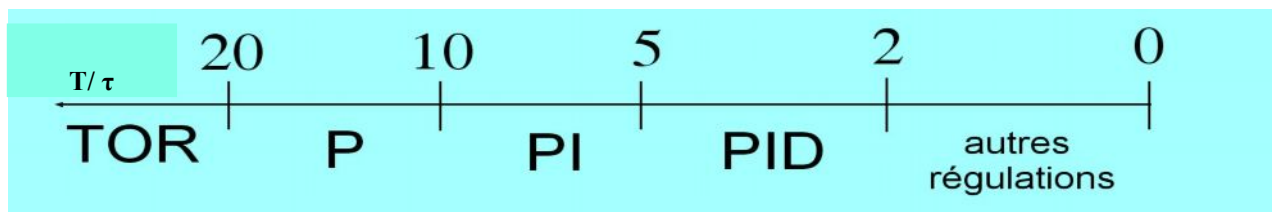
Un procédé de fonction de transfert du premier ordre, ne pose pas de problème de stabilité ; il peut être contrôlé par un régulateur à action proportionnelle avec ou sans action intégrale. Mais si un procédé similaire comporte un retard important, alors la stabilité en chaîne fermée devient un problème ; le régulateur devra posséder une action de plus et deviendra un régulateur PID.

Les deux modèles de Broïda proposés sont très utilisés industriellement ; Il s'agit du modèle de Broïda :

$H_1(s)$  concernant un procédé autoréglant, et d'un modèle intégrateur comprenant un retard pur  $H_2(s)$ .

$$H_1(s) = K \frac{e^{-\tau s}}{1 + Ts} \quad , \quad H_2(s) = \frac{k e^{-\tau s}}{s}$$

L'échelle de choix présentée (figure 1.12 a et b) permet de savoir quel type de régulation est nécessaire à mettre en place, et cela en fonction des constantes de la fonction de transfert du procédé.



(a) : procédé stable.

	P		PI		PID	
	Série Mixte Paral.	Série Mixte	Paral.	Série	Paral.	Mixte
$K_R$	$\frac{0,8}{k \cdot \tau}$	$\frac{0,8}{k \cdot \tau}$	$\frac{0,8}{k \cdot \tau}$	$\frac{0,85}{k \cdot \tau}$	$\frac{0,9}{k \cdot \tau}$	$\frac{0,9}{k \cdot \tau}$
$T_i$		$5 \cdot \tau$	$\frac{k \cdot \tau^2}{0,15}$	$4,8 \cdot \tau$	$\frac{k \cdot \tau^2}{0,15}$	$5,2 \cdot \tau$
$T_d$				$0,4 \cdot \tau$	$\frac{0,35}{k}$	$0,4 \cdot \tau$

(b) : procédé instable.

Figure 12 : Choix du type de régulation

## 6. Conclusion

Le régulateur PID (action proportionnelle - intégrale - dérivée filtrée) est une combinaison d'un régulateur P, un régulateur I et un régulateur D.

La robustesse est sans doute le paramètre le plus important et délicat. On dit qu'une régulation est robuste si elle fonctionne toujours même si le modèle change ses paramètres.

Il faut trouver des compromis, le régulateur idéal n'existe pas. En général on se fixe un cahier des charges à respecter sur la robustesse, en termes de dépassement, temps de réponse, et l'erreur statique suite à un changement de consigne ou de perturbations.

La régulation par correcteur PID est sans doute l'une des régulations les plus utilisées. Il existe des méthodes très précises qui permettent de déterminer les coefficients du correcteur.

# III - Cahier des charges

Pour caractériser les performances d'un procédé en boucle fermée, on utilise trois paramètres :

- stabilité
- précision
- rapidité

## 1 - Stabilité

### Définition :

Un procédé asservi ou non est stable si à une variation bornée du signal d'entrée correspond une variation bornée du signal de sortie. Une variation d'un signal est dite bornée lorsqu'elle est constante en régime permanent.

### Critères géométriques de stabilité

Ces critères géométriques permettent par l'étude de la représentation de la fonction de transfert en boucle ouverte FTBO dans les plans de Bode, Nyquist ou Black, de déterminer la stabilité d'un système en boucle fermée.

Considérons un procédé en boucle fermée à retour unitaire de fonction de transfert réglante  $H(s)$  et perturbatrice  $L(s)$ .

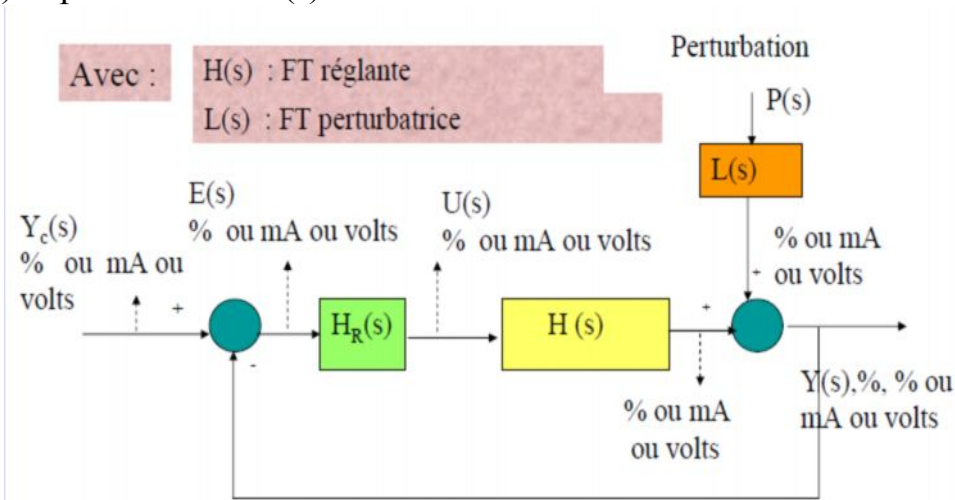


Figure 1.13 : Schéma d'un procédé en boucle fermée.

Avec

$$Y(s) = FTBF(s).Y_c(s) + LBF(s)P(s)$$

$$FTBF(s) = \frac{FTBO(s)}{1+FTBO(s)} ; LBF(s) = \frac{L(s)}{1+FTBO(s)} \quad \text{et } FTBO(s) = H_R(s).H(s)$$

## 2 - Précision

On souhaite que la sortie du système, soit la plus proche possible de l'entrée. Pour cela, il suffit que l'écart  $\varepsilon_0$  soit le plus faible possible.

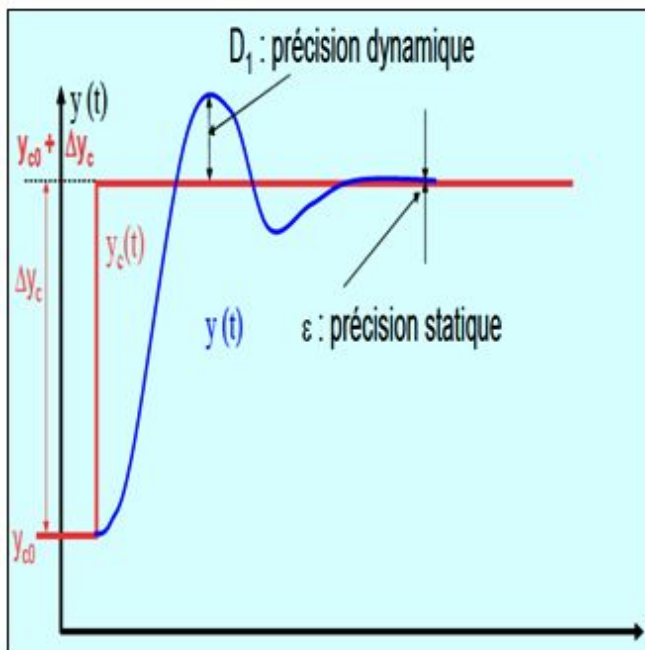


Figure 1.14 : Réponse à un échelon de consigne

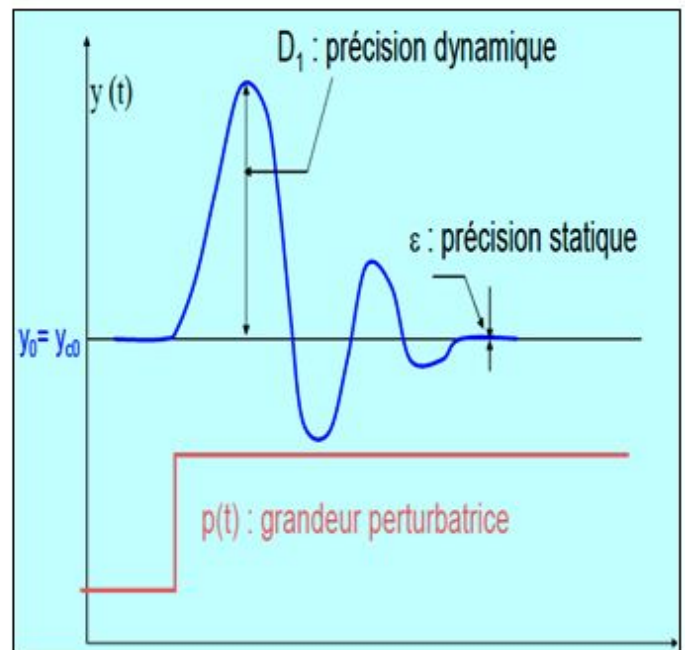


Figure 1.15 : Réponse à un échelon de perturbation

La précision dynamique : est caractérisée par le dépassement  $D1$  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.

La précision statique:

Soit  $e(t)$  l'écart entre la consigne  $y_c(t)$  et la mesure de la grandeur réglée  $y(t)$ . On a aussi  $E(s) = Y_c(s) - Y(s)$ . Donc l'écart résiduel ou écart statique, en régime permanent est :

$$\varepsilon = \lim_{t \rightarrow \infty} E(t) = \lim_{s \rightarrow 0} S.E(s)$$

## 3 - Rapidité

Le critère standard de rapidité utilisé est le temps de réponse à 5% de la sortie lorsque le système est soumis à une entrée en échelon. Pour un système bouclé c'est la FTBF(s) qu'il faut considérer, l'entrée est la consigne  $y_c(t)$  et la sortie est la mesure  $y(t)$  : grandeur réglée. La réponse à une entrée en échelon d'un système dynamique linéaire stable se présente en général sous la forme suivante :

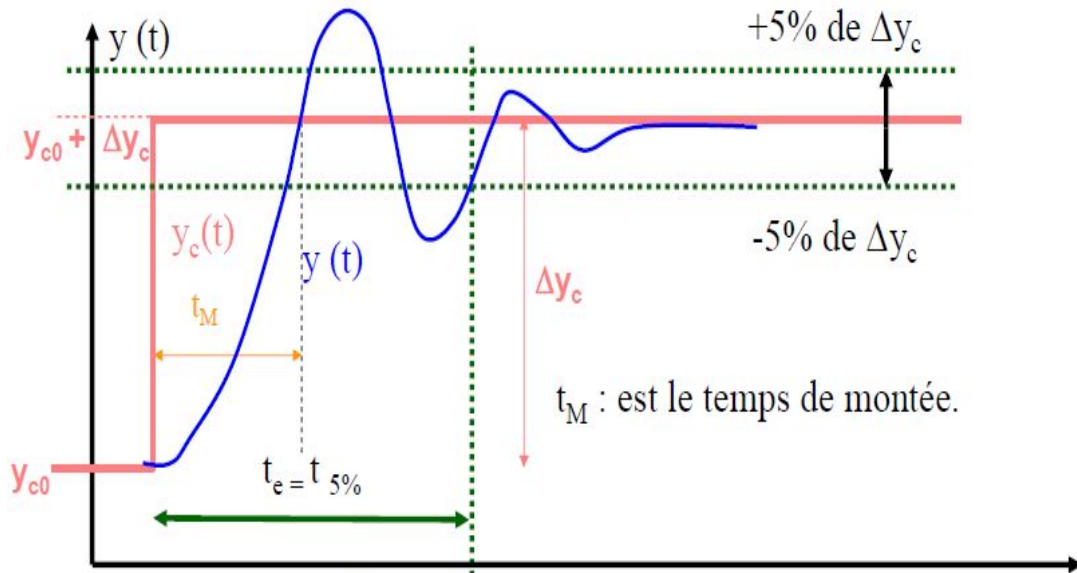


Figure 1.16 : Temps de réponse 5% d'un système.

### Conclusion :

La définition des performances d'un système est une étape très intéressante qui va permettre à l'opérateur de trouver un compromis entre les trois paramètres (stabilité, précision et rapidité) en respectant les exigences de procédé.

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

## IV - Avantages et Inconvénients des régulateurs PID et Floue

On peut comparer dans le tableau ci-dessous le comportement des deux régulateurs PID et Floue dans les différentes situations :



Comparatif résumé des principales familles de régulateurs					
Critères	Type de régulation				
	PID	Prédictif	Modèle interne	Flou	PID auto-adaptatif
Simplicité d'utilisation en monovariante	bien	médiocre	moyen	moyen	bien
Simplicité d'utilisation en multivariable	bien	bien	moyen	bien	médiocre
Mesures à échantillonnage lent et irrégulier	médiocre	moyen	bien	médiocre	médiocre
Performances en multivariable	médiocre	moyen	bien	moyen	médiocre
Processus à non-minimum de phase	médiocre	bien	bien	moyen	médiocre
Processus instable	moyen	moyen	moyen	médiocre	médiocre
Processus non linéaire au sens paramétrique	médiocre	moyen	bien	bien	moyen
Processus non linéaire à structure brutalement évolutive	moyen	moyen	bien	moyen	moyen
Processus à paramètres évolutifs dans le temps	médiocre	moyen	moyen	moyen	bien
Contraintes sur la commande	médiocre	bien	moyen	médiocre	médiocre
Contraintes sur une grandeur réglée	moyen	bien	bien	moyen	moyen
Processus non modélisable mathématiquement	moyen	médiocre	médiocre	bien	médiocre

**Tableau 1.3 : Comparatif des différents critères des régulateurs PID et Floue.**

**Conclusion :** le choix de régulateur reste la clé de réussite de la régulation industrielle, un bon réglage d'une boucle de régulation est celui qui répondra au meilleur compromis global du cahier des charges.

# PARTIE II : APPLICATION



## Présentation de Matlab/Simulink :

**MATLAB** est le « *noyau dur* » d'un logiciel de mathématiques spécialement étudié pour traiter des matrices. Un langage de programmation spécifique permet de faire des manipulations sur des matrices et toutes sortes de calculs et de présentation graphique de ces calculs. Différentes « *boîtes à outils* » peuvent y être adjointes pour simuler le comportement des systèmes linéaires ou non. Dans ce cas, il n'est pas nécessaire d'utiliser le langage de programmation car une interface graphique permet d'avoir accès de façon intuitive à des schémas réalisant une fonction particulière.

**SIMULINK** est un outil d'analyse des systèmes linéaires qui permet de décrire sous forme de schéma simulink le système à étudier, de lancer une simulation temporelle et de récupérer les résultats pour les exploiter. Simulink est une interface graphique qui facilite l'analyse des systèmes dans le domaine temporel. Cette approche est, en effet, facilitée, dans le sens où, les systèmes ne sont plus décrits par des lignes de code Matlab mais simplement définis à partir d'un bloc diagramme, dont tous les éléments sont prédéfinis dans des bibliothèques de blocs élémentaires, qu'il suffit d'assembler. Lorsque le schéma bloc du système que l'on étudie est représenté sous Simulink, il est possible d'analyser sa réponse temporelle (à des entrées aussi diverses sous Simulink peut qu'un échelon, une rampe, un signal rectangulaire, un sinus, un signal aléatoire...) en lançant une simulation.

Ce module permet de construire des schémas fonctionnels à partir d'éléments contenus dans des bibliothèques qu'il suffit de « *draguer* » avec la souris pour les déposer sur la « *feuille* » de travail et construire son système :

- ❖ la bibliothèque « *source* » fournit des échelons, des GBF, etc. ...
- ❖ les bibliothèques « *continu* » et « *discret* » proposent des dérivateurs, des intégrateurs, des fonctions de transfert pour construire des systèmes à temps continu ou échantillonné (discret),
- ❖ la bibliothèque « *math* » fournit des gains, des multiplicateurs, des sommateurs,
- ❖ on trouve également des multiplexeurs pour « *vectoriser* » des signaux ainsi que des oscilloscopes pour visualiser les résultats de modélisation.

Le module **CONTROL SYSTEM TOOLBOX** fournit des outils d'analyse des systèmes (Bode, etc. ...)

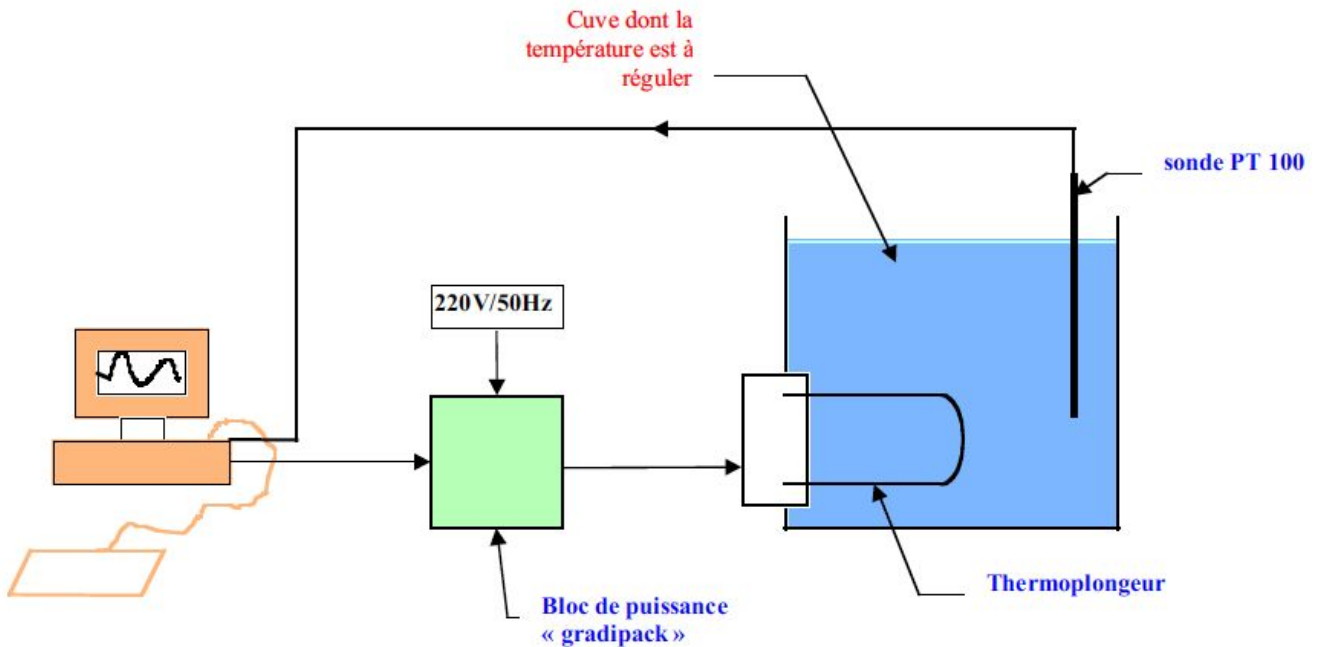
Le module **FUZZY LOGIC TOOLBOX** permet de générer des blocs utilisant la logique floue.

**On développera particulièrement dans la suite de ce rapport la méthode permettant d'élaborer un « *correcteur flou* » après avoir donné les éléments nécessaires à la compréhension.**

# I - Commande de la température d'eau dans une cuve : procédé 1.

## 1. Présentation et modélisation du procédé 1

### 1-1- Schéma de principe.



Le schéma de principe ci-dessus représente le synoptique de la régulation de température à réaliser, et dont le schéma fonctionnel est précisé ci-dessous :

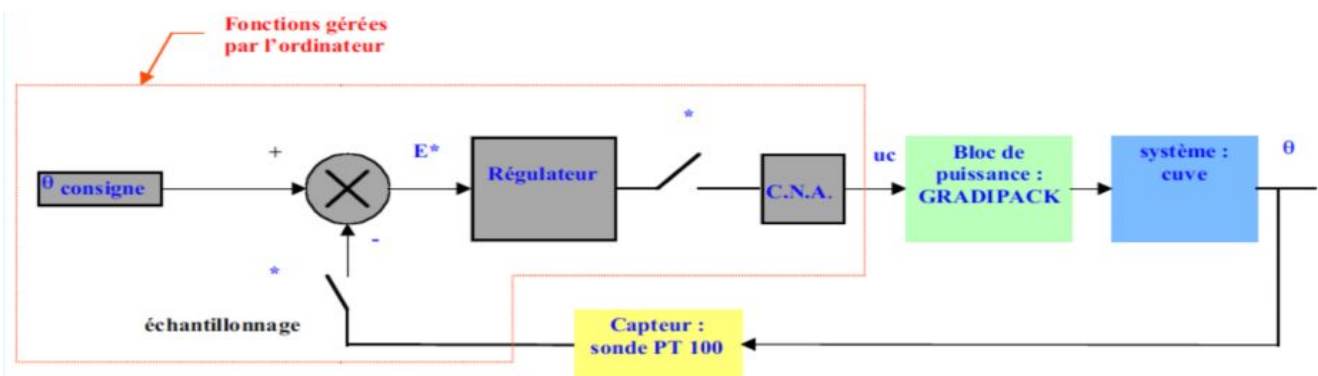


Figure 2.1 : Schéma fonctionnel du procédé 1.

Il s'agit de réguler la température de l'eau d'une **cuve** (seau d'une contenance de quelques litres) équipée d'un **thermoplongeur** apportant l'énergie calorifique. Une **sonde de température (PT100)** permet de mesurer la température réelle de l'eau. Il est bon de prévoir un brassage de l'eau dans la cuve afin d'homogénéiser sa température.

La puissance nominale du **thermoplongeur (1 kW)** est modulée grâce à un bloc de puissance « **GRADIPACK** » (dénomination commerciale). Il s'agit d'un ensemble de 2 thyristors montés tête-bêche (gradateur) permettant de faire varier la puissance électrique dissipée par variation de l'angle de retard à l'amorçage des thyristors. Celle-ci est contrôlée

par une tension de commande 0 -10V continue ( $U_{com}$ ) qui permet de faire varier la tension efficace ( $U_{eff}$ ) appliquée au thermoplongeur.

Le rôle du micro-ordinateur sera de faire la mesure de la température de la cuve à partir du signal délivré par la sonde PT100 et sa mise à l'échelle, de comparer cette valeur avec la température de consigne, de traiter l'écart entre ces 2 valeurs et de fournir une tension de commande utile pour le « GRADIPACK ». Tout cela nécessite la mise en œuvre d'une carte d'entrées-sorties analogiques.

## 1-2- Modélisation du procédé

### Cuve à chauffer :

On supposera que la capacité calorifique de la cuve à chauffer se réduit à l'eau qu'elle contient. En prenant un volume de 3 litres, on obtient :

$$C = m.c = 3 * 4.18E3 \approx 1.25E4 JK^{-1}$$

### Contrôle du thermoplongeur:

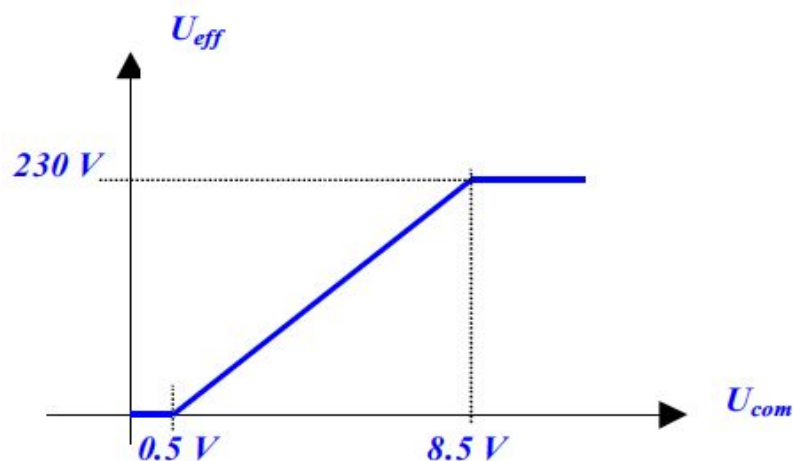


Figure 2.2: La modélisation du fonctionnement « GRADIPACK »

La tension efficace appliquée au thermoplongeur passe de 0 à 230V, si la tension de commande évolue de 0.5V à 8.5V. On pourra donc écrire :

$$U_{eff} = \frac{230}{8} \cdot (U_{com} - 0.5)$$

La puissance de chauffe sera donnée par :

$$P = P_N \cdot \frac{(U_{eff})^2}{(U_N)^2}$$

Soit en tenant compte de l'expression précédente :

$$P = \frac{P_N}{8^2} (U_{com} - 0.5)^2 = 15.6 (U_{com} - 0.5)^2 \text{ (en Watts).}$$

### Evaluation des pertes thermiques :

Les pertes thermiques peuvent se mettre sous la forme  $P_{th} = K.(\theta - \theta_{ext})$  (Loi de FOURIER).

L'expérience montre qu'elles sont faibles. On les estimera à 50 W pour un  $\Delta\theta = 25^\circ\text{C}$ . D'où :

$$K = 2WK^{-1}$$

### Bilan de puissance sur la cuve :

La puissance fournie par le thermoplongeur :  $P_T$ , sert à élever la température de l'eau et participe aux pertes soit :

$$P_T = C. \frac{d(\theta - \theta_{ini})}{dt} + K.(\theta - \theta_{ext})$$

Avec

$\theta$  : température de l'eau.

$\theta_{ini}$  : température initiale de l'eau.

$\theta_{ext}$  : température extérieure à la cuve .

## 2. Commande du procédé 1

### 2-1- Construction du régulateur flou

#### Les ensembles flous

On va exploiter 3 variables, l'erreur (notée **e**), entrée du régulateur, mais aussi la variation de l'erreur (notée  $\Delta e$ ), et la sortie (notée **s**). Les valeurs de **e** de  $\Delta e$  sont mises à jour à chaque période d'échantillonnage.

L'entrée **e** a été partitionnée en 3 valeurs linguistiques et  $\Delta e$  en deux valeurs linguistiques. Elles peuvent être positives ou négatives, situées sur un intervalle de [-10,30] et [-200,200] respectivement pour **e** et  $\Delta e$ .

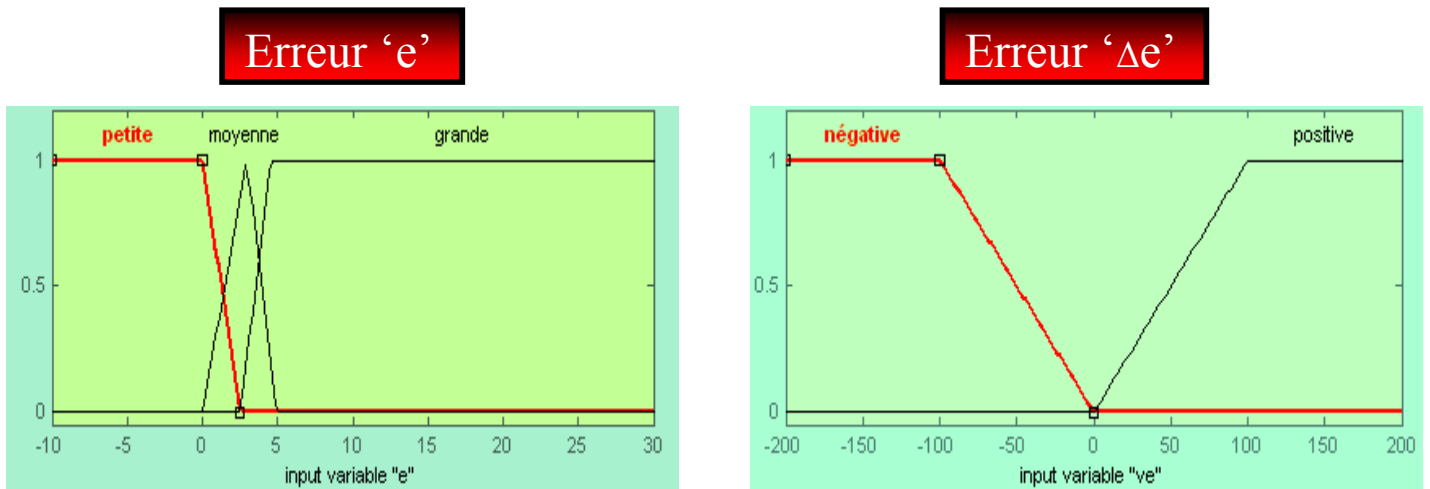
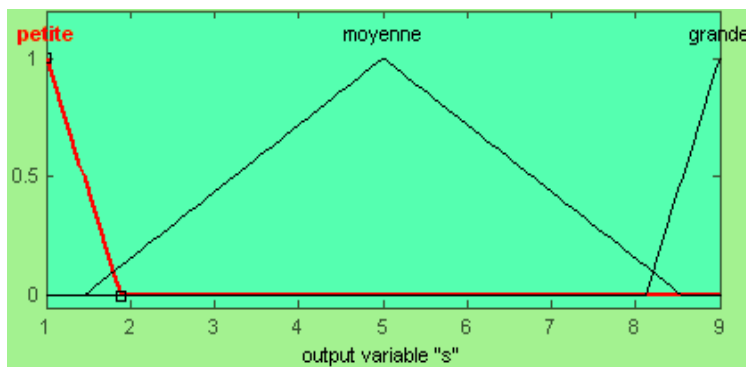


Figure 2.3 : Fuzzification des deux entrées  $e$  et  $\Delta e$

La sortie  $s$  a été partitionnée en 3 valeurs linguistiques ; elle pourra n'être que positive dans le cas où la partie puissance est commandée par une tension (1, 9V).



Erreur 's'

Figure 2.4 : Fuzzification de sortie  $s$

### Les règles Floues

Le rôle de l'expert est ici présent , car c'est lui qui va fixer les règles de la commande qui vont porter **uniquement** sur les valeurs linguistiques des variables.

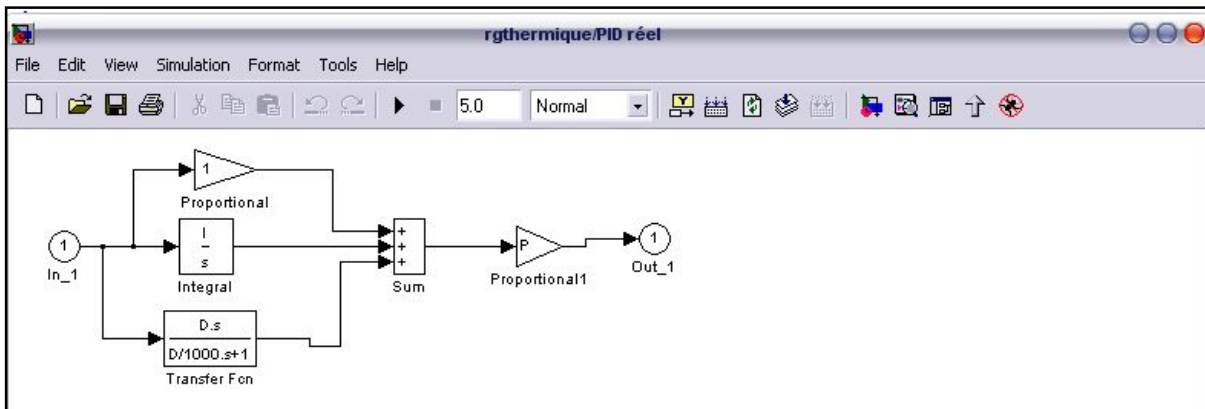
La liste des règles utilisées est la suivante :

- ✓ R1 : **SI** ( $e$ ) est **petite** **ALORS** ( $s$ ) est **petite**
- ✓ R2 : **SI** ( $e$ ) est **moyenne** **ALORS** ( $s$ ) est **moyenne**
- ✓ R3 : **SI** ( $e$ ) est **grande** **ALORS** ( $s$ ) est **grande**
- ✓ R4 : **SI** ( $e$ ) est **petite** **ET** ( $\Delta e$ ) est **négative** **ALORS** ( $s$ ) est **petite**
- ✓ R5 : **SI** ( $e$ ) est **moyenne** **ET** ( $\Delta e$ ) est **positive** **ALORS** ( $s$ ) est **grande**

### 2-2- Le régulateur PID



Le bloc Simulink représentant le régulateur PID (mixte) est le suivant :



**Figure 2.5 :**  
Schéma Simulink des composants principaux de

## PID

Le réglage des paramètres PID été effectué par la méthode expérimentale de Ziegler-Nichols qui nous donne les valeurs suivantes :

$$K_p = 1$$

$$T_i = 1$$

$$T_d = 1,25$$

### 2-3- schéma simulink du procédé 1

Le schéma fonctionnel ci dessous présente la modélisation du système à étudier à partir des éléments de base décrits plus haut. On peut y reconnaître :

- ❖ des échelons,
- ❖ des sommateurs,
- ❖ un produit,
- ❖ des oscilloscopes,
- ❖ un intégrateur,
- ❖ des multiplexeurs,...

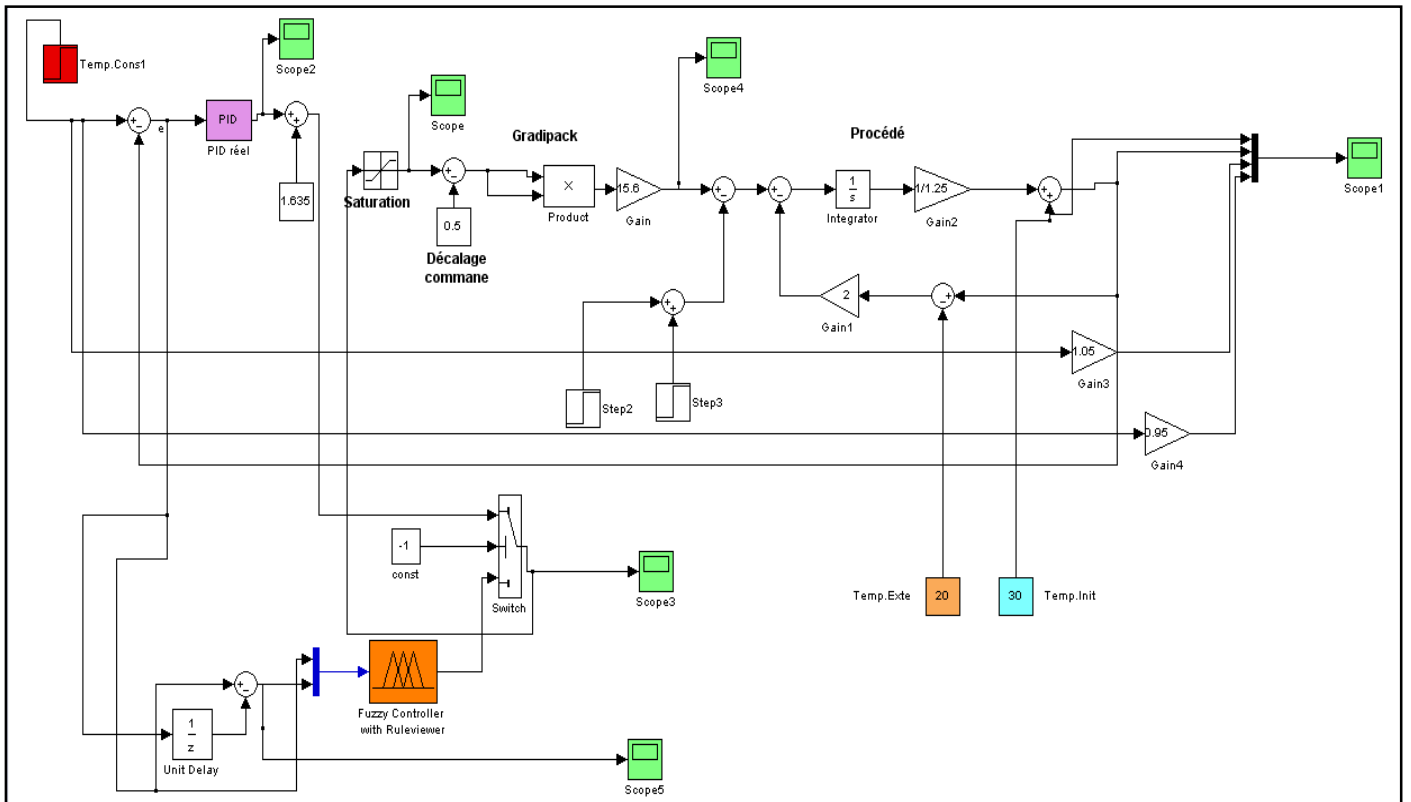


Figure 2.6 : Schéma Simulink pour la simulation du procédé 1.

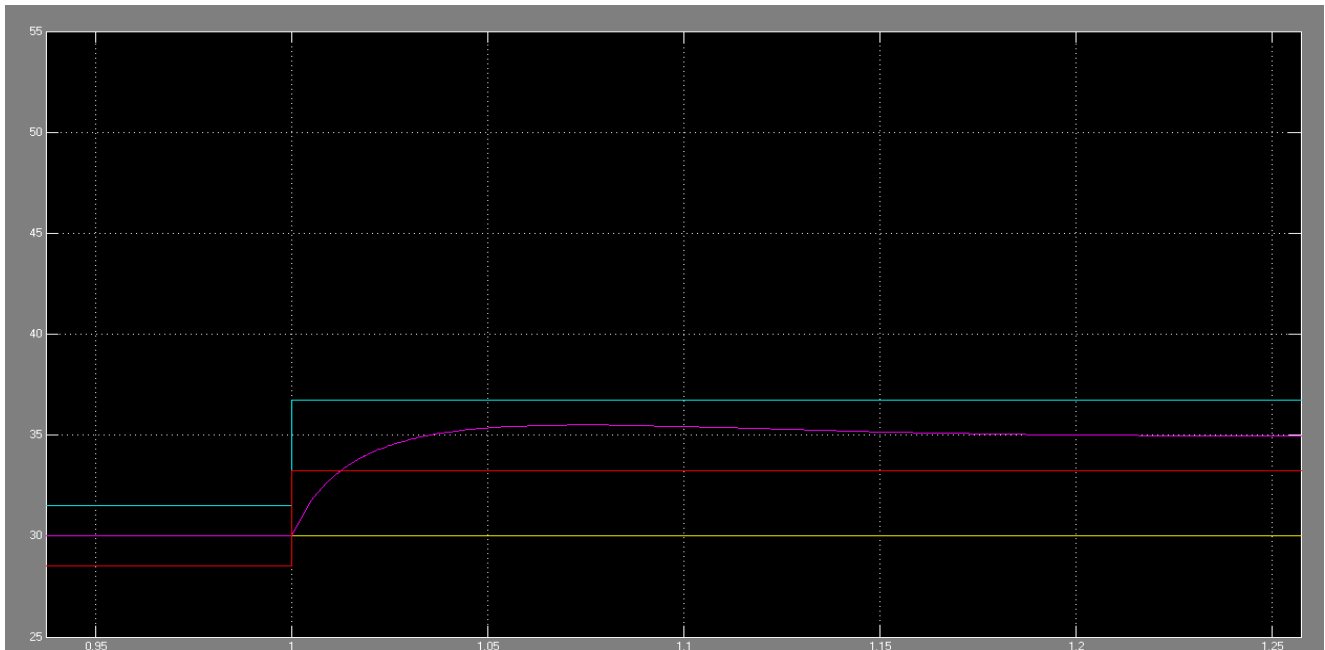
### 3. Simulation

On utilise le schéma Simulink (figure 2.6) dans lequel on colle un bloc "Fuzzy Logic Controller" qu'on trouve dans la boîte "SIMULINK Fuzzy". On doit y spécifier un nom de régulateur parmi ceux chargés précédemment dans l'espace de travail. Ce bloc doit être précédé de blocs gain chargés de la normalisation des signaux et suivi d'un gain ajustable de l'amplification du régulateur.

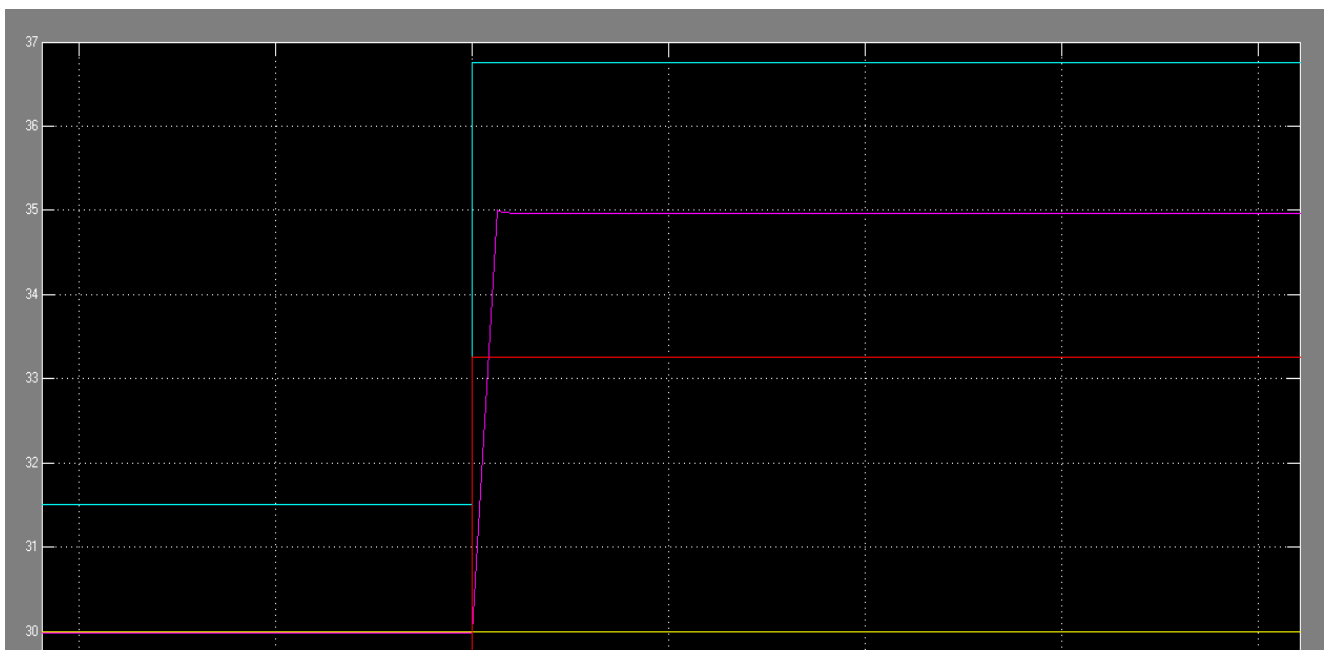
### 3-1 Commande en poursuite par PID et logique Floue

Point de fonctionnement nominal : température d'eau dans la cuve 30°C, température ambiante 20°C.

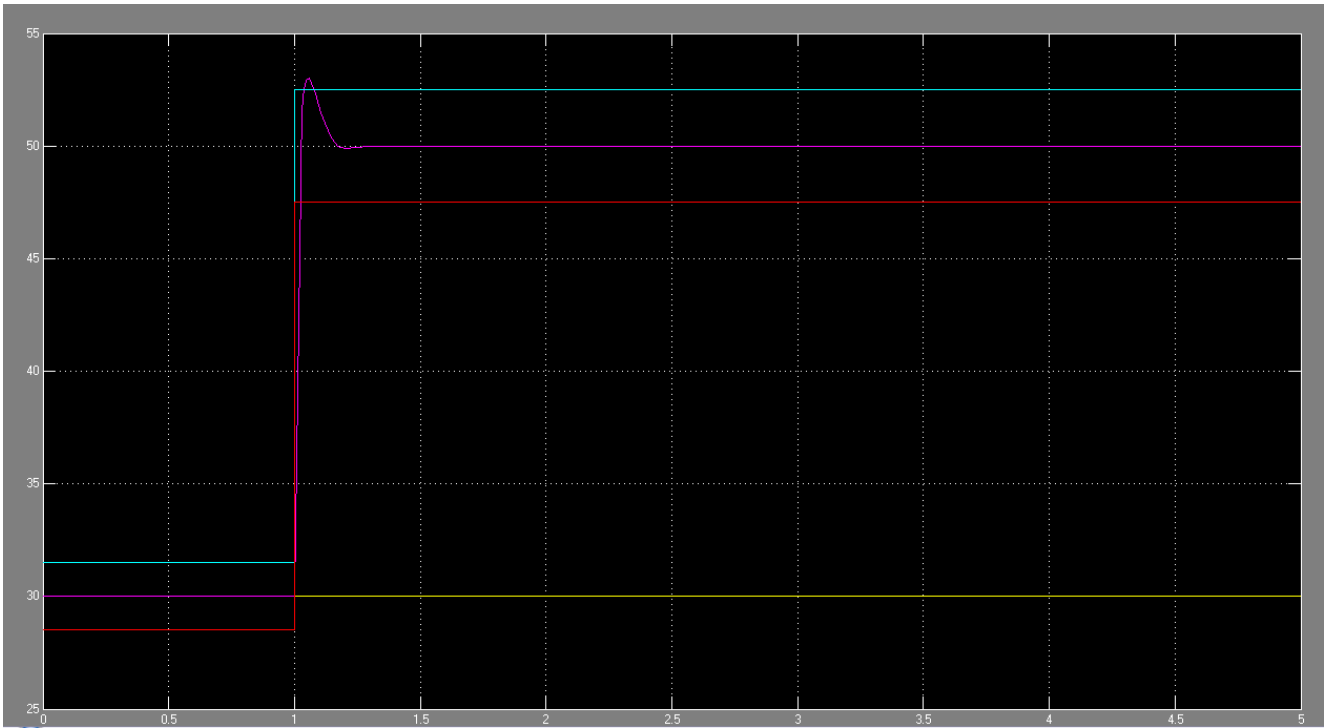
On effectue une série de changement de consigne (Figures 2.7- 2.12) ; et on simule le comportement du procédé par les deux commandes PID et logique Floue.les résultats de simulation sont reportés sur le tableau 2.1.



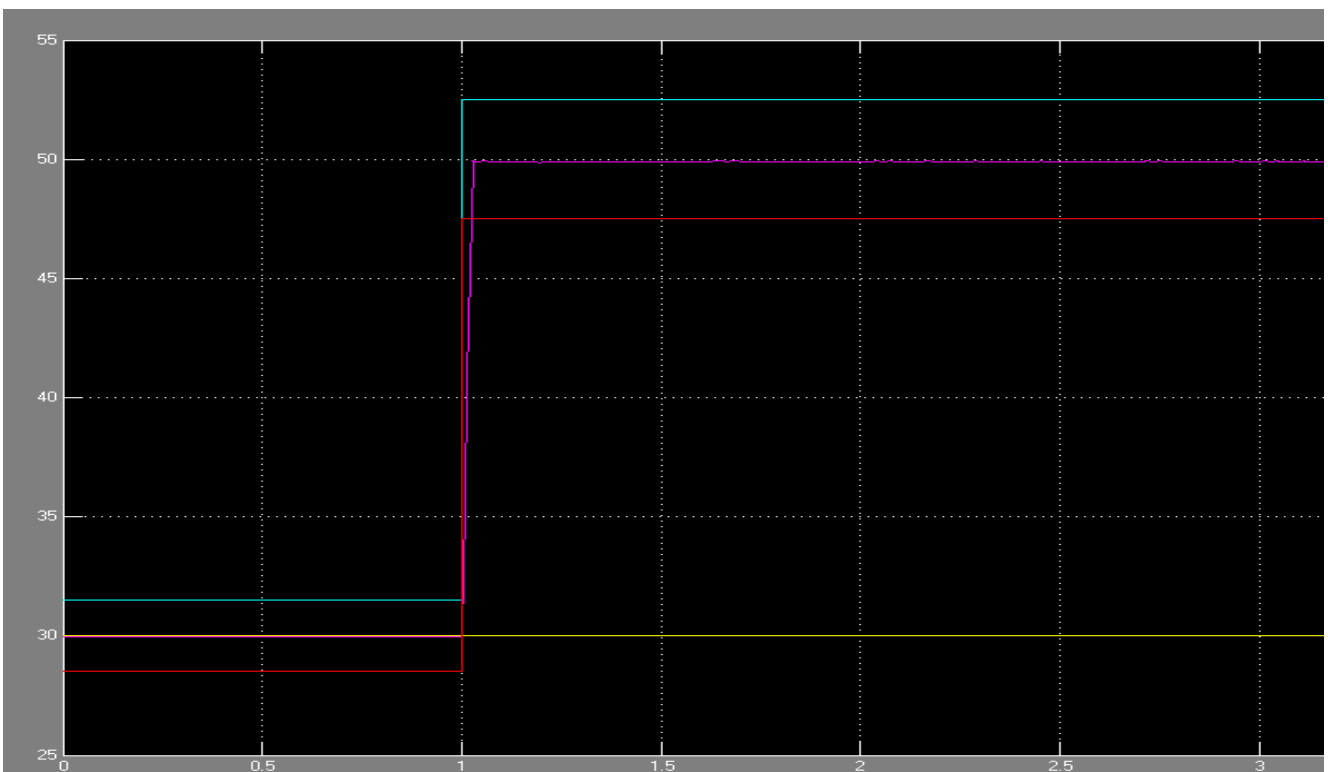
**Figure 2.7: Commande PID, changement de consigne de 30°C à 35°C.**



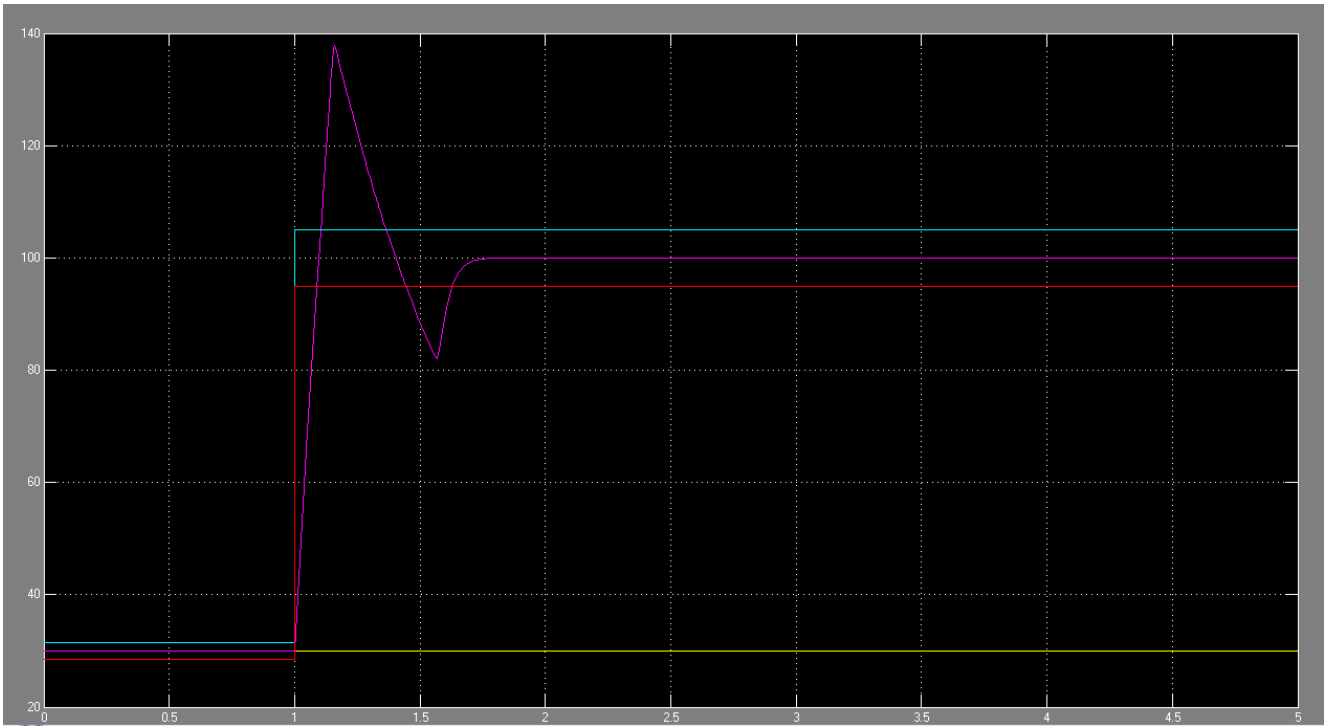
**Figure 2.8 : Commande Floue, changement de consigne 30°C à 35°C.**



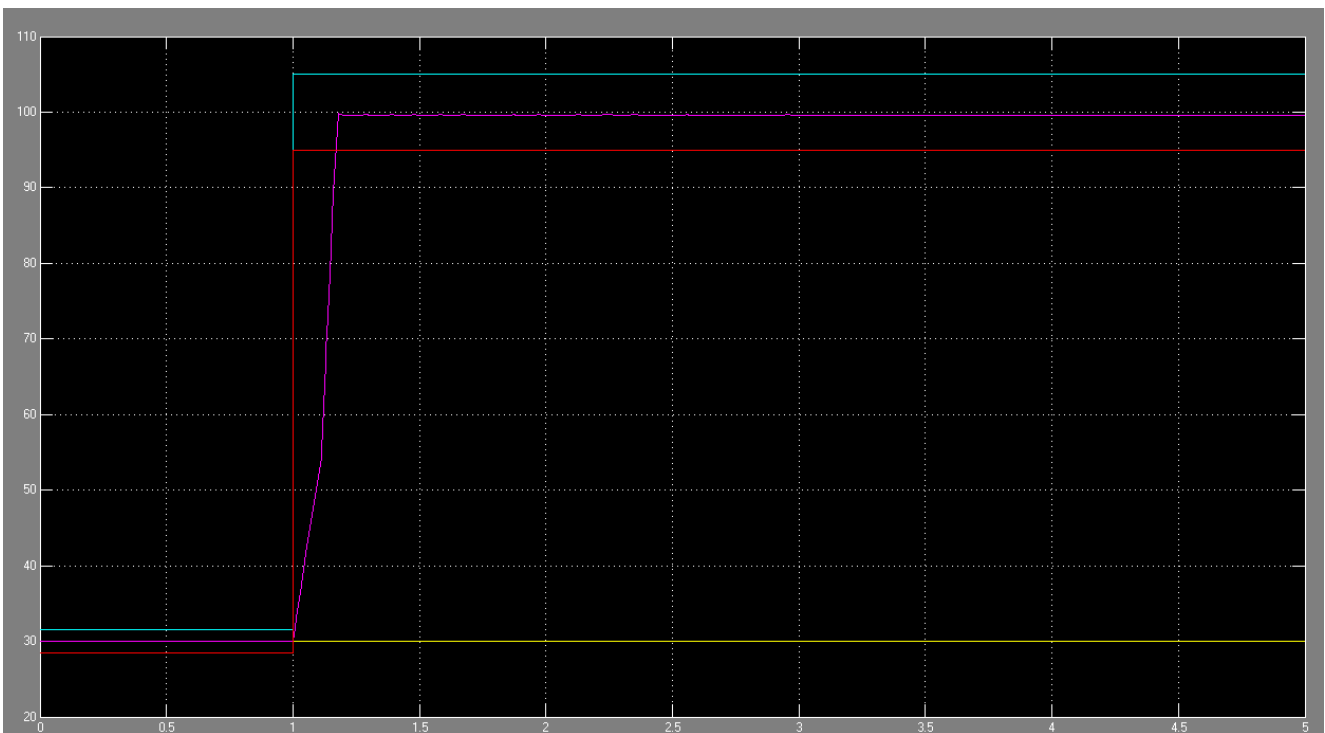
**Figure 2.9 : Commande PID , changement de consigne de 30°C-50°C .**



**Figure 2.10 : Commande Floué , changement de consigne 30°C-50°C.**



**Figure 2.11 : Commande PID , changement de consigne 30°C-100°C**



**Figure 2.12 : Commande Floue , changement de consigne 30°C-100°C**

Tableau 2.1 : résultat d'une commande en poursuite par PID et logique floue du procédé 1

Le régulateur	changement de consigne		30°C - 35°C	30°C - 50°C	30°C - 80°C	30°C - 100°C		
	<b>PID</b>	$\epsilon$	= 0,006 °C	$\epsilon$	= 0,0144 °C	$\epsilon$	= 0,01 °C	$\epsilon$
$t_{5\%}$		= 0,004 s	$t_{5\%}$	= 0,023 s	$t_{5\%}$	= 0,06 s	$t_{5\%}$	= 0,205 s
D		= 0 °C	D	= 0,34 °C	D	= 2,89 °C	D	= 6 °C
<b>Floue</b>	$\epsilon$	= 0,041 °C	$\epsilon$	= 0,094 °C	$\epsilon$	= 0,02 °C	$\epsilon$	= 0,02 °C
	$t_{5\%}$	= 0,01 s	$t_{5\%}$	= 0,024 s	$t_{5\%}$	= 0,068 s	$t_{5\%}$	= 0,159 s
	D	= 0 °C	D	= 0 °C	D	= 0 °C	D	= 0 °C

En calculant les critères de performances (précision, temps de réponse à 5% et dépassement). On constate que les deux régulateurs donnent des résultats efficaces pour les petits et moyens échelons de consigne. Mais pour les grands échelons c'est à dire en s'éloignant trop du régime nominal, et donc on sort du domaine linéaire, le PID perd ses performances, en particulier en terme de dépassement ou précision dynamique. Par contre le régulateur floue reste robuste.

## II - Commande de la température d'une serre agricole : procédé 2

### 1 - Présentation du procédé 2



Figure 2.13 : Une serre agricole.

Il s'agit de réguler la température d'une serre agricole (figure 2.13). On mesure la température avec un capteur qui fournit la valeur de la grandeur à régler.

On suppose que le système est linéaire, et ayant une fonction de transfert  $H(s)=1/(s+2)$  avec un retard égale à 1s. C'est un système de contrôle de température avec un élément de chauffage et de ventilation. La température désirée est de 20°C.

La température extérieure est de 10°C, et son effet intervient au niveau de l'intégrateur. On modélise le système avec SIMULINK. Pour la régulation d'un système pareil, on propose deux solutions :

- ✚ 1<sup>er</sup> cas : Régulation PID.
- ✚ 2<sup>ème</sup> cas : Régulateur flou.

## 2 - Commande du Procédé 2

### 2-1- Construction du régulateur flou

#### Les ensembles flous

On va exploiter 3 variables, l'erreur (notée **e**), entrée du régulateur, mais aussi la variation de l'erreur (notée  $\Delta e$ ) et la sortie (notée **s**). Les valeurs de **e** de  $\Delta e$  sont mises à jour à chaque période d'échantillonnage.

L'entrée **e** a été partitionnée en 3 valeurs linguistiques (chaud, Ok et froid) ainsi que  $\Delta e$ . Elles peuvent être positives ou négatives, situe sur un intervalle de [-10,14] et [-15,15] respectivement pour e et  $\Delta e$ .

La sortie **s** a été partitionnée en 5 valeurs linguistiques ; elle pourra être positive ou négative.

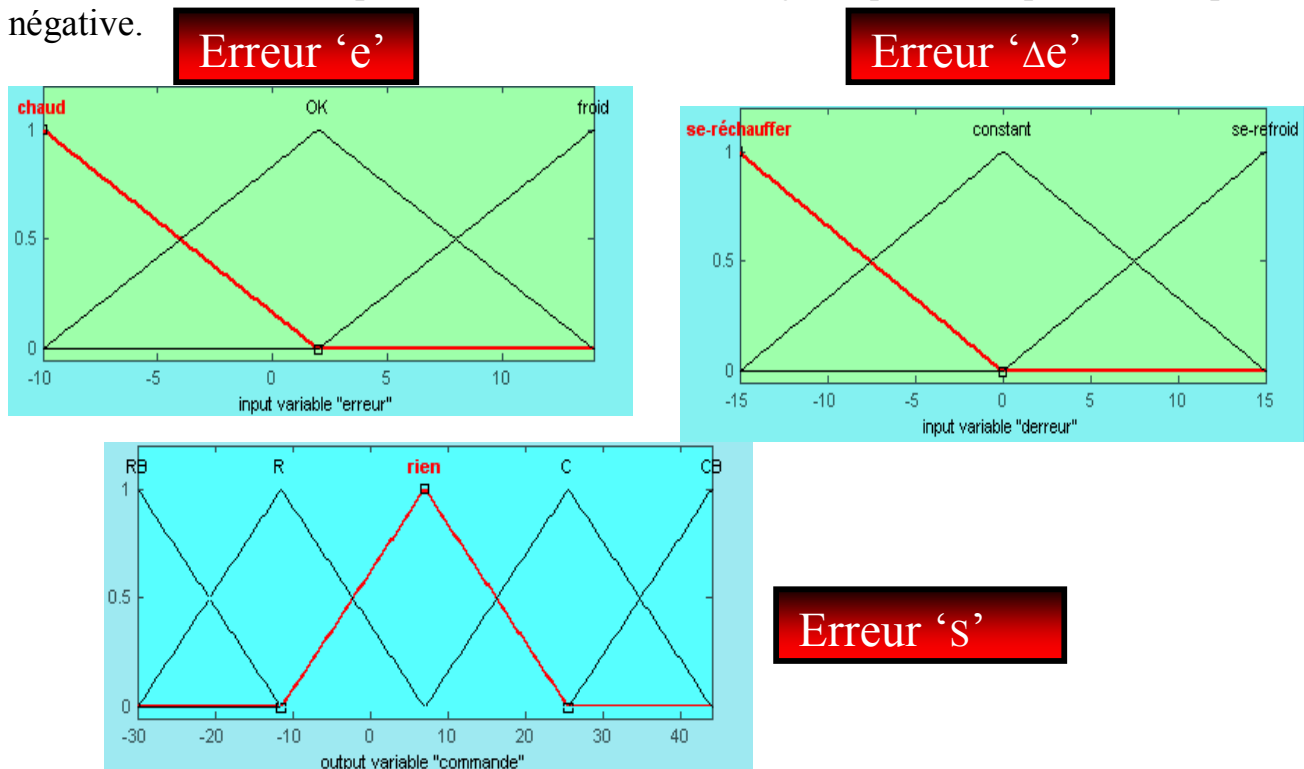


Figure 2.14 : Fuzzification des variables d'entrée et de sortie.



## Les règles Floues

La liste des règles utilisées est la suivante :

- ✓ R1 : **SI (e)** est **chaud** **ET ( $\Delta e$ )** est **se-refroid** **ALORS (s)** est **R**
- ✓ R2 : **SI (e)** est **chaud** **ET ( $\Delta e$ )** est **constant** **ALORS (s)** est **R**
- ✓ R3 : **SI (e)** est **chaud** **ET ( $\Delta e$ )** est **se-réchauffer** **ALORS (s)** est **RB**
- ✓ R4 : **SI (e)** est **OK** **ET ( $\Delta e$ )** est **se-refroid** **ALORS (s)** est **C**
- ✓ R5 : **SI (e)** est **OK** **ET ( $\Delta e$ )** est **constant** **ALORS (s)** est **rien**
- ✓ R6 : **SI (e)** est **OK** **ET ( $\Delta e$ )** est **se-réchauffer** **ALORS (s)** est **R**
- ✓ R7 : **SI (e)** est **froid** **ET ( $\Delta e$ )** est **se-refroid** **ALORS (s)** est **CB**
- ✓ R8 : **SI (e)** est **froid** **ET ( $\Delta e$ )** est **constant** **ALORS (s)** est **C**
- ✓ R9 : **SI (e)** est **froid** **ET ( $\Delta e$ )** est **se-réchauffer** **ALORS (s)** est **C**

Avec :

- Refroidir beaucoup : RB
- Refroidir : R
- Réchauffer : C
- Réchauffer beaucoup : CB

### 2-2- Le régulateur PID

On règle le PID selon les paramètres suivants :

Step=20°C; Step1=-1.5°C; Integrator: CI=10; Lower saturation limit=10; Transport Delay=0; et  $K_p = 1$   $T_i = 0,15$   $T_d = 0$

### 2-3- Schéma Simulink du procédé 2

Pour la simulation de notre procédé, on va utiliser pour chaque régulateur, soit PID ou logique Floue un schéma simulink particulier.

Pour le PID :

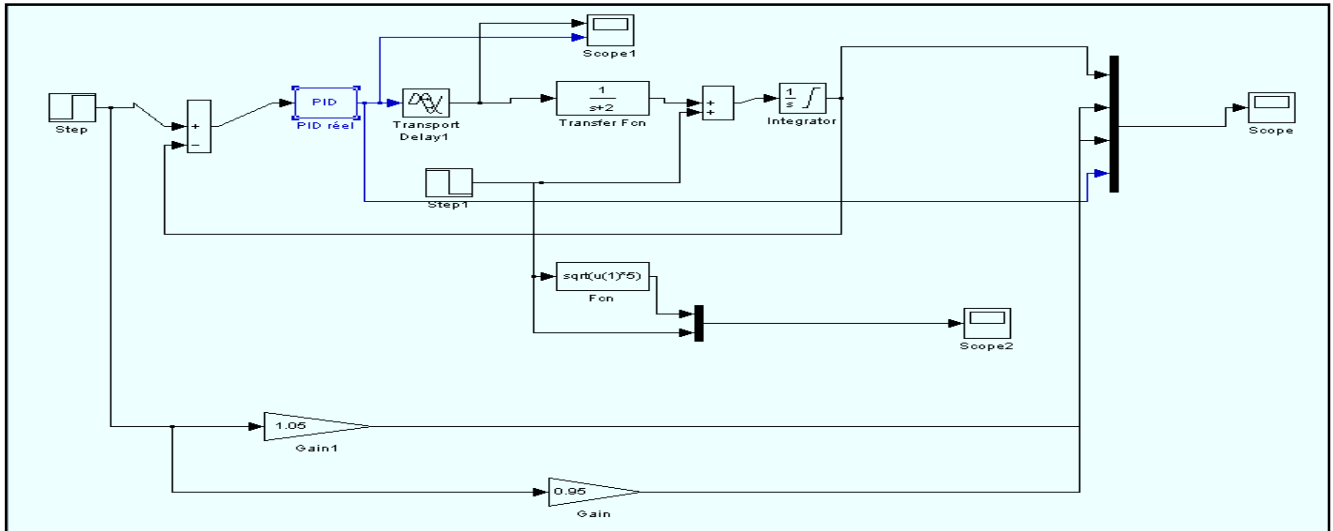


Figure 2.15 : Schéma Simulink PID.

### Pour le Régulateur Floue

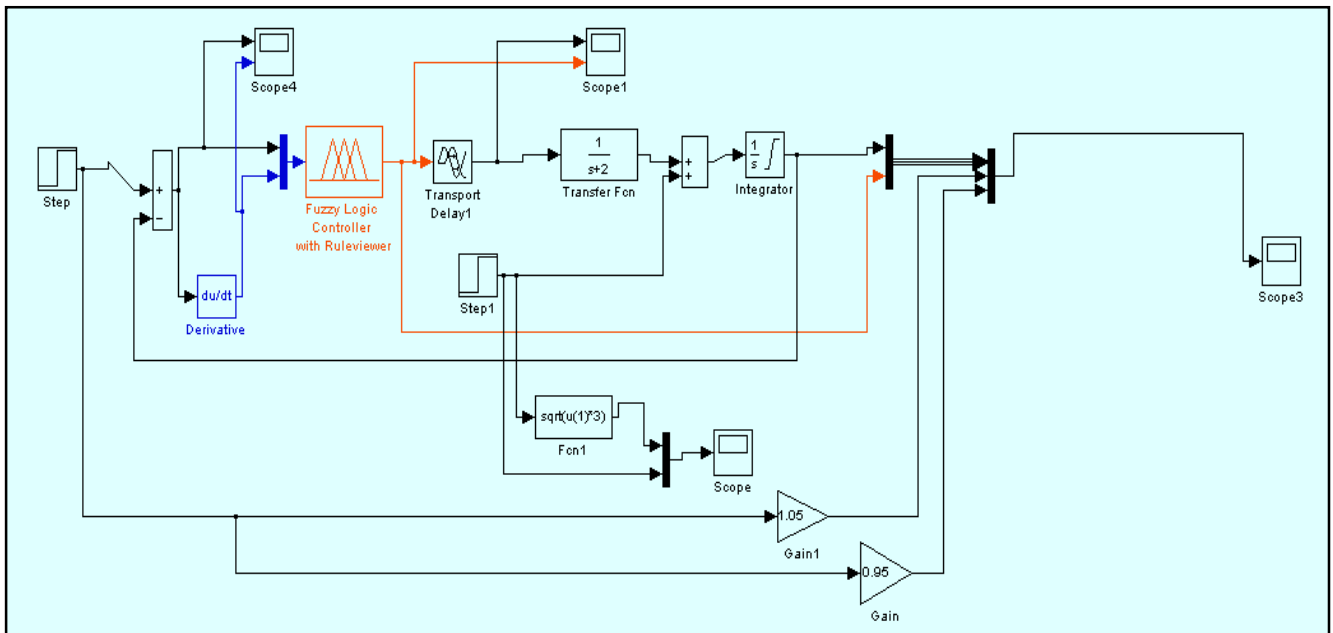


Figure 2.16 : Schéma Simulink , Régulateur Floue.

## 3. Simulation

### 3-1- Commande en régulation par PID et logique Floue

Point de fonctionnement nominal : température de serre 10 °C.  
 Pour des échelons de consigne et de perturbation constants, La température est fixée à 20°C et la perturbation est fixée à -1,5.

On effectue une série de changement de retard (Figures 2.17 – 2.22) et on simule le comportement du procédé par les deux commandes PID et logique Floue.

Les résultats de simulation sont reportés sur le tableau suivant :

Le régulateur \ changement de retard	1	1,5	2
	<b>PID</b>	$\varepsilon = 0 \text{ } ^\circ\text{C}$	$\varepsilon = 0 \text{ } ^\circ\text{C}$
$t_{5\%} = 7 \text{ s}$		$t_{5\%} = 13,5 \text{ s}$	$t_{5\%} = 202,2 \text{ s}$
$D = 1,3 \text{ } ^\circ\text{C}$		$D = 4 \text{ } ^\circ\text{C}$	$D = 7,4 \text{ } ^\circ\text{C}$
<b>Floue</b>	$\varepsilon = 0,7 \text{ } ^\circ\text{C}$	$\varepsilon = 0,72 \text{ } ^\circ\text{C}$	$\varepsilon = 0,7 \text{ } ^\circ\text{C}$
	$t_{5\%} = 3,5 \text{ s}$	$t_{5\%} = 6,4 \text{ s}$	$t_{5\%} = 63,5 \text{ s}$
	$D = 0 \text{ } ^\circ\text{C}$	$D = 2,9 \text{ } ^\circ\text{C}$	$D = 6,4 \text{ } ^\circ\text{C}$

Tableau 2.2 : Simulation avec un changement de retard.

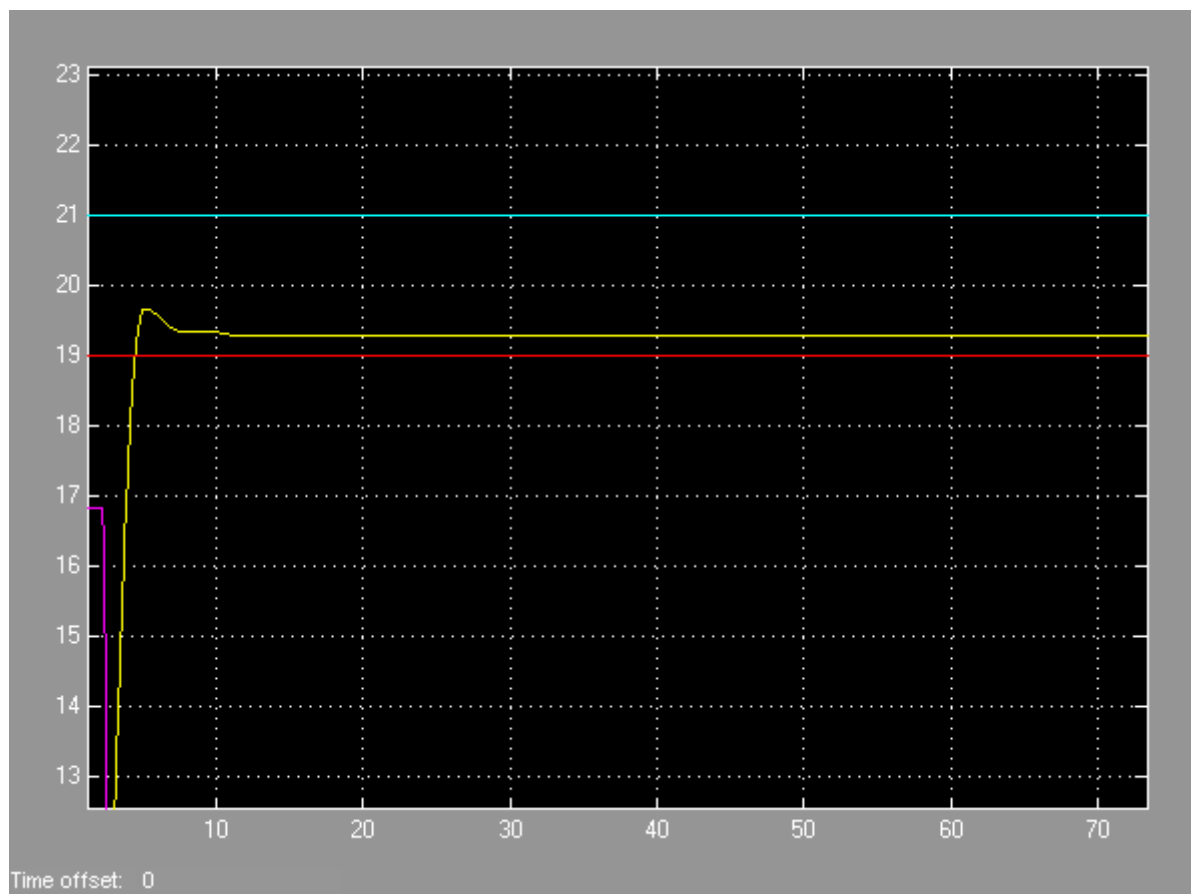
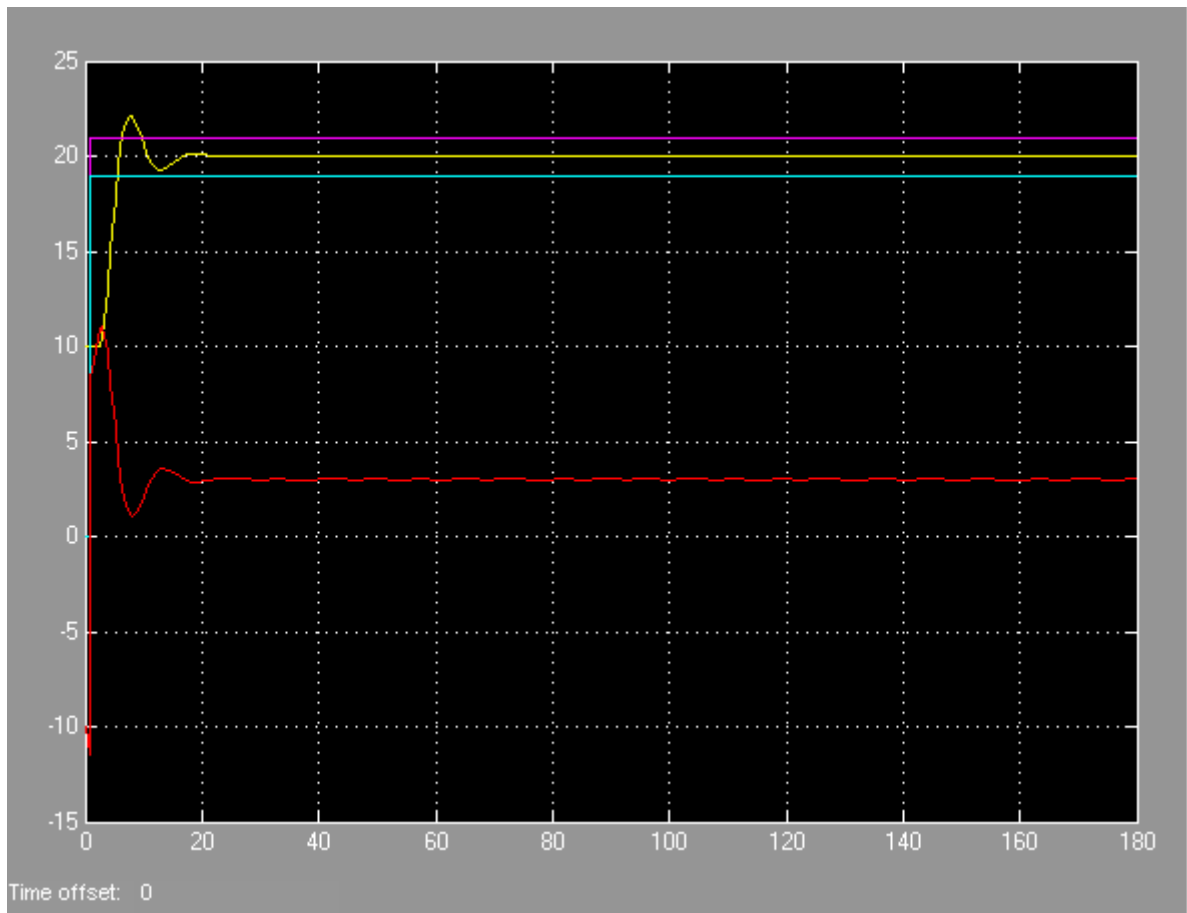
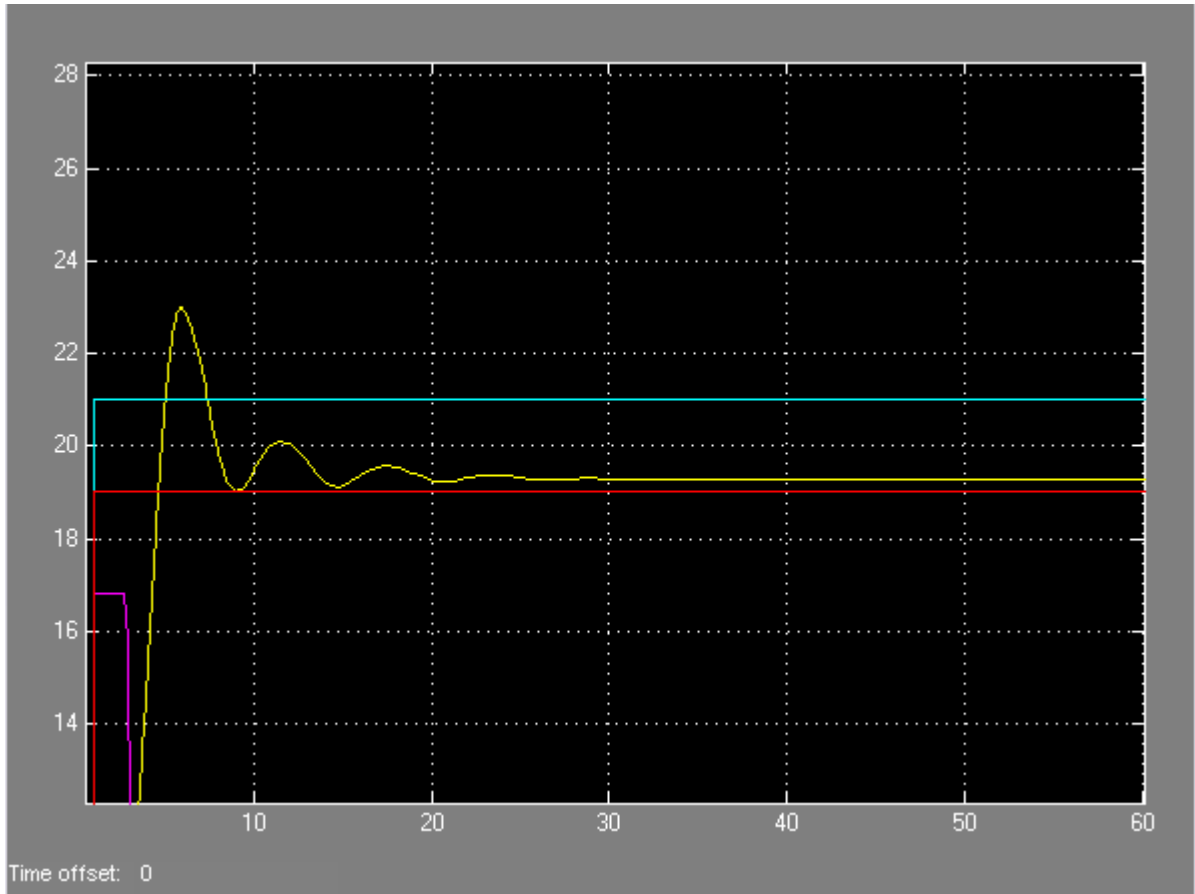


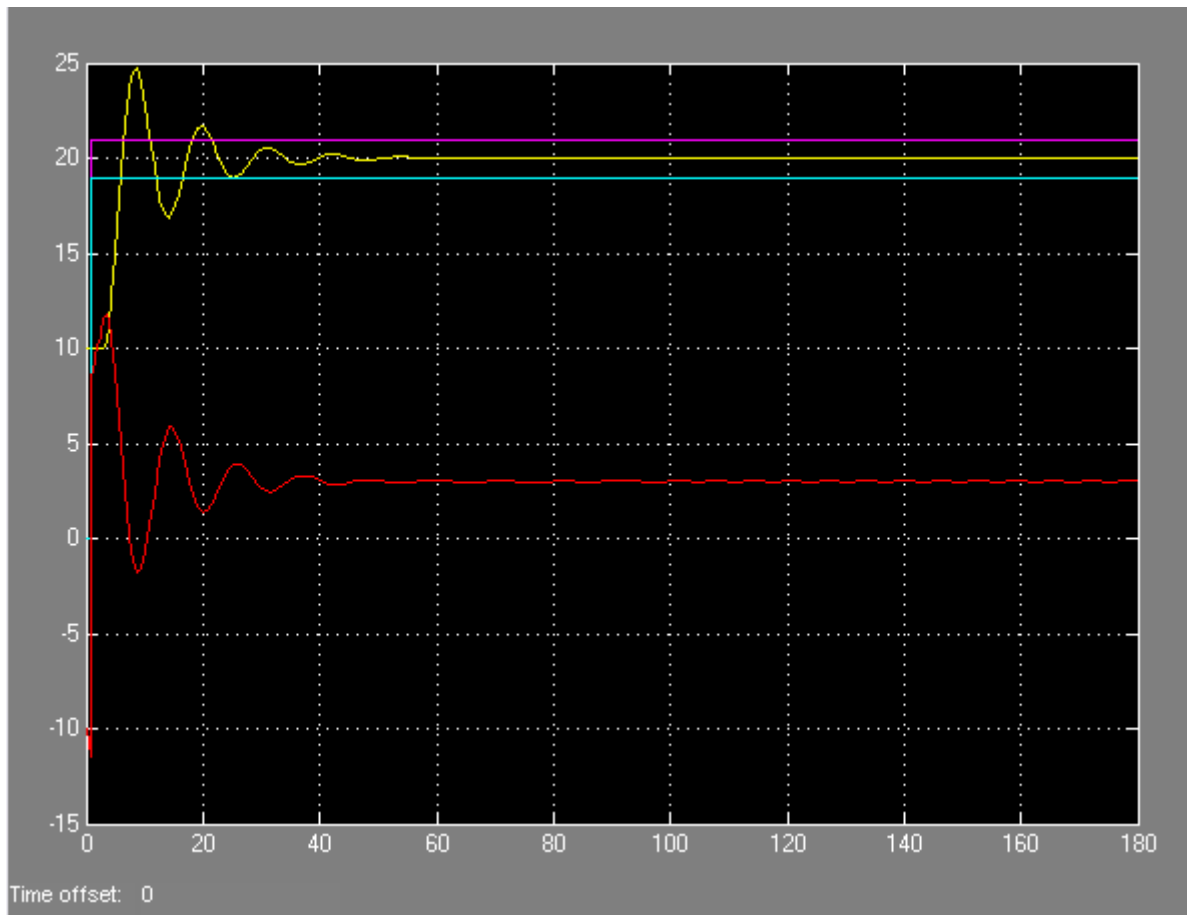
Figure 2.17 : Commande fluide pour un retard =1s.



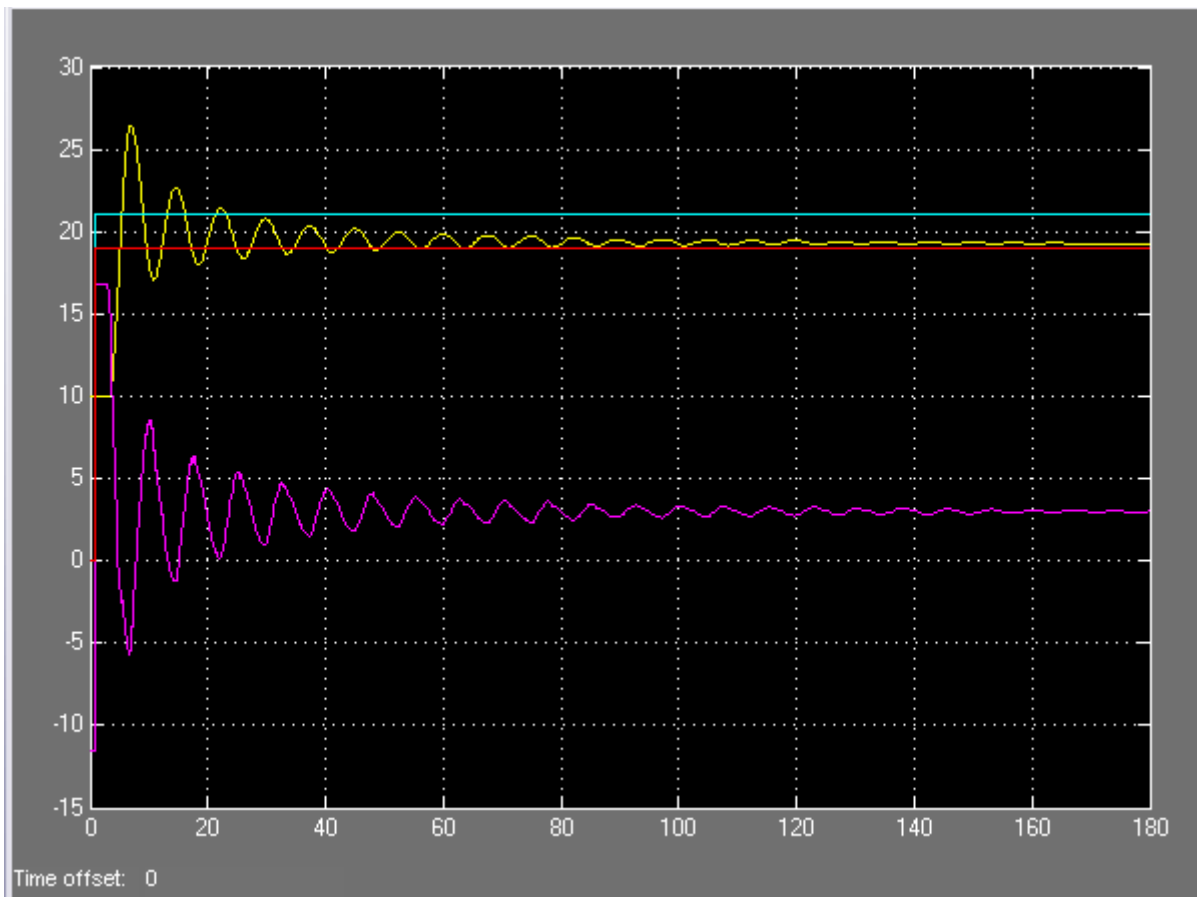
**Figure 2.18 : Commande PID pour un retard =1s.**



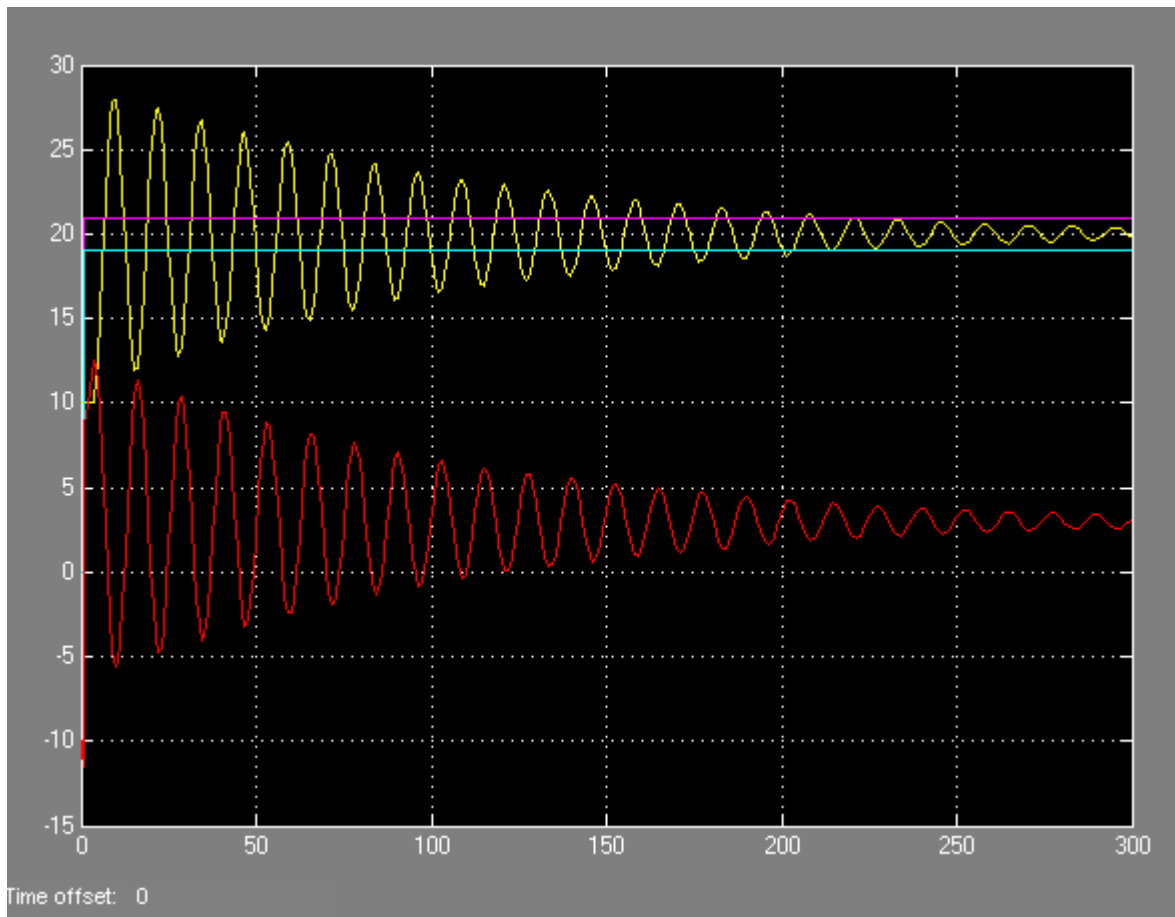
**Figure 2.19 : Commande flou pour un retard =1,5s.**



**Figure 2.20 : Commande PID pour un retard= 1,5s.**



**Figure 2.21 : Commande flou pour un retard= 2s.**



**Figure 2.22 : Commande PID pour un retard =2s.**

En calculant les critères de performances (précision, temps de réponse à 5% et dépassement). On constate que les deux régulateurs donnent des résultats acceptables pour les petits et moyens changements de retard, mais pour des grands retards le PID perd ses performances, en particulier en terme de dépassement ou temps de réponse. Par contre le régulateur flou reste robuste.

## **Conclusion**

D'après les deux résultats de simulation des procédés précédents on peut déduire que la Commande floue donne des résultats précieux surtout lorsque on effectue des changements de consigne (procédé 1) et de retard (procédé 2). Par contre la commande classique PID perd ses performances notamment en terme (stabilité, dépassement et la précision dynamique), et le régulateur flou reste le plus robuste.



# CONCLUSION GÉNÉRALE

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

Le but de notre travail est d'étudier en simulation l'application d'une telle commande à deux procédés thermiques, puis de faire une comparaison avec la commande analogique PID des mêmes procédés. Le premier procédé est une régulation de température d'eau dans une cuve par thermoplongeur et pour lequel on dispose du modèle temporel [1]. Le deuxième procédé est une régulation de température d'une serre agricole et pour lequel on dispose du modèle dans le domaine de Laplace [2].

Dans un premier temps, nous avons simulé, à l'aide du logiciel Simulink, la commande en poursuite (changement de consigne) par logique floue et par PID du premier procédé. Pour des échelons croissants de consigne, les résultats de simulation ont montré que le PID perd ses performances en particulier en termes de dépassement ou précision dynamique alors le régulateur floue reste robuste.

Dans un deuxième temps les simulations des mêmes commandes appliquées au deuxième procédé ont été effectuées pour différentes valeurs du retard que présente ce procédé, et pour des échelons constants de la consigne et de la perturbation. Les résultats de simulation ont montré que plus le retard augmente et plus le PID perd ses performances notamment en termes de stabilité alors que régulateur floue garde des performances correctes.

Ces deux résultats nous ont permis de confirmer l'avantage d'une commande floue par rapport à une commande classique PID, notamment lorsque le procédé est non linéaire, ou présentait un retard important. Dans ce cas on peut dire que la commande floue constituera un complément précieux à la commande classique PID.

# Références

- [1] : P.BRUNET, E.FERRERE ; Réalisation d'un T.P de régulation thermique (niveau BAC + 2) en utilisant l'environnement MATLAB pour la partie commande.
- [2] : M. RABI ; Cours de la régulation industrielle, 2010, ESTF.
- [3] : B. Le Ballais ; Matlab / Simulink, Application à l'automatique linéaire, 2001, ellipse.