

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ MOULOUD MAMMÉRI, TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE  
DEPARTEMENT D'ELECTRONIQUE

**Mémoire de fin d'études**  
Présenté en vue de l'obtention  
du Diplôme d'Ingénieur d'Etat en Electronique

Option : *Contrôle*

*Thème:*

*Simulation des correcteurs  
PID (Classique et Flou)*

Proposé et dirigé par :

M<sup>elle</sup> AIT OUAZZOU H.

Présenté par :

BEN YOUNES Yazid

OURAMDANE Khaled

ZEMIH Youcef

Année universitaire 2008/2009

Soutenu le 02/07/2009

# SOMMAIRE

<i>Introduction générale</i> .....	1
------------------------------------	---

## **Chapitre I Principe de l'Asservissement.**

I.1 Introduction.....	3
I.2 Les systèmes asservis.....	3
I.3 Schéma fonctionnel d'un asservissement.....	4
▪ Transmetteur de consigne.....	5
▪ Capteur.....	5
▪ Comparateur.....	5
▪ Amplificateur d'écart.....	5
▪ Correcteur (Régulateur).....	5
▪ Actionneur.....	5
▪ Le procédé.....	5
I.4 Performances d'un asservissement.....	5
I.4.1 Stabilités d'un système d'entrée $e(t)$ et de sortie $s(t)$ .....	6
I.4.2 Condition de stabilité.....	7
I.4.3 Stabilité des systèmes asservis.....	7
I.4.4 Précision.....	7
I.4.5 Rapidité.....	7
I.4.6 Qualité.....	8
I.5 Différentes étapes dans la réalisation d'un asservissement.....	8
▪ Etape1.....	8
▪ Etape2.....	8
▪ Etape3.....	8
▪ Etape4.....	9
▪ Etape5.....	9
I.6 Conclusion.....	9

## **Chapitre II Régulateurs PID classique.**

II.1 Introduction.....	10
II.2 Aspects fonctionnels.....	10
II.2.1 Action Proportionnelle P.....	10
II.2.2 Action Proportionnelle Dérivée (PD).....	11

# SOMMAIRE

II.2.3 Correcteur Proportionnel Intégral PI.....	12
II.2.4 Correcteur Proportionnel Intégral Dérivé PID.....	13
II.2.5 Fonction de transfert du correcteur PID standard.....	14
II.3 Caractéristiques des correcteurs.....	15
II.3.1 Correcteur proportionnel.....	15
II.3.2 Correcteur Proportionnel Dérivé (PD) .....	15
II.3.3 Correcteur Proportionnel Intégral (PI).....	15
II.3.4 Correcteur Proportionnel Intégral Dérivé (PID).....	16
II.4 Différentes méthodes de la synthèse des paramètres du PID.....	16
II.4.1. Méthode de Ziegler-Nichols.....	16
II.4.2. Méthode de Cohen-Coon.....	18
II.4.3. Réglage en fonction de la Réglabilité.....	19
II.4.4 Réglage par optimisation de critères.....	20
II.4.5 Réglage basé sur la structure IMC (Internal Model Control).....	22
II.5 Conclusion.....	23

## **Chapitre III Régulateurs PID flous.**

III.1 Introduction.....	24
III.2 Historique.....	25
III.3 Applications.....	25
III.4 Elément de base de la logique floue.....	26
III.4.1 Variables linguistiques.....	26
III.4.2 Fonctions d'appartenances.....	27
III.4.3 Dédutions floues (inférences).....	29

# SOMMAIRE

III.4.4 Opérateurs de la logique floue.....	30
III.5 Réglage par logique floue.....	32
III.5.1 Schéma de principe d'un réglage par logique floue.....	32
III.5.2 Configuration interne d'un régulateur flou.....	33
III.5.2.1 Fuzzification.....	34
III.5.2.2 Inférences.....	35
III.5.2.3 Défuzzification.....	36
III.6 Réglage d'un régulateur flou.....	37
III.7 Régulateurs par logique floue de type PD, PI et PID.....	38
III.7.1 Structure des régulateurs .....	39
III.7.1.1 Correcteur flou de type PD .....	39
III.7.1.2 Correcteur flou de type PI.....	41
III.7.1.3 Correcteur flou de type PID.....	41
III.7.2 Relation de passage.....	42
III.7.3 Réalisation numérique et équivalente.....	43
III.8 Propriété d'un régulateur par logique floue.....	45
III.9 Avantages et inconvénients du réglage par logique floue.....	45
III.9.1 Avantages.....	45
III.9.2 Inconvénients.....	45
III.10 Conclusion .....	46

# SOMMAIRE

## Chapitre IV Simulation.

IV.1 Introduction.....	47
IV.2 Résultat de simulation d'un correcteur classique .....	47
IV.2.1 Résultat de simulation d'un correcteur proportionnel.....	47
IV.2.2 Résultat de simulation d'un correcteur Proportionnel Intégral.....	48
IV.2.3 Résultat de simulation d'un correcteur PID Classique.....	49
IV.3 Résultats de simulation d'un régulateur flou .....	60
IV.3.1 Résultat de simulation d'un régulateur PI flou sous simulink.....	64
IV.3.2 Résultat de simulation d'un régulateur PD flou sous simulink.....	65
IV.3.3 Résultat de simulation d'un régulateur PID flou sous simulink.....	66
IV.3.4 Exemple numérique (PI flou) .....	67
IV.4 Conclusion.....	69
 <i>Conclusion générale</i> .....	 70

## V Annexe.

## *Quelques notations*

$e$  : l'erreur.

$de$  : Dérivé de l'erreur.

$ie$  : l'intégrale de l'erreur.

$G_e, G_{de}, G_{ie}, G_{cu}, G_u$  : Paramètre d'adaptation du régulateur flou.

$U$  : Grandeur de commande.

$U_{com}$  : Signal de commande.

$U_{pd}$  : Sortie du régulateur PD.

$U_{pi}$  : Sortie du régulateur PI.

$U_{pid}$  : Sortie du régulateur PID.

$K_p$  : Gain de l'action proportionnelle.

$K_i$  : Gain de l'action intégrale.

$K_d$  : Gain de l'action dérivée.

$T$  : Période d'échantillonnage.

$X$  : Variable linguistique.

$X_R$  : Variable de sortie.

$S$  : Système à régler.

$Y_m$  : Grandeur interne ou mesurée.

$Y$  : Grandeur à régler.

$W$  : Grandeur de consigne.

$V$  : Une perturbation

$\mu(x)$  : Fonction d'appartenance.

$\mu_{rés}(x)$  : Fonction d'appartenance résultante.

$X_R^*$  : Signal de sortie.

$C(s)$  : Fonction de transfert du correcteur.

$T_i$  : Constante d'intégration.

$r(t)$  : Signal de commande de sortie du correcteur.

$r_s(t)$  : Signal de décentrement de l'actionneur.

$K_c$  : Gain proportionnel.

## *Liste des abréviations*

P : Correcteur proportionnel.

P.I : Correcteur proportionnel intégral.

P.I.D : Correcteur proportionnel intégral dérivée.

ISE : Intégrale du carré de l'erreur.

IAE : Intégral de la valeur absolue de l'erreur.

ITAE : Intégral de la valeur absolue de l'erreur pondérée par le temps.

OCM : organe de commande.

RLF : régulateur par logique floue.

***Introduction***  
***Générale***



## Introduction générale

Le second chapitre est consacré au réglage conventionnel (réglage par régulateur standard PID). Nous citons les différentes actions de ces régulateurs, leurs structures et caractéristiques ainsi que les techniques de réglage des régulateurs PID.

Nous détaillons dans le troisième chapitre le principe de la commande floue des processus industriels.

Nous présentons dans le quatrième chapitre quelques résultats de simulation des deux correcteurs PID classique et flou.

Nous terminons par une conclusion et quelques perspectives .

**Chapitre I**  
**Principe de**  
**l'Asservissement**

## I.1 INTRODUCTION

En automatique, lorsque l'on souhaite atteindre une certaine vitesse, température, position, angle..., il est donc très souvent nécessaire d'avoir recours à un asservissement, c'est à dire un système capable d'atteindre et de maintenir une consigne en utilisant une mesure. Il s'agit donc d'un système bouclé, dont il reste à déterminer la fonction permettant de corriger la commande en fonction de la consigne initiale et de l'erreur mesurée.

## I.2 LES SYSTEMES ASSERVIS. [3]

Un asservissement est un système de commande en boucle fermée dont le signal de commande dépend de l'écart entre la grandeur consigne et la grandeur sortie du système.

L'objectif d'un asservissement est de rendre cet écart le plus faible possible c'est à dire la sortie doit suivre à chaque fois la consigne.

Dans un asservissement on distingue deux fonctions :

**Fonction de régulation :** Elle a pour objectif de maintenir la sortie proche de la consigne, quelque soit les perturbations.

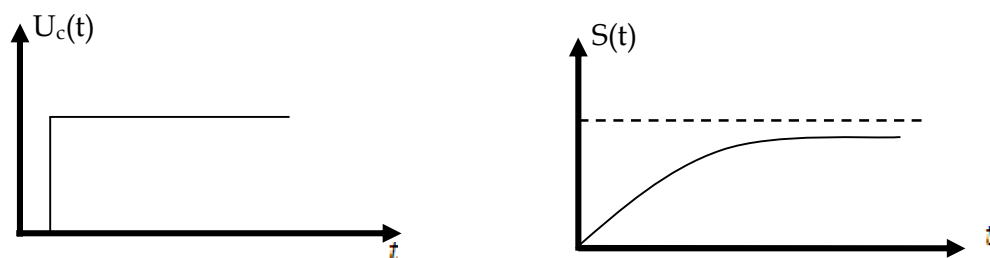


Figure I.1 Fonction de régulation d'un système asservi.

### Fonction de poursuite

Son objectif est que la sortie puisse atteindre la consigne malgré ses variations.

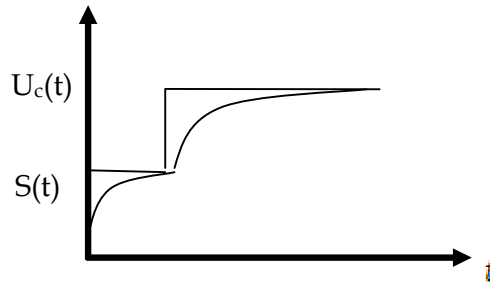


Figure I.2 Fonction de poursuite d'un système asservi.

En pratique, souvent un asservissement doit assurer ces deux fonctions simultanément.

### I.3 SCHEMA FONCTIONNEL D'UN ASSERVISSEMENT. [5]

Le schéma fonctionnel ou le diagramme fonctionnel d'un asservissement est une représentation sous forme de bloc.

D'une manière générale, il se présente comme suit :

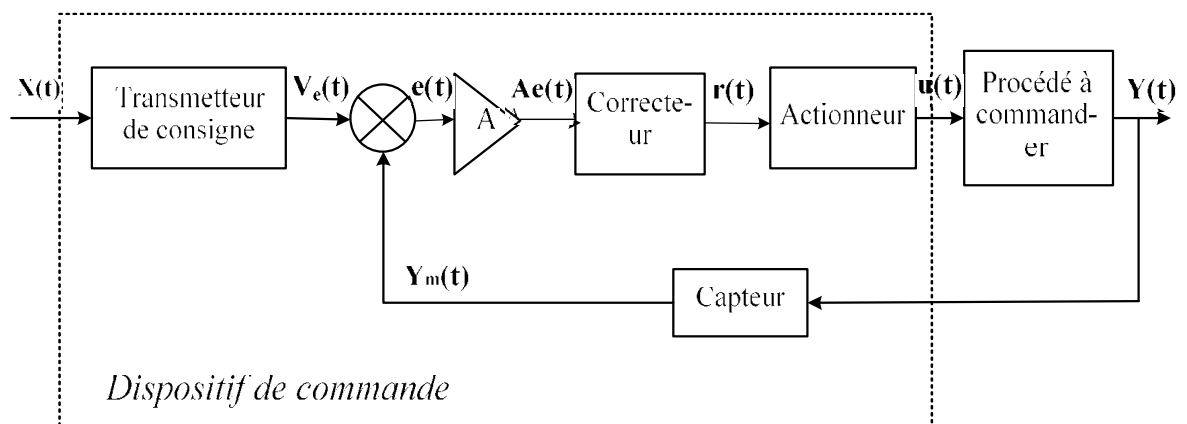


Figure I.3 Schéma fonctionnel d'un asservissement.

- **Transmetteur de consigne**

C'est un élément dont le rôle est de convertir une consigne qui est de nature physique en une grandeur de nature électrique comprise par les dispositifs de la chaîne d'asservissement, on l'appelle aussi Afficheur de consigne.

- **Capteur**

C'est un dispositif de mesure qui permet de convertir la sortie de grandeur physique en une information en général de type électrique.

Exemple

- Thermocouple : capteur de température.
- Dynamo tachymétrique : capteur de vitesse de rotation d'un moteur électrique.

- **Comparateur**

Il délivre l'écart entre la consigne convertie et la sortie mesurée.

- **Amplificateur d'écart**

Il amplifie l'écart afin que la chaîne d'action soit sensible aux écarts de faibles amplitudes.

- **Correcteur**

C'est un dispositif inséré dans la chaîne d'asservissement dont le rôle est d'améliorer les performances afin d'atteindre les exigences du cahier de charge.

- **Actionneur**

C'est un dispositif qui permet de transmettre la commande fournie par le correcteur ou le procédé.

- **Le procédé**

C'est le système physique à commander.

## I.4 PERFORMANCES D'UN ASSERVISSEMENT

L'un des objectifs essentiels qui peuvent être fixés lors de la mise en œuvre d'un système asservi est de réaliser un dispositif avec de bonnes performances. Par exemple, dans le cas d'une machine outil, il est particulièrement souhaité qu'il n'y ait pas d'erreur d'exécution de cette machine par rapport à un opérateur.

Généralement, pour évaluer leurs performances, les systèmes sont testés à l'aide d'une commande d'entrée du type échelon. Pour un système dont la grandeur d'entrée est  $E$  et la sortie  $S$ , sa réponse temporelle à un échelon d'entrée  $u(t)$  présente l'allure décrite dans la Figure I.4. Cette réponse indicielle est généralement oscillatoire et elle est caractéristique de nombreux systèmes bouclés. Considérons la réponse transitoire de la Figure et notons tout d'abord la grandeur de sortie  $s(t)$  de façon instantanée. La réponse transitoire rallie une première fois la consigne à l'instant  $t_r$  puis effectue un premier dépassement  $D$  exprimé généralement en %. Dans le cas de la Figure I.4, il y a une erreur statique. Cette erreur est dite de position et elle est décrite par  $\varepsilon$ .

Ainsi, la réponse indicielle d'un système asservi peut permettre de qualifier ses performances en mettant en évidence sa précision statique, son dépassement et le temps que mettra le système pour rallier sa position définitive ou par approcher celle-ci par exemple à 5% près.

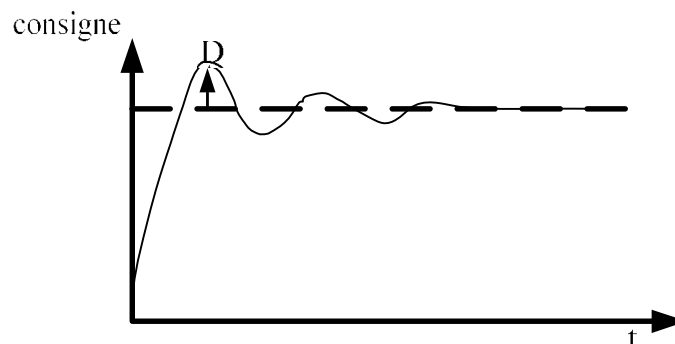


Figure I.4 Réponse indicielle à un échelon.

L'étude des asservissements linéaires permet aussi, grâce à des techniques d'analyse, d'améliorer les performances dynamiques de ces systèmes réduisant leur temps de réponse, en augmentant leur précision, etc...

#### I.4.1 Stabilités d'un système

Le système est stable par rapport à la relation entrée-sortie si pour toute entrée bornée  $|e(t)| \leq M_e$ , la sortie correspondante est bornée c'est-à-dire :  $|s(t)| \leq M_s$ , c'est la BIBO stabilité (Bounded Input-Bounded Output)

### I.4.2 Condition de stabilité

Un système de fonction de transfert  $G(p)$  est stable si :

- les pôles de  $G(p)$  sont à partie réelle négative.
- $G(p)$  ne possède pas de pôle à l'origine.
- Les pôles imaginaires purs (partie réelle=0) de  $G(p)$  sont simples.

### I.4.3 Précision

La précision d'un système linéaire, en valeur relative, par rapport à la grandeur de référence  $e(t)$ , est définie par son signal d'erreur  $\varepsilon(t)$ .

Selon que le système asservi se trouve en régime statique ou en régime dynamique, on parle de précision statique [régime permanent variable :  $e(t) = t$ ].

### I.4.4 Rapidité

La rapidité d'un système linéaire est définie par le temps de réponse à 5% ou à 95%. Le temps de réponse à 5% est le temps mis par la réponse du système pour atteindre 95% de sa valeur finale.

Si  $t_r(5\%)$  est grand  $\longrightarrow$  système lent.

Si  $t_r(5\%)$  est petit  $\longrightarrow$  système rapide.

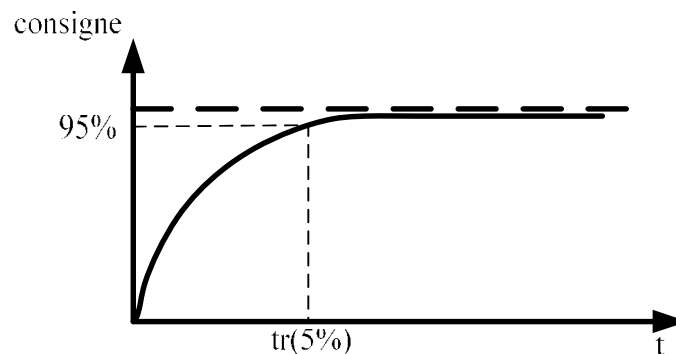


Figure I.5 Réponse indicielle à un échelon.  $t_r$ : temps de réponse.

### I.4.5 Qualité

La fonction de coût est une expression mathématique qui fournit une appréciation objective de la qualité de la régulation effectuée par le système dans des conditions déterminées. La régulation optimale est la régulation pour laquelle la fonction de coût adoptée est minimale, compte tenu de certaines limitations imposées.

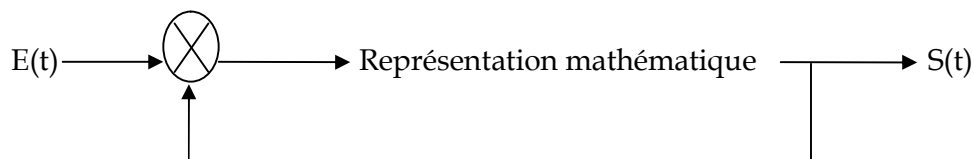
## I.5 DIFFERENTES ETAPES DE REALISATION D'UN ASSERVISSEMENT

- **Données**
  - ✓ processus à asservir
  - ✓ Cahier de charge comportant les spécificiques des performances a atteindre
- **Etape1**
  - ✓ définir la grandeur de sortie à commander et la grandeur en entrée qui sera l'unité de commande



- **Etape2**
  - ✓ Obtenir un modèle mathématique du système (équations différentielles, fonction de transfert).

- **Etape3**
  - ✓ analyse mathématique d'un asservissement





- ✓ déterminer les performances d'un asservissement :
  - Stabilité.
  - Précision statique.
  - Précision dynamique.
  - Analyse temporelle et analyse fréquentielle.

#### ▪ Etape4

- ✓ Le calcul d'un correcteur inséré dans la chaîne d'asservissement nous permet d'atteindre les performances exigées dans le cahier de charge.

→ Tests de validation et de vérification.

#### ▪ Etape5

- ✓ conception, réalisation pratique et insertion réelle du correcteur dans la chaîne d'asservissement.

## I.6 CONCLUSION

Plusieurs problèmes existent lors de la commande d'un système asservi, pour y remédier à ces problèmes, le principe est d'insérer dans l'asservissement un dispositif appelé « **Régulateur** », afin d'améliorer les performances de cet asservissement c'est-à-dire :

- Comment rendre stable un asservissement instable ?
- Comment diminuer ou annuler les écarts statiques ?
- Comment augmenter la rapidité d'un asservissement jugé trop lent ?
- Comment diminuer les oscillations et les dépassements d'un asservissement mal amorti ?

# **Chapitre II**

## **Régulateurs PID**

### **Classique**

## II.1 INTRODUCTION

Bien que le contrôle proportionnel ait été employé par les Grecs en 250 avant J.C, pour réaliser un système de contrôle de niveau d'eau basé sur un principe similaire à celui de la chasse d'eau, on peut considérer que le régulateur PID à 3 modes (proportionnel, intégral et dérivé) daté des années 1930 où il est devenu commercialement disponible.

Nous présentons dans cette partie, le principe d'un régulateur PID classique. Nous expliquons les différentes actions qui sont : proportionnelle, intégrale et dérivée.

## II.2 LES ACTIONS ELEMENTAIRES

### II.2.1 Action Proportionnelle - P

On parle d'action proportionnelle lorsque le signal de commande  $r(t)$  varie de façon proportionnelle par rapport au signal d'erreur.

$$r(t) = k_c e(t) + r_b \quad \text{II.1}$$

$r_b$  : Signal de décentrement de l'actionneur qui est égale au signal agissant lorsque  $e(t)$  est nulle.

Il est ajusté pour que la sortie mesurée coïncide avec la sortie désirée à l'état stationnaire. En général, le correcteur est simplement de la forme :

$$c(s) = k_c \quad \text{II.2}$$

Avec A.O : amplificateur opérationnel idéal.

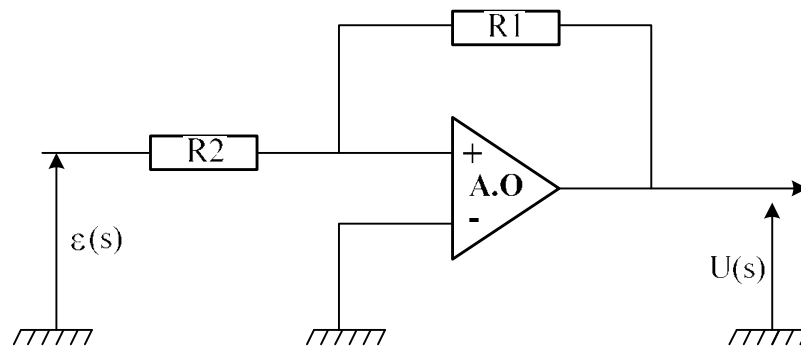


Figure II.1 Schéma de la réalisation électronique P

## II.2.2 Action Proportionnelle Dérivée - PD

### Principe de l'action dérivée

L'objectif de l'action dérivée est d'anticiper les variations à venir du signal de mesure (l'action dérivée a un effet prédictif).

### Fonction de transfert du PD

$$c(s) = k_c (1 + T_d s) \quad \text{II.3}$$

En basses fréquences, le régulateur PD se comporte comme un régulateur proportionnel simple. Lorsque, la fréquence est élevée, l'action dérivée l'emporte donc il se comporte comme un filtre passe haut.

### Remarque

- On n'utilise pas le régulateur PD seul mais on ajoute une action intégrale.
- Il existe une autre forme approchée du correcteur PD dite :

" Correcteur à avance de phase ", de fonction de transfert [2]

$$C_{ap}(s) = k_c \frac{1 + aT_d s}{1 + T_d s} \quad ; \quad a > 0 \quad \text{II.4}$$

qui est pratiquement réalisable.

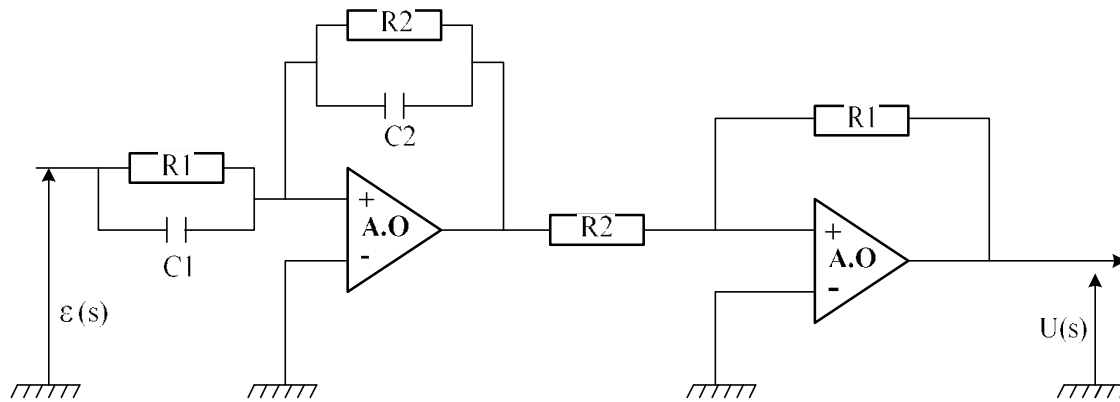


Figure II.2 Schéma de la réalisation électronique d'un correcteur à avance de phase

### II.2.3 Correcteur Proportionnel Intégral - PI

Le signal de commande est composé de la somme de deux termes d'actions proportionnelle et intégrale.

$$r(t) = K_c(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau) \tag{II.5}$$

La fonction de transfert du correcteur PI est de la forme :

$$C(s) = K_c(1 + \frac{1}{T_i s}) \tag{II.6}$$

L'action intégrale a pour effet de modifier la sortie du correcteur tant qu'une erreur existe à la sortie du procédé.

#### Remarque

- L'action intégrale prend en compte (intègre) le passé.
- Le correcteur proportionnel intégral présente l'avantage d'éliminer l'écart entre la sortie et la consigne.
- Un tel correcteur se comporte comme filtre passe bas.
- Ce type de correcteur possède une forme approchée dite :  
 " **Correcteur à retard de phase** ", de fonction de transfert :

$$C_{rp}(s) = k_c \frac{1 + T_i s}{1 + b T_s s} \quad ; \quad b > 0 \quad \text{II.7}$$

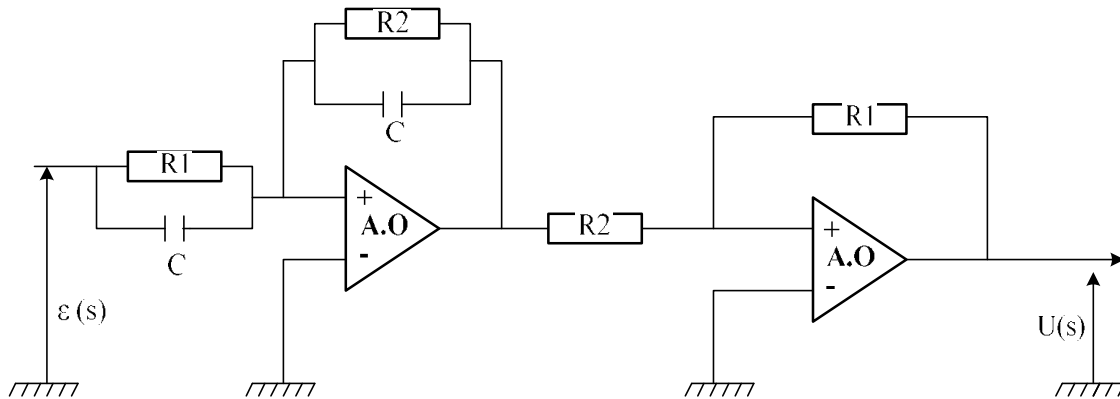


Figure II.3 Schéma de la réalisation électronique d'un correcteur à retard de phase

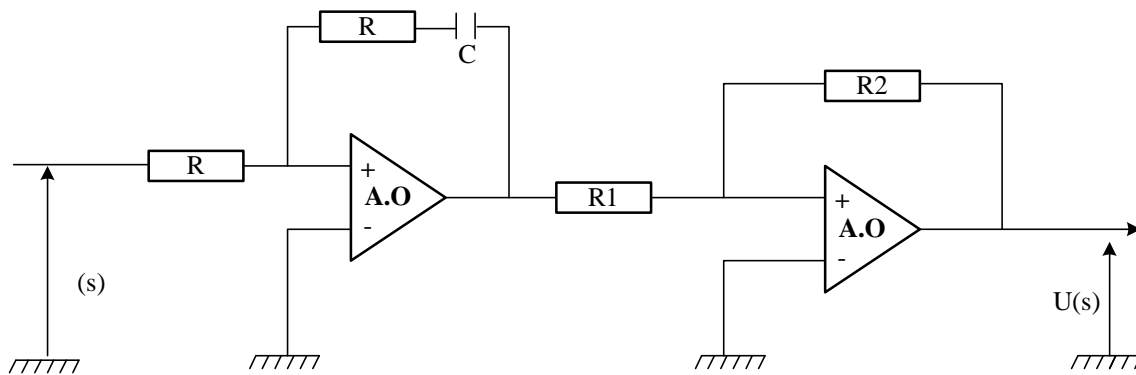


Figure II.4 Schéma de la réalisation électronique PI

### II.2.4 Correcteur Proportionnel Intégral D érivé - PID

La sortie agissante du correcteur proportionnel intégral dérivé est égale :

$$r(t) = K_c \left( e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i s} \int_0^t e(\tau) d\tau \right) \quad \text{II.8}$$

La fonction de transfert du PID est de la forme :

$$C(s) = K_c \left( 1 + T_d s + \frac{1}{T_i s} \right) \quad \text{II.9}$$

La version théorique du correcteur étudié ne peut être véritablement réalisée car le degré du numérateur est supérieur au degré du dénominateur.

Une version améliorée de ce correcteur théorique est proposée, pratiquement réalisable dite : correcteur PID réel de fonction de transfert [5]

$$C(s) = K_c \left( \frac{1 + T_i s}{T_i} \right) \left( \frac{1 + T_d s}{1 + \beta T_d s} \right) \tag{II.10}$$

II.2.5 Fonction de transfert du correcteur PID standard [2]

$$C(s) = K_c \left( 1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d s}{N}} \right) \tag{II.11}$$

Souvent N est très grand, pour simplifier les calculs, il est intéressant de considérer la fonction de transfert en boucle fermée avec un correcteur.

**Remarque**

Le correcteur PID est approché par une combinaison d'un correcteur à avance de phase et à retard de phase, sa fonction de transfert est de la forme :

$$c(s) = \frac{(1 + T_1 s)(1 + T_2 s)}{(1 + T_3 s)(1 + T_4 s)} \quad T_1 T_2 = T_3 T_4 \tag{II.12}$$

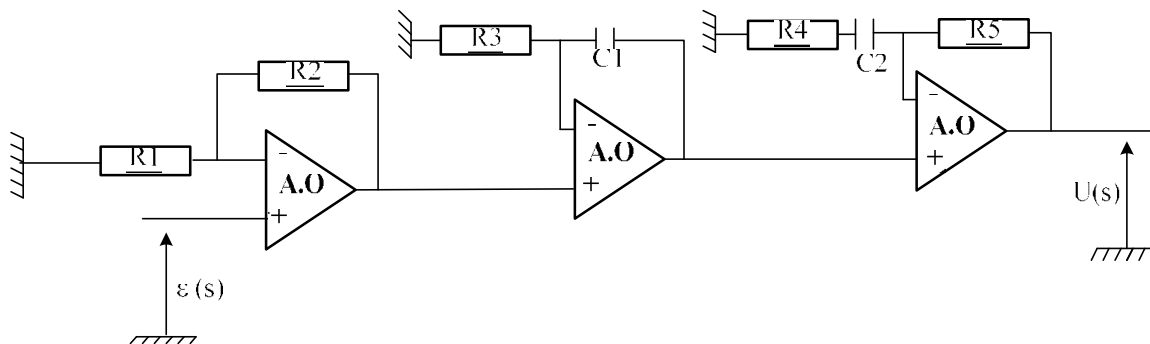


Figure II.5 Schéma électronique d'un PID série

## II.3 CARACTERISTIQUES DES CORRECTEURS

### II.3.1 Correcteur proportionnel

- Il possède un seul paramètre de réglage  $K_c$  (gain du correcteur).
- Augmenter  $K_c$  engendre une constante de temps plus petite ce qui implique que le système devient rapide mais l'augmenter de manière excessive peut provoquer l'instabilité du système ce qui nous oblige à faire un choix judicieux sur le paramètre.
- L'écart entre l'entrée (consigne) et la sortie est diminuée, ce qui augmente la précision du système (précision statique).

### II.3.2 Correcteur Proportionnel Dérivé - PD [5]

- Il se comporte comme un filtre passe haut.
- En basse fréquence, il se résume à une action proportionnelle simple.
- L'action dérivée à un effet stabilisateur.
- L'inconvénient qu'il n'est pas pratiquement réalisable ce qui justifie son utilisation rare.
- Il amplifie les bruits en hautes fréquences.

### II.3.3 Correcteur Proportionnel Intégral - PI

- En hautes fréquences, il se comporte comme un régulateur proportionnel simple.
- Son rôle est assimilé à un filtre passe bas.
- La présence de l'intégrateur engendre la suppression de l'écart entre la consigne et la sortie.
- Amélioration de la précision statique contrairement à la précision dynamique qui reste inchangée.
- La réponse peut devenir plus rapide en augmentant le gain mais risque d'être oscillatoire
- La diminution de la constante de temps intégrale rend la réponse plus rapide.

- L'action intégrale permet d'atténuer l'effet des perturbations.

### II.3.4 Correcteur Proportionnel Intégral Dérivé - PID

- Il permet d'améliorer la précision dynamique et statique.
- Un tel correcteur est moins sensible aux bruits et aux perturbations.
- L'action dérivée est limitée en hautes fréquences ce qui est un avantage car cela limite le bruit en hautes fréquences.

En résumé, le correcteur **PID** résulte et englobe les avantages de l'ensemble des correcteurs **P**, **PI**, **PD**. Son utilisation large est due aux performances qu'il peut assurer lors d'une correction ou d'une régulation.

## II.4 DIFFERENTES METHODES DE LA SYNTHES DES PARAMETRES DU PID. [2]

### II.4.1 Méthode de Ziegler - Nichols

J.G Ziegler et N.B Nichols ont été les premiers à montrer comment on pouvait choisir les paramètres d'un régulateur PID à partir d'une réponse à un échelon du procédé. Leur méthode, même si elle n'est pas la meilleure, jouit d'une notoriété telle qu'il n'est pas possible de l'ignorer.

L'idée consiste à approximer la réponse du procédé à un échelon unitaire, que l'on suppose apériodique, par un modèle du type (**Figure II.4**).

$$G(P) = \frac{K \cdot e^{-\theta \cdot P}}{1 + \tau \cdot P} \quad \text{II.13}$$

Le coefficient de pente R est défini comme étant :  $R = \frac{K}{\tau}$

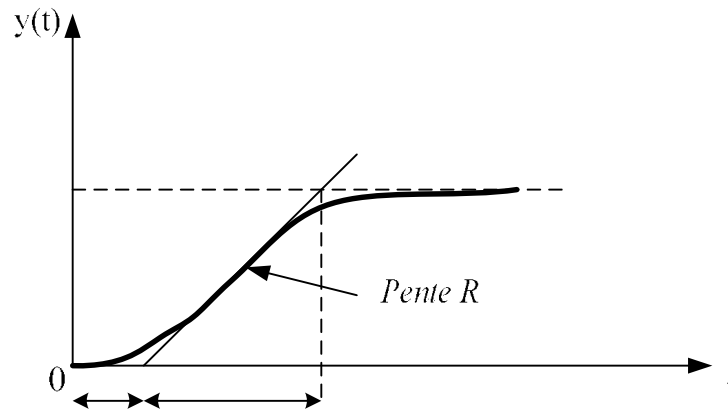


Figure II.6 Réponse d'un procédé stable et aperiodique à un échelon

Il suffit ensuite d'appliquer les relations du **Tableau II.1** suivant que l'on utilise un régulateur P, PI ou PID. Ces relations ont été développées empiriquement pour donner une réponse en boucle fermée oscillante, avec un dépassement initial de l'ordre de 30 à 40 % et avec un rapport d'amplitude des oscillations de 1/4 (rapport entre le dépassement de deux pics de même signe). pour un système de deuxièmes ordre, ceci correspond à un coefficient d'amortissement  $\varepsilon=0.2$  et un premier d'épassement de 50%.

De telles spécifications sont adaptées pour le comportement en régulation mais conduisent à des changements de consigne plutôt mouvementés. Cette approche est aussi valable pour un processus intégrateur. Le modèle recherché est de la forme :

$$G(P) = R \frac{e^{-\theta \cdot p}}{P} \tag{II.14}$$

Les actions P, I et D peuvent ensuite être calculées à partir du tableau suivant :

Type de régulateur	$K_c$	$\frac{1}{T_i}$	$T_d$
P	$\frac{1}{\theta \cdot R}$	-	-
PI	$\frac{0.9}{\theta \cdot R}$	3.336	-
PID	$\frac{1.2}{\theta \cdot R}$	$2\theta$	$0.5\theta$

Tableau II.1 Réglage de Ziegler-Nichols à partir d'une réponse

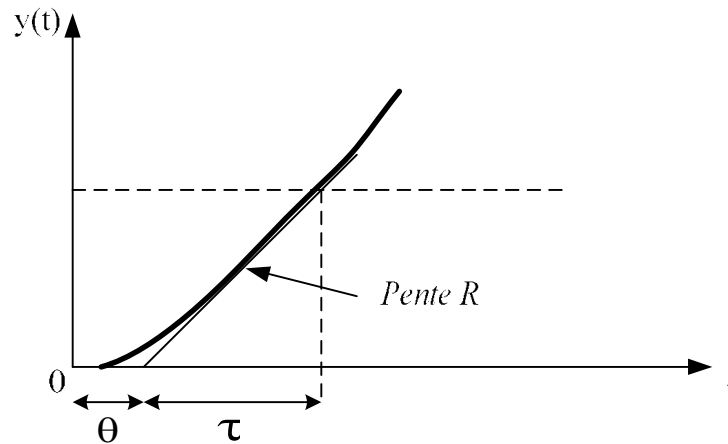


Figure II.7 Réponse d'un procédé instable à un échelon

II.4.2 Méthode de Cohen-Coon

La méthode de Cohen-Coon est très proche de celle de Ziegler-Nichols. Nous la citons ici par référence. Pour un modèle de la forme :

$$G(P) = \frac{k * e^{-\theta \cdot p}}{1 + \tau \cdot p} \tag{II.15}$$

Les actions P, I et D se calculent en s' appuyant sur les relations du tableau ci-dessous :

Type de régulateur	$K_c$	$\frac{1}{T_i}$	$T_d$
P	$\frac{1}{K} \frac{\tau}{\theta} \left[ \frac{3 \cdot \tau + \theta}{3 \cdot \tau} \right]$	-	-
PI	$\frac{1}{K} \frac{\tau}{\theta} \left[ \frac{10.8 \tau + \theta}{12 \cdot \tau} \right]$	$\theta \frac{30 + 3(\theta/\tau)}{9 + 20(\theta/\tau)}$	
PID	$\frac{1}{K} \frac{\tau}{\theta} \left[ \frac{16 \cdot \tau + 3\theta}{12 \cdot \tau} \right]$	$\theta \frac{32 + 6(\theta/\tau)}{13 + 8(\theta/\tau)}$	$\frac{4\theta}{11 + 2(\theta/\tau)}$

Tableau II.2 Réglage de Cohen - Coon à partir d'une réponse indicielle

### II.4.3 Réglage en fonction de la Réglabilité

Comme nous venons de le voir, les méthodes de Ziegler -Nichols et celle de Cohen-Coon sont difficiles à appliquer en pratique car elles conduisent à un comportement très oscillant, particulièrement gênant lors des changements de consignes.

C'est la raison pour laquelle les instrumentistes utilisent une version adoucie de ces réglages, basée sur le même type de modélisation, à savoir un modèle de type premier ordre avec retard.

Les coefficients du PID, déterminés pour obtenir une réponse bien amortie, se calculent en fonction du "coefficient de réglabilité" :  $r_c = \frac{\theta}{\tau}$

ou  $\tau$  est la constante de temps du procédé et  $\theta$  son retard.

Notons que ce rapport intervient aussi dans les réglages de Ziegler -Nichols et de Cohen-Coon. Il traduit l'importance du retard par rapport à la constante de temps du système (**Figure II.6**). Plus ce rapport est élevé, et plus les performances qu'on peut espérer avec un régulateur PID seront limitées, car plus son réglage devra être amorti.

Le **Tableau II.3** donne les relations pour calculer les coefficients du régulateur et déterminer son type. On réservera le PID pour les cas où le système possède un retard important que l'action dérivée peut compenser. Par contre, dans les autres cas, un simple PI suffira.

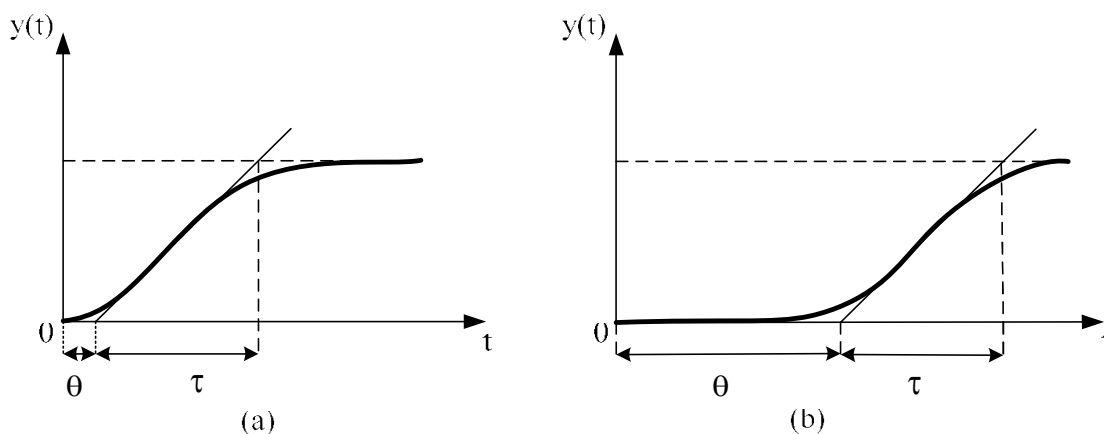


Figure II.8 Systèmes à forte réglabilité (a) et à faible réglabilité (b)



Par rapport au critère ISE, ce critère donne plus de poids aux faibles erreurs.

- Le critère ITAE (Integral of the time-weighted absolute error)

$$ITAE = \int_0^{\infty} |\epsilon(t)| t \cdot dt \tag{II.18}$$

Ce critère privilégie une faible erreur statique au prix d'un dépassement initial qui peut être important, puisqu'il a lieu pour de faibles valeurs de t.

$$K_c = \frac{1}{K} a \left[ \frac{\theta}{\tau} \right]^b$$

$$T_i = K_c \frac{\tau}{c+d} \left[ \frac{\theta}{\tau} \right]^{-c} \quad \text{pour le réglage " asservissement "}$$

$$T_i = K_c \frac{\tau}{c} \left[ \frac{\theta}{\tau} \right]^{-c} \quad \text{pour le réglage " régulation "}$$

$$T_d = \frac{\tau}{K_c} e \left[ \frac{\theta}{\tau} \right]^f$$

Elles sont valables si  $0.1 \leq \frac{\theta}{\tau} \leq 1$ . Dans les cas où le retard est très faible par rapport à la constante de temps

	Réglage Asservissement			Réglage Régulation		
	ISE	IAE	ITAE	ISE	IAE	ITAE
A	1.26239	1.13031	0.98384	1.3466	1.31509	1.3176
B	-0.83880	-0.81314	-0.49851	-0.9308	-0.8826	-0.7937
C	6.03560	5.7527	2.71348	1.6585	1.2587	1.12499
D	-6.0191	-5.7241	-2.29778	-1.25738	-1.3756	-1.42603
E	0.47617	0.32175	0.21443	0.79715	0.5655	0.49547
F	0.24572	0.17707	0.16768	0.41941	0.4576	0.41932

**Tableau II.4 Coefficients pour les réglages ISE, IAE et ITAE d'un PID**

II.4.5 Réglage basé sur la structure IMC (Internal Model Control)

La structure du régulateur à modèle interne, que nous présentons, est dans certaines conditions équivalente à un régulateur PID. Il est alors possible d'utiliser cette approche comme méthode de réglage de régulateur PID.

Modèle	$K_c$	$\frac{1}{T_i}$	$T_d$
$\frac{K}{1 + \tau p}$	$\frac{\tau}{\tau_c}$	$\tau$	-
$\frac{K \cdot e^{-\theta p}}{1 + \tau p}$	$\frac{\tau}{\tau_c + \theta}$	$\tau$	-
$\frac{K}{(1 + \tau_1 p)(1 + \tau_2 p)}$	$\frac{\tau_1 + \tau_2}{\tau_c}$	$\tau_1 + \tau_2$	$\frac{\tau_1 \tau_2}{\tau_1 + \tau_2}$
$\frac{K(1 + \tau_1 p) \cdot e^{-\theta p}}{(1 + \tau_1 p)(1 + \tau_2 p)}$	$\frac{\tau_1 + \tau_2 - \tau_3}{\tau_c + \theta}$	$\tau_1 + \tau_2 - \tau_3$	$\frac{\tau_1 \tau_2}{\tau_1 + \tau_2 - \tau_3} - \tau_3$
$\frac{K}{1 + 2\epsilon \tau p + \tau^2 p^2}$	$\frac{2\epsilon \tau p}{\tau_c}$	$2\epsilon \tau p$	$\frac{\tau}{2\epsilon}$
$\frac{K(1 + \tau_3 p) \cdot e^{-\theta p}}{1 + 2\epsilon \tau p + \tau^2 p^2}$	$\frac{2\epsilon \tau p}{(\tau_c + \tau_3 + \theta)} + \frac{\tau_3 \theta}{(\tau_c + \tau_3 + \theta)^2}$	$2\epsilon \tau + \frac{\tau_3 \theta}{(\tau_c + \tau_3 + \theta)}$	$\frac{\tau_3 \theta}{(\tau_c + \tau_3 + \theta)} + \frac{\tau^2}{2\epsilon \tau + \frac{\tau_3 \theta}{(\tau_c + \tau_3 + \theta)}}$
$\frac{K}{\bar{p}}$	$\frac{1}{\tau_c}$	-	-
$\frac{K \cdot e^{-\theta p}}{p}$	$\frac{2\tau_c + \theta}{(\tau_c + \theta)^2}$	$2\tau_c + \theta$	-

Tableau II.5 Réglage PID basé sur la structure modèle interne

Les relations données dans le **Tableau II.5** sont calculées par un régulateur idéal, c'est-à-dire sans filtrage de la dérivée. Ils sont utilisables avec un PID à dérivées filtrées si  $N > 10$ . Le paramètre  $\tau_c$  correspond à la constante de temps désirée en boucle fermée.

Nous précisons toutefois, que cette approche est basée sur une compensation des pôles du système par ceux du régulateur, dont nous avons présenté les avantages et inconvénients plus haut. Par ailleurs, pour obtenir le réglage PID pour les systèmes avec retard.

## II.5 CONCLUSION

Le contrôleur PID pneumatique s'est répandu dans l'industrie dans les années 40. Il a été remplacé par le contrôleur électronique dans les années 50 et les premières applications des ordinateurs ont eu lieu au début des années 60 mais ce n'est qu'avec l'avènement du circuit intégré numérique dans les années 70 que l'électronique digitale s'est répandue.

De nos jours, la puissance disponible sur les automates permet d'envisager l'utilisation d'algorithmes beaucoup plus sophistiqués que le PID à 3 modes des années 30. De plus, d'autres types de PID ont pris place dans le domaine de l'industrie, comme exemple nous citerons le PID flou qui fera l'objet du chapitre suivant.

# **Chapitre III**

## **Régulateurs PID**

### **Flou**

### III.1 INTRODUCTION

La logique floue a été développée dans l'objectif de proposer un cadre formel pour travailler sur des notions imprécises. Aujourd'hui, elle est de grande actualité. Son domaine d'application devient de plus en plus important, notamment celui du réglage et de la commande des processus industriels liés à l'énergie, les transports, la robotique, l'électroménager, les machines-outils...etc.

Le contrôle flou a connu un certain intérêt dans de nombreux domaines depuis quelques années, notamment dans le monde industriel, en particulier grâce à ses avantages tels que les performances dynamiques, la robustesse ou encore la possibilité de prendre en compte une connaissance expérimentale du processus.

Cette technique a été rapidement introduite dans des produits grand public comme les appareils photos, les machines à laver..., ce qui a contribué sans doute à renforcer l'image de faisabilité du contrôle flou.

La commande par la logique floue peut s'appliquer à tout domaine de la commande traditionnelle. De plus, elle peut opérer lorsque les procédés à commander sont mal connus ou difficiles à décrire précisément. Son intérêt devient plus évident dans le cadre de la régulation des systèmes complexes, pour lesquels les informations sont exprimées d'une façon vague et imprécise.

Dans ce chapitre, nous présentons le principe général et la théorie de base de la logique floue. Cela englobe des aspects de la théorie des possibilités qui fait intervenir des sous-ensembles d'appartenance appelés sous-ensembles flous caractérisant les différentes grandeurs du système à commander, et le raisonnement flou qui emploie un ensemble de règles floues établies par le savoir faire humain et dont la manipulation permet la génération de la commande adéquate ou la prise de décision. Ensuite, on va décrire les notions générales et l'architecture algorithmique et structurelle d'une commande floue, ou nous faisons le point sur :

- La Fuzzification
- Les Inférences floues

- La Défuzzification

Et enfin, on citera les propriétés, les avantages et les inconvénients de ce type de réglage.

## III.2 HISTORIQUE

Les bases théoriques de la logique floue ont été établies en 1965 par le professeur Lotfi A. Zadeh de l'université de Californie de Berkeley USA qui a éprouvé le besoin de formaliser la représentation et les traitements des connaissances imprécises ou approximatives [1].

En 1974 à l'Université de Londres, M. Mamdani a présenté le premier régulateur flou qu'il a utilisé sur une chaudière à vapeur qui est de grande complexité. Le succès de cette première expérimentation a donné lieu à d'autres applications en Europe. Une des premières applications industrielles a été la commande d'un four à ciment de Danemark en 1980. Depuis, des applications concernant des domaines aussi variés que la commande de machines-outils, d'appareils électroménagers, de caméra..., ont été développées [7].

A partir de 1985, ce sont les japonais qui commencent à utiliser la logique floue dans des produits industriels pour résoudre des problèmes de réglages et de commande.

## III.3 APPLICATIONS

Les principaux domaines de recherche et application de la logique floue sont les suivants : [9].

- Appareils électroménagers (lave-linge, aspirateurs, autocuiseurs, ...etc.).
- Systèmes audio-visuels (appareils de photos autofocus, caméscope à stabilisateur d'image, photocopieur,.....).
- Systèmes automobiles embarqués (ABS, suspension, climatisation, ...etc. ).
- Systèmes de transport (train, métro, ascenseur,...).

- Systèmes de diagnostic, décision, reconnaissance.
- Systèmes de contrôle / commande dans la plupart des domaines industriels de production, transformation.
- Instrumentation : capteurs, instruments de mesure, reconnaissance de voix et de caractère.

### III.4 ELEMENT DE BASE DE LA LOGIQUE FLOUE

La logique floue repose sur une série de notion fondamentales et une multitude de variantes pour les règles de traitement. Dans cette section, on se limitera aux éléments qui sont nécessaires pour effectuer un réglage par logique floue.

Les éléments de base de la logique floue : [1].

- Les variables linguistiques,
- Les fonctions d'appartenances,
- Les déductions floues (inférences),
- Les operateurs,
- Les sous-ensembles flous,

#### III.4.1 Variables linguistiques

Les variables linguistiques sont associées aux entrées / sorties d'un contrôleur. Une variable linguistique est une variable dont la valeur linguistique est un mot ou une phrase d'une langue naturelle ou artificielle. Pour bien comprendre cet élément, on introduit l'exemple suivant :

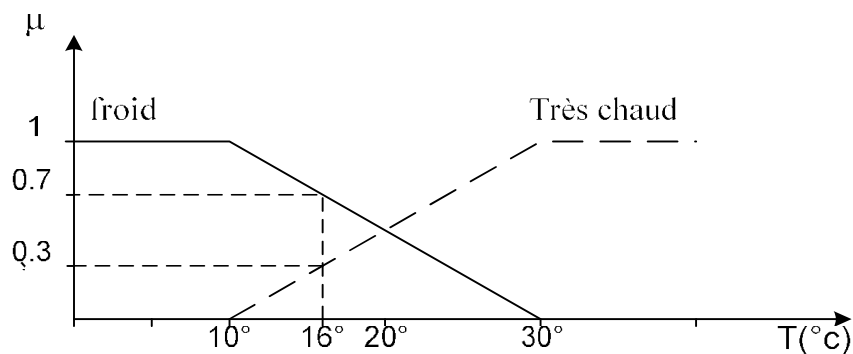
- On désire mesurer la température de l'air ambiant d'une pièce. Dans ce cas, la " température  $T$  " est la variable linguistique.
- On peut distinguer, dans un premier cas, deux valeurs " froid " et " très chaud " de cette variable " température ".

- Dans un autre cas, on peut introduire une subdivision plus fine avec quatre valeurs " froid ", " tiède ", " chaud " et " très chaud " de la variable linguistique " température ".
- On constate que ces variables linguistiques ne peuvent pas être traitées mathématiquement (pour traiter des déductions floues par ordinateur) ; cela nécessite donc de les soumettre à une définition à l'aide de fonctions d'appartenance.

### III.4.2 Fonction d'appartenance

Il s'agit d'établir une relation entre le degré de vérité de la variable floue et la grandeur d'entrée correspondante, c'est-à-dire attribuer à chaque valeur de la variable linguistique des fonctions d'appartenances  $\mu$ , dont la valeur varie entre 0 et 1 en tenant compte de la classification en un certain nombre de sous-ensembles flous.

Dans l'exemple précédent, on peut présenter les fonctions d'appartenance comme suit :



**Figure III.1** fonction d'appartenance avec deux sous-ensembles flous pour la variable linguistique (température).

- La température 16°C appartient avec un facteur d'appartenance  $\mu=0.7$  au sous-ensemble (froid) et avec  $\mu=0.3$  au sous-ensemble (très chaud).
- Concernant le deuxième cas, où nous avons introduit deux autres sous-ensembles flous, nous obtenons la représentation suivante des fonctions d'appartenance :

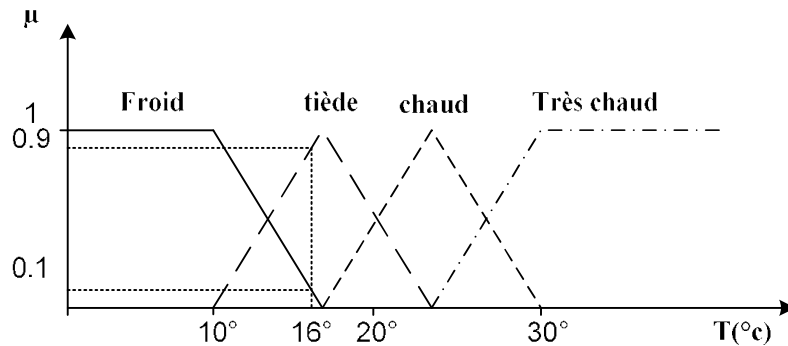


Figure III.2 fonction d'appartenance avec quatre sous-ensembles flous pour la variable linguistique (température)

- Dans ce cas, la température  $16^\circ\text{C}$  appartient avec  $\mu=0.1$  à l'ensemble (froid), avec  $\mu=0.9$  à l'ensemble (tiède), et elle n'appartient pas aux sous-ensembles (chaud) et (très chaud).

On peut choisir différentes formes de la fonction d'appartenance. En effet, il existe plusieurs formes mais le plus souvent, on utilise les formes trapézoïdales ou triangulaires qui sont suffisantes pour délimiter des sous-ensembles flous dans le cas de réglage par logique floue.

La Figure III.3 montre quelques formes de la fonction d'appartenance :

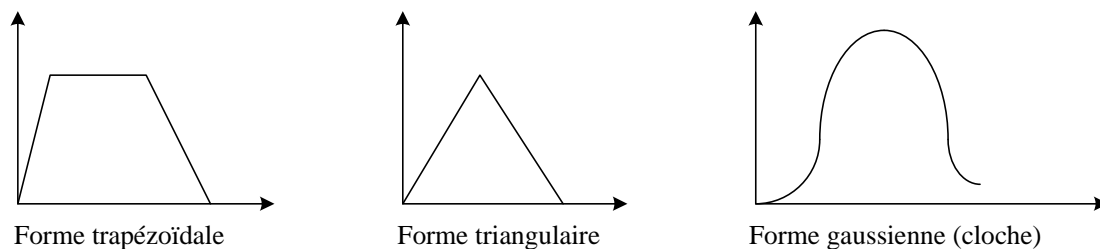


Figure III.3 différentes formes de la fonction d'appartenance.

### III.4.3 Déduction floues (inférences)

Les règles d'inférence permettent de relier les variables floues d'entrées aux variables floues de sorties à l'aide des différents opérateurs.

On distingue deux genres de règles :

#### - Inférences avec une seule règle

Ce cas se présente lorsqu'il faut comparer plusieurs concurrents (objets et personnes) dans une certaine situation et en choisir l'optimum.

$Y = [x_1 \text{ OU } (x_1 \text{ ET } x_2) \text{ OU } \dots] \text{ ET}$

avec Y : critère de prise de décision

$x_1, x_2, \dots, x_n$  : variables linguistiques

#### - Inférences avec plusieurs règles

Ce cas se présente lorsqu'une ou plusieurs variables nécessitent une prise de décision différente suivant les valeurs qu'atteignent ces variables.

Cette problématique se présente essentiellement pour des problèmes de réglage et de commande et la prise de décision conduit à l'exécution d'une opération.

Dans ce cas, les règles s'expriment sous la forme générale :

Opération 1 : = SI condition 1, ALORS opération 1 OU

SI condition 2, ALORS opération 2 OU

SI condition n, ALORS opération n.

Pour le cas de réglage de la température d'un local, les règles d'inférence seront par

Exemple :

Variation de la position de la valve :

SI écart positif, ALORS ouvrir la valve, OU

SI écart zéro, ALORS ne pas bouger la valve, OU

SI écart négatif, ALORS fermer la valve.

### III.4.4 Opérateurs de la logique floue

Dans les règles d'inférence, interviennent les opérateurs ET et OU. L'opérateur ET s'applique aux variables à l'intérieur d'une règle, tandis que l'opérateur OU lie les différentes règles.

De plus, il existe des opérateurs qu'on citera avec leurs différentes réalisations dans le tableau suivant :

Opérateur	Réalisation
NON	$\mu_c(x) = 1 - \mu_a(x)$
ET	$\mu_c(x) = \min(\mu_a(x), \mu_b(x))$ $\mu_c(x) = \mu_a(x) * \mu_b(x)$
OU	$\mu_c(x) = \max[\mu_a(x), \mu_b(x)]$ $\mu_c(x) = [\mu_a(x) + \mu_b(x)]/2$
ET flou	$\mu_c(x) = \gamma \min[\mu_a(x), \mu_b(x)] + (1 - \gamma) * [\mu_a(x) + \mu_b(x)]/2$
OU flou	$\mu_c(x) = \gamma \max[\mu_a(x), \mu_b(x)] + (1 - \gamma) * [\mu_a(x) + \mu_b(x)]/2$
Opérateur min-max	$\mu_c(x) = \gamma \min[\mu_a(x), \mu_b(x)] + (1 - \gamma) * \max[\mu_a(x), \mu_b(x)]$
Opérateur $\gamma$	$\mu_c(x) = [\mu_a(x), \mu_b(x)]^{1-\gamma} * \{1 - [1 - \mu_a(x)] * [1 - \mu_b(x)]\}^\gamma$

Les Figures suivantes mettent en évidence les réalisations des opérateurs NON, ET et OU :

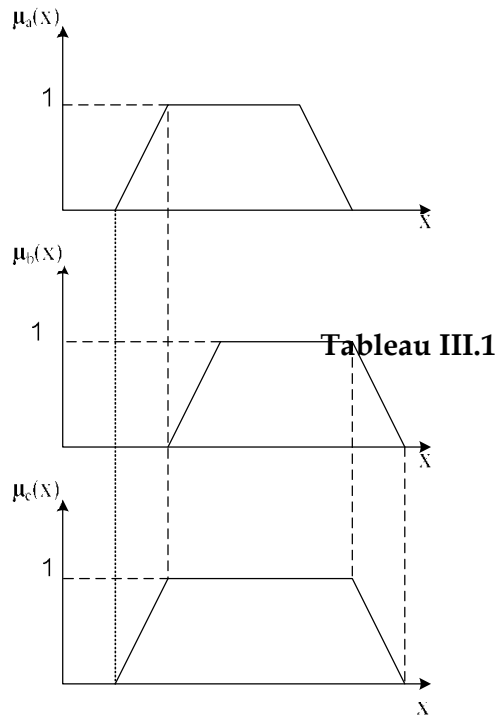


Figure III.4 Opérateur OU réalisé par la formation du max

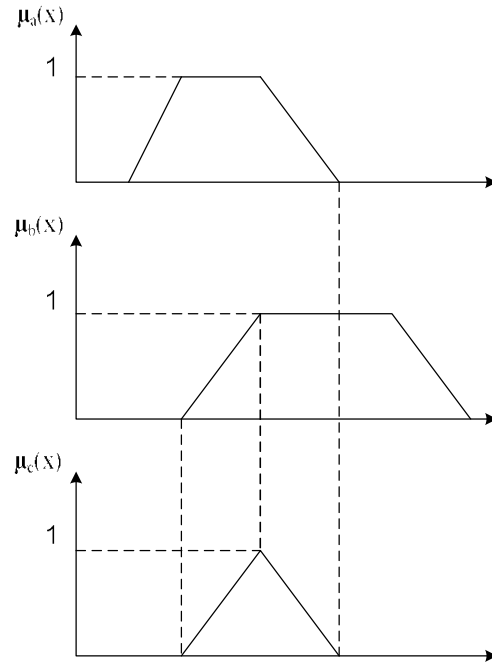


Figure III.5 opérateur ET réalisé par la formation du min.

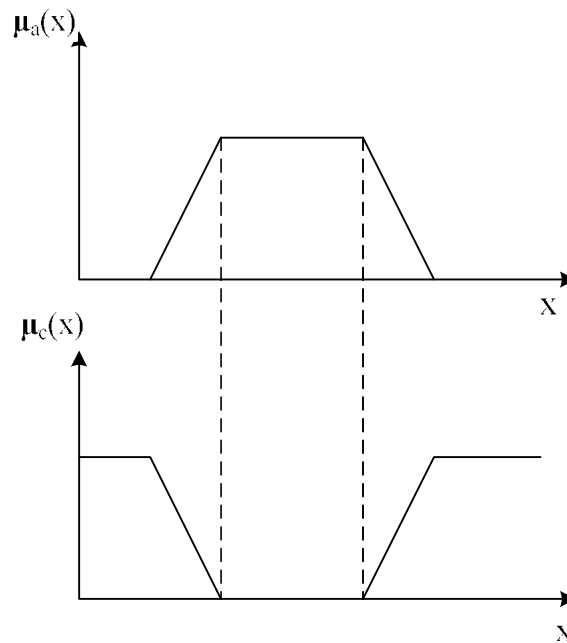


Figure III.6 opérateur NON.

### III.5 REGLAGE PAR LOGIQUE FLOUE

Une des plus importantes applications de la logique floue est celle du réglage et de la commande industrielle. En effet, cette méthode permet d'obtenir une loi de réglage très efficace sans devoir faire des études théoriques approfondies sur le système.

#### III.5.1 Schéma de principe d'un réglage par logique floue [6]

La Figure III.7 représente le schéma de principe d'un réglage par la logique floue où :

**S** : système à réguler, **OCM** : organe de commande,

**RLF** : régulateur par logique floue, il fournit le signal de commande  $U_{cm}$ .

$U_{cm}$  : signal de commande,  $Y_m$  : grandeur(s) interne(s) ou mesurée(s),

**W** : grandeur de commande, **V** : une perturbation.

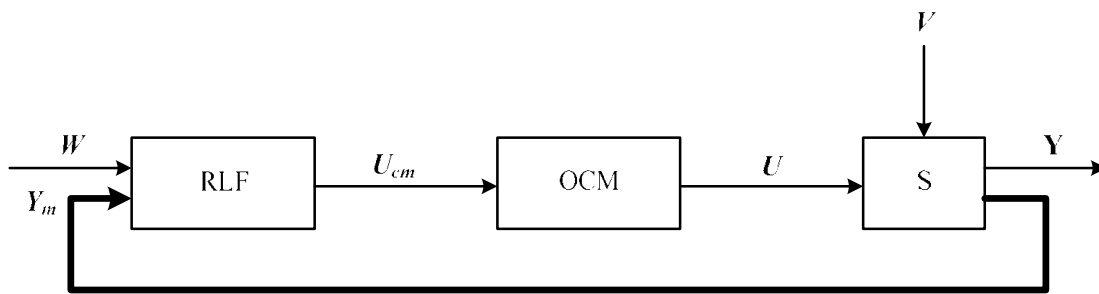


Figure III.7 schéma de principe d'un réglage par logique floue.

### III.5.2 Configuration interne d'un régulateur flou

Le principe d'un régulateur flou est le suivant :

La commande est déduite d'un raisonnement élaboré à partir de l'état observé du procédé et d'une liste de règles décrivent la manière selon laquelle le régulateur doit fonctionner. L'ensemble des opérateurs ne porte non pas sur les valeurs réelles mais sur des valeurs qualitatives exprimées par des termes linguistiques ou par un code approprié.

La Figure III.9 présente le schéma classique d'une commande floue et de son régulateur qui est composé de trois modules : [4]

- L'interface réelle-floue (la Fuzzification),
- La déduction (l'interface floue),
- L'interface floue-réelle (la Défuzzification),

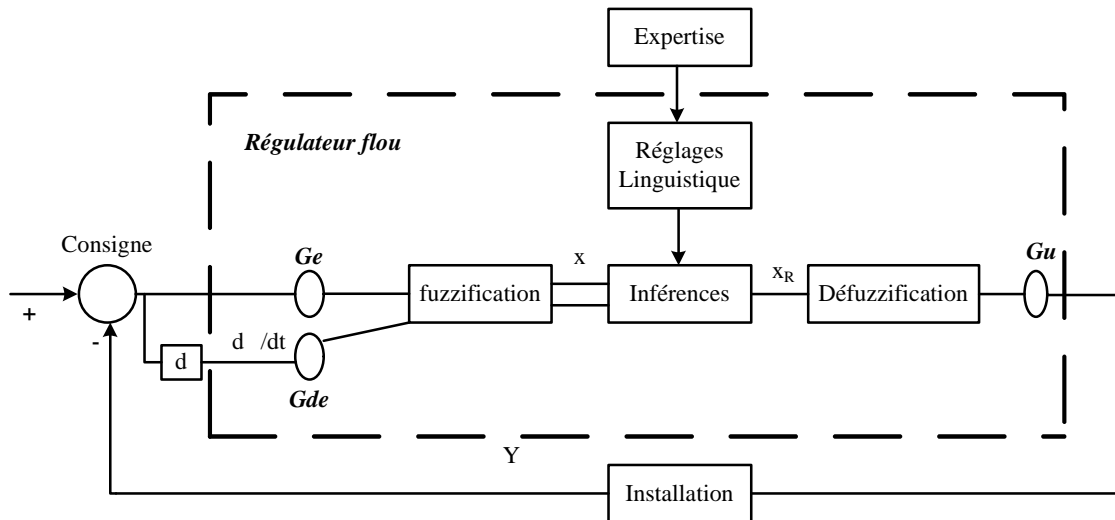


Figure III.9 schéma d'une commande floue et de son régulateur.

III.5.2.1 Fuzzification

Le bloc de Fuzzification contient en général un traitement de données préliminaires. La Fuzzification consiste à définir les fonctions d'appartenances pour les différentes variables. Ainsi, il s'agit de la conversion analogique/digitale. C'est-à-dire, le passage des grandeurs physiques (grandeurs déterminées) en variables linguistiques (variables floues) qui peuvent être exploitées par le bloc d'inférence [9]. Ces sous-ensembles flous sont caractérisés par des désignations standard illustrées dans le tableau suivant :

Symbole	Signification
NG	Négatif Grand
NM	Négatif Moyen
NP	Négatif Petit
EZ	Environ Zéro
PP	Positif Petit
PM	Positif Moyen
PG	Positif Grand

Tableau III.2 désignation standard des sous-ensembles

### III.5.2.2 Inférences

Le moteur d'inférence est le deuxième module d'un régulateur par logique floue. Il consiste à lier les grandeurs mesurées, qui sont les variables d'entrée  $x$  (transformées en variables linguistiques à l'aide de la Fuzzification), à la variable de sortie  $x_r$ , et à appliquer des règles de types (**SI** la condition **ALORS** l'action). Cette étape permet d'obtenir une information floue pour la variable de sortie  $x_r$  du régulateur [1].

Il existe plusieurs méthodes d'inférence, les plus utilisées pour le réglage par logique floue sont :

- Méthode d'inférence max-min : type Mamdani ;
- Méthode d'inférence max-produit : type Larsen ;
- Méthode d'inférence somme-produit : type Zadeh ;

Les trois méthodes sont résumées dans le tableau III.3

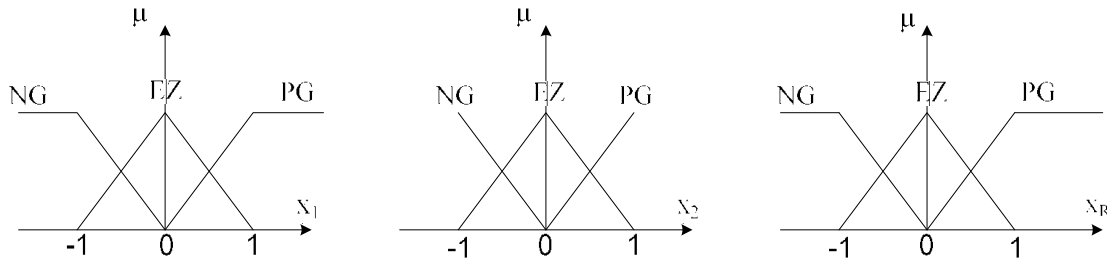
Méthode	Opérateurs sur prémisses (au niveau de la condition)		Opérateurs d'implication (au niveau de la conclusion)	Opérateurs D'agrégation (entre deux règles)
	ET	OU	ALORS (IMP)	
<b>Max-Min</b>	Min	Max	Min	Max
<b>Max-Prod</b>	Min	Max	Prod	Max
<b>Som-Prod</b>	Prod	Som	Prod	Som

*Tableau III.3 principales méthodes d'inférences*

Ces méthodes permettent le traitement numérique des inférences.

Afin de mettre en évidence ce traitement numérique, on fera appel à un cas particulier avec deux variables d'entrée  $x_1$  et  $x_2$  et une variable de sortie  $x_R$ . Chacune d'elles est décomposée en trois sous-ensembles **NG**, **EZ**, **PG** et définie par des fonctions d'appartenance. Comme la montre la Figure III .10.

On a comme exemple:



**Figure III.10** Fonction d'appartenance pour l'exemple servant à la description des méthodes d'inférence.

### III.5.2.3 Défuzzification

Cette opération consiste à transformer la grandeur floue de la variable de sortie  $\mu_{cm}$  en une valeur numérique qu'on peut appliquer au procédé (information floue en une information déterminée). De plus, on doit souvent prévoir un traitement du signal et la conversion digital / analogique [1].

Afin d'extraire la valeur numérique  $\mu_{cm}$  à appliquer au système à commander, il existe plusieurs approches pour réaliser cette transformation :

- Défuzzification par la technique du centre de gravité.
- Défuzzification par la technique de la moyenne pondérée.
- Défuzzification par la technique du maximum.

La plus utilisée est celle du centre de gravité, car elle permet d'éviter de grandes discontinuités et supprime toute ambiguïté. Mais son inconvénient réside dans sa complexité.

### - Méthode du centre de gravité

Cette méthode consiste à tracer sur le même diagramme les différentes courbes correspondantes à chacune des règles et à calculer le centre de la zone commune (Voir Figure III.11). [1].

L'abscisse du centre de gravité peut être déterminé à l'aide de la relation suivante :

$$x_R^* = \frac{\int_{-1}^1 x_R \mu_{Res}(x_R) dx_R}{\int_{-1}^1 \mu_{Res}(x_R) dx_R} \quad \text{III.1}$$

Le numérateur donne le moment de la surface, tandis que le dénominateur correspond à la surface.

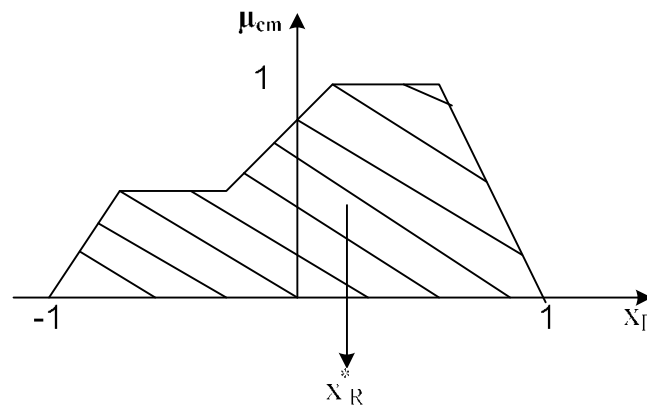


Figure III.11 Défuzzification par centre de gravité.

### Remarque

La Fuzzification, les inférences et la Défuzzification provoquent une caractéristique non linéaire du régulateur par logique floue. Il s'agit d'un avantage par rapport à un régulateur linéaire.

## III.6 REGLAGE D'UN REGULATEUR FLOU

Lors de la conception d'un réglage par logique floue, le régulateur doit subir différents réglages.

Les différents paramètres du régulateur qui peuvent être réglés sont les prémisses, les fonctions d'appartenances, la conclusion des règles, la méthode de

Défuzzification et les facteurs d'échelle. Ces derniers jouent un rôle important dans la stabilité et l'obtention de bonnes performances [4].

Dans cette section, on étudiera le réglage par les facteurs d'échelles du régulateur par logique floue.

On considère ici, le cas où les variables d'entrée sont l'erreur et sa dérivée.

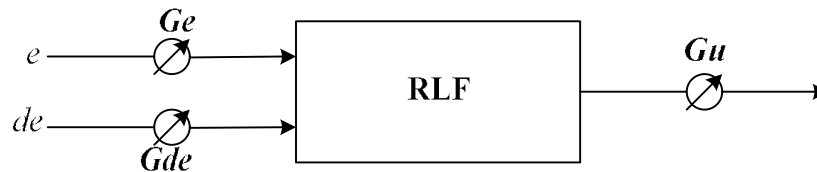


Figure III.12 Facteur d'échelle.

$Ge$  : gain de l'erreur, influe sur l'amplitude de la réponse.

$Gde$  : gain de la dérivée de l'erreur, influe sur la rapidité du système.

$Gu$  : gain de la commande.

Les trois facteurs d'échelle  $Ge$ ,  $Gde$  et  $Gu$  sont fixés pour que l'étalement de chaque variable réelle corresponde à l'étendue normalisée des univers de discours. Généralement, ils sont déterminés par tâtonnement en faisant des essais de simulation et varier ces facteurs jusqu'à ce qu'on ait trouvé un phénomène de réglage convenable. [1].

### III.7 REGULATEURS PAR LOGIQUE FLOUE DE TYPES PD, PI, PID [9].

La logique floue est aussi applicable pour générer des correcteurs de type PID non linéaires. Selon la sortie du correcteur flou concerne la commande ou sa variation, nous obtenons des équivalents structuraux non linéaires des contrôleurs classiques PD, PI ou même des PID, en augmentant le nombre d'entrées des contrôleurs. Dans cette section, nous allons décrire la structure de ces correcteurs.

La Figure III.13 montre des correcteurs PD, PI, PID par logique floue ou on précise le type des entrées et des sorties :

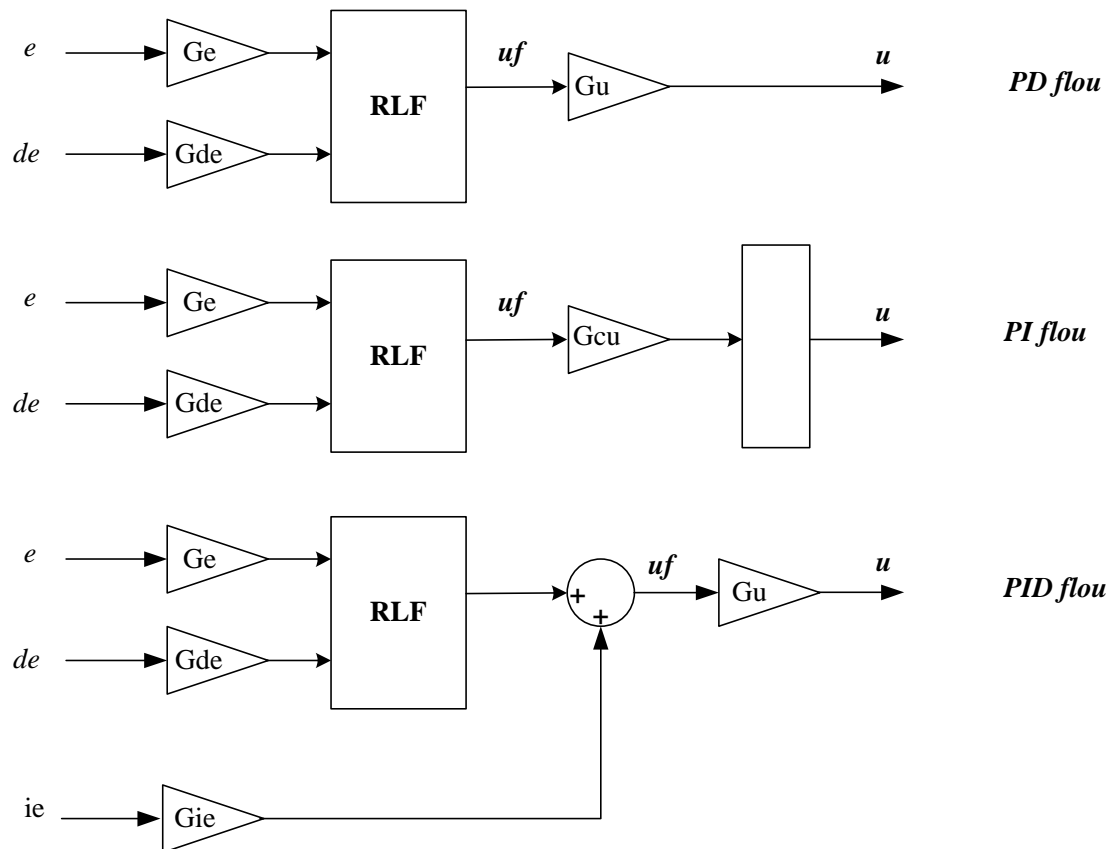


Figure III.13 schéma simplifié du correcteur flou de type PD, PI et PID

### III.7.1 Structure des régulateurs

La présente section a pour but de finaliser la description des correcteurs à base de logique floue [10].

#### III.7.1.1 Correcteur flou de type PD

La Figure III.14 montre la structure du régulateur PD flou. Les deux entrées sont l'erreur et sa dérivée qui sont normalisées au moyen des gains de normalisation  $G_e$  et  $G_{de}$  respectivement. Un gain de dénormalisation  $G_u$  est affecté à la sortie.

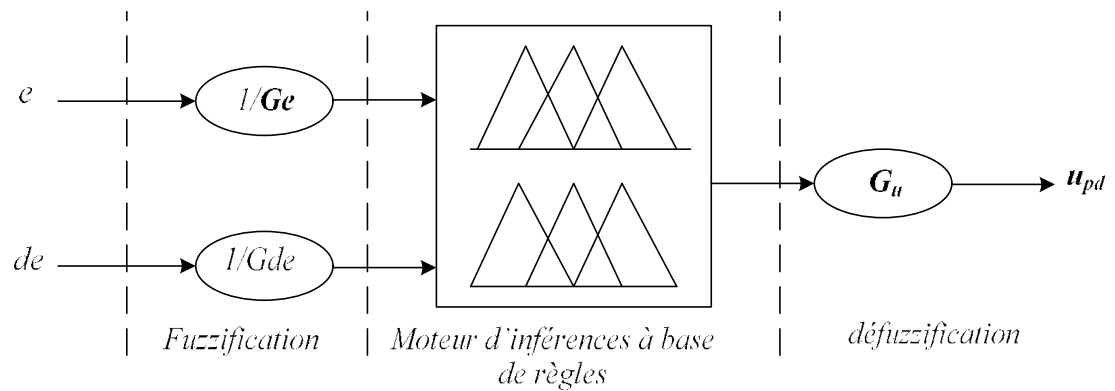


Figure III.14 structure du régulateur flou, type PD.

A partir de cette structure, deux correcteurs de type PI d'une part et de type PID d'autre part, vont pouvoir être construits. Ces correcteurs utilisent le même moteur d'inférence flou décrit précédemment.

La première étape consiste à définir la base de chacun de ces correcteurs, c'est-à-dire le moteur d'inférence flou.

La sortie du PD flou notée  $u_{pd}$  peut être définie à partir des fonctions  $k_p$  et  $k_d$  qui sont respectivement les gains de l'action proportionnelle et dérivée du moteur d'inférence flou, variant selon le point de fonctionnement, c'est-à-dire des valeurs de  $e$  et  $de$ .

$$u_{pd} = k_p(e, de) \cdot e + k_d(e, de) \cdot de \quad \text{III.2}$$

Cette équation peut être réécrite en définissant des fonctions  $K_1$  et  $K_2$  de l'erreur  $e$  et de la dérivée de l'erreur  $de$ .

$$u_{pd} = G_u \left( \frac{1}{G_e} \cdot e(t) k_1(e, de) + \frac{1}{G_{de}} \frac{de(t)}{dt} k_2(e, de) \right) \quad \text{III.3}$$

A partir de cette description du moteur d'inférence flou, il devient aisé de construire les correcteurs.

### III.7.1.2 Correcteur flou de type PI

Pour réaliser un correcteur de type PI, il suffit d'intégrer la sortie du moteur d'inférence flou comme indiqué à la Figure III.15.

En notant  $u_{pi}$  la sortie du contrôleur flou de type PI, il vient :

$$u_{pi} = \int u_{pd} dt \quad \text{III.4}$$

$$u_{pi} = \frac{G_u}{G_e} \int k_1(e, de) \cdot e(t) dt + \frac{G_u}{G_e} k_2(e, de) \cdot \frac{de(t)}{dt} dt \quad \text{III.5}$$

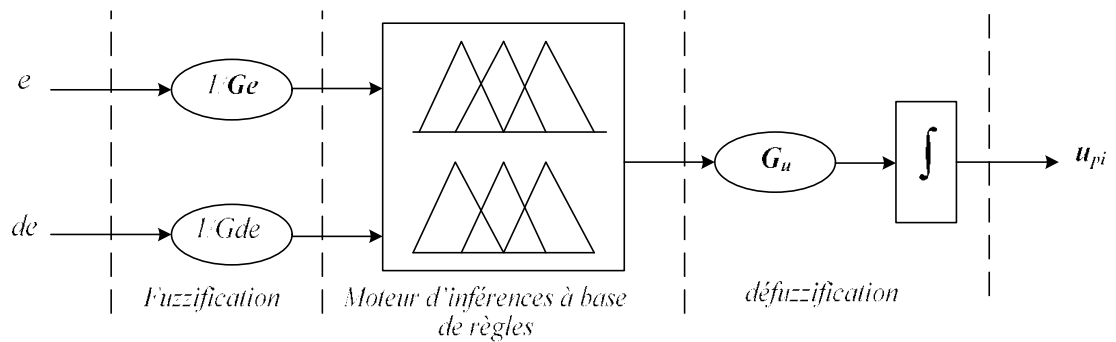


Figure III.15 structure du régulateur flou, type PI.

### III.7.1.3 Correcteur flou de type PID

Pour réaliser un correcteur de type PID flou, une partie intégrale va être ajoutée en parallèle au moteur d'inférence flou.

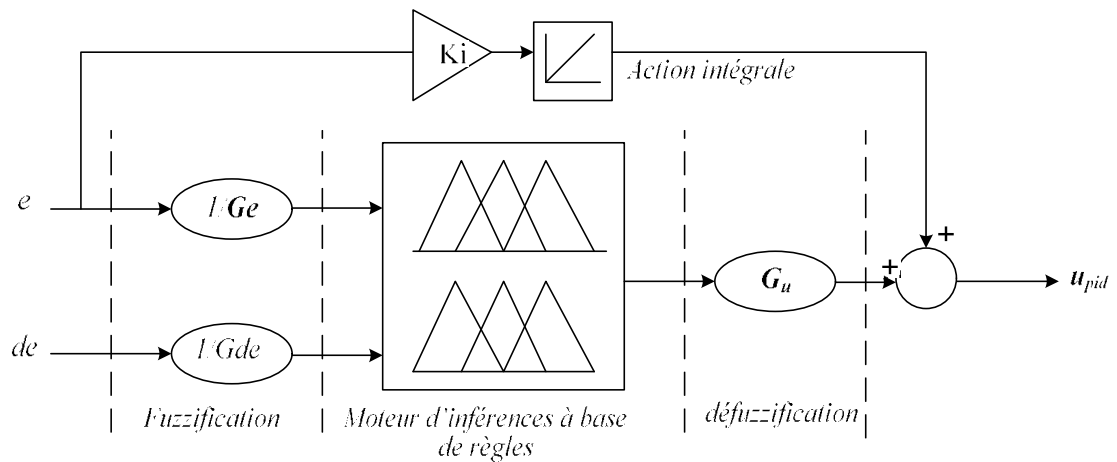


Figure III.16 Structure du régulateur flou, type PID

L'avantage de ce type de structure, basée sur le moteur flou PD est qu'il est possible de construire un correcteur PID sans avoir à calculer la dérivée seconde de l'erreur, qui risquerait d'amplifier de façon trop importante le bruit de mesure.

En notant  $u_{pid}$  la sortie du contrôleur flou du type PID, on aura :

$$u_{pid} = k_i \int e(t) dt + u_{pd} \quad \text{III. 6}$$

Soit :

$$u_{pid} = k_i \int e(t) dt + G_u \left( \frac{1}{G_e} \cdot e(t) k_1(e, de) + \frac{1}{G_{de}} \frac{de(t)}{dt} k_2(e, de) \right) \quad \text{III. 7}$$

### III.7.2 Relation de passage

Le Tableau III.4 résume les relations donnant les paramètres d'adaptation d'un correcteur flou en fonction des paramètres d'un correcteur PID.

		Correcteur PID classique		
		Kc	1/Ti	Td
Corre- cteur flou	FP	Ge. Gu	-	-
	FPI	Gde. Gcu	Ge/Gde	-
	FPD	Ge. Gu	-	Gde/ Ge
	FPID	Ge.Gu	Gie/Ge	Gde/Ge

**Tableau III.4** relation entre les gains flous et classiques.

Avec :  $G_{ie}$  le facteur d'échelle appliqué au régulateur flou type PID, et  $G_{cu}$  celui appliqué au régulateur flou type PI.

### III.7.3 Réalisation numérique et équivalence

En pratique, le correcteur flou est réalisé numériquement. Soit  $T$  la période d'échantillonnage du calculateur. En notant  $k$  l'instant  $t$  d'échantillonnage, il vient :

$$e(t) \rightarrow e(k)$$

$$\frac{de(t)}{dt} \rightarrow \frac{e(k)-e(k-1)}{T}$$

Un PID classique délivre une commande  $u(t)$  donnée par l'équation :

$$u(t) = k_p \left[ \frac{de}{dt} + \frac{1}{T_i} \int_0^T e(t). dt + T_d \frac{d^2e}{dt^2} \right] \quad \text{III 8}$$

Avec les valeurs échantillonnées à la période  $T$ , la grandeur  $du/dt$  est exprimée à l'instant  $t = n$ , sous la forme d'équation aux différences.

$$\frac{du}{dt} = \frac{u_n - u_{n-1}}{T} = K_p \left[ \frac{e_n - e_{n-1}}{T} + \frac{1}{T_i} e_n + T_d \left( \frac{e_n - 2e_{n-1} + e_{n-2}}{T^2} \right) \right] \quad \text{III 9}$$

$$u_n - u_{n-1} = K_p \left[ (e_n - e_{n-1}) + \frac{T}{T_i} e_n + T_d \left( \frac{e_n - 2e_{n-1} + e_{n-2}}{T} \right) \right] \quad \text{III.10}$$

$$u_n = u_{n-1} + K_p \left[ (e_n - e_{n-1}) + \frac{T}{T_i} e_n + T_d \left( \frac{e_n - 2e_{n-1} + e_{n-2}}{T} \right) \right] \quad \text{III.11}$$

En posant  $u_n = u_{n-1} + u$ , on déduit la variation de la commande.

$$u = K_p \left[ (e_n - e_{n-1}) + \frac{T}{T_i} e_n + T_d \left( \frac{e_n - 2e_{n-1} + e_{n-2}}{T} \right) \right] \quad \text{III.12}$$

On constate que les valeurs  $(e_n - e_{n-1})$ ,  $e_n$  et  $(e_n - 2e_{n-1} + e_{n-2})$  sont respectivement relatives aux parties proportionnelle, intégrale et dérivée de la commande. En se basant sur cette analyse, on peut définir un ensemble d'états par l'erreur, sa dérivée et sa dérivée seconde. Chaque état est une situation particulière qui est reliée à une commande par une règle linguistique conditionnelle. Par exemple :

**SI** (l'erreur  $e$  est GN) **ET** (la dérivée de l'erreur est PP) **ET** (la dérivée seconde de l'erreur est ZE) **ALORS** (le changement de la commande  $\Delta u$  est MN).

A chaque proposition entre parenthèse est associé le degré d'appartenance au sous-ensemble correspondant.

### Remarque

La logique floue a été aussi utilisée pour adapter les paramètres d'un régulateur conventionnel, de type PID par exemple, énonçant des règles de type :

" **SI** le temps de montée est trop faible (par rapport à une valeur prescrite) **ALORS** (diminuer légèrement le paramètre  $k_d$ ) ".

### III.8 PROPRIETES D'UN REGLAGE PAR LOGIQUE FLOUE

Les propriétés essentielles d'un réglage par logique floue peuvent être résumées de la manière suivant :

- Mesure de plusieurs grandeurs : la grandeur régler et d'autre grandeurs caractérisant le comportement dynamique du système à régler ;
- Comportement non linéaire sans dynamique ;
- Aptitude à régler convenablement surtout des systèmes à règles avec comportement dynamique compliqué, dont la modélisation est difficile,
- Modélisation du système à régler superflue (pouvant tout de même être utile) ;
- Connaissance approximative des paramètres du système à régler.

### III.9 AVANTAGES ET INCONVENIENTS DU REGLAGE PAR LOGIQUE FLOUE

Evidemment, le réglage par logique floue réunit un certain nombre d'avantages et d'inconvénients.

#### III.9.1 Avantages

- la non nécessité d'une modélisation (cependant, il peut être utile de disposer d'un modèle convenable) ;
- La possibilité d'implémenter des connaissances (linguistiques) par l'opérateur du processus ;
- l'obtention fréquente de meilleures présentations dynamiques (régulateur non linéaire) ;
- l'emploi possible aussi pour des processus rapides (grâce à des processus dédiés) ;
- La maîtrise du système à régler avec un comportement complexe (fortement non linéaire et difficile à modéliser).

#### III.9.2 Inconvénients

- la précision de réglage souvent peu élevée ;
- la possibilité d'apparition de cycles limites à cause du fonctionnement non linéaire ;

- le manque de directives précises pour la conception d'un réglage (choix de grandeurs à mesurer, détermination de la Fuzzification, des inférences et de la Défuzzification) ;
- la cohérence des inférences non garantie à priori (apparition possible de règles d'inférence contradictoires).

### III.10 CONCLUSION

Dans ce chapitre, nous avons présenté le principe de la commande par logique floue ainsi que la principale démarche pour la conception d'un régulateur flou et les bases nécessaires à la compréhension des méthodes à base de logique floue.

Les outils fournis par la logique floue permettent une modélisation des phénomènes pouvant en un certain sens s'approcher du raisonnement humain. La commande floue apporte donc des améliorations (performances, robustesse) par rapport aux commandes linéaires classiques. Elle est bien adaptée à la commande des systèmes pour lesquels les modèles disponibles sont souvent complexes ou approximatifs. La logique floue reste toujours le sujet de nombreuses recherches et de la communication Homme-machine.

# ***Chapitre IV***

## ***Simulation***

## IV.1 INTRODUCTION

Dans ce chapitre, nous présentons quelques résultats de simulation des régulateurs PID classique et PID flou.

Concernant le PID classique, nous donnons les courbes de la réponse indicielle en utilisant quatre méthodes de réglage :

- La méthode de Ziegler Nichols,
- La méthode de Cohen Coon,
- La méthode de réglage par la réglabilité
- La méthode d'optimisation des critères.

Nous avons utilisé le Logiciel Matlab pour la programmation.

## IV.2 Résultat de simulation d'un correcteur classique

### IV.2.1 Le correcteur Proportionnel - P

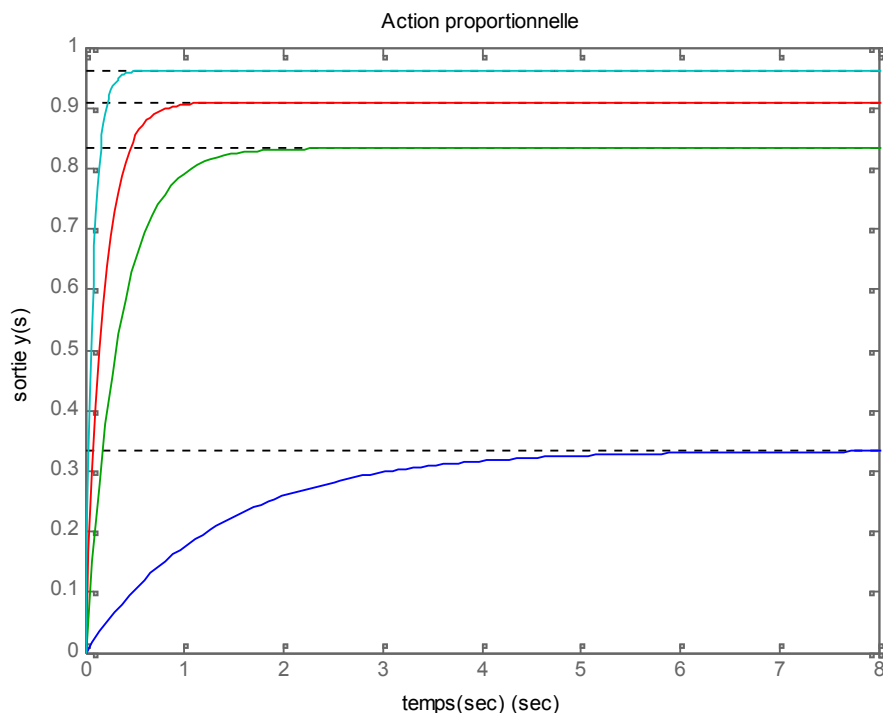


Figure IV.1 Simulation d'un correcteur type P sur un système du 1<sup>er</sup> ordre

La fonction de transfert du système est  $G(s) = \frac{0,5}{2s + 1}$

La Figure IV.1 montre que l'augmentation de Kp engendre une constante de temps plus petits ce qui implique que le système est plus rapide et l'écart statique diminue.

### IV.2.2 Le correcteur Proportionnel Intégral - PI

#### - La méthode de Ziegler Nichols

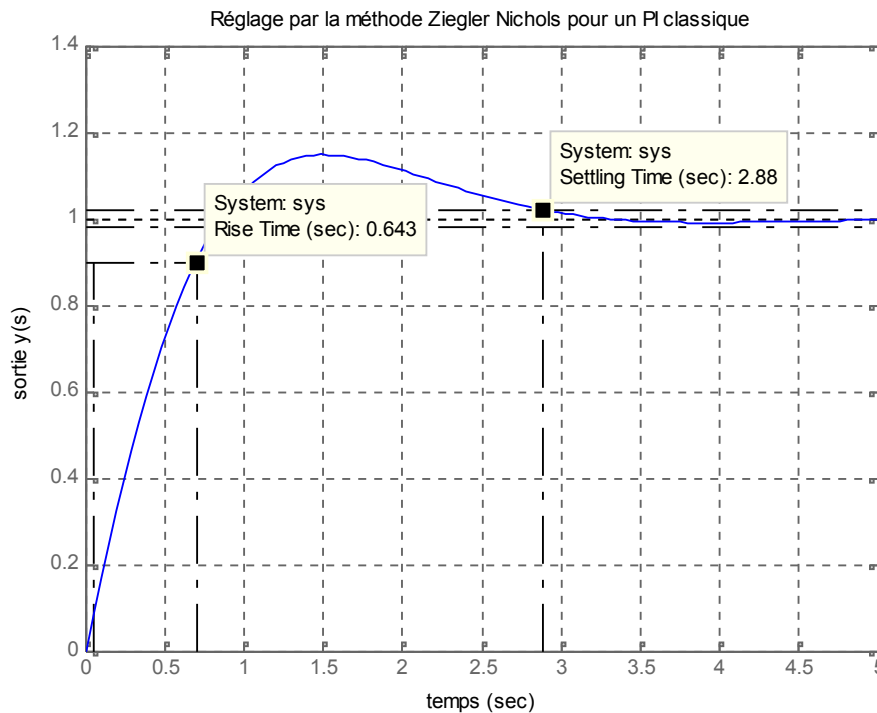


Figure IV.2 Simulation d'un correcteur PI classique d'un système du 1<sup>er</sup> ordre

La fonction de transfert du système est  $G(s) = \frac{0,5}{2s + 1} e^{-0.5s}$

Le temps de montée :  $t_m=0.643$  s , Le temps de réponse :  $t_r=2.88$  s

La Figure IV.2 montre le réglage d'un PI classique par la méthode de Ziegler Nichols. Nous constatons que le temps de réponse du système est petit.

### IV.2.3 Le correcteur PID Classique

#### - La Méthode de Ziegler Nichols

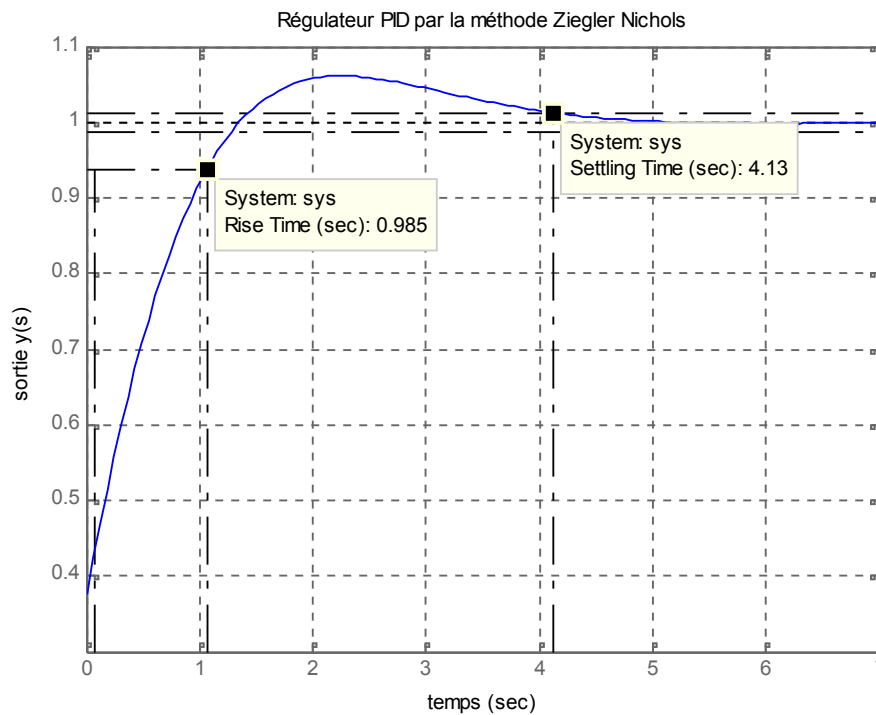


Figure IV.3 Réglage par Ziegler Nichols. La réponse indicielle.

Le temps de montée :  $t_m=0.985$  s

Le temps de réponse :  $t_r=4.13$  s

## - La Méthode de COHEN COON

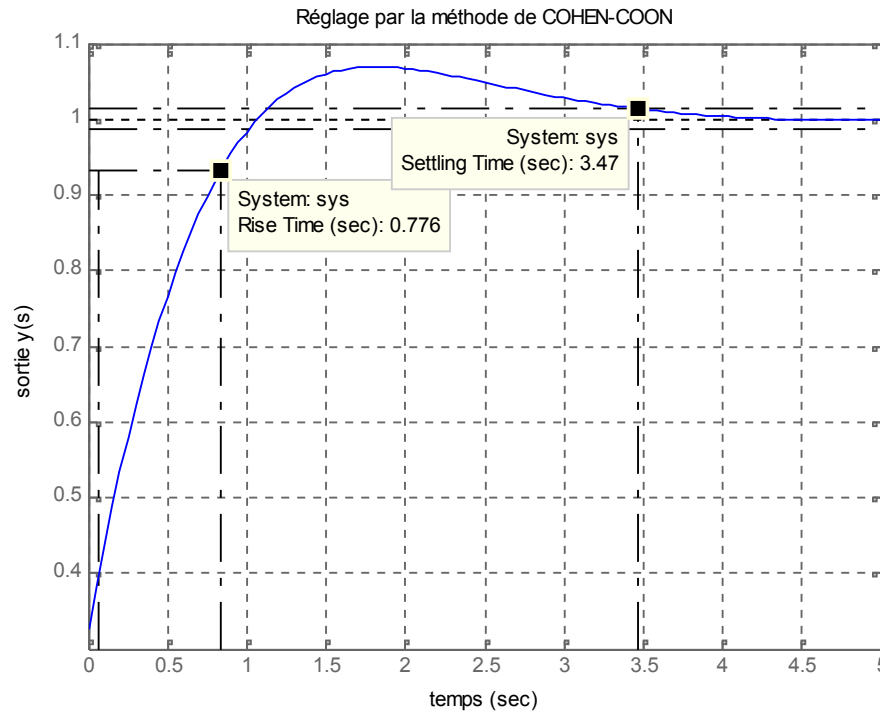


Figure IV.4 Réglage par Cohen Coon : La réponse indicielle.

Le temps de montée :  $t_m=0.776$  s

Le temps de réponse :  $t_r=3.47$ s

Les Figures IV.3 et IV.4 montrent le réglage d'un PID classique par les méthodes de Ziegler Nichols et Cohen Coon.

Nous constatons que les deux systèmes ont un temps de réponse rapide.

## - La Méthode par réglage en fonction de la Réglabilité

Le coefficient de réglabilité :

Soit un système de fonction de transfert :

$$G(s) = \frac{K}{Ts + 1} e^{-es}$$

Tel que :

K : gain, T : constante de temps (sec), e : retard.

On définit alors le coefficient de réglabilité comme suit :  $rc = e/T$

Pour K=0.5, e=0.5, T=10.

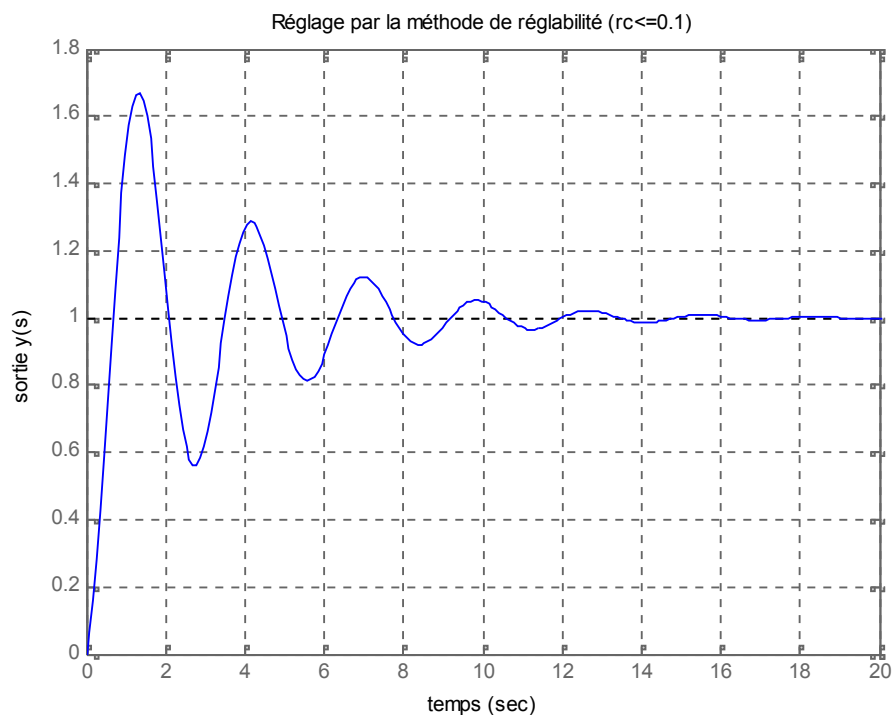


Figure IV.5 Réglage par la méthode de réglabilité ( $rc \leq 0.1$ )  
La réponse indicielle

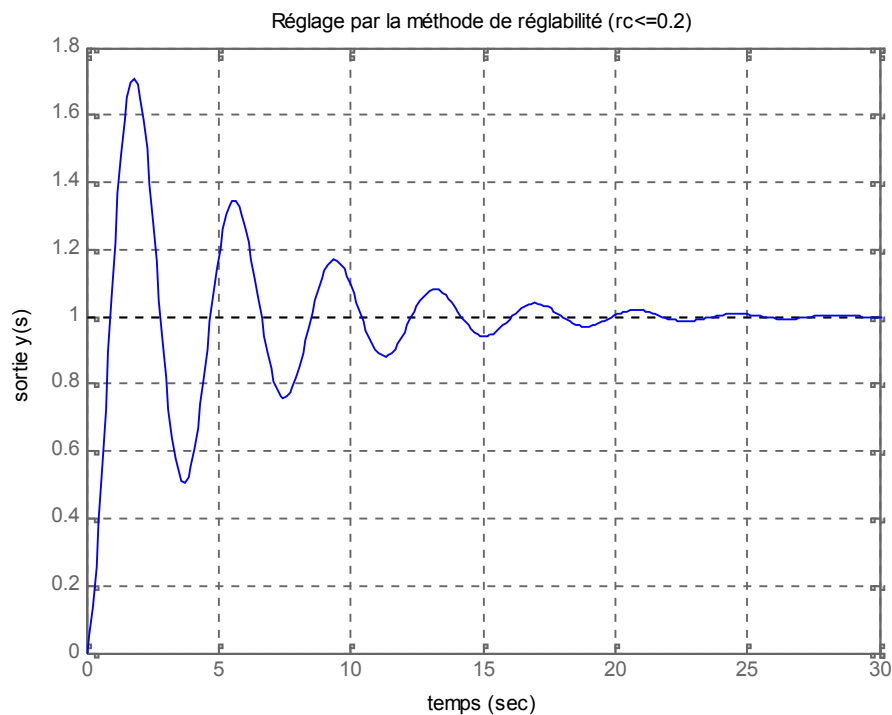
La fonction de transfert du procédé alors, sera :

$$rc = 0,05$$

Le temps de montée :  $t_m=0.49$  s.

Le temps de réponse :  $t_r=12.9$  s.

Pour  $K=0.5$ ,  $e=1.8$ ,  $T=10$ .



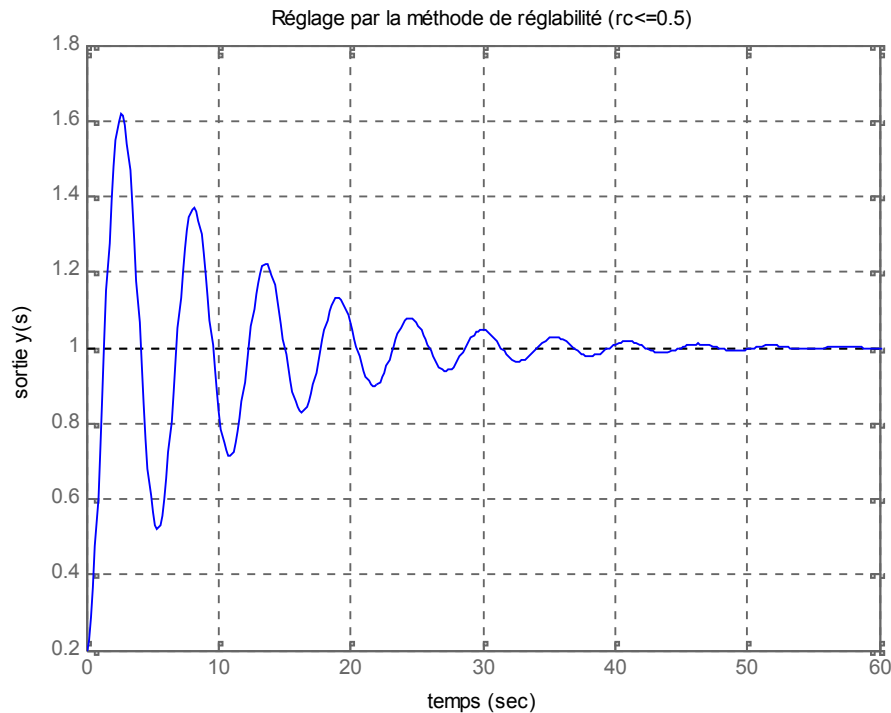
**Figure IV.6 Réglage par la réglabilité ( $rc \leq 0,2$ )  
La réponse indicielle**

La fonction de transfert du procédé alors, sera :  $G(s) = \frac{0,5}{10s + 1} e^{-1,3s}$

$$rc = 0,18$$

Le temps de montée :  $t_m=0.659$  s.    Le temps de réponse :  $t_r=19.4$ s.

Pour  $K=0.5$ ,  $e=4.5$ ,  $T=10$ .



**Figure IV.7 Réglage par la réglabilité ( $r_c \leq 0,5$ )  
La réponse indicielle**

La fonction de transfert du procédé alors, sera :  $G(s) = \frac{0,5}{10s + 1} e^{-4.5s}$

$r_c = 0,45$

Le temps de montée :  $t_m = 0.94$  s.

Le temps de réponse :  $t_r = 41.1$  s.

Les Figures IV.5, IV.6 et IV.7 représentent le réglage du système par la méthode de réglabilité pour différentes valeurs de  $r_c$ .

Nous remarquons que plus  $r_c$  augmente plus l'amplitude du dépassement augmente et le système devient instable.

## - La Méthode d'Optimisation de critère

Soit un système de fonction de transfert

$$G(s) = \frac{1}{20s+1} \cdot e^{-5s}$$

### ▪ Régulation

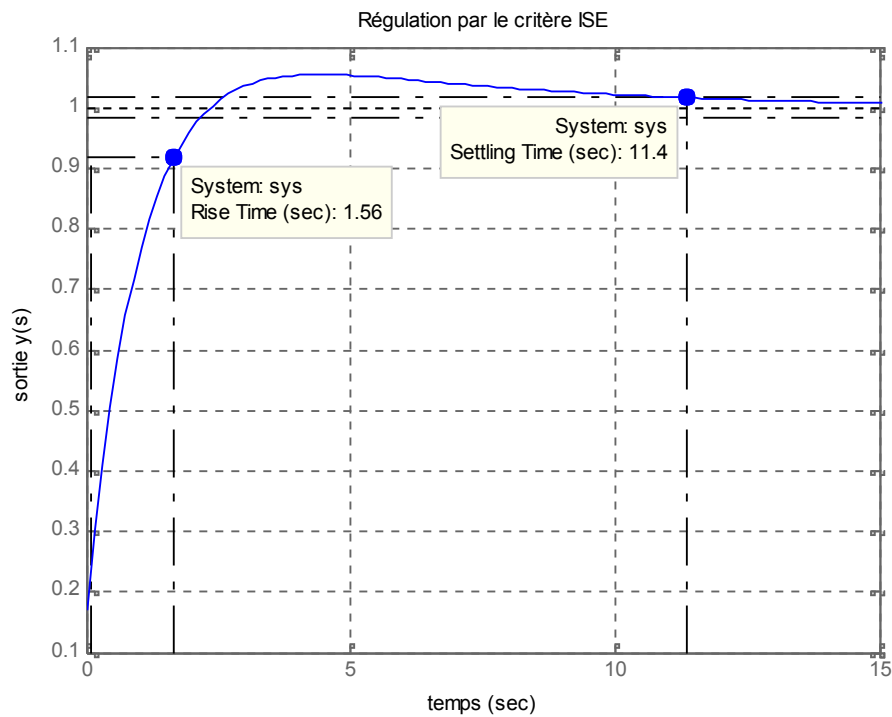
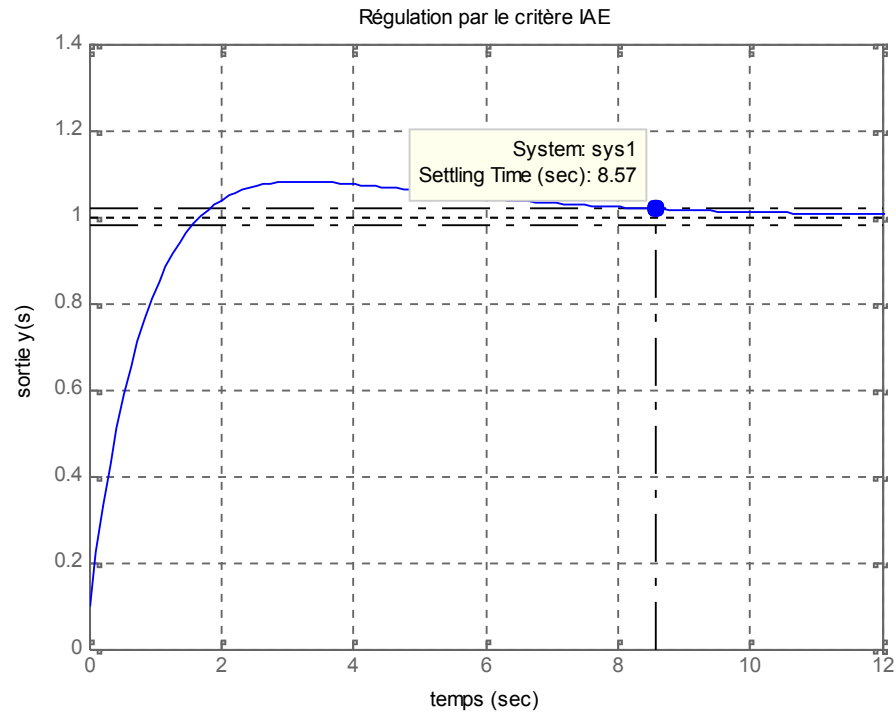


Figure IV.8 Régulation par le critère ISE. La réponse indicielle

Le temps de montée :  $t_m=1.56$  s.

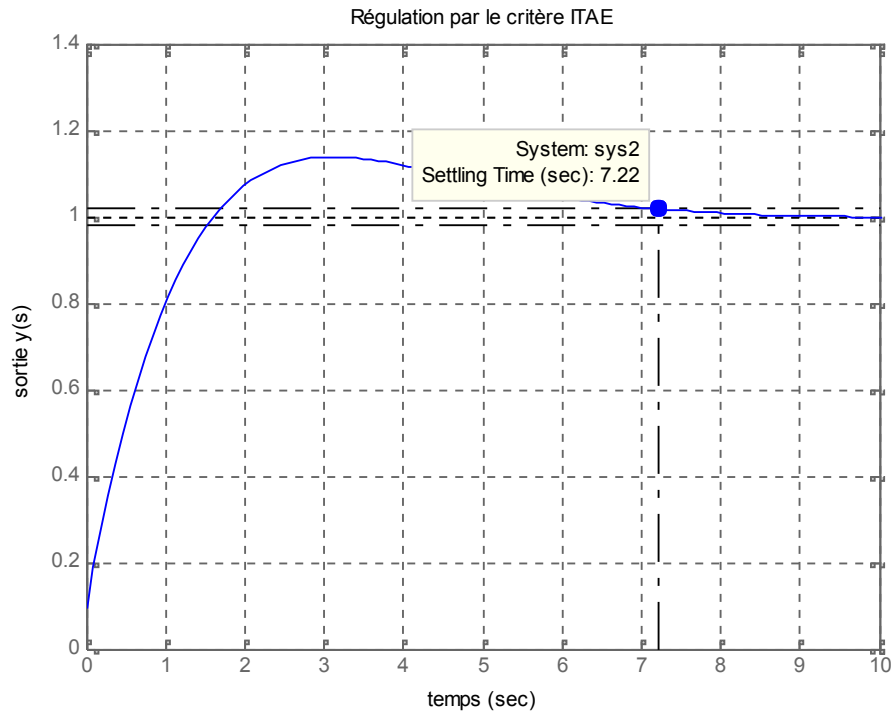
Le temps de réponse :  $t_r=11.4$  s.



**Figure IV.9 Régulation par le critère IAE : La réponse indicielle**

Le temps de montée :  $t_m=1.18$  s.

Le temps de réponse :  $t_r=8.57$ s.



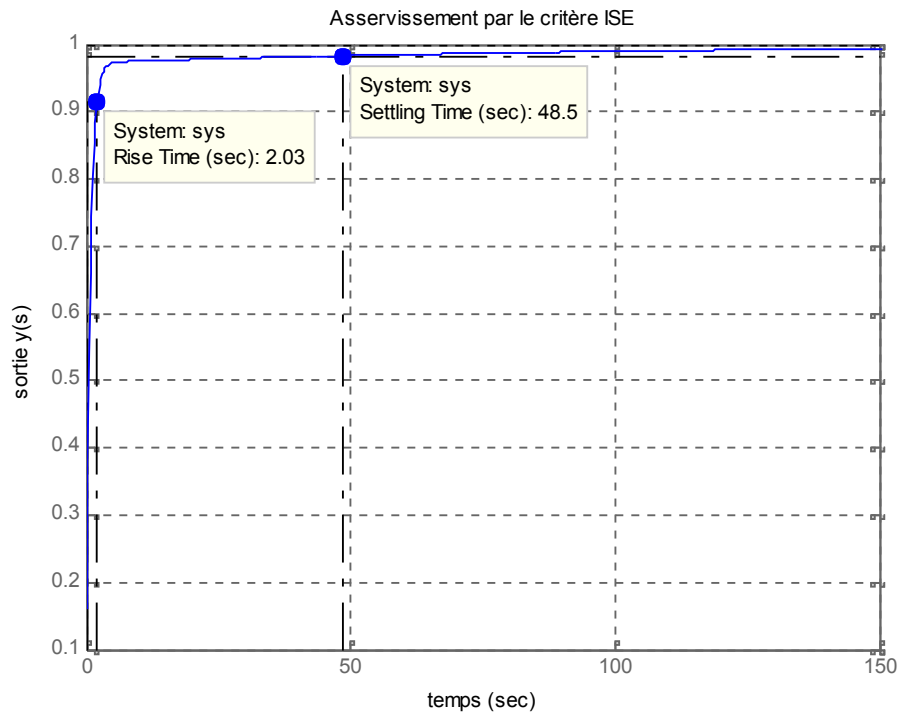
**Figure IV.10 Régulation par le critère ITAE.  
La réponse indicielle.**

Le temps de montée :  $t_m = 1.2$  s.

Le temps de réponse :  $t_r = 7.22$  s.

Les Figures IV.8, IV.9 et IV.10 présentent la méthode d'optimisation de critères par ISE, IAE et ITAE. Les réglages obtenus avec chacun des trois critères pour le mode Régulation. Le critère ITAE est celui qui donne la réponse la moins la plus rapide.

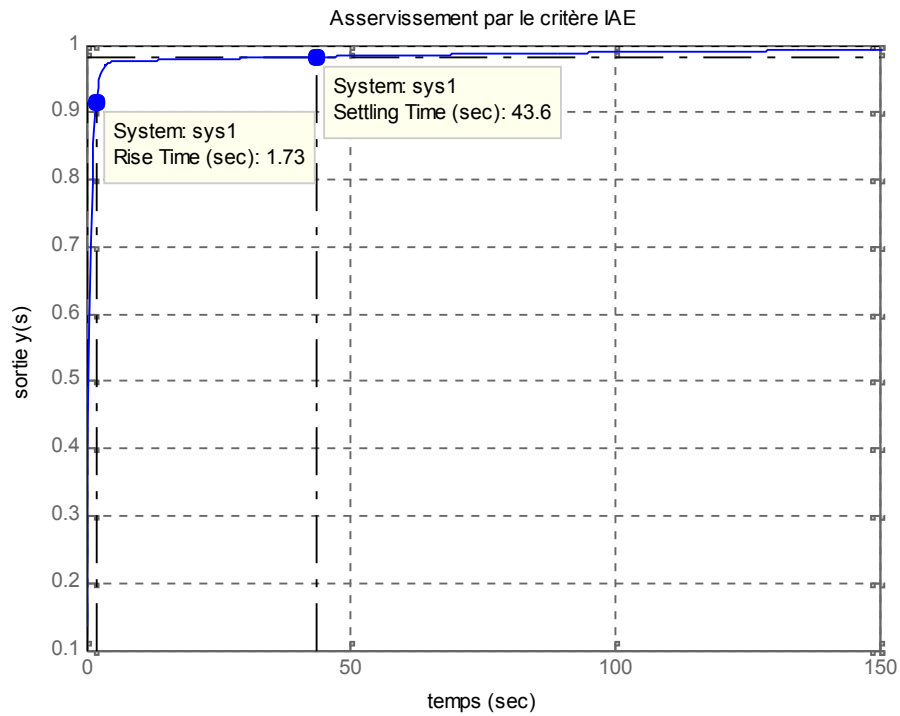
- *Asservissement*



*Figure IV.11* Asservissement par le critère ISE : réponse à un échelon unitaire.

Le temps de montée :  $t_m=2.03$  s.

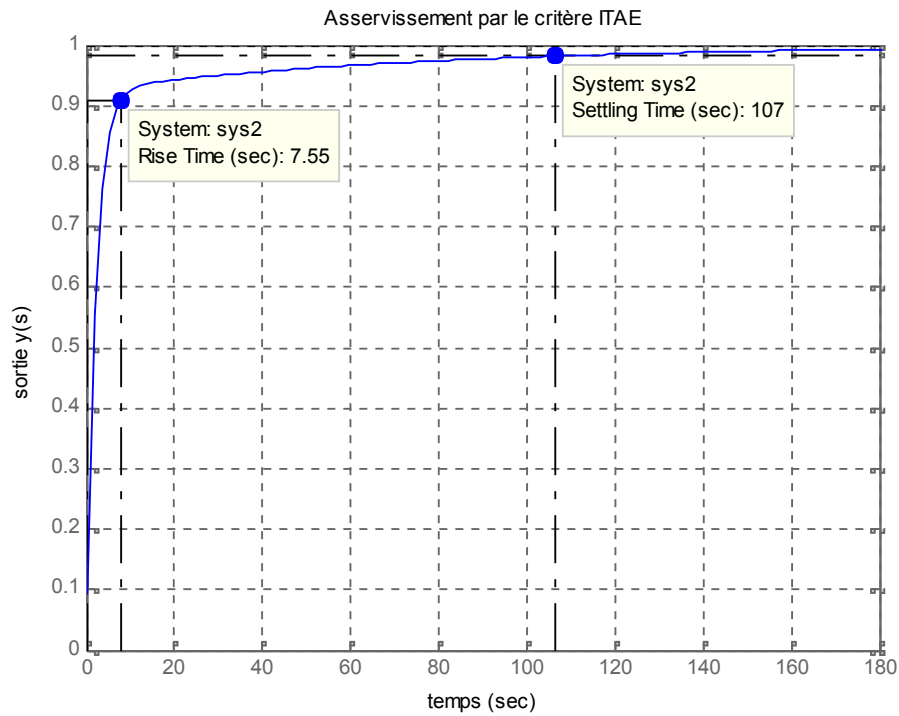
Le temps de réponse :  $t_r=48.5$  s.



**Figure IV.12 Asservissement par le critère IAE.  
La réponse indicielle.**

Le temps de montée :  $t_m=1.73$  s.

Le temps de réponse :  $t_r=43.6$  s.



**Figure IV.13 Asservissement par le critère ITAE.  
La réponse indicielle.**

Le temps de montée :  $t_m=7.55$  s.

Le temps de réponse :  $t_r=107$  s.

Les Figures IV.11, IV.12 et IV.13 représentent la méthode d'optimisation de critères par ISE, IAE et ITAE. Les réglages obtenus avec chacun des trois critères pour le mode Asservissement. Le critère IAE est celui qui donne la réponse la plus rapide.

### IV.3 Résultat de simulation d'un régulateur flou

-sortie du processus

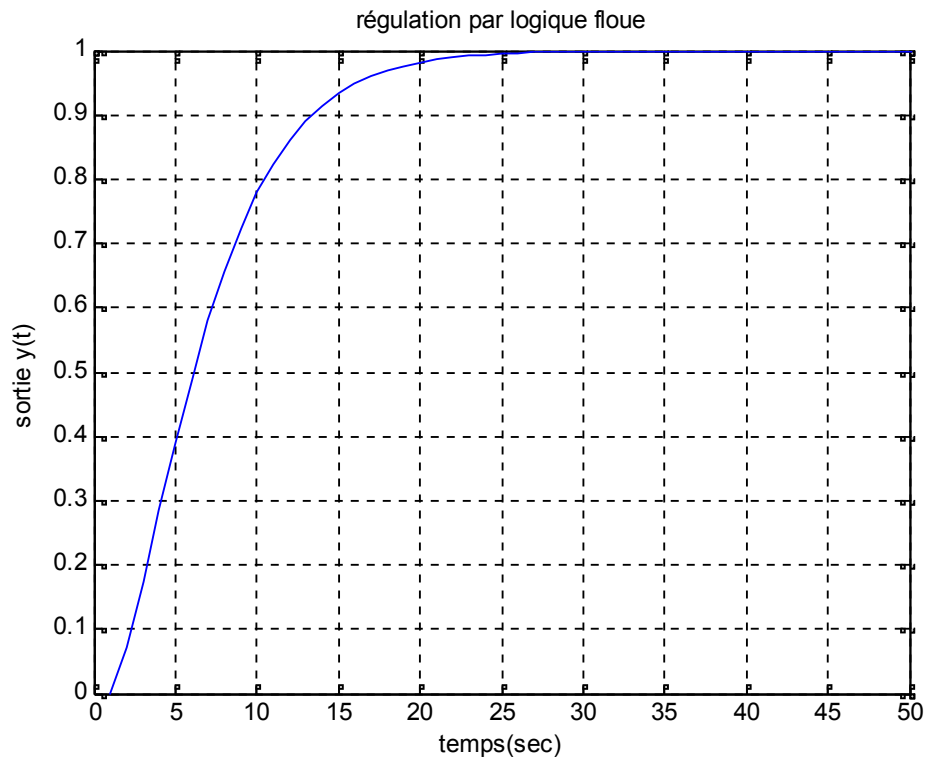


Figure IV.14 Sortie du système régulé e par la logique floue

-signal de commande appliqué au processus

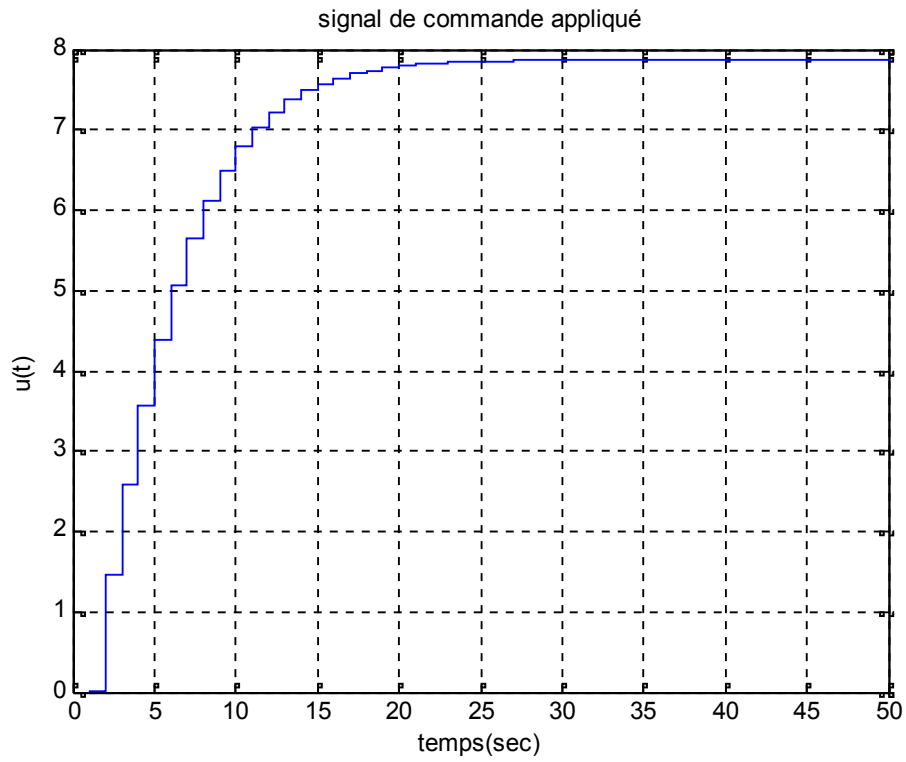


Figure IV.15 Signale de commande appliqué

-signal de sortie du régulateur flou

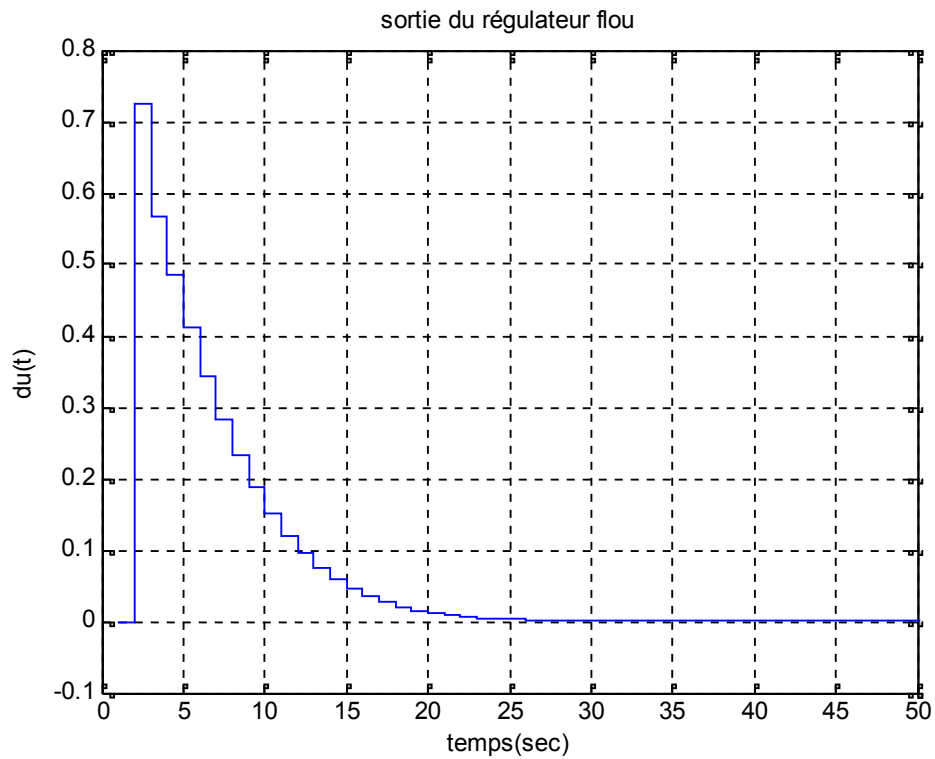


Figure IV.16 Sortie du régulateur flou

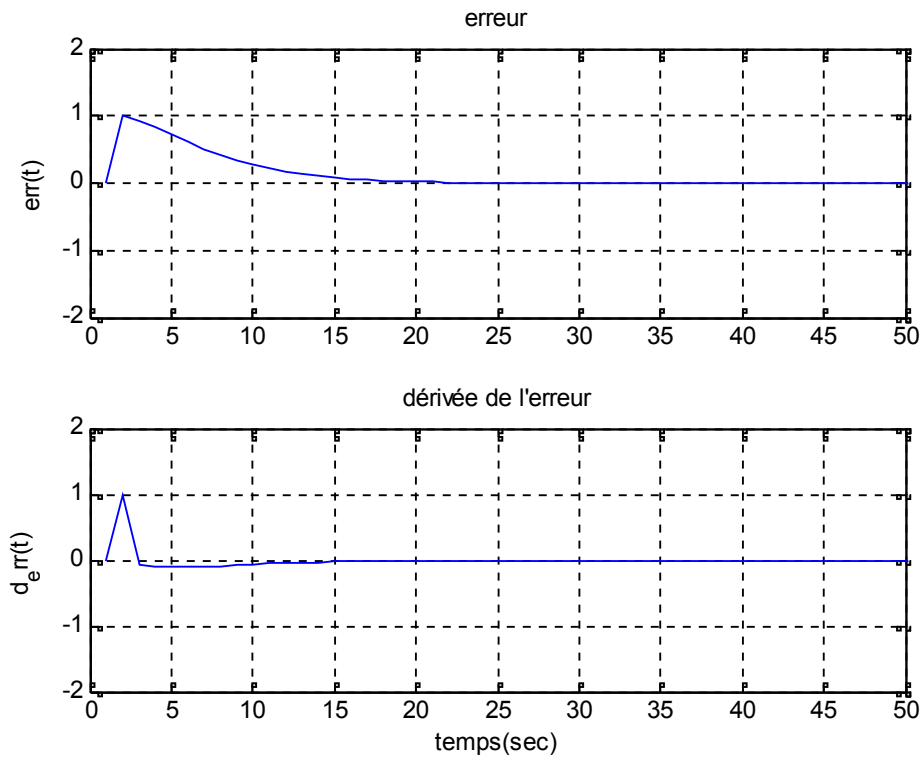
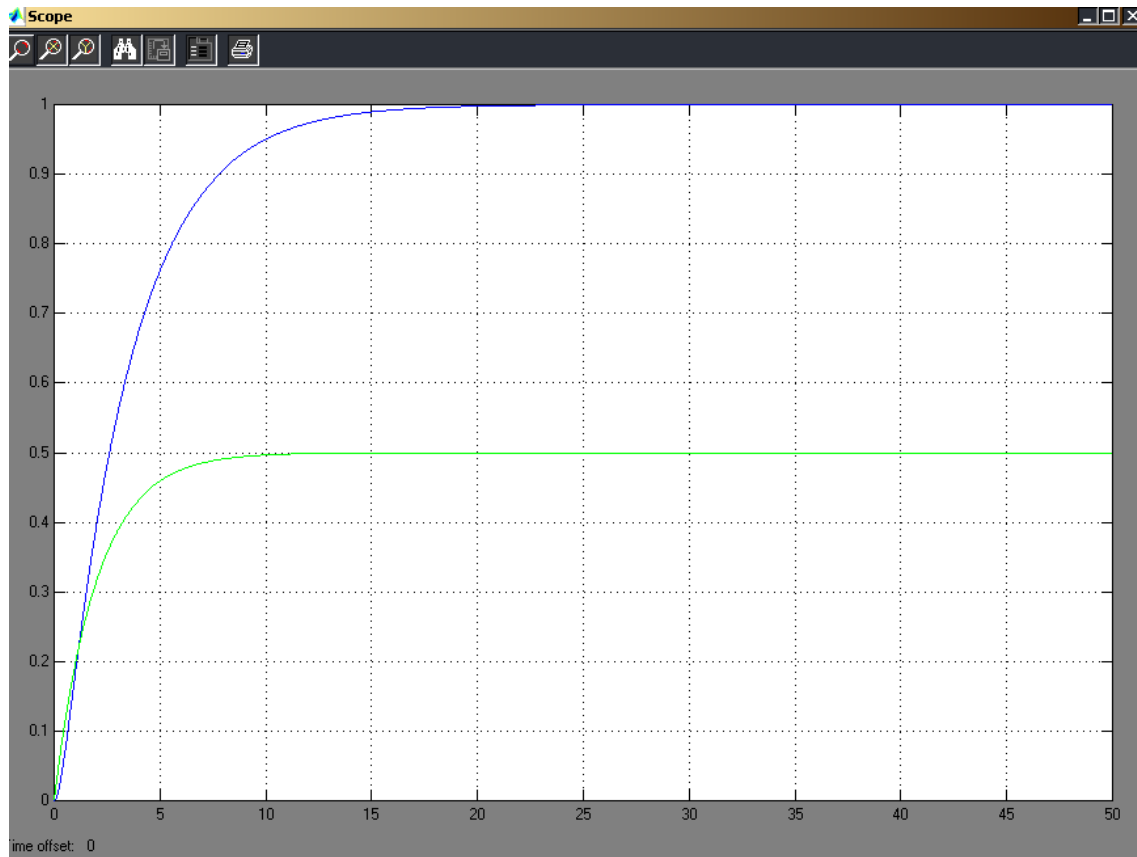
**-signal d'erreur et variation de l'erreur**

Figure IV.17 Variation de l'erreur et la dérivée de l'erreur pendant la correction

Les Figures IV.15, IV.16 et IV.17 représentent les différents signaux de commande du régulateur fluo, on remarque que le signal de sortie du système converge vers la valeur de la consigne et l'écart statique s'annule.

### V.3.1 Résultat de simulation d'un régulateur PI flou

— Réponse du système non corrigé  
— Réponse du système corrigé



**Figure IV.18 Correction d'un système de premier ordre par un régulateur PI flou**

La fonction de transfert du système est  $G(s) = \frac{0,5}{2s+1}$

Le temps de réponse  $t_r = 10s$ .

Les facteurs associés aux entrées :

$$G_e = 0,5$$

$$G_{de} = 1,5$$

$$G_{ut} = 10$$

## IV.3.2 Résultat de simulation d'un régulateur PD flou

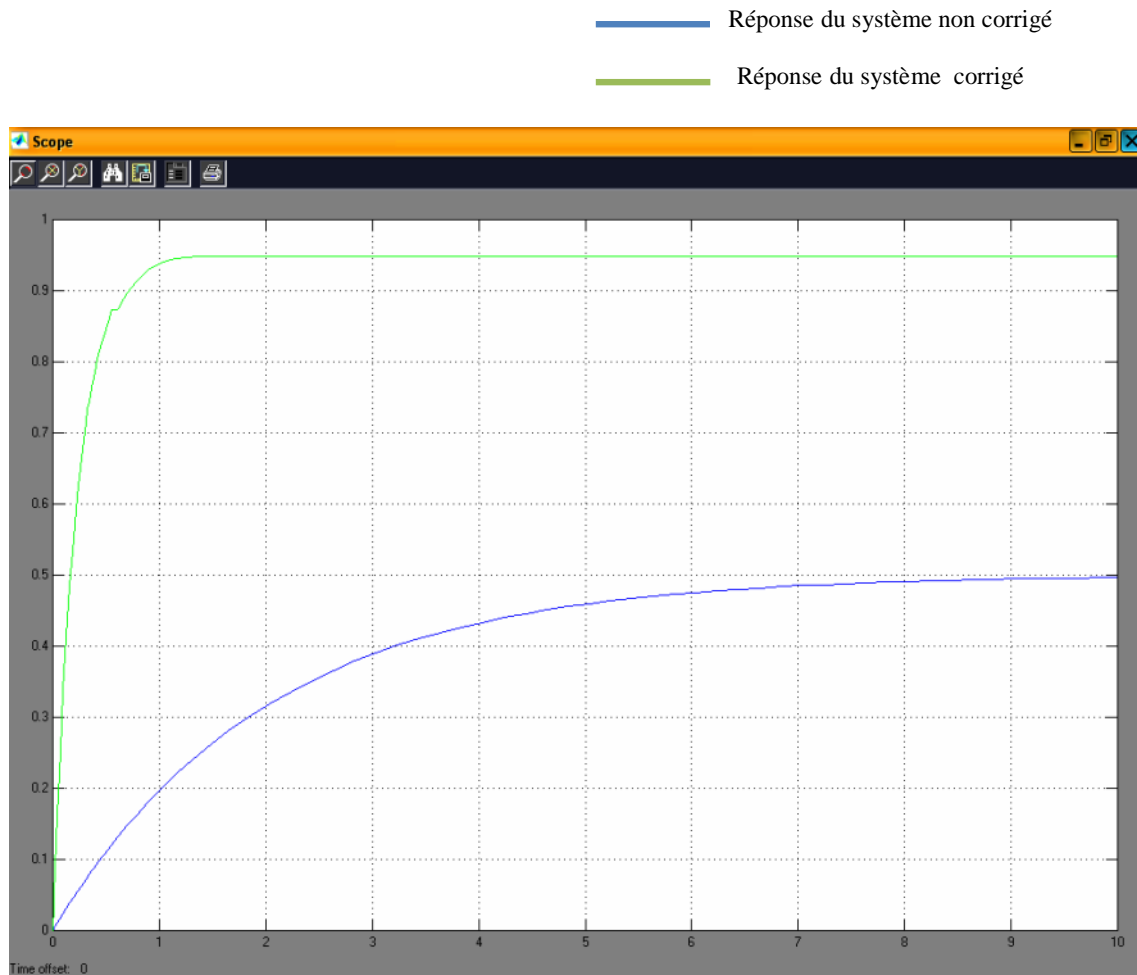


Figure IV.19 Correction d'un système de premier ordre par un régulateur PD flou

La fonction de transfert du système est  $G(s) = \frac{0,5}{2s+1}$

Le temps de réponse  $t_r = 1s$ .

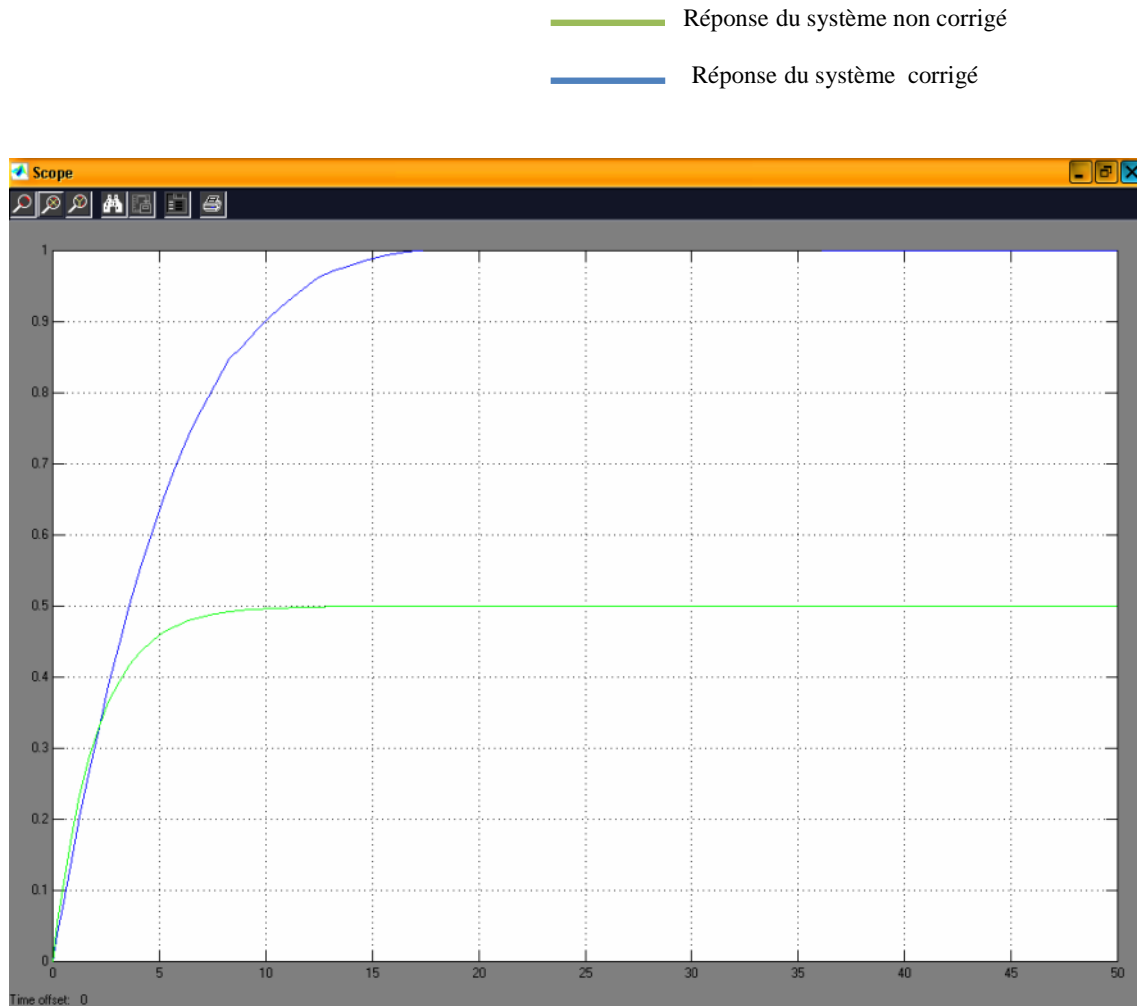
Les facteurs associés aux entrées :

$$G_e = 0,9$$

$$G_{d_e} = 0,1$$

$$G_u = 56$$

## IV.3.3 Résultat de simulation d'un régulateur PID flou



*Figure IV.20* Correction d'un système de premier ordre par un régulateur PID flou

La fonction de transfert du système est  $G(s) = \frac{0,5}{2s+1}$

Le temps de réponse  $t_r = 12s$ .

Les facteurs associés aux entrées :

$$G_e = 0,5 \quad G_{\alpha e} = 1,5 \quad G_u = 4,1 \quad G_{i_e} = 0,11$$

#### IV.3.4 Exemple numérique (PI flou)

Les conditions initiales :  $u(1) = 0.01$ ,  $err(1) = 0$ ,  $err(2)=0$ ,  $y(1) = 0$

Les équations :

$$err(k)=y_d(k)-y(k-1);$$

$$d\_err(k)=err(k)-err(k-1)$$

$$u(k)=u(k-1)+G_u * d\_err(k)$$

$$y(k)=0.6065*y(k-1)+0.05*u(k)$$

**1<sup>ère</sup> itération** : pour  $k=2$

$$err(2)=1.$$

$$d\_err(2)=1.$$

$$u(2)=1.46.$$

$$y(2)=0.073002.$$

**2<sup>ème</sup> itération** : pour  $k=3$

$$err(3)=0.927.$$

$$d\_err(3)=-0.73002.$$

$$u(3)=2.5945.$$

$$y(3)=0.174.$$

**3<sup>ème</sup> itération** : pour  $k=4$

$$err(4)=0.826.$$

$$d\_err(4)=-0.101.$$

$$u(4)=3.5661.$$

$$y(4)=0.28383.$$

4<sup>ème</sup> itération : pour  $k=5$

$$\text{err}(5)=0.71617.$$

$$d\_err(5)=-0.10983.$$

$$u(5)=4.3874.$$

$$y(5)=0.39151.$$

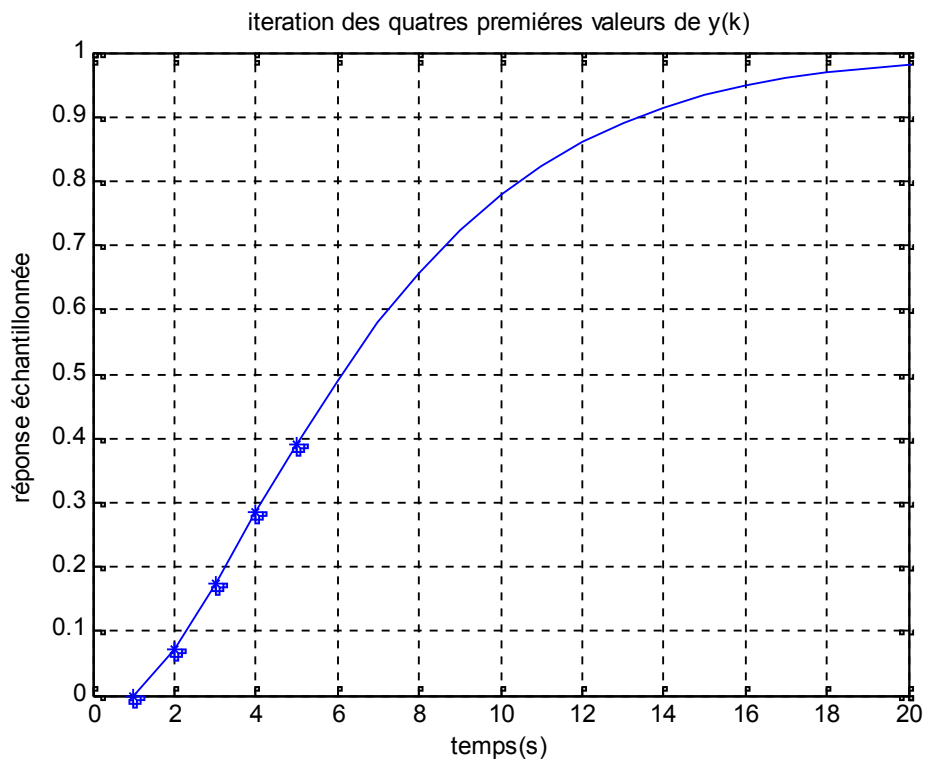


Figure IV.21 Itération des quatre premières valeurs du signal de sortie  $y(k)$

## IV.4 CONCLUSION

D'après les résultats obtenus précédemment, nous constatons que le PID est efficace. Il permet de corriger la stabilité, la rapidité et la précision d'un système mais pas les trois performances à la fois. Le système à asservir doit toujours obéir à un cahier de charge exigé. De plus, on ne peut pas améliorer deux performances simultanément.

Le PID flou donne de bon résultat par rapport au PID classique. Son principal avantage est qu'il ne nécessite pas de modèles, il garantit de bonnes performances en terme de stabilité et rapidité.

***Conclusion***

***Générale***

## Conclusion générale

Nous avons présenté dans ce mémoire les méthodes de réglage et de la commande des systèmes industriels par régulateurs PID classiques et flous. Nous avons également détaillé la procédure de synthèse des correcteurs les plus communément utilisés en industrie.

L'objectif de notre travail est de présenter une étude comparative entre deux approches de réglage PID (classique et flou) et de simuler leur fonctionnement.

Les méthodes de réglage conventionnelles se basent sur une modélisation adéquate de système à régler et un traitement analytique à l'aide de fonction de transfert ou de représentation d'état. Cela nécessite souvent des notions de mathématique assez avancées. Par contre dans le réglage par logique floue, on exige beaucoup plus de connaissances, d'expériences, et d'applications acquises par les opérateurs.

La combinaison entre les deux approches, conventionnelle et réglage par logique floue, permet de tirer profit, d'une part des performances offertes par les algorithmes conventionnels, et d'autre part de la souplesse de la logique floue qui se traduit par sa progressivité dans la prise de décision.

Ainsi, les résultats de simulations obtenus nous ont permis d'affirmer que l'application de la logique floue pour la commande offre aussi des avantages multiples, mais la régulation traditionnelle tels que les correcteurs PID gardent toujours leurs places et leurs terrains d'applications.



# ***Annexe***

## 1. Programme de simulation montrant l'action proportionnelle

```
%Action proportionnelle  
  
clc,clear all close all  
  
kp1 = 1  
kp2 = 10  
kp3 = 20  
kp4 = 50  
  
G1 = tf(0.5 * kp1, [2 1])  
G2 = tf(0.5 * kp2, [2 1])  
G3 = tf(0.5 * kp3, [2 1])  
G4 = tf(0.5 * kp4, [2 1])  
  
sys1 = feedback(G1,1,-1)  
sys2 = feedback(G2,1,-1)  
sys3 = feedback(G3,1,-1)  
sys4 = feedback(G4,1,-1)  
  
step(sys1)  
  
hold on  
step(sys2)  
  
hold on  
step(sys3)  
  
hold on  
step(sys4)  
  
hold on  
title('Action proportionnelle')  
  
xlabel('temps (sec)')  
  
ylabel('sortie y(t)')
```

## 2. Programme de simulation d'un PI classique

Utilisation de la méthode de Ziegler Nichols

```
%Action proportionnelle integrale
clc,clear all close all
K = 0,5
e = 0,5
T = 2
G = tf(K,[T 1])
R = K/I'
Kc =  $\frac{0,9}{e * R}$ 
Ki = 3,33
num = [Kc Ki]
den = [1 0]
C = tf(num,den)
f = C * G
sys = feedback(f,1,-1)
step(sys)
title('Reglage par la méthode de Ziegler Nichols pour un PI classique')
xlabel('temps (sec)')
ylabel('sortie y(t)')
```

### 3. Programme de simulation des différentes méthodes de réglage des paramètres d'un PID classique

#### 3.1 Méthode de Ziegler Nichols

```
clc,clear all alose all
k= 0.5
e= 0.5
T= 2
G= tf(k,[T 1])
R= k/T
kc= 1.2/(e * R)
ki= 2 * e
kd= 0.5 * e
num= [kd kc ki]
den= [1 0]
C= tf(num,den)
f= C * G
sys= feedback(f,1,-1)
step(sys)
title('Réglage par la méthode Ziegler Nichols')
xlabel('temps (sec)')
ylabel('sortie y(t)')
```

#### 3.2 Méthode de COHEN-COON

```
clc,clear all close all
k= 0.5
e= 0.5
T= 2
G= tf(k,[T 1])
kc= (T/(k * e)) * (((16 * T) + (3 * e))/(12 * T))
```

## Annexe

```
ki = e * (32 + (6 * (e/T))) / (13 + (8 * (e/T)))
kd = (4 * e) / (11 + (2 * (e/T)))
num = [kd kc ki]
den = [1 0]
C = tf(num, den)
f = C * G
sys = feedback(f, 1, -1)
step(sys)
title('Réglage par la méthode de COHEN – COON')
xlabel('temps (sec)')
ylabel('sortie y(t)')
```

### 3.3 Méthode par Réglage en fonction de la Réglabilité

```
clc, clear all close all
k = input('donner k = ')
e = input('donner e = ')
T = input('donner T = ')
G = tf(k, [T 1])
rc = e/T
if rc <= 0.1
    Kc = 5/k
    ki = T
    C = tf([Kc ki], [1 0])
    f = C * G
    sys = feedback(f, 1, -1)
    step(sys)
    title('Réglage par la méthode de réglabilité (rc <= 0.1)')
xlabel('temps (sec)')
ylabel('sortie y(t)')
```

```

elseif rc < - 0.2
    Kc = 0,5/(k * rc)
    ki = T
    C = tf([Kc ki], [1 0])
    f = C * G
    sys = feedback(f,1,-1)
    step(sys)
    title('Réglage par la méthode de réglabilité (rc <= 0.2)')
    xlabel('temps (sec)')
    ylabel('sortie y(t)')

elseif rc <= 0.5
    Kc = (0.5 * (1 + 0.5 * rc))/(k * rc)
    ki = T * (1 + 0.5 * rc)
    kd = T * 0.5 * rc / (0.5 * rc + 1)
    C = tf([kd Kc ki], [1 0])
    f = C * G
    sys = feedback(f,1,-1)
    step(sys)
    title('Réglage par la méthode de réglabilité (rc <= 0.5)')
    xlabel('temps (sec)')
    ylabel('sortie y(t)')
else
    ('Impossible à régler avec un PID')
end

```

### 3.4 Méthode d'Optimisation de critère

- Par Régulation

```
clc, clear all close al
```

```

k = 1
e = 5
T = 20
G = tf(k, [T 1])

```

### %calcul des paramètres du PID par le critère ISE

```

a = 1.3466;
b = -0.9308;
c = 1.6585;
d = -1.25738;
e = 0.79715;
f = 0.41941;
Kc = (a/k) * (e/T)^b
Ti = (Kc * T)/(c * (e/T)^d)
Td = (T * e/Kc) * ((e/T)^f)
C = tf([Td Kc 1/Ti], [1 0])
f = C * G
sys = feedback(f, 1, -1)
figure(1)
step(sys)
title('Régulation par le critère ISE')
xlabel('temps (sec)')
ylabel('sortie y(t)')

```

### %calcul des paramètres du PID par le critère IAE

```

a = 1.31509;
b = -0.8826;
c = 1.2507;
d = -1.3756;
e = 0.5655;
f = 0.4576;
Kc = (a/k) * (e/T)^b
Ti = (Kc * T)/(c * (e/T)^d)

```

## Annexe

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```
Td = (T * e/Kc) * ((e/T)^f)
C = tf([Td Kc 1/Ti], [1 0])
f = C * G
sys1 = feedback(f, 1, -1)
figure(2)
step(sys1)
title('Régulation par le critère IAE')
xlabel('temps (sec)')
ylabel('sortie y(t)')
```

### %calcul des paramètres du PID par le critère ITAE

```
a = 1.3176;
b = -0.7937;
c = 1.12499;
d = -1.42603;
e = 0.49547;
f = 0.41932;
Kc = (a/k) * (e/T)^b
Ti = (Kc * T)/(c * (e/T)^d)
Td = (T * e/Kc) * ((e/T)^f)
C = tf([Td Kc 1/Ti], [1 0])
f = C * G
sys2 = feedback(f, 1, -1)
figure(3)
step(sys2)
title('Régulation par le critère ITAE')
xlabel('temps (sec)')
ylabel('sortie y(t)')
```

- Par Asservissement

clc, clear all close all

## Annexe

$k = 1$

$e = 5$

$T = 20$

$G = \text{tf}(k, [T \ 1])$

**%calcul des paramètres du PID par le critère ISE**

```
a = 1.26239;  
b = -0.83880;  
c = 6.03560;  
d = -6.0191;  
e = 0.47617;  
f = 0.24572;  
Kc = (a/k) * (e/T)^b  
Ti = (Kc * T)/(c + d * (e/T))  
Td = (T * e/Kc) * ((e/T)^f)  
C = tf([Td Kc 1/Ti],[1 0])  
f = C * G  
sys = feedback(f,1,-1)  
figure(1)  
step(sys)  
title('Asservissement par le critère ISE')  
xlabel('temps (sec)')  
ylabel('sortie y(t)')
```

**%calcul des paramètres du PID par le critère IAI**

```
a = 1.13031;  
b = -0.81314;  
c = 5.7527;  
d = -5.7241;  
e = 0.32175;  
f = 0.17707;  
Kc = (a/k) * (e/T)^b
```

## Annexe

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```
Ti = (Kc * T)/(c + d * (e/T))
Td = (T * e/Kc) * ((e/T)^f)
C = tf([Td Kc 1/Ti],[1 0])
f = C * G
sys1 = feedback(f,1,-1)
figure(2)
step(sys)
title('Asservissement par le critère IAE')
xlabel('temps (sec)')
ylabel('sortie y(t)')
```

### %calcul des paramètres du PID par le critère ITAE

```
a = 0.98384;
b = -0.49851;
c = 2.71348;
d = -2.29778;
e = 0.21443;
f = 0.16768;
Kc = (a/k) * (e/T)^b
Ti = (Kc * T)/(c + d * (e/T))
Td = (T * e/Kc) * ((e/T)^f)
C = tf([Td Kc 1/Ti],[1 0])
f = C * G
sys2 = feedback(f,1,-1)
figure(3)
step(sys)
title('Asservissement par le critère l'IAE')
xlabel('temps (sec)')
ylabel('sortie y(t)')
```

## 4 Programme de simulation d'un RLF

### 4.1 Programme de création d'un régulateur PI flou

```

%création d'un système flou dans un fichier M
sys_flou_1 = newfis('regul_flou 1');

%définition des variables d'entrée: l'erreur et sa variation
interv_err = [-10 10]
interv_derr = [-10 10]
sys_flou_1 = addvar(sys_flou_1, 'input', 'err', interv_err)

%définition des ensembles flou de l'entrée 1
base1 = [-20 -10 0]; %base de la première fonction triangulaire
sys_flou_1 = addmf(sys_flou_1, 'input', 1, 'Négative', 'trimf', base1);
base = base1 + 10;
sys_flou_1 = addmf(sys_flou_1, 'input', 1, 'Nulle', 'trimf', base);
base = base + 10;
sys_flou_1 = addmf(sys_flou_1, 'input', 1, 'Positive', 'trimf', base);

%ajout de la 2ème variable d'entrée: dérivée de l'erreur
sys_flou_1 = addvar(sys_flou_1, 'input', 'd_err', interv_derr)

%définition des ensembles flou de l'entrée 2
sys_flou_1 = addmf(sys_flou_1, 'input', 2, 'Négative', 'trimf', base1);
base = base1 + 10;
sys_flou_1 = addmf(sys_flou_1, 'input', 2, 'Nulle', 'trimf', base);
base = base + 10;
sys_flou_1 = addmf(sys_flou_1, 'input', 2, 'Positive', 'trimf', base);

```

## Annexe

```
%définition de la variable de sortie du régulateur
interv_cde = [-10 10];
sys_flou_1 = addvar(sys_flou_1,'output','cde',interv_cde);

%définition des 5 fonctions d'appartenances triangulaires de la sortie
base2 = [-15 -10 -5]; %base de la première fonction triangulaire
sys_flou_1 = addmf(sys_flou_1,'output',1,'GN','trimf',base2);
base = base2 + 5;
sys_flou_1 = addmf(sys_flou_1,'output',1,'N','trimf',base);
base = base + 5;
sys_flou_1 = addmf(sys_flou_1,'output',1,'Z','trimf',base);
base = base + 5;
sys_flou_1 = addmf(sys_flou_1,'output',1,'P','trimf',base);
base = base + 5;
sys_flou_1 = addmf(sys_flou_1,'output',1,'GP','trimf',base);

%édition de la matrice des règles floues:
regles = [1 1 1 1 1
          1 2 2 1 1
          1 3 3 1 1
          2 1 2 1 1
          2 2 3 1 1
          2 3 4 1 1
          3 1 3 1 1
          3 2 4 1 1
          3 3 5 1 1];
sys_flou_1 = addrule(sys_flou_1,regles)

%sauvegarder le système flou sur disque sous le nom 'regul_flou 1. fis'
writetis(sys_flou_1,'regul_flou 1');
```

## 4.2 Programme de régulation par un PI flou

%utilisation du régulateur flou dans une loi de commande

**close all**

%programme de définition du régulateur

triang1

sys\_fou\_1

%signal de consigne

yd = 1 \* ones(1,200);

%initialisation de la commande

u(1) = 0.01;

err = zeros(1,2);

y(1) = 0;

%gain

Gu = 2;

%programmation du régulateur

**for** k = 2:200

    %erreur et dérivée de l'erreur

    err(k) = yd(k) - y(k - 1);

    d\_err(k) = err(k) - err(k - 1);

    %commande à appliquer

    x = [err(k) d\_err(k)];

## Annexe

```
du(k) = evalfis(x,sys_flou_1);
u(k) = u(k-1) + Gu * du(k);

%lecture de la sortie du processus
y(k) = 0.6065 * y(k-1) + 0.05 * u(k);

end

%Affichage des résultats

figure(1)
%sortie du processus
plot(y)
grid
title('régulation par logique floue')
xlabel('temps (sec)')
ylabel('sortie y(t)')

figure(2)
%signal de commande appliqué au processus
stairs(u)
xlabel('temps (sec)')
ylabel(' u(t)')
grid
title('signal de commande appliqué')

figure(3)
%signal de sortie du régulateur flou
stairs(du);
title(' sortie du régulateur flou')
xlabel('temps (sec)')
ylabel(' du(t)')
```

**grid**

figure(4)

**subplot(211)**

%signal d'erreur

**plot(err),axis([0 length(err) - 2 2])**

**title('erreur')**

**xlabel('temps (sec)')**

**ylabel('err(t)')**

**grid**

**subplot(212)**

%variation de l'erreur

**plot(d\_err),axis([0 length(d\_err) - 2 2])**

**title('dérivée de l'erreur')**

**xlabel('temps (sec)')**

**ylabel('d\_err(t)')**

**grid**

## 5 Schémas de simulation des différents régulateurs PID sous simulink

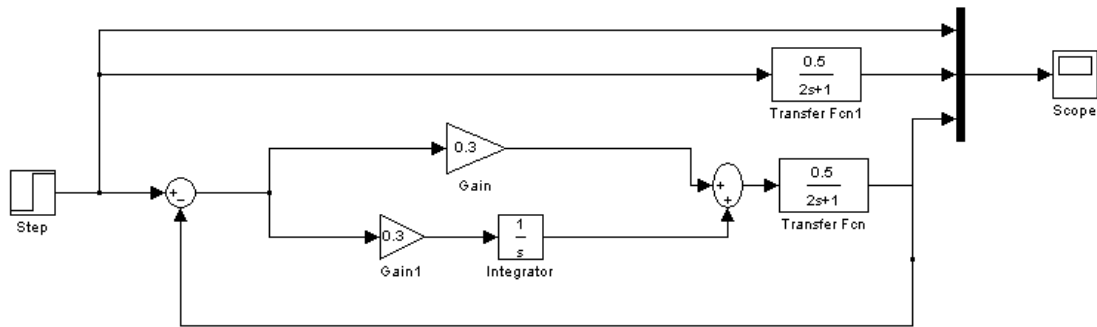


Figure 1 Schéma de simulation d'un PI classique sous Simulink.

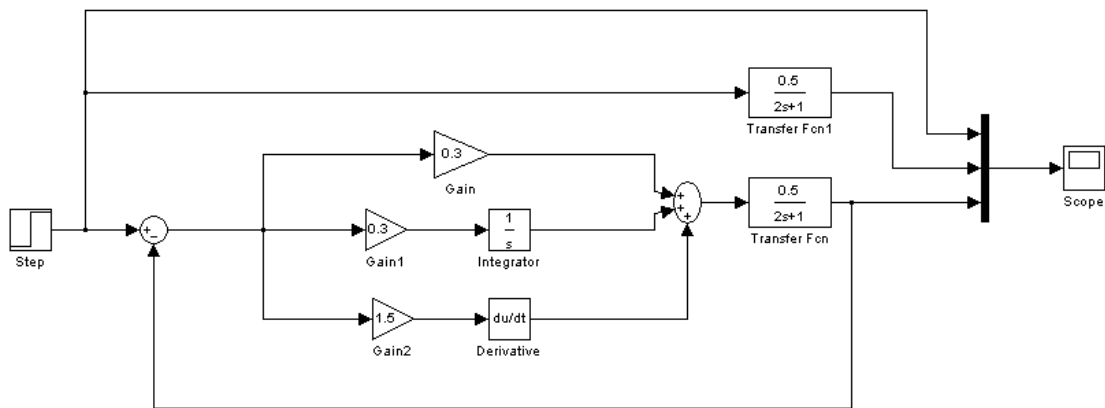


Figure 2 Schéma de simulation d'un PID classique sous Simulink.

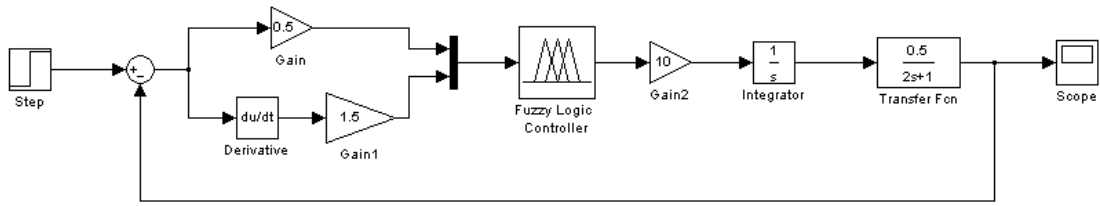


Figure 3 Schéma de simulation d'un PI flou sous Simulink.

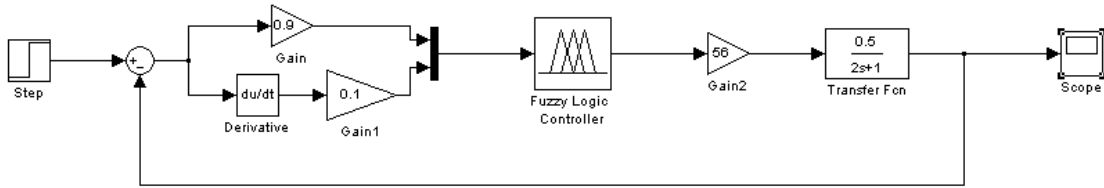


Figure 4 Schéma de simulation d'un PD flou sous Simulink.

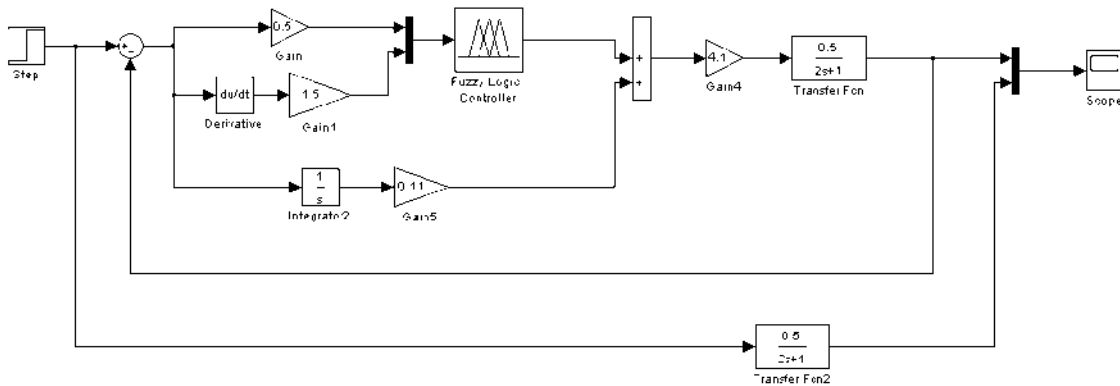


Figure 5 Schéma de simulation d'un PID flou sous Simulink.

# Bibliographie

- [1] **Hansruedi Buhler**, " Réglage par logique floue ", Edition Presse polytechniques et universitaire Romandes, Lausanne, 1994.
- [2] **Jean-Marie Flaus**, "la régulation industrielle - régulateurs PID, prédictif et flous "
- [3] " système linéaires – système asservis" université de Savoie DEUG STPI unité U32
- [4] **A. Rachid**, " systèmes de régulation ", Edition Masson, Paris 1996.
- [5] **Maurice Rivoire et Louis Ferrier**, " Asservissement, Régulation-commande analogique", Edition EYROLLES, juin 2006.
- [6] **G.Bouchebaba et N.Djehali**, " Commande floue d'un changeur de chaleur à contre-courant ».Mémoire d'ingénieur, UMMTO 2001.
- [7] **Gilles Fraisse**, Thèse de doctorat, " Application de la logique floue à régulation central du chauffage en régime intermittent ", Institut national des sciences appliquées.
- [8] **Pascal Maussion**, Thèse, " logique floue pour l'optimisation de la commande ", Laboratoire d'électrotechnique et d'électronique industrielle (LEEI)
- [9] **K. Kheddoudja**, " Commande d'un moteur à courant alternatif en utilisant les outils de la logique floue ". Mémoire de magister en automatique appliquée, UMMTO, 2005.
- [10] **J. Faucher**, Thèse de doctorat, " Réglage et commande à base de logique floue ". Ecole nationale supérieure d'électronique, électrotechnique, d'informatique et des télécommunications, 2006.