

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJI MOKHTAR - ANNABA  
BADJI MOKHTAR – ANNABA UNIVERSITY



جامعة باجي مختار - عنابة

Faculté : Sciences de L'ingéniorat

Département : Electronique

Domaine : Sciences et Techniques

Filière : Automatique.

Spécialité : Automatique et système

## Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème :

**Ajustage en ligne des régulateurs PID non linéaires et à ordre fractionnaire**

Présenté par : *SADOUN Mohammed Salah Eddine*

Encadrant : *BOULEBTATECHE Ibrahim*

**Jury de Soutenance :**

CHAKER Karima	MCB	Université Badji Mokhtar	Président
BOULEBTATECHE Ibrahim	MCB	Université Badji Mokhtar	Encadrant
GHERBI Soufiane	Professeur	Université Badji Mokhtar	Examineur

Année Universitaire : 2020/2021

## **REMERCIEMENTS**

Je remercie avant tout, Allah le Tout- Puissant de m'avoir donné courage et volonté pour réaliser ce travail et d'avoir guidé mes pas vers le savoir et la connaissance.

Je tiens à exprimer ensuite ma profonde gratitude et mon remerciement à mon encadreur Mr. BOULEBTATECHE Brahim de m'avoir accompagné avec persévérance dans l'élaboration de ce travail. Et de m'avoir encadré, orienté, aidé et conseillé, pour sa patience, sa disponibilité et surtout ses judicieux conseils qui ont contribué à alimenter ma réflexion.

Aux membres de jury qui m'ont fait l'honneur d'avoir bien accepté d'examiner le contenu du présent travail et m'avoir enrichie de leurs critiques constructives et jugements scientifiques.

Je remercie enfin toutes les personnes qui m'ont soutenu et aidé et qui ont contribué de près ou de loin à l'élaboration de ce travail.

## DEDICACE

Je dédie ce travail :

**A ma chère maman**, je ne saurais t'exprimer ma reconnaissance pour tous les sacrifices que tu as faits pour nous, pour toutes les valeurs que tu nous as inculqué, pour ta présence et ton soutien dans tout ce que je fais dans ma vie, tu es sans doute mon exemple de la parfaite. Merci pour tout.

**A mon cher papa**, je ne trouverai jamais les mots justes pour exprimer ma gratitude et le respect que j'éprouve à ton égard. Je te remercie pour tes encouragements, pour ton soutien, pour avoir mis à ma disposition tous les moyens pour réussir et pour tes sacrifices et ton travail dur afin qu'on ne manque de rien. Merci.

**A ma très chère sœur** ; Amina Wissal, mon bras droit ; je te remercie d'avoir toujours été là pour moi ; pour tes encouragements et soutien moral.

# Table des matières

<b>Introduction générale</b> .....	1
I. Introduction .....	1
II. Les objectifs de la commande sont multiples .....	1
III. Les méthodes de détermination des paramètres du contrôleur .....	1
III.1. Les méthodes empiriques .....	1
III.1. Les méthodes analytiques .....	2
IV. Bases de calcul .....	2
IV.1. Indices de performance .....	2
IV.1.1. Pourcentage de dépassement D .....	2
IV.1.2. Intégrales faisant intervenir l'erreur .....	2
IV.1.3. Les paramètres du domaine temporel.....	3
<b>Chapitre 1 : le régulateur PID Classique</b> .....	<b>4</b>
1. Introduction .....	4
2. Méthodes de définir les paramètres de contrôleur .....	6
3. Intérêts du PID.....	7
4. Avantages et inconvénients du PID.....	7
<b>Chapitre 2 : le régulateur PID d'ordre fractionnaire FOPID</b> .....	<b>8</b>
1. Introduction .....	8
2. Caractéristiques du FOPID .....	8
3. Structure du FOPID .....	8
4. Intérêts du FOPID.....	9
5. Avantages et inconvénients du FOPID.....	9
<b>Chapitre 3 : le régulateur PID non linéaire</b> .....	<b>11</b>
1. Définition .....	11
2. Lignes directrices pour obtenir les paramètres du contrôleur NL-PID .....	12
3. Essai de stabilité .....	12
4. Implémentation d'un NL-PID .....	13
5. Rôle des différentiateurs de suivi non linéaire .....	14
<b>Chapitre 4 : Etude de Simulation</b> .....	<b>15</b>
1. Introduction .....	15
2. Etude de système de contrôle de niveau d'un réservoir conique .....	15

2.1. Description du système .....	15
2.2. Utilisation des différents régulateur PID classique, FOPID, NL-PID.....	17
2.2.1. Contrôleur proportionnel intégral dérivé .....	17
2.2.1.a. dimensionnement .....	17
2.2.1.b. Optimisation en boucle fermée via Simulink .....	20
3. Implantation des régulateurs .....	21
3.1. Contrôleur propositionnel intégral dérivée .....	21
3.2. Contrôleur non linéaire propositionnel intégral dérivée .....	22
3.3. Contrôleur fractionnaire ordre propositionnel intégral dérivée .....	23
4. Algorithme PSO pour l'ajustement des régulateurs .....	24
4.1. Introduction .....	24
4.2. Optimisation par essaim de particules PSO .....	24
4.2.1. Formulation générale.....	24
4.3. Déroulement de l'algorithme .....	25
5. Optimisation des gains des contrôleurs par la technique PSO .....	26
5.1. Contrôleur proportionnel intégral dérivée .....	28
5.1.1. Conception du contrôleur PID.....	28
5.1.2. Ajustement du régulateur PID par l'algorithme PSO .....	29
5.2. Contrôleur d'ordre proportionnel intégral proportionnel dérivée .....	30
5.3. Contrôleur non linéaire proportionnel intégral proportionnel dérivée NL-PID .....	31
<b>Conclusion générale.....</b>	<b>32</b>

## Table des figures

<b>Figure 01</b>	Intégrale faisant intervenir l'erreur .....	3
<b>Figure 02</b>	La réponse type d'un procédé stable .....	5
<b>Figure 03</b>	Structure d'un FOPID .....	8
<b>Figure 04</b>	Représentation graphique du contrôleur PID d'ordre fractionnaire (FOPID) .....	9
<b>Figure 05</b>	Schéma d'un système avec contrôleur FOPID (Fractional Order PID) .....	10
<b>Figure 06</b>	Schéma fonctionnel du système contrôlé NL-PID .....	11
<b>Figure 07</b>	La structure traditionnelle du contrôleur NLPID .....	13
<b>Figure 08</b>	Réservoir conique .....	15
<b>Figure 09</b>	Réponse indicielle du système en boucle ouverte .....	18
<b>Figure 10</b>	Réponse indicielle référence tracking .....	19
<b>Figure 11</b>	Réponse indicielle en boucle fermée .....	20
<b>Figure 12</b>	Schéma du contrôleur PID .....	21
<b>Figure 13</b>	Réponse indicielle du correcteur PID .....	21
<b>Figure 14</b>	Schéma d'un contrôleur FOPID .....	22
<b>Figure 15</b>	Réponse indicielle du correcteur FOPID .....	22
<b>Figure 16</b>	Schéma d'un contrôleur NL-PID .....	23
<b>Figure 17</b>	Réponse indicielle du correcteur NL-PID .....	23
<b>Figure 18</b>	bloc diagramme d'implantation des PSO pour l'ajustement du PID .....	29
<b>Figure 19</b>	Schéma fonctionnel d'un FOPID avec PSO .....	30
<b>Figure 20</b>	Schéma de structure .....	32

## Liste des Tableaux

<b>Tab 1</b> Actions du contrôleur .....	5
<b>Tab 2</b> Effet de PID sur la réponse du système .....	6

## Liste des abréviations

<b>PID</b> :	Proportionnelle, Intégrale et Dérivative
<b>FOPID</b> :	Fractional Order PID
<b>NL-PID</b> :	non linéaire Proportionnelle, intégrale dérivée
<b>NL-FOPID</b> :	PID non linéaire d'ordre fractionnaire
<b>PSO</b> :	Particle Swam Optimization (OEP : optimisation par essaims de particules)
<b>FO</b> :	Fractional Order (OF : Ordre fractionnaire)
<b>FT</b> :	Fonction de transfert
<b>FOMCON</b> :	Modélisation et contrôle par ordre fractionnaire
<b>FPID</b> :	Contrôleur de dérivé intégral proportionnel fractionnaire
<b>MIGO</b> :	Optimisation du gain intégral MS contraint

## **Résumé :**

On a parlé en général des contrôleurs (Proportionnel intégral dérivé classique (PID), FOPID, NL-PID), définir les objectifs (la robustesse, la rapidité et le critère de précision) et les méthodes de détermination des paramètres des contrôleur (méthodes empirique et méthode analytique), on a défini les indices de performances

Etude de système a l'état ouvert et le système a l'état fermé, implémentation des différents régulateur (PID, FOPID, NL-PID)

Application de système de régulation au niveau d'un réservoir conique

Optimisation en boucle ouverte et en boucle fermé avec Matlab et Simulink

Optimisation des gains par l'algorithme PSO

## **Mot clés :**

PID (Contrôleurs classique), FOPID (Contrôles PID d'ordre fractionnaires), NLPID (Contrôleurs PID non linéaire).

Algorithme : PSO (Particle Swam Optimization), (OEP : optimisation par essais de particules).

Matlab. Simulink.

## **Abstract :**

We talked in general about the controllers (Proportional Integral Derivative Classic (PID), FOPID, NL-PID), define the objectives (robustness, speed and accuracy criterion) and the methods of determination of the parameters of the controllers (empirical method and analytical method), we defined the performance indices

Study of open and closed loop systems, implementation of different controllers (PID, FOPID, NL-PID)

Application of control system to a conical tank

Open loop and closed loop optimization with Matlab and Simulink

Optimisation of the gains by the PSO algorithm

## **Keyword :**

PID (Classical PID Controllers), FOPID (Fractional Order PID Controllers), NLPID (Nonlinear PID Controllers).

Algorithm: PSO (Particle Swam Optimization).

Matlab. Simulink.



## ملخص

تحدثنا بشكل عام عن المتحكمات (مشتق كلاسيكي متكامل نسبي، **PID, FOPID, NL-PID** مع تحدد الأهداف (المتانة، السرعة ومعيار الدقة) وطرق تحديد معالم وحدات التحكم (الطرق التجريبية والطريقة التحليلية)، حددنا أداء المؤشرات دراسة النظام في الحالة المفتوحة والنظام في الحالة المغلقة وتنفيذ مختلف المنظمين ، تطبيق نظام تنظيم على مستوى الخزان المخروطي الحلقة المفتوحة والحلقة المغلقة الأمثل مع **Matlab** تحسين المكاسب بواسطة الخوارزمية **PSO**

## الكلمات المفتاحية

ذات الترتيب **PID**، وحدات تحكم (ذات الترتيب الجزئي **PID** وحدات تحكم) **FOPID**، (الكلاسيكية **PID** وحدات تحكم) **PID** (تحسين مسابح الجسيمات). سيمولينك **PSO**: الخوارزمية. (غير الخطية **PID** وحدات تحكم) **NLPID**، (الجزئي غير الخطي

# Introduction Générale

## I. Introduction

Le contrôleur Proportionnel-Intégrateur-Dérivateur (PID) est le régulateur le plus utilisé dans la commande des processus industriels.

Actuellement et malgré les progrès dans les domaines de la théorie du contrôle, les régulateurs PID sont encore très largement utilisés dans l'industrie.

Les raisons majeures de sa large acceptation sont :

- Sa capacité de commander la majorité des processus.
- Ses actions sont bien comprises.
- Son implémentation est très simple.

Le PID est obtenu par l'association de trois actions (Proportionnel, Intégral et Dérivée) qui sont la conception et le réglage du correcteur PID

Il a été un sujet de recherche depuis le jour où Ziegler et Nichols ont présenté leur méthode de réglage en 1942, qui consiste à déterminer les coefficients  $K_r$ ,  $T_i$  et  $T_d$  du régulateur, afin d'obtenir une réponse adéquate du procédé et de la régulation. [1]

Ces deux auteurs ont proposé deux approches expérimentales destinées à ajuster rapidement les paramètres des régulateurs :

- a. La première nécessite l'enregistrement de la réponse indicielle du système en boucle ouverte.
- b. La deuxième demande d'amener le système en boucle fermée à sa limite de stabilité.

## II. Les objectifs de la commande sont multiples

La robustesse, la rapidité et la précision sont les principaux objectifs d'une régulation.

- La robustesse est sans doute le paramètre le plus délicat. On dit qu'un système est robuste si sa régulation réagit à toutes les perturbations.
- La rapidité du contrôleur dépend du temps de montée et du temps d'établissement du régime permanent.
- Le critère de précision est basé sur l'erreur statique.

## III. Les méthodes de détermination des paramètres du contrôleur

Ils se divisent en deux catégories : méthodes empiriques et méthodes analytiques :

### III.1. Les méthodes empiriques

Elles sont basées sur des essais expérimentaux sur le procédé. Elles sont efficaces, simples à mettre en œuvre ; mais leur application devient longue sur les processus lents puisque le réglage se fait successivement (action proportionnelle ensuite action intégrale puis l'action dérivée).

### **III.2. Les méthodes analytiques**

Ce sont les techniques de calcul basées sur le modèle du procédé à commander, telles que, la commande optimale LQG, la commande  $H_\infty$ , la commande d'ordre fractionnaire (FO) et la commande à modèle interne

## **IV. Bases de calcul**

### **IV.1. Indices de performance**

Pour définir la qualité de la régulation, on se base en général sur l'analyse de la réponse indicielle de l'ensemble régulateur et du système (boucle à l'état ouvert).

Différents indices de performance peuvent être évalués à partir de la réponse temporelle.

De façon générale, on cherche à quantifier la différence entre la réponse réelle du système asservi et une valeur idéale qui serait un échelon.

Les indices couramment utilisés sont définis de la façon suivante :

#### **IV.1.1. Pourcentage de dépassement « D »**

Avant de se stabiliser, la sortie du système passe par un régime transitoire oscillant de part et d'autre de la valeur finale. On définit le pourcentage de dépassement par :

$$d\% = [y_{\max} - y(\infty) \times 100\%] / y_{\max} \quad (01)$$

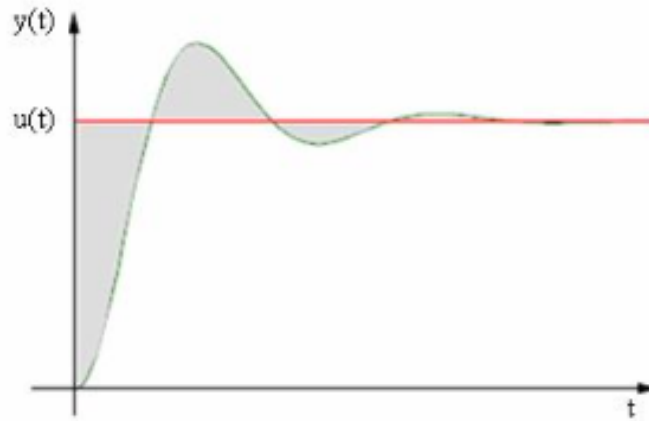
#### **IV.1.2. Intégrales faisant intervenir l'erreur**

Pour évaluer la différence existante entre la réponse réelle et une réponse idéale de type échelon.

On calcule l'intégrale d'un terme positif faisant intervenir l'erreur.

De cette façon l'indice calculé augmente avec l'écart entre la réponse réelle et la réponse idéale.

En pratique, l'intégrale est calculée sur un intervalle  $[0, T]$  suffisamment étendu pour contenir tout le régime transitoire.



**Figure (01) :** Intégrale faisant intervenir l'erreur.

L'erreur  $\varepsilon(t)=y(t)-u(t)$ ; où  $y(t)$ : signal de sortie,  $u(t)$ : signal d'entrée **(02)**

- L'intégrale de la valeur absolue de l'erreur  $\varepsilon(t)$  est donnée par :

$$IAE = \int_0^T I_{\varepsilon}(t) dt \quad \text{(03)}$$

Cet indice exprime la surface générée par la différence entre la valeur de consigne et la valeur réelle.

- On utilise également l'intégrale de l'erreur quadratique, définie par :

$$ISE = \int_0^T I_{\varepsilon^2}(t) dt \quad \text{(04)}$$

- Pour pénaliser les systèmes dont le régime transitoire dure trop longtemps, on utilise également l'intégrale du produit de l'erreur par le temps, donnée par :

$$ITAE = \int_0^T t \cdot I_{\varepsilon}(t) dt \quad \text{(05)}$$

- Et également l'intégrale du produit de l'erreur quadratique par le temps, donnée par :

$$ISTE = \int_0^T t \cdot I_{\varepsilon^2}(t) dt \quad \text{(06)}$$

Il existe une liste plus complète de mesures de performances d'un système asservi.

#### IV.1.3. Les paramètres du domaine temporel de la réponse du système :

- Le dépassement de crête ( $M_p$ ).
- Le temps de montée ( $t_r$ ).
- Le temps de stabilisation ( $t_s$ ).
- L'erreur d'état stable (Ess) Steady-state error

$$d\% = [y_{\max} - y(\infty) \times 100\%] / y_{\max} \quad (01)$$

# CHAPITRE 1 : Le régulateur PID Classique

## 1. Introduction

La régulation en PID est une commande classique mais largement utilisée en industrie ; elle permet de régler un grand nombre de grandeurs physiques.

Vue la simplicité de l'implémentation dans des calculateurs de plus en plus performants,

Le PID est un système d'auto régulation (boucle fermée), qui cherche à réduire l'erreur entre la consigne (SP) et la mesure (PV).

Ce compensateur est utilisé en commande des procédés industriels et des moteurs, dans le monde de l'aéronautique ; l'automobile, pétrochimie, alimentaires, robotique etc...

Le PID est obtenu par l'association de trois actions (Proportionnel, Intégral et Dérivée) Pour avoir une

$$C(s) = G + \frac{1}{T_i} \cdot \frac{1}{s} + T_d \cdot s \quad (7)$$

Bonne réponse du procédé et de la régulation ; on doit déterminer les coefficients K, Ti et Td qui sont respectivement l'action proportionnelle, intégrale et dérivée du régulateur.

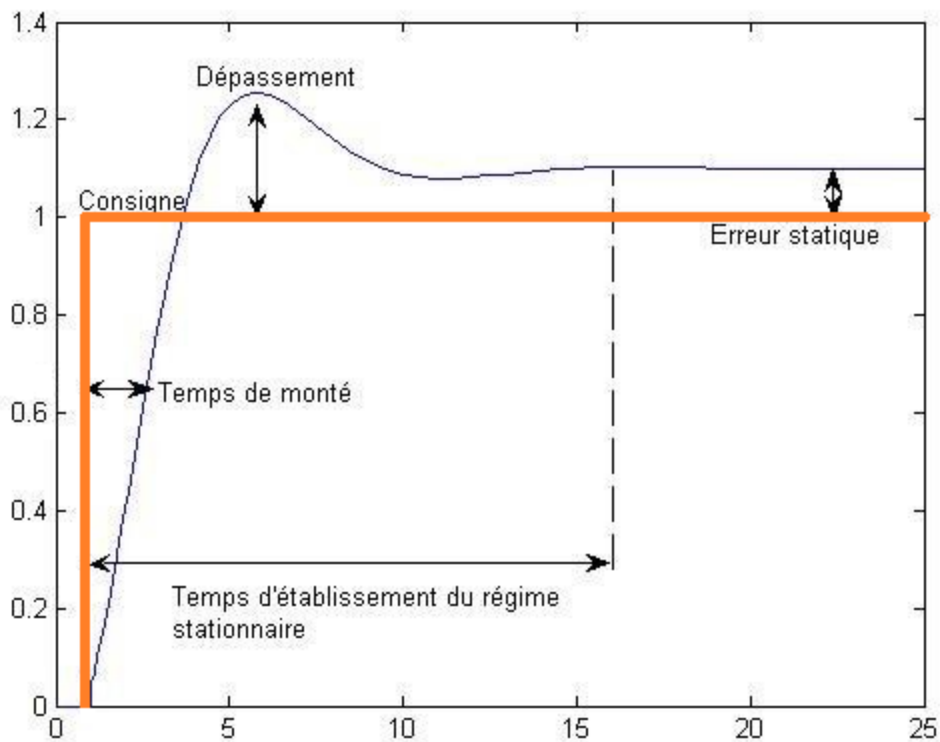
Le réglage d'un PID consiste à déterminer les coefficients afin d'obtenir une réponse adéquate du procédé et de la régulation. Les objectifs sont d'être **robustes**, **rapide** et **précis**.

Il faut pour cela :

En mode de régulation (consigne fixe) choisir des réglages permettant à la grandeur réglée (mesure) de retourner dans un temps raisonnable à sa valeur de consigne.

En mode d'asservissement (consigne variable), choisir des réglages permettant de limiter le ou les éventuels **dépassements** (overshoot) de la grandeur réglée (mesure).

La réponse type d'un procédé stable est la suivante :



**Figure (02) :** La réponse type d'un procédé stable

Dans le cas des systèmes simples, les paramètres du PID influencent sur la réponse du système de la manière suivante :

	Précision	Stabilité	Rapidité
P	↗	↘	↗
I	↗	↘	↘
D	↘	↗	↗

**Tableau (01) :** Actions du contrôleur ( $K_p$  ;  $K_i$  ;  $K_d$ )

Les actions des gains proportionnel, intégral et dérivatif, chaque action dispose d'avantages et d'inconvénients, c'est la combinaison des 03(trois) avec le bon paramétrage qui donne au correcteur PID son efficacité.

## 2. Méthodes d'ajustage des paramètres d'un Contrôleur

Il existe plusieurs méthodes pour connaître les paramètres du contrôleur, les méthodes de réglage les plus utilisées en théorie sont :

- Les méthodes de ZieglerNichols (en boucle ouverte et boucle fermée).
- La méthode de P. Naslin (polynômes normaux à amortissement réglable).
- La méthode du lieu de Nyquist inverse (utilise le diagramme de Nyquist).
- Le diagramme de Black permet d'en constater très visuellement les effets.

Les méthodes de réglage les plus utilisées dans la pratique sont :

- L'identification par modèle de Broïda pour les systèmes stables.
- Le modèle intégrateur retardé pour les systèmes instables.
- La méthode par approches successives, qui répond à une procédure rigoureuse : on règle d'abord l'action P seule pour avoir un dépassement de 10 à 15 % puis l'action dérivée de façon à « raboter » au mieux le dépassement précédent, enfin on ajuste si nécessaire l'action intégrale en se fixant un dépassement final compris entre 5 et 10 %.

### Remarque :

Le régulateur PID permet de réaliser des performances telles que la stabilité, la rapidité et la précision grâce à la combinaison des trois actions P, I et D (la structure peut être série, parallèle ou mixte). Ceci peut se faire en ajustant les paramètres associés à ces actions.

D'après les expériences le régulateur PID représente des effets sur la réponse ; Ils sont présentés dans le tableau suivant (T 02) :

<b>Paramètre</b>	<b>Temps de montée</b>	<b>Temps de stabilisation</b>	<b>Dépassement</b>	<b>Erreur statique</b>
<b>Kp</b>	Diminue	Augmente	Augmente	Diminue
<b>Ki</b>	Diminue	Augmente	Augmente	Annule
<b>Kd</b>		Diminue	Diminue	

**Tableau (02) :** effet de PID sur la réponse du système

2- Les actions des gains proportionnel, intégral et dérivatif, chaque action dispose d'avantages & d'inconvénients, c'est la combinaison des 3 avec le bon paramétrage qui donne au correcteur PID son efficacité.

### **3. Intérêts du PID**

Les PID permettent de réguler efficacement les procédés industriels en se basant sur le retour et la comparaison de leur propre erreur grâce à la structure en boucle fermée

### **4. Avantages & inconvénients du PID**

#### **Avantages du PID :**

- Régulation flexible a trois paramètres différents
- Précision et rapidité
- Ne nécessite pas une modélisation dynamique du processus.
- Fiabilité

#### **Inconvénients du PID :**

- Nécessité d'implémenter les bons paramètres.
- La méthode de calcul peut prendre du temps.
- Trois paramètres au lieu de cinq Peut être pas insuffisamment



# CHAPITRE 2 : Le régulateur PID d'Ordre Fractionnaire FOPID

## 1. Introduction

La conception et le réglage du correcteur PID a été un sujet de recherche depuis le jour où Ziegler et Nichols ont présenté leur méthode de réglage en 1942.[1]

Bien que toutes les techniques existantes pour le réglage des paramètres du correcteur PID, un travail de recherche continu et intensif est encore en cours pour le rehaussement de la qualité et l'amélioration des performances de la commande.

En automatique, ce n'est qu'au début des années 1990 que le régulateur CRONE (Commande Robuste d'Ordre Non Entier) a été proposé par OUSTALOUP [2].

En profitant des propriétés avantageuses des systèmes d'ordre fractionnaire, ce régulateur permettait d'assurer la Robustesse de la commande dans une bande de fréquences donnée.

Depuis cette initiative, la commande d'ordre fractionnaire captiva l'intérêt de beaucoup de chercheurs.

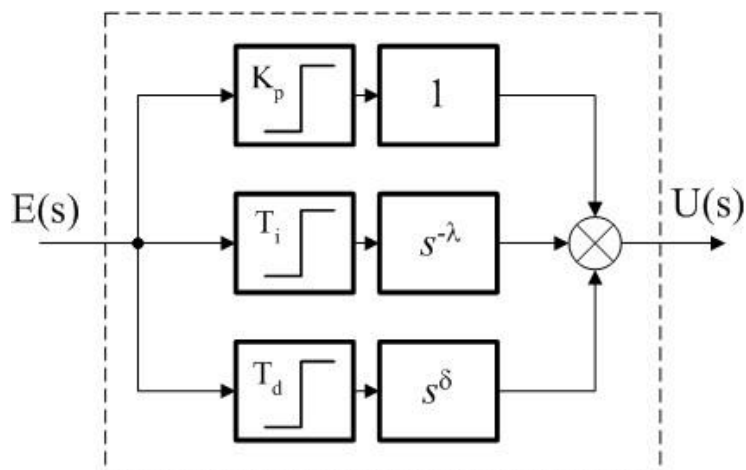
En 1999, PODLUBNY [3] a proposé le régulateur  $PI\lambda D\mu$ , une généralisation du correcteur PID classique, comprenant une intégration fractionnaire d'ordre  $\lambda$  et une dérivation fractionnaire d'ordre  $\mu$ , élargissant ainsi le champ d'application du calcul fractionnaire à la théorie de la commande ce qui a orienté plusieurs chercheurs à un Nouvel axe de recherche qui est le réglage du correcteur  $PI\lambda D\mu$  d'ordre fractionnaire [3] [4]

## 2. Caractéristiques du FOPID

Similaire au PID dans l'ensemble, le FOPID dispose de quelques additions (voir structure ci-dessous) :

## 3. Structure du FOPID

Le schéma du FOPID est similaire à celui du PID parallèle classique.



**Figure (03) :** Structure d'un FOPID

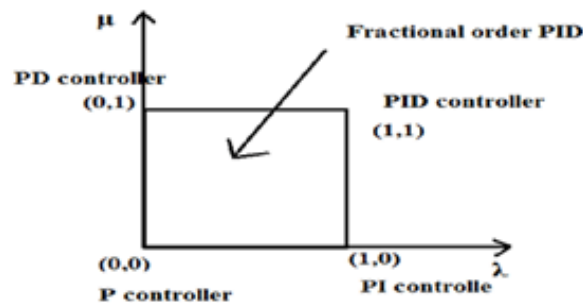
- Equation de la Fonction de Transfert d'un régulateur d'ordre fractionnel (FOPID) en boucle fermée :

$$G_{FOPID}(s) = \frac{U(s)}{E(s)} = k_p + k_i \frac{1}{s^\lambda} + k_d s^\mu, \quad (\lambda, \mu > 0) \quad (8)$$

$\frac{1}{s^\lambda}$  : opérateur intégrale d'ordre  $\lambda$        $0 < \lambda < 1$

$s^\mu$  : opérateur intégrale d'ordre  $\mu$        $0 < \mu < 1$

- Représentation graphique :



**Figure (04) :** Représentation graphique du contrôleur PID d'ordre fractionnaire (FOPID)

#### 4. Intérêts du FOPID

Les FOPID permettent de réguler en plus efficacement les procédés industriels que les PID classiques et ce grâce aux deux paramètres supplémentaires qu'ils ont. En se basant sur le retour et la comparaison de leur propre erreur grâce à la structure en boucle fermée de la même manière que le modèle classique.

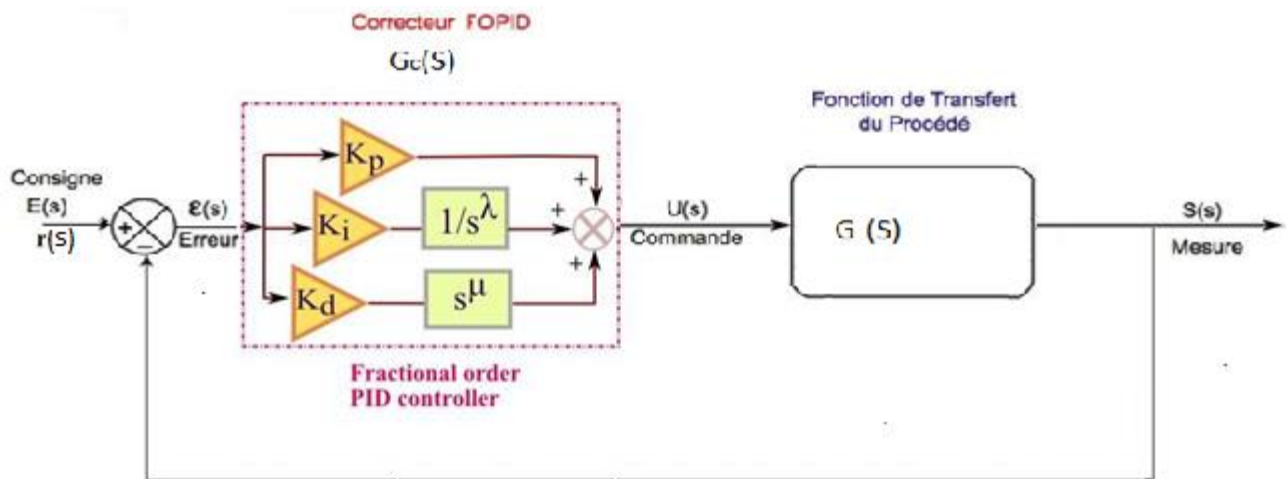
#### 5. Avantages & inconvénients du FOPID

##### Avantages :

- Encore plus de précision & de stabilité que le PID classique.  
Mêmes avantages que le PID classique

##### Inconvénients :

- Difficulté à trouver les bons paramètres et à ajuster



**Figure (05) :** Schéma d'un système avec contrôleur FOPID (Fractional Order PID)

## CHAPITRE 3 : Le régulateur PID Non Linéaire NL-PID

### 1. Définition

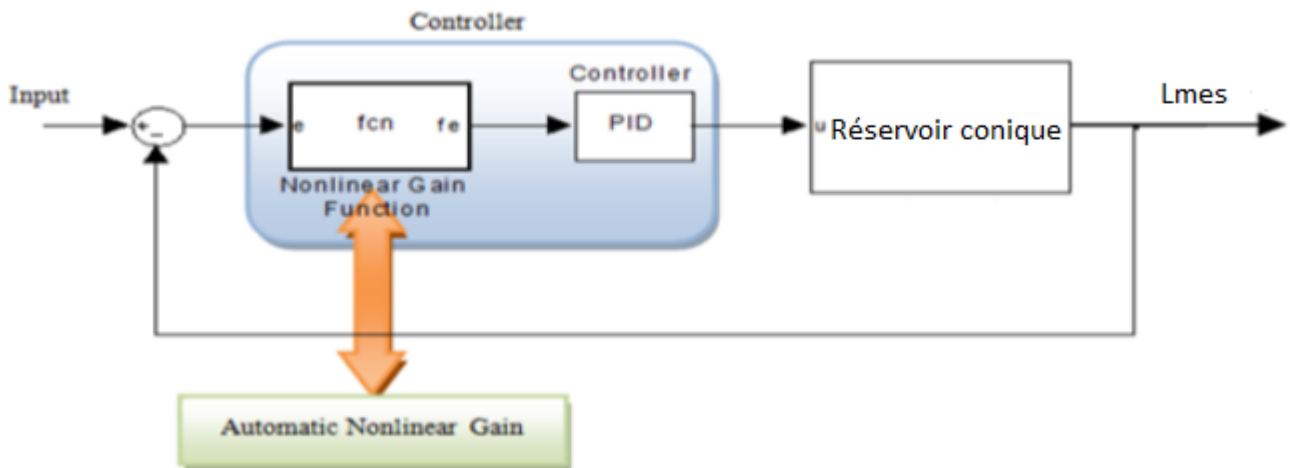
En général, le contrôleur PID non linéaire (NL-PID) est développé par un gain non linéaire borné par secteur,  $K(e)$  qui est intégré en série (cascade) avec régulateur PID, [5].

Le schéma fonctionnel du système de contrôle NL-PID est présenté dans la figure (06).

Le réglage de gain auto-accordé,  $K(e)$  se comporte comme une fonction d'erreur non linéaire,  $e(t)$  qui est compris entre  $0 \leq K(e) \leq K(e_{\max})$  comme indiqué en (10) et (11).

Ceux-ci peuvent déterminer la gamme de choix pour le gain non linéaire,  $K(e)$ . [5]

La sortie formée de cette fonction non linéaire est reconnue comme une erreur mise à l'échelle,  $fe$  et l'équation de  $fe$  sont présentées en (12). Alternativement, L'équation globale du contrôleur NPID peut être vue comme dans (13).



**Figure (06) :** Schéma fonctionnel du système contrôlé NL-PID

Gain non linéaire  $K(e)$  :

$$K(e) = \frac{\exp(\alpha e) + \exp(-\alpha e)}{2} \quad (09)$$

$$\text{Error, } e = \begin{cases} e & ; \text{ if } |e| \leq e_{\max} \\ e_{\max} * \text{sign}(e) & ; \text{ else } |e| > e_{\max} \end{cases} \quad (10)$$

$$K(e_{\max}) = \frac{1}{|G(j\omega)|} \quad (11)$$

Erreur échelle  $f_e$

$$f_e = K(e) * e(t) \quad (12)$$

Fonction de transfert générale du contrôleur NL-PID [5]

$$G_{NL-PID}(s) = [K_p * f_e] + [K_i/s * f_e] + [K_d s * f_e] \quad (13)$$

## 2. Lignes directrices pour obtenir les paramètres du contrôleur NL-PID

Pour concevoir le contrôleur NL-PID ; on doit déterminer :

- Les paramètres du contrôleur PID.
- La variation de taux de gain non linéaire.
- La valeur de plage de variation erreur, max.

Un contrôleur PID non linéaire est développé en définissant une fonction non linéaire qui donne les paramètres du contrôleur en fonction de l'erreur calculée et de certains paramètres de réglage.

## 3. Essai de stabilité : [5]

$\alpha$  : le taux de variation de non linéaire gain

$e_{max}$  : est la plage de variation d'erreur.

Les deux valeurs de  $\alpha = 0,5$  et  $e_{max} = 3$ .

La sélection des paramètres  $\alpha$  et  $e_{max}$  est choisie en fonction de la valeur maximale admissible du gain non linéaire,  $K(e)$  qui est sélectionné en fonction de la plage de stabilité.

Remarque : [5]

1. Le cas de valeur fixe du contrôleur PID, le tracé de Nyquist pourrait être utilisé pour le test de stabilité.
2. Le cas du NL-PID contrôleur où la valeur de l'erreur mise à l'échelle,  $f_e$  garder changea, le tracé de Nyquist n'est plus valable.

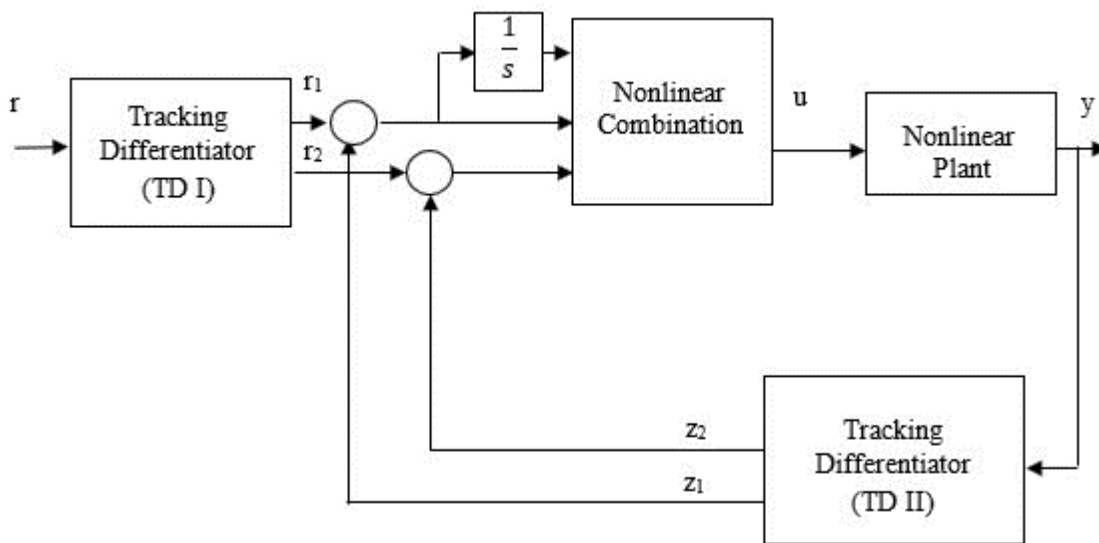
Par conséquent, Critère de stabilité de Popov ou critère de stabilité de Lyapunov pourrait être employé à ce cas. [4]

#### 4. Implémentation d'un NL-PID

En utilisant un différentiateur de suivi non linéaire, un contrôleur PID standard est transformé en PID non linéaire (NLPID) [5] comme le montre la figure 1.

Le premier différentiateur de suivi (TD(I)) est utilisé comme générateur de profil de processus transitoire, tandis que le second différentiateur de suivi (TD(II)) est utilisé comme observateur d'état pour obtenir la sortie de suivi  $z_1$  et son différentiel  $z_2$ .

Les signaux d'erreur, d'intégrale et différentielle sont produits en comparant le profil de processus transitoire à la sortie de TD(II).



**Figure (07) :** La structure traditionnelle du contrôleur NLPID

Après quelques recherches sur les structures traditionnelles et les propriétés essentielles du différentiateur de suivi non linéaire. Une sorte de différentiateur de suivi non linéaire du second ordre a été proposé pour les systèmes du second ordre :

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -R \operatorname{sign} \left( x_1 - v(t) + \frac{x_2 |x_2|}{2R} \right) \end{aligned} \quad (14)$$

Où  $x_1$  est la trajectoire désirée et  $x_2$  est sa dérivée. Notez que le paramètre  $R$  dépend de l'application et qu'il est défini en conséquence pour accélérer ou ralentir le profil transitoire. Ensuite,  $x_2$  est désigné comme le « différentiateur de suivi » de  $v(t)$ .

Afin d'éviter le broutage près de l'origine, en changeant la fonction signe en fonction de saturation linéaire sat,

Alors la modification est représentée par :

$$x_1 = x_2$$

$$\dot{x}_1 = -R \operatorname{sat}\left(x_1 - v(t) + \frac{x_2 |x_2|}{2R}, \delta\right) \quad (15)$$

$$\text{ou} \quad \operatorname{sat}(A, \delta) = \begin{cases} \operatorname{sign}(A), & |A| > \delta \\ \frac{A}{\delta}, & |A| \leq \delta \end{cases} \quad (16)$$

La règle de contrôle prend :

$$u = \beta_1 \operatorname{fal}(e, \alpha_1, \delta_1) + \beta_2 \operatorname{fal}(\dot{e}, \alpha_2, \delta_2) + \beta_3 \operatorname{fal}\left(\int e, \alpha_3, \delta_3\right) \quad (17)$$

$$\text{ou: } \alpha_1, \alpha_2 \text{ and } \alpha_3 \in [0.5 \ 1]$$

## 5. Rôle des différentiateurs de suivi non linéaire (NTD) [5]

Deux NTD de premier ordre sont utilisés dans la structure de contrôle, l'un pour le point de consigne et l'autre dans chemin de retour pour la variable de processus (mesure).

Un différentiateur de suivi donne l'estimation de la dérivée de l'entrée et de l'entrée suivie. Son utilisation rend l'action différentielle plus rapide et arrête l'influence du signal de bruit jusqu'à dans une certaine mesure.

## CHAPITRE 4 : Etude de cas et Résultats de Simulation

### 1. Introduction

Dans le cadre de ce travail, nous traitons le problème de contrôle de processus :

- Etude du système de contrôle de niveau d'un réservoir conique.
- Utilisation des différents régulateurs (PID classique ; FOPID.NL- PID ; NL-FOPID)
- Définir les performances de chaque type du régulateur.
- Comparer les performances des régulateurs.

### 2. Etude du système de contrôle de niveau d'un réservoir conique : [5]



**Figure (08)** : réservoir conique

#### 2.1. Description du système [6]

Le réservoir conique mesure 30 cm de hauteur. L'élément de contrôle final est une pompe à moteur à courant continu qui peut fournir un débit d'entrée maximal de 1600 lph. Un transmetteur de pression différentielle est utilisé pour mesurer le niveau par le principe  $DP = \rho GH$  (18)

Le processus est modélisé en utilisant la méthode d'identification de la boîte noire en tant que modèle de processus de premier ordre avec temps mort (FOPDT). Voir la courbe dans le chapitre antécédent.



Un processus au niveau du réservoir conique est un processus non linéaire en raison de sa surface variable par rapport à sa hauteur et rend le contrôle de niveau très difficile

Le changement de la forme conduit à la non linéarité

Le processus non linéaire est modélisé sous forme de modèles linéaires par morceaux à certains points de fonctionnement

Le modèle de processus obtenu est modélisé et définit comme fonction du système suivante :

$$P(s) = \frac{0.925}{25.05s + 1} e^{-1.09s} \quad (19)$$

L'équation de transfert (19) nous indique qu'il y'a présence d'un retard.

Le terme de retard d'entrée est donné dans le domaine temporel par :

$$u(t) = ud(t - L). \quad (20)$$

Donc on peut dire d'elle se compose de 02 deux parties est sous la forme : [8]

(21)

$$P(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \dots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \dots + a_0 s^{\alpha_0}} e^{-Ls}$$

La forme générale est donc (21)

D'habitude on prend :  $\beta_0 = \alpha_0 = 0$  pour que le gain statique du système soit donné par :

$$K = b_0/a_0, \text{ et } L \in \mathbb{R}^+.$$

Ce qui donne un modèle particulier de ce type qui est le premier ordre fractionnaire plus le temps de retard modèle :

$$P(s) = \frac{K}{1 + Ts^\alpha} e^{-LS} \quad (22)$$

Où K est le gain statique,  $L > 0$  est le retard,  $T > 0$  est la constante de temps,

et  $\alpha \in (0, 2)$  est la puissance fractionnaire de l'opérateur.

Pour le cas ou  $\alpha = 1$  est devenu la base de nombreuses règles de réglage

Ce qui nous ramène à dire que la fonction du système :

$$P(s) = \frac{0.925}{25.05s + 1} e^{-1.09s} \quad (19)$$

Possède les caractéristiques suivantes :

Gain statique	: K=0,925
Retard	: L=1,09s
Constante du temps	: T=25,05
Puissance fractionnaire de l'opérateur	: $\alpha=1$

## 2.2. Utilisation des différents régulateurs (PID classique , FOPID,NL- PID)

### 2.2.1 Contrôleur Proportionnel Intégral Dérivé (PID) :

Il existe plusieurs façons d'ajuster les paramètres d'un contrôleur

Dans cette partie on utilise :

- Dimensionnement du régulateur en boucle ouverte à l'aide de Matlab.
- Une méthode empirique (par exemple les méthodes de Chien-Hrones-Reswick)
- Optimisation en boucle fermée.
- Optimisation par l'algorithme PSO

### 2.2.2. Dimensionnement :

Utiliser Matlab pour d'abord visualiser le système voire les performances en boucle ouverte et ensuite essayer de corriger le système avec un correcteur PID.

Application de la fonction suivante

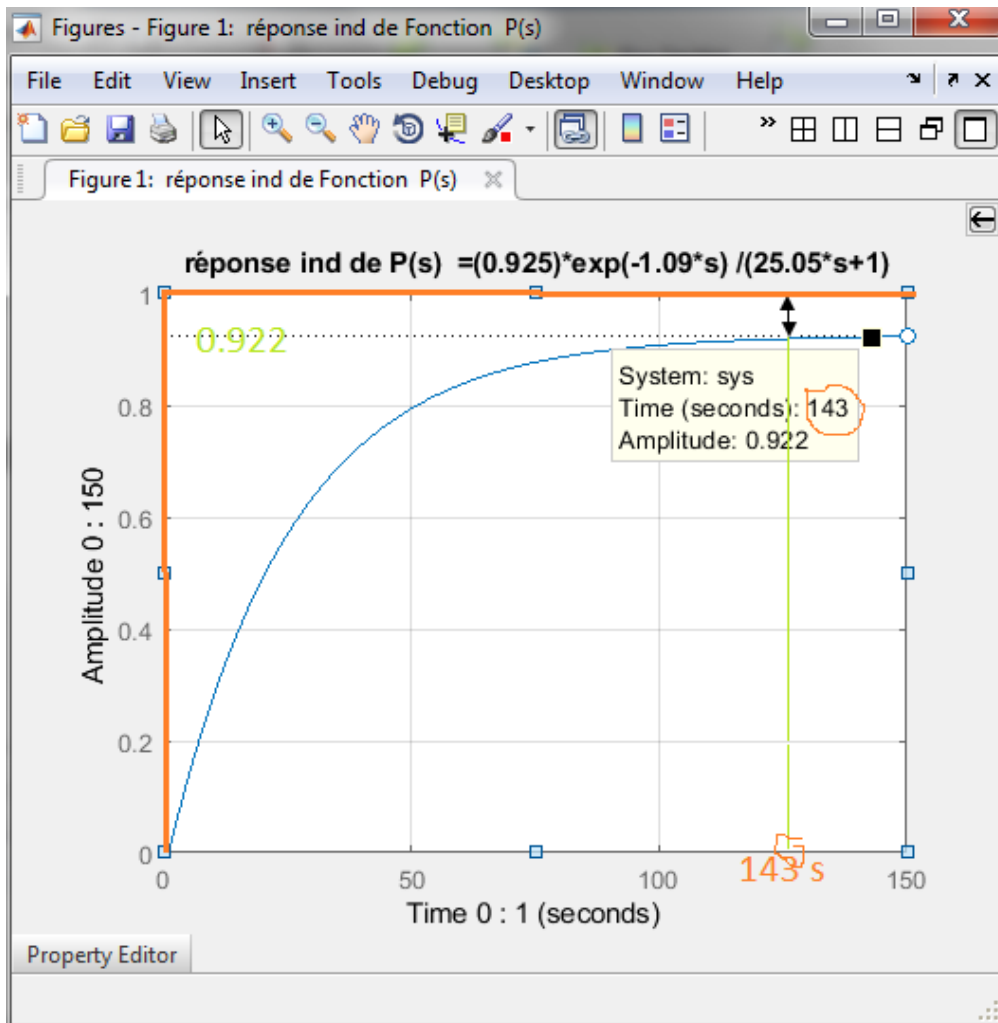
Création de fonction de transfert sur Matlab :

$$P(s) = \frac{0.925}{25.05s + 1} e^{-1.09s} \quad (19)$$

### Code 01:

```
s=tf('s'); Ps= (0.925) *exp(-1.09*s) /(25.05*s+1) ; step(Ps)
```

On aura la courbe (14) :



**Figure (09) :** réponse indicielle du système en boucle ouverte

On remarque le système se stabilise après un temps de 143 secondes mais toujours avec une erreur statique amplitude 0,922

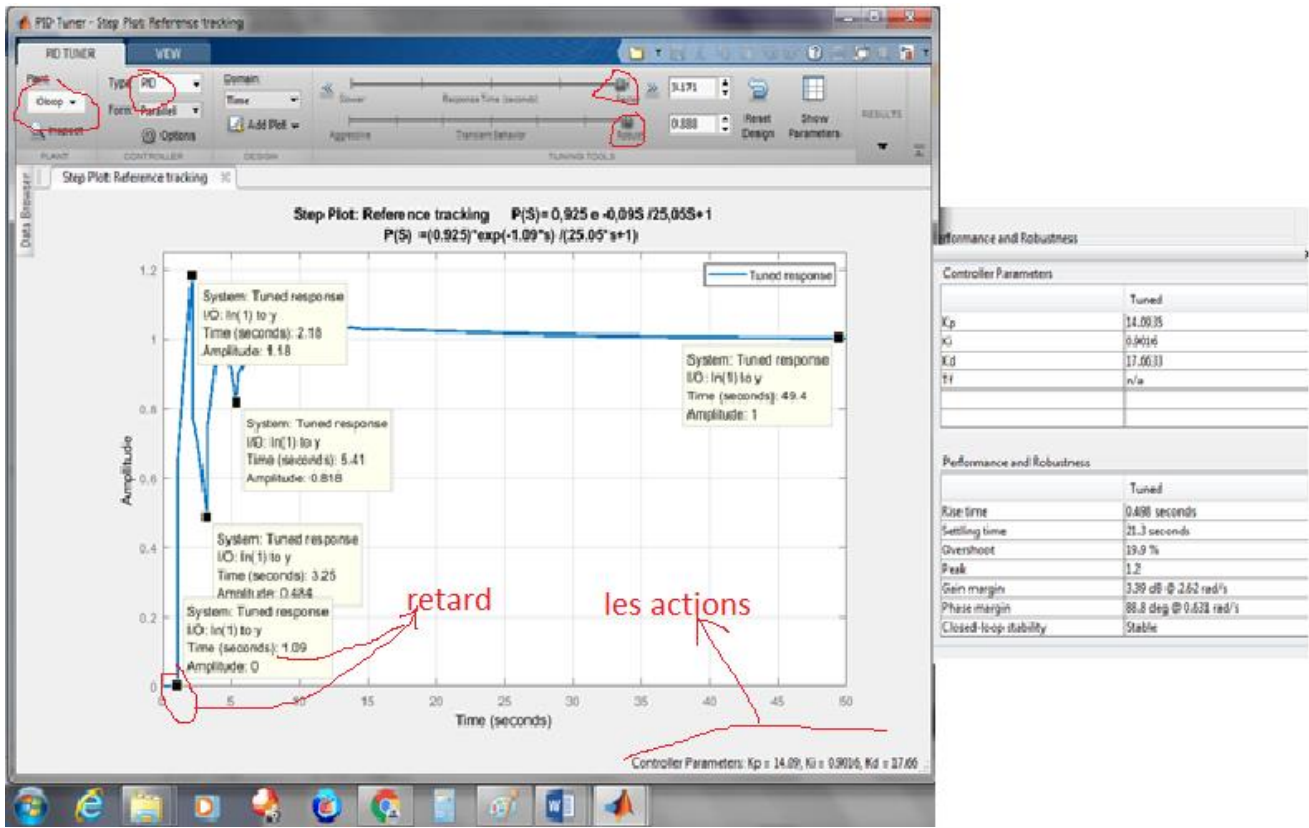
Puis, on cherche le comportement du système face à un signal de saut c'est à dire les performances en boucle ouverte :

- **Step (Oloop)** : la réponse indicielle de la fonction de transfert du système en boucle ouverte.

En exécutant le programme ci –dessous on aura une interface avec des valeurs arbitraires puis avec la possibilité d'ajuster les actions du régulateur (figure 15).

**code 02:**

$s=tf('s');$  Oloop =  $tf(0.925) * \exp(-1.09*s) / (25.05*s+1)$   
 $\% \text{step(Oloop)}$   
 Pidtool (Oloop)



**Figure (10) :** Réponse indicielle référence tracking (PID Tuner)

**Interpretation:**

- La réponse nous confirme Le retard de 1,09 seconde pour amplitude 0,00 voir la légende et l'indication en rouge
- Le système présente un grand amortissement ce qui explique sa non linéarité
- On remarque que la présence des grands dépassements de 0,4 jusqu'à +1,2 pour un signal unitaire et il y'a toujours un dépassement

**Conclusion :** Dans cette étape appelée dimensionnement on peut :

- Choisir quel type de régulateur. PI ; PD ; PID
- Faire varier les actions.
- Décider sur le type (Série ou parallèle)

Prenons notre cas :

$$P(s) = \frac{0.925}{25.05s + 1} e^{-1.09s} \quad (19)$$

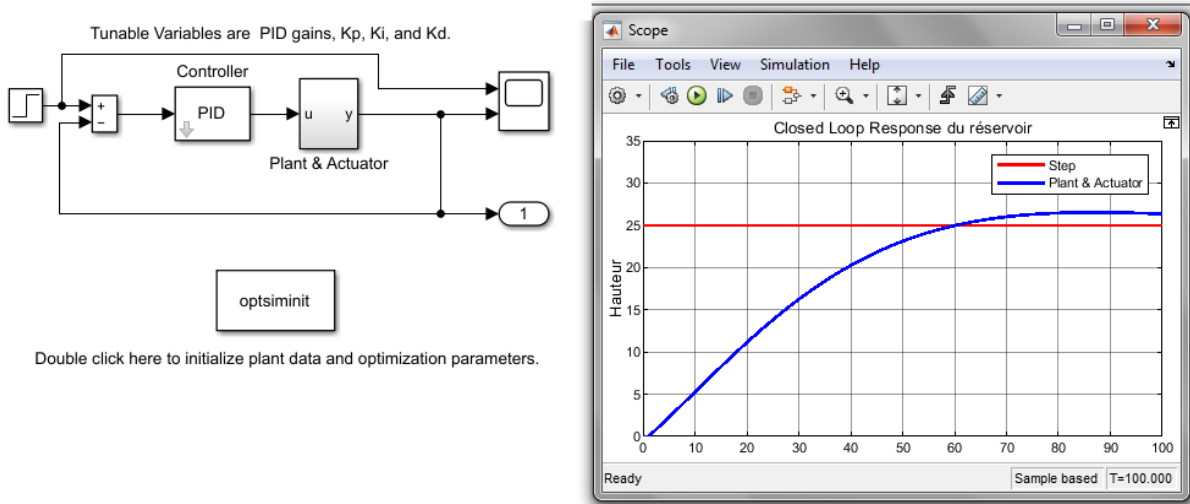
Gain statique :  $K=0,925$   
Retard :  $T=1,09s$   
Constante du temps :  $\tau=25,05$

### 2.2.3. Optimisation en boucle fermée (Via simulink par la commande optsim) :

>> optsim en boucle fermée :

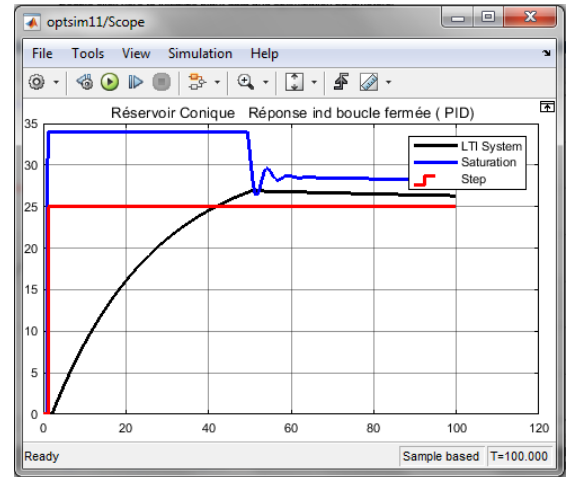
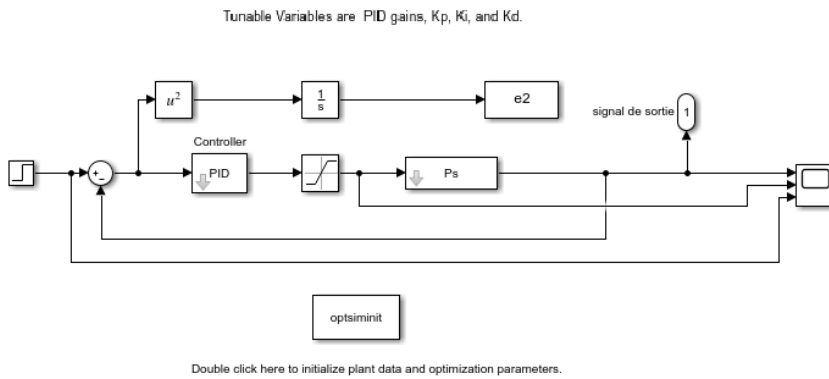
Réponse pour un échelon de 25 (valeur nominale prise pour les calculs : hauteur max du réservoir =30)

#### Méthode 1 :



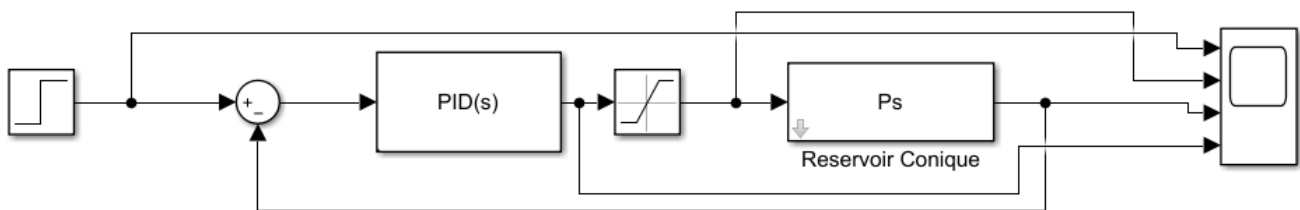
**Figure (11) :** réponse indicielle en boucle fermée

## Méthode 2 : PID avec saturation et extension

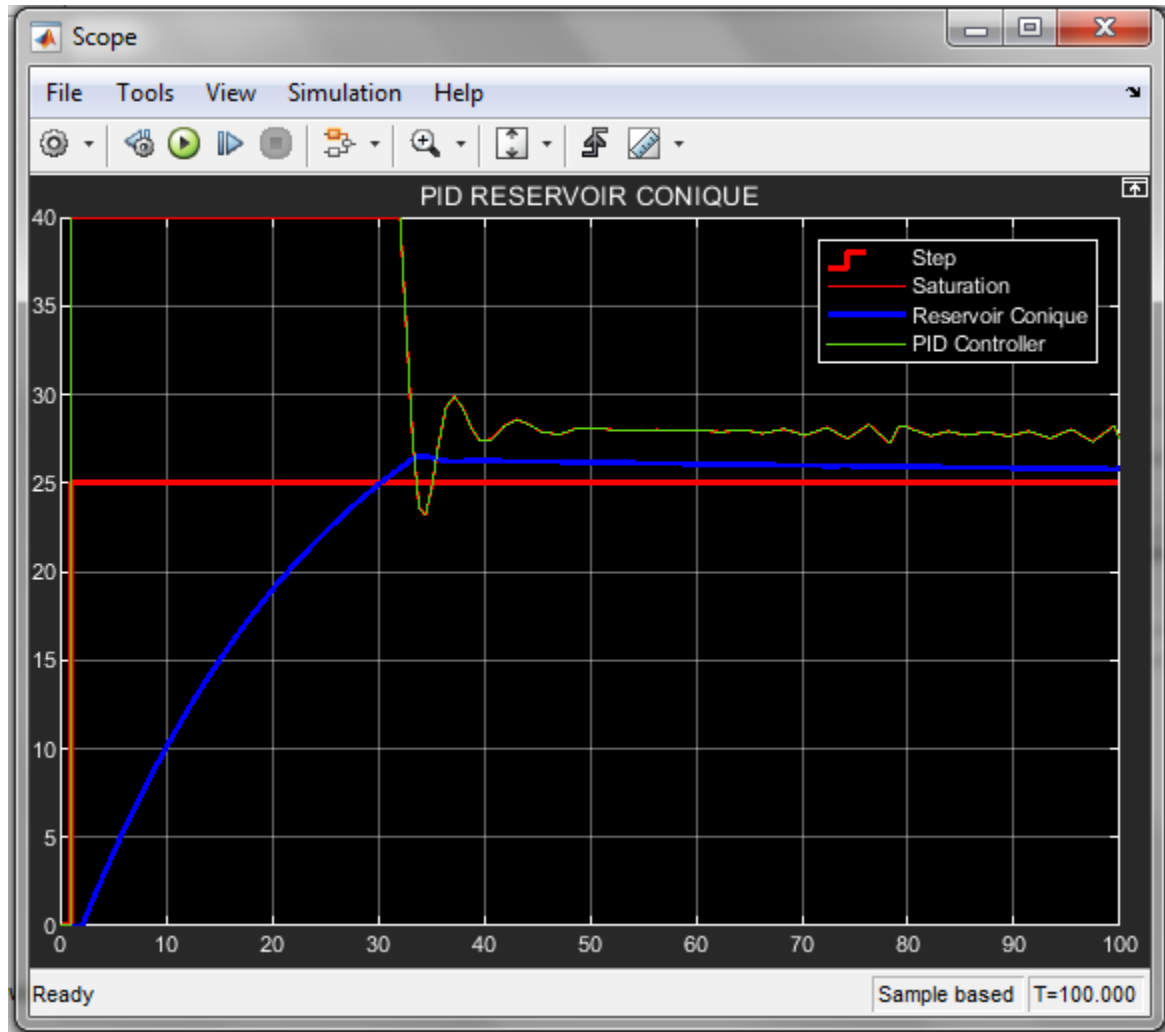


## 3. Implantation des régulateurs

### 3.1. Contrôleur proportionnel intégral dérivée (PID) :

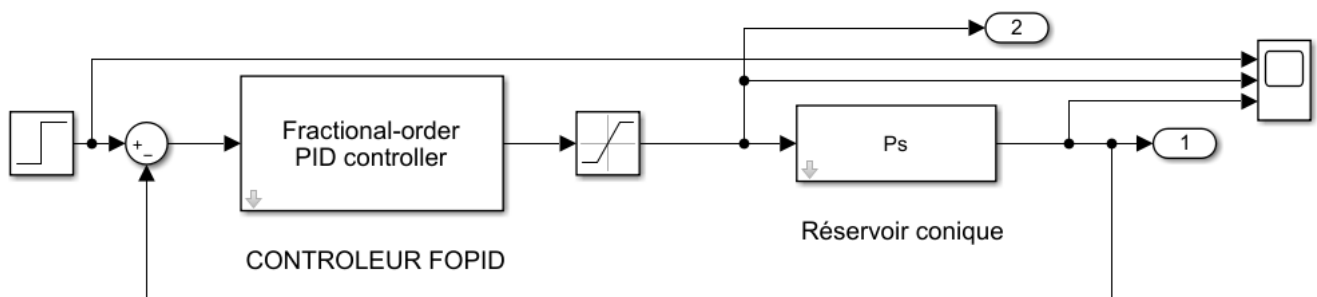


**Figure (12) :** schéma du contrôleur PID

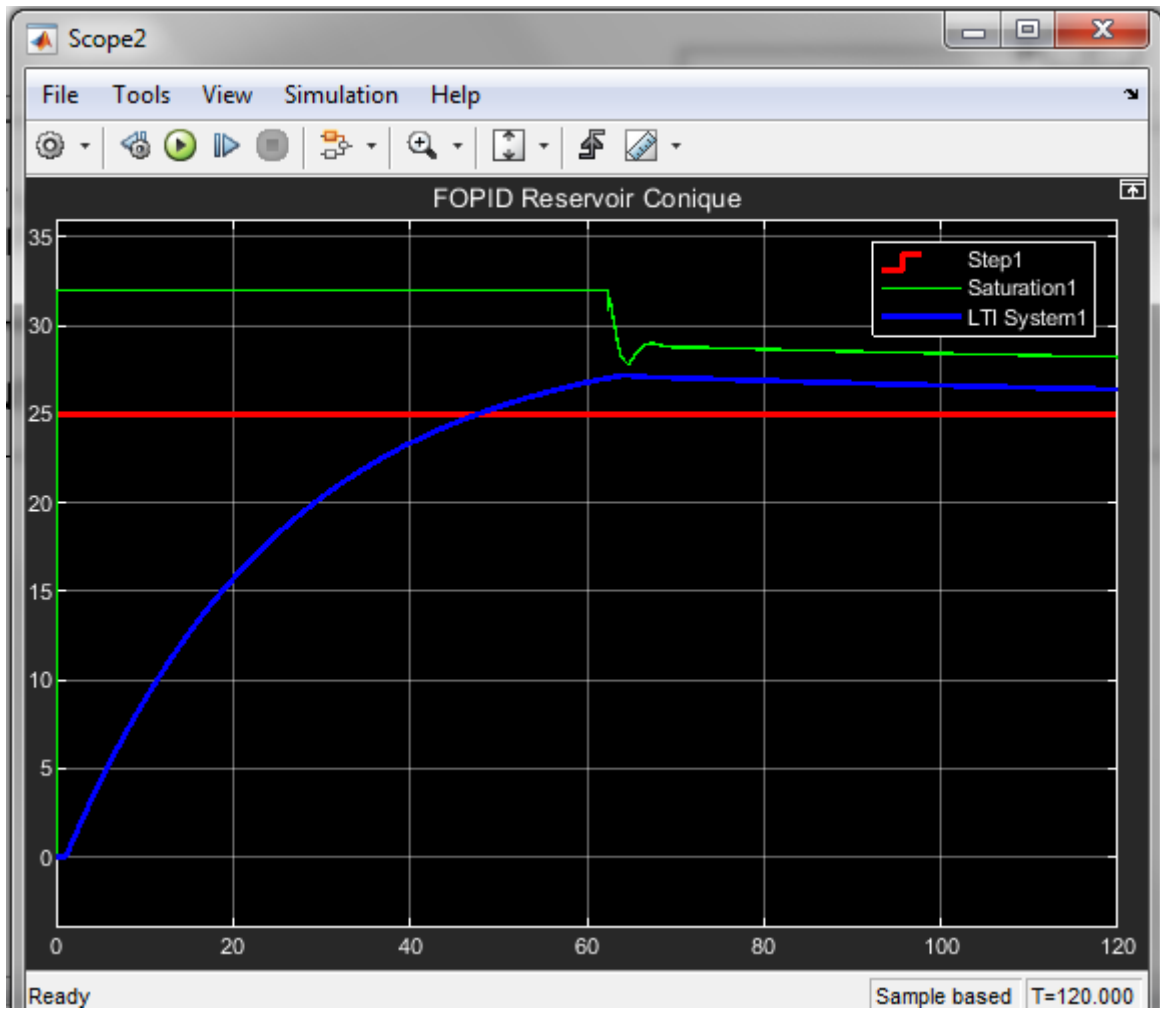


**Figure (13) :** Réponse indicielle du correcteur PID

### 3.3. Contrôleur Fractionnaire Ordre Proportionnel Intégral Dérivée (FOPID) :



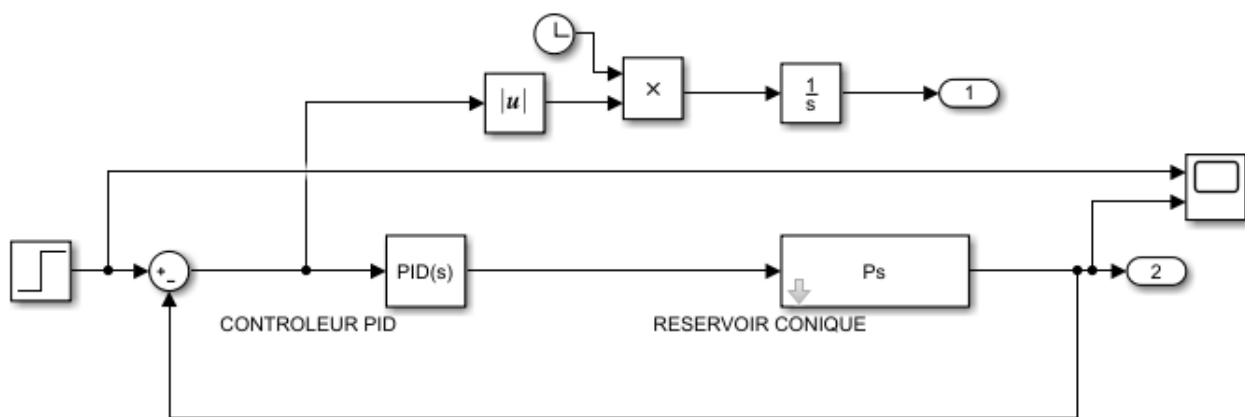
**Figure (14) :** schéma d'un contrôleur FOPID



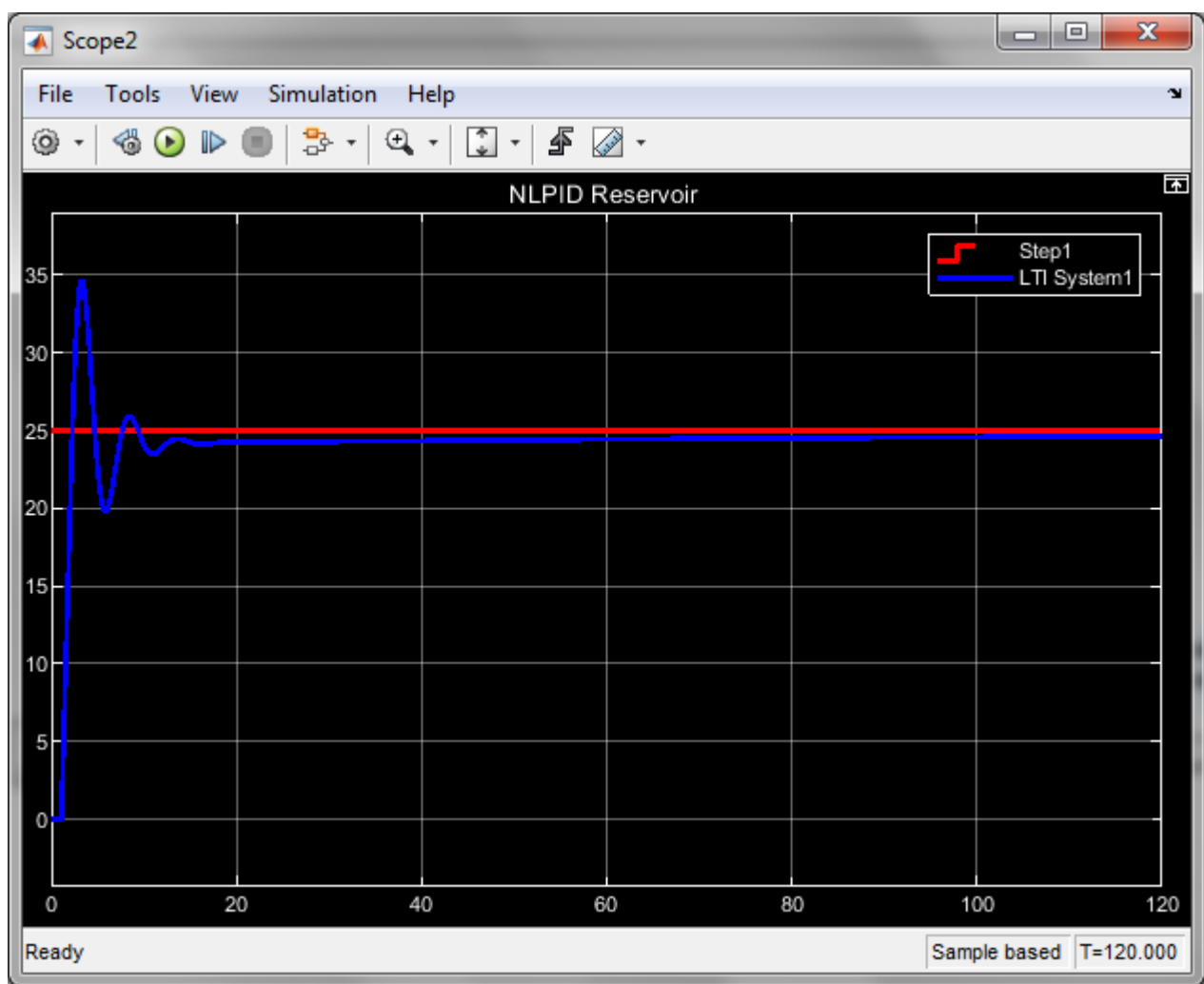
**Figure (15) :** Réponse indicielle du correcteur FOPID



### 3.4. Contrôleur Non Linéaire Propositionnel Intégral dérivée (NL-PID) :



**Figure (16) :** schéma d'un contrôleur NL-PID



**Figure (17) :** Réponse indicielle du correcteur NL-PID

## 4. Algorithme PSO pour l'ajustement des régulateurs : [7][8][9][10]

### 4.1. Introduction :

Dans cette on donne le principe général de cet algorithme ainsi que son application dans la détermination des paramètres d'un régulateur

L'optimisation par essaim de particules (PSO) est une méthode développée à partir du travail en groupe qui est basée sur le comportement du déplacement des oiseaux ou des poissons pour la recherche de la nourriture. Lorsque les oiseaux sont à la recherche de nourriture, d'un endroit à l'autre, il y a toujours un oiseau qui peut sentir la nourriture et trouver l'endroit où elle peut être trouvée. Comme s'il y'a un échange d'informations entres eux à tout moment, ils finiront par affluer vers l'endroit où la nourriture peut être trouvée

### 4.2. Optimisation par essaim de particules PSO :

#### 4.2.1. Formulation générale :

Le PSO est constitué de  $n$  particules

La position de chaque particule représente une solution dans l'espace de recherche.

Les particules changent d'état et se déplacent selon les trois composantes (une composante d'inertie ; une composante cognitive ; une composante sociale)

#### • Les équations de calcul sont définies comme suit :

Le vecteur vitesse est calculé à partir de l'équation suivante :

$$v_{ij}(k) = wv_{ij}(k-1) + c_1r_1(p_{ij}(k-1) - x_{ij}(k-1)) + c_2r_2(g_j(k-1) - x_{ij}(k-1)) \quad (23)$$

La position à l'itération  $k$  de la particule  $i$  est alors définie par :

$$x_{ij}(k) = x_{ij}(k-1) + v_{ij}(k) \quad (24)$$

$i= 1,2,3,\dots,N_P \quad j=1,2,3,\dots,N_d \quad k= 1,2,3,\dots,k_{\max}$

$N_P$  : est le nombre de particules de l'essaim .

$N_d$  : est le nombre de variables (c-à-d dimensions d'une particule) ex PID =3

$k_{\max}$  : est le nombre maximal d'itération

$v_{ij}(k)$  : est la vitesse de la  $j^{\text{ieme}}$  composante de la  $i^{\text{ieme}}$  particule de l'essaim, à la  $k^{\text{ieme}}$  itération

$p_{ij}$  : est la  $j^{\text{ieme}}$  composante de la meilleure position occupée par la  $i^{\text{ieme}}$  particule de

l'essaim enregistrée dans les itérations précédentes ( local best).

$g_i$  : est la  $j^{\text{ème}}$  composante de la meilleure position occupée par la  $i^{\text{ème}}$  particule globale de l'essaim ( global best).

$x_{ij}(k)$  : est la  $j^{\text{ème}}$  coordonnée de la position actuelle de la particule  $i$  à la  $k^{\text{ème}}$  itération.

- **Les coefficients et variables :**

$\omega$  est un coefficient d'inertie ;il peut être variable

$C1$  et  $c2$  sont deux coefficients d'accélération, définies par la relation  $c1 + c2 \leq 4$  qui sont fixés par l'utilisateur ; mais généralement égale à 2.

$R1$  et  $r2$  sont deux nombres aléatoires tirés uniformément dans  $[0,1]$  à chaque itération

### **4.3. Déroulement de l'algorithme**

Considérons le problème de minimisation :

$$\begin{cases} \text{Min } f(x) \\ x \in D \end{cases} \quad (27)$$

La fonction fitness associée est :

$$\text{fitness}(x_j) = f(x_j) \quad (28)$$

Soit  $(x)$  la fonction objective à optimiser (fitness) et  $n$  le nombre de particules.

Les étapes essentielles de l'optimisation par essaim de particules sont présentées par l'algorithme suivant :

1- Initialisation aléatoire de la population et des vitesses des particules.

2- Trouver la meilleure valeur de fitness  $f$  à  $k=0$ .

3- Traitement

Répéter jusqu'à fin d'itération.

Répéter pour chaque particule.

Générer la nouvelle valeur de la vitesse en utilisant l'équation (23).

Calculer la nouvelle position en utilisant l'équation (24).

Evaluation de la valeur de fitness.

Trouver la meilleure position pour chaque particule.

Fin.

Trouver la meilleure position globale Fin

## 5. Optimisation des gains des contrôleurs par la technique PSO

### Introduction

Les contrôleurs basés sur le domaine temporel visent à minimiser les indices de performance

L'algorithme PSO en se basant sur la minimisation du critère est utilisé.

- Pour déterminer les paramètres optimaux du contrôleur ( $K_p, K_i, K_d \dots$  etc).

- Et les paramètres du domaine temporel suivants :

(Le dépassement de crête ( $M_p$ ) ; le temps de montée ( $t_r$ ) ; le temps de stabilisation ( $t_s$ ) ; l'erreur d'état stable (Ess) Steady-state error) de la réponse du système .

On doit avoir :

1. L'équation de commande  $U_c$  à laquelle le régulateur est synthétisé

2. On choisit un ou plusieurs on peut même combiner plusieurs indices :

Les indices couramment utilisés sont des entiers du carré de l'erreur (ISE), l'intégrale de la valeur absolue de l'erreur (IAE), et l'intégral de la valeur absolue de l'erreur (ITAE). Voir les formules ...

**L'erreur  $\varepsilon(t)=y(t)-u(t)$ ; où  $y(t)$ : signal de sortie,  $u(t)$ : signal d'entrée**

Donc la fonction à optimiser est donnée par :

$$\text{L'équation : } \varepsilon(t) = L_{\text{ref}} - L_{\text{mes}} \quad (29)$$

**$L_{\text{ref}}$ = échelon** : Niveau de référence

**$L_{\text{mes}}$**  niveau mesuré

L'échelon unité est la fonction  $u(t)$  telle que :

$$u(t) = 0 \text{ pour } t < 0 \text{ et } u(t) = 1 \text{ pour } t > \text{ ou } = 0.$$

On a alors :  $u(t) \rightarrow U(S) = 1/S$  Compte tenu de la linéarité de la transformée de Laplace, tout échelon (non unitaire), d'amplitude  $A$ , aura pour transformée de Laplace :  $f(t) = Au(t) \rightarrow F(S) = A/S$

$L_{\text{ref}} = A/S$  fonction de transfert échelon de valeur nominale  $A$

$A=25$  est utilisée pour avoir le maximum d'information et pas faire de trop plein (valeur maximale donnée c'est 30)

$\varepsilon(S)$  dans l'indice choisi( ITAE,ITE,ITS.....)

**Remarque :** Selon le Schéma d'implémentation du contrôleur proposé, on définit tous les paramètres en utilisant

$$\text{L'équation : } \quad \varepsilon (t) = L_{\text{ref}} - L_{\text{mes}} \quad (29) \text{ et l'équation de l'indice choisit}$$

• **La fonction de transfert du système est :**

$$P(s) = \frac{0.925}{25.05s + 1} e^{-1.09s} \quad (19)$$

Possède les caractéristiques suivantes :

Gain statique	: K=0,925
Retard	: L=1,09s
Constante du temps	: T=25,05
Puissance fractionnaire de l'opérateur	: $\alpha=1$

- **Les paramètres du domaine temporel :** (Le dépassement de crête (Mp). Le temps de montée (tr). Le temps de stabilisation (ts) de la réponse du système sont évalués :
  - A l'aide des indices de performances cités dans l'introduction.
  - Ils dépendent des paramètres(actions) du régulateur.

• **La fonction de la vitesse de déplacement de la particule est donnée par :**

$$(k + 1) = \omega v(k) + c1 . r1[ p_{\text{best}i}(k) - x_i(k) ] + c2 . r2[ g_{\text{best}i}(k) - x_i(k) ] \quad (23)$$

• **La fonction de déplacement de la particule :**

$$(k + 1) = x(k) + v_i(k + 1) \quad (24)$$

• **La fonction fitness de la fonction objective :**

C'est cette fonction que l'algorithme d'optimisation va devoir optimiser (trouver un optimum) pour améliorer les performances de la boucle de régulation telles que la précision dynamique, le dépassement, l'erreur statique, ... etc.

On aura donc un problème d'optimisation sous contraintes.

Mathématiquement, on cherche à minimiser

$$f \text{ sur } E ; \text{ c'est à dire, on cherche } x^* \in E \text{ tel que : } f(x^*) = \min f(x)$$

-Dans la conception et l'analyse de systèmes de contrôle les indices de performance sont calculés pour être utilisées comme mesures quantitatives pour évaluer les performances d'un système.

$$O(K_p, K_i, K_d, \lambda, \mu, \dots) = \beta(M_p + E_{ss}) + (1 - \beta)(t_s + t_r) \quad (25)$$

### O (toutes les composantes (variables) de la particule) :

Puisque la fonction objectif (O) doit être minimisée, la fonction de fitness (f) est définie par :

$$f = 1/O \quad O(\text{Variables (paramètres) du régulateur}) = \beta(M_p + E_{ss}) + (1-\beta)(t_s + t_r) \quad (31)$$

#### • La fonction de transfert du type de régulateur choisit selon le cas :

**NB :** Chaque particule contient un certain nombre de composantes

Exemple pour un PID la particule à trois composante  $K_p$  ;  $K_i$  ; et  $K_d$  ; auxquelles les deux équations (la vitesse du déplacement de la particule et de son déplacement pour la mise à jour selon le critère) sont utilisées pour chaque composante.

## 5.1. Contrôleur Proportionnel Intégral Dérivée PID

### 5.1.1. Conception du contrôleur PID :

Un Régulateur PID contrôle l'erreur entre la sortie mesurée du système et la consigne désirée. A partir de cette erreur, un signal de contrôle est calculé pour ajuster les performances du processus.

L'équation différentielle du contrôleur PID est :

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) dt + K_d \cdot \frac{d}{dt} e(t) \quad (32)$$

La superposition de ces trois actions constitue le mécanisme d'ajustement des performances du processus.

La fonction de transfert continue du contrôleur PID est obtenue par la transformation de Laplace suivante :

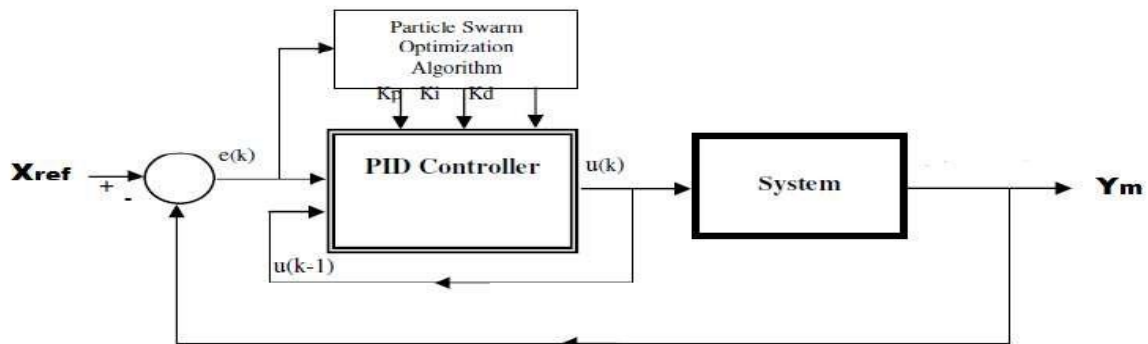
$$(s) = K_p + \frac{k_i}{s} + K_d s \quad (33)$$

À partir des paramètres initiaux choisie aléatoirement, l'algorithme d'optimisation minimise l'indices ITAE, donné par l'équation (5)

Pour chaque itération afin de déterminer un ensemble optimal des paramètres  $K_p$ ,  $K_i$  et  $K_d$ .

### 5.1.2. Ajustement du régulateur PID par l'algorithme PSO :

La structure générale utilisé pour l'auto-détermination des paramètres du régulateur par l'algorithme PSO est représenté sur la figure (16) où le régulateur PID est représenté sur la fig (17).



**Figure (18) :** Bloc diagramme d'implantation des PSO pour l'ajustement du PID.

Les trois paramètres du régulateur PID forme une particule

Chaque particule a sa propre position et vitesse pour son déplacement dans l'espace de recherche

Les particules sont initialisées aléatoirement et sont mises à jour, selon les équations (23)(24)

Algorithme d'apprentissage par PSO

**Les étapes de l'algorithme PSO peuvent être décrites comme suit [7] :**

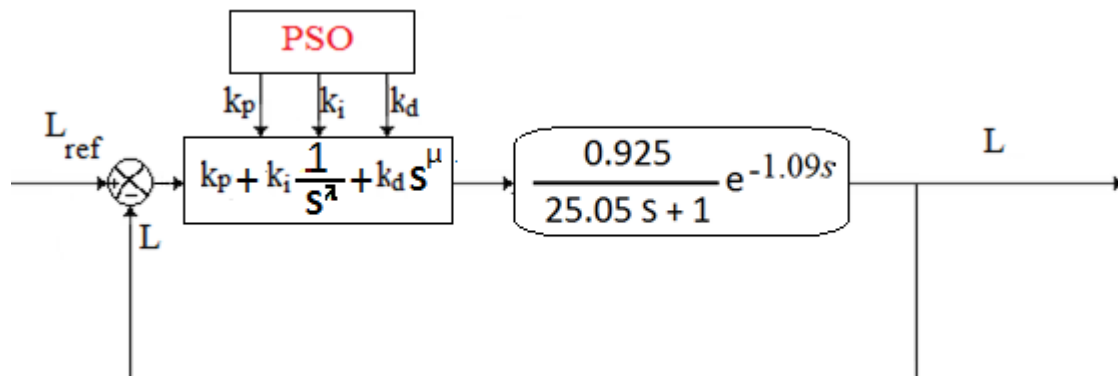
- **Etape 1 :** Initialisation des paramètres  $Kp_m^0$ ,  $Ki_m^0$ ,  $Kd_m^0$  de chaque particule aléatoirement dans la plage autorisée. La quantité  $pbest_m$  est évalué pour chaque particule. La valeur  $gbest$  est la meilleure valeur des  $pbest_m$ .
- **Etape 2 :** La valeur de la fonction objective (fitness) est calculée pour chaque particule en utilisant l'équation Si la valeur est meilleure que le  $pbest$  actuel de la particule, la valeur  $pbest$  est remplacée par la valeur actuelle. Si la meilleure valeur de  $pbest$  est meilleure que la valeur actuelle de  $gbest$ , cette dernière sera remplacée par la meilleure des  $pbest$ .
- **Etape 3 :** Mise à jour des paramètres selon les équations (23A .23. B.23C.24A.24B.24C).
- **Etape 4 :** Si le nombre d'itérations actuel atteint le nombre d'itération maximum prédéterminé, quitter, sinon, passez à l'étape 2

Pour un contrôleur PID ; la fonction objectif (O) qui doit être minimisée est :

$$O(K_p, K_i, K_d) = \beta(M_p + E_{ss}) + (1-\beta)(t_s + t_r) \quad \text{avec } \beta = 0,92 \quad (31)$$

La fonction de fitness (f) est définie par :  $f = 1/O$

## 5.2. Contrôleur d'Ordre Proportionnel Intégral Proportionnel Dérivée FOPID



**Figure (19) :** schéma fonctionnel d'un FOPID avec PSO

La fonction de transfert du contrôleur PID d'ordre fractionnaire G(s) est définie comme en :

$$G_{FOPID}(s) = \frac{U(s)}{E(s)} = k_p + k_i \frac{1}{s^\lambda} + k_d s^\mu, \quad (\lambda, \mu > 0) \quad (08)$$

Pour pénaliser les systèmes dont le régime transitoire dure trop longtemps, on utilise l'intégrale du produit de l'erreur par le temps, donnée par :

$$ITAE = \int_0^T t \cdot |e(t)| dt \quad (05)$$

Pour Le contrôleur PID d'ordre fractionnaire ; on a besoin de régler les cinq paramètres ( $K_p, K_i, K_d, \lambda, \mu$ ) de manière appropriée pour optimiser les performances du système

Évaluation des performances du contrôleur  $PI^\lambda D^\mu$

Pour chaque particule de la population, qui représente cinq paramètres du contrôleur, le système est simulé. Sur la base de la réponse du système, et sont évalués

Les quatre paramètres du domaine temporel dépendent de cinq paramètres du contrôleur ( $K_p, K_i, K_d, \lambda, \mu$ ).



Le système est bon lorsque ces quatre paramètres sont au minimum.

Le critère de performance (O) du régulateur PI<sup>λ</sup>D<sup>μ</sup> est défini par :

$$O(K_p, K_i, K_d, \lambda, \mu) = \beta(M_p + E_{ss}) + (1 - \beta)(t_s + t_r) \quad (32)$$

Puisque la fonction objectif (O) doit être minimisée, la fonction de fitness (f) est définie par :

$$F = 1/O$$

On accorde plus d'importance à la réduction du dépassement de crête (overshoot) et la meilleure valeur pour  $\beta$  est de 0,92.

### 5.3. Contrôleur Non linéaire Proportionnel Intégral Proportionnel Dérivée NL-PID

Un contrôleur PID non linéaire (NL-PID) à deux degrés de liberté est utilisé pour ce processus non linéaire du premier ordre avec un retard.

Les paramètres du contrôleur NPID seront réglés indépendamment et arbitrairement

Une fonction non linéaire  $f(e, \alpha, \delta)$  est pour le NL-PID est définie par :

$$f(e, \alpha, \delta) = \begin{cases} \text{sign}(e) \cdot |e|^\alpha, & \text{lorsque } |e| > \delta \\ \delta^{\alpha-1} e, & \text{lorsque } |e| \leq \delta \end{cases} \quad (33)$$

Ce qui précède est une fonction hautement non linéaire avec les paramètres  $\alpha$  et  $\delta$

$\alpha$  = constante détermine la plage linéaire

$$\text{sign}(A, \delta) = \begin{cases} \text{sign}(A), & \text{when } |A| > 0 \\ \frac{A}{\delta}, & \text{when } |A| \leq \delta, \delta > 0 \end{cases} \quad (34)$$

La loi de contrôle  $U_c$  peut être écrite sous forme de contrôle PID parallèle est :

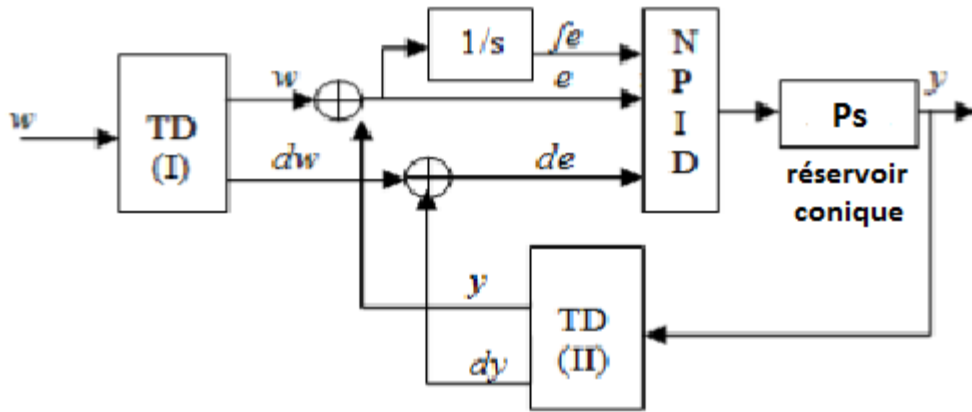
$$U_c = K_p e_f(e_p, \alpha_p, \delta_p) + K_I \int e_I dt + K_d \frac{de_d}{dt} \quad (35)$$

C'est-à-dire que le contrôleur est synthétisé par  $U_c$

$0 < \alpha_p < 1$  le gain doit être inversement proportionnel à l'erreur

$1 \leq \alpha_I < 1$  pour éviter la saturation

$\alpha_d \leq 1$



**Figure (20) :** schéma de structure d'un FOPID

Gain non linéaire  $K(e)$  :

$$K(e) = \frac{\exp(\alpha e) + \exp(-\alpha e)}{2} \quad (09)$$

$$\text{Error, } e = \begin{cases} e & ; \text{ if } |e| \leq e_{\max} \\ e_{\max} * \text{sign}(e) & ; \text{ else } |e| > e_{\max} \end{cases} \quad (10)$$

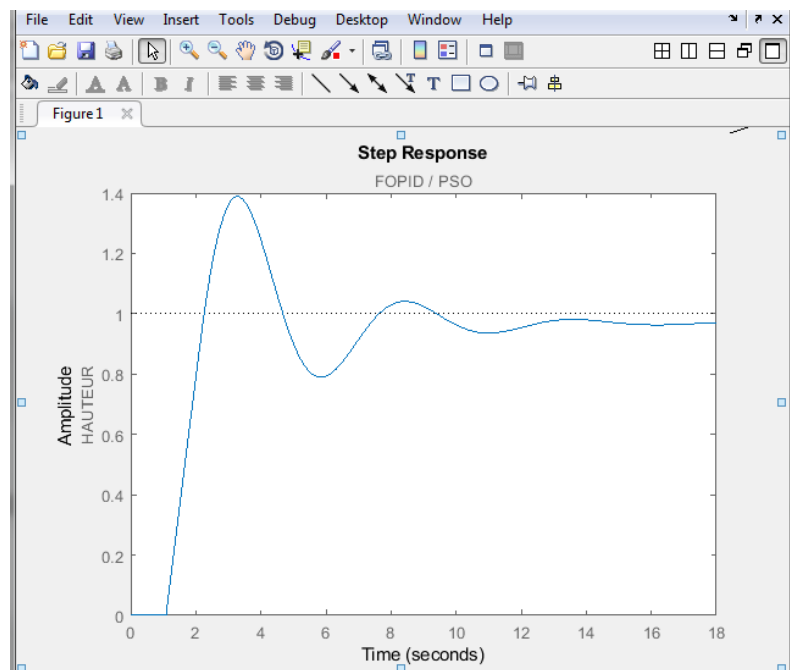
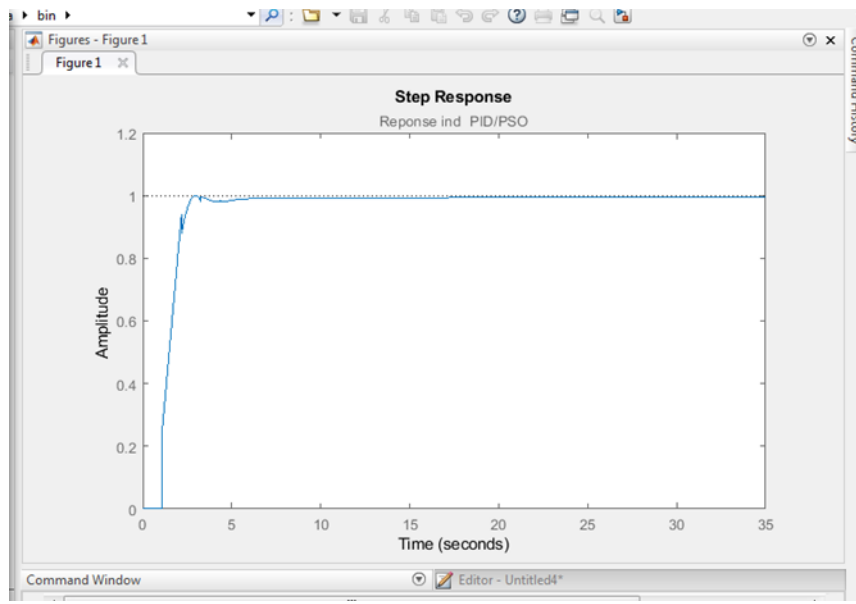
$$K(e_{\max}) = -\frac{1}{|G(j\omega)|} \quad (11)$$

Erreur échelle  $f_e$  :

$$f_e = K(e) * e(t) \quad (12)$$

Fonction de transfert générale du contrôleur NL-PID :

$$G_{\text{NL-PID}}(s) = [K_p * f_e] + [K_i / s * fe] + [K_d s * f_e] \quad (13)$$



## CONCLUSION GENERALE

Le travail présenté dans ce mémoire s'inscrit dans le cadre de l'ajustage des paramètres des contrôleurs PID classiques, à ordre fractionnaire (FOPID) et non linéaire (NLPID) en utilisant la technique du Particle Swarm Optimization (PSO) appliquée à système de bacs coniques.

La première partie a fait l'objet d'un rappel sur les différents contrôleurs en question en présentant leur principe et leurs caractéristiques. On peut remarquer que la complexité des contrôleurs FOPID ET NLPID donne plus de flexibilité sur le choix des paramètres et par conséquent les performances de la régulation sont améliorées.

Ce travail nous a permis de pratiquer des concepts que nous avons étudiés durant notre cursus et d'autres que nous avons découverts dans ce projet.

Notre travail reste primitif et donc sujet à des améliorations, comme par exemple, la comparaison avec d'autres techniques méta heuristiques d'optimisation, son extension à d'autres types de systèmes dynamiques.

Autrement le contrôleur idéal (PID, FOPID, NL-PID, PID floue...) n'existe pas pour vraiment satisfaire toutes les exigences (robustesse, rapidité et précision)

## BIBLIOGRAPHIE

- [1]: J.G.Ziegler, N.B.Nichols\_optimum settings for automatic controllers\_trans.ASME 64 (1942) 759-768.
- [2]: Oustaloup, A., 'La commande CRONE', Hermès, Paris, 1991
- [3]: Podlubny,I., 'Fractional order systems and PID $\mu$ controllers', IEEE Transactions on automatic control, Vol. 44, No. 1, pp 208-214, 1999
- [4]: Fractional Order Control - A Tutorial YangQuan Chen, Ivo Petr'as' and Dingyu' Xue, 2009 American Control Conference Hyatt Regency Riverfront, St. Louis, MO, USA June 10-12, 2009
- [5]: Design and PLC Implementation of Nonlinear PID Controllers for Control of Nonlinear Processes, NOMZAMO TSHEMESE
- [6]: *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*  
e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 9, Issue 3 Ver. III (May – Jun. 2014), PP 59-64  
[www.iosrjournals.org](http://www.iosrjournals.org)
- [7]: A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System  
Zwe-Lee Gaing, Member, IEEE TRANSACTIONS ON ENERGY CONVERSION, VOL. 19, NO. 2, JUNE 2004

[8]: Gases Brownian Motion Optimization: an Algorithm for Optimization (GBMO), Marjan Abdechiri a,\*, Mohammad Reza Meybodi b, Helena Bahrami a a Electronic, Computer & IT Department, Qazvin Azad University, Qazvin, Iran b Computer Engineering and Information Technology Department, Amirkabir University of Technology, Tehran, Iran, 2021

[9]: Particle Swarm Optimization (PSO) Based Turbine Control, Faculty of Engineering and Applied Science, University of Ontario Institute of Technology, Oshawa, Canada 2 Faculty of Energy Systems and Nuclear Science, University of Ontario Institute of Technology, Oshawa, Canada

## ANNEXES

Le code génère des valeurs aléatoires puis adapte les valeurs des gains pour que la correction se fasse, et en grâce à un apprentissage sur plusieurs générations prenant le retour du système

Niveau de réservoir conique :  $H_{\max} = 30\text{cm}$

Saturation 0 : 34 pour éviter le bruit

Scope 0 : 35

$y = 0 : 120$

$x$  (hauteur) = 0 : 35

Ces valeurs données dans Simulink pour avoir le maximum d'information et éviter le débordement de réservoir

---

```
%% Reservoir conique PSO Controleur PID
% Initialization
% Parameters
close all
clear all
clc
iterations = 100;
W = 0.9;
C1 = 2.0;
C2 = 2.0;
n = 81;

% ---- initial swarm position ----
index = 1;
for i = 1 : 9
    for j = 1 : 9
        for k = 1 : 9
            particle(index, 1, 1) = i;
            particle(index, 1, 2) = j;
            particle(index, 1, 3) = k;
            index = index + 1;
        end
    end
end
particle(:, 4, 1) = 1000;           % best value so far
particle(:, 2, :) = 0;           % initial velocity
```

```

%% Iterations
for iter = 1 : iterations
    %-- evaluating position & quality ---
    for i = 1 : n
        particle(i, 1, 1) = particle(i, 1, 1) + particle(i, 2, 1)/1.3;    %update x position
        particle(i, 1, 2) = particle(i, 1, 2) + particle(i, 2, 2)/1.3;    %update y position
        particle(i, 1, 3) = particle(i, 1, 3) + particle(i, 2, 3)/1.3;    %update z position

        % Parametres du Regulateur
        Kp = particle(i, 1, 1);
        Ki = particle(i, 1, 2);
        Kd = particle(i, 1, 3);
        Gc=tf([Kd Kp Ki],[1 0]);

        %% Transfer Function of PID
        s=tf('s');
        Ps=(0.925)*exp(-1.09*s) / (25.05*s+1);
        Q=feedback(Gc*Ps,1);
        Z=stepinfo(Q);
        Tr=Z.RiseTime;
        Ts=Z.SettlingTime;
        % Z.Overshoot=350;

        Mp=Z.Overshoot;
        Ess=Z.Undershoot;
        % e=Z.SettlingMax;
        %beta= 0.92;

        O= 0.92*(Mp+Ess)+0.08*(Ts+Tr)
    end
end

```

```

O= 0.92*(Mp+Ess)+0.08*(Ts+Tr)
F= 1/O % fitness evaluation

if O < particle(i, 4, 1) % if new position is better
particle(i, 3, 1) = particle(i, 1, 1); % update best x,
particle(i, 3, 2) = particle(i, 1, 2); % best y positions
particle(i, 3, 3) = particle(i, 1, 3); % best z positions
particle(i, 4, 1) = O; % and best value

end
end
[niveau, gbest] = min(particle(:, 4, 1)); % global best position
%--- updating velocity vectors
for i = 1 : n
particle(i, 2, 1) = rand*W*particle(i, 2, 1) + C1*rand*(particle(i, 3, 1)
- particle(i, 1, 1)) + C2*rand*(particle(gbest, 3, 1) - particle(i, 1, 1));
%x velocity component
particle(i, 2, 2) = rand*W*particle(i, 2, 2) + C1*rand*(particle(i, 3, 2)
- particle(i, 1, 2)) + C2*rand*(particle(gbest, 3, 2) - particle(i, 1, 2));
%y velocity component
particle(i, 2, 3) = rand*W*particle(i, 2, 3) + C1*rand*(particle(i, 3, 3)
- particle(i, 1, 3)) + C2*rand*(particle(gbest, 3, 3) - particle(i, 1, 3));
%z velocity component
end
%% Plotting the swarm
clf
plot(particle(:, 1, 1), particle(:, 1, 2), 'x')
axis([-2 150 -2 150]);
pause(0)
end

```

---

**Code FOPID /PSO avec courbe et workspace des parametres**



```

%% Reservoir conique PSO Controleur FOPID
% Initialization
% Parameters
close all
clear all
clc
iterations = 100;
W = 0.9;
C1 = 2.0;
C2 = 2.0;
n = 81;

% ---- initial swarm position ----
index = 1;
for i = 1 : 9
    for j = 1 : 9

        particle(index, 1, 1) = i;
        particle(index, 1, 2) = j;

        index = index + 1;
    end
end

particle(:, 4, 1) = 1000;           % best value so far
particle(:, 2, :) = 0;             % initial velocity

```

```

%% Iterations
for iter = 1 : iterations
    %-- evaluating position & quality ---
    for i = 1 : n
        particle(i, 1, 1) = particle(i, 1, 1) + particle(i, 2, 1)/1.3;    %update x position
        particle(i, 1, 2) = particle(i, 1, 2) + particle(i, 2, 2)/1.3;    %update y position

        % Parametres du Regulateur
        Kp = 23.60;
        Ki = 0.19;
        Kd = 0.17;
        lamda= particle(i, 1, 1);
        omega= particle(i, 1, 2);

        s=tf('s');
        NumGc=Kd*s^(lamda+omega)+Kp*s^lamda+Ki;
        DenGc=s^lamda ;
        Gc=tf(NumGc/DenGc);
        s=tf('s')
        Ps=(0.925)*exp(-1.09*s) / (25.05*s+1)
        Q=feedback(Gc*Ps,1);

        Z=stepinfo(Q);
        Tr=Z.RiseTime;
        Ts=Z.SettlingTime;
        % Z.Overshoot=350;
        step(Q)
        Mp=Z.Overshoot;
        Ess=Z.Undershoot;
        % e=Z.SettlingMax;
        %beta= 0.92;
    end
end

```

```

O= 0.92*(Mp+Ess)+0.08*(Ts+Tr)
F= 1/O % fitness evaluation

if O < particle(i, 4, 1) % if new position is better
particle(i, 3, 1) = particle(i, 1, 1); % update best x,
particle(i, 3, 2) = particle(i, 1, 2); % best y postions

particle(i, 4, 1) = O; % and best value

end
-end
[niveau, gbest] = min(particle(:, 4, 1)); % global best position
%--- updating velocity vectors
]for i = 1 : n
particle(i, 2, 1) = rand*W*particle(i, 2, 1) + C1*rand*(particle(i, 3, 1)
- particle(i, 1, 1)) + C2*rand*(particle(gbest, 3, 1) - particle(i, 1, 1));
%x velocity component
particle(i, 2, 2) = rand*W*particle(i, 2, 2) + C1*rand*(particle(i, 3, 2)
- particle(i, 1, 2)) + C2*rand*(particle(gbest, 3, 2) - particle(i, 1, 2));
%y velocity component

-end
%% Plotting the swarm
clf
plot(particle(:, 1, 1), particle(:, 1, 2), 'x')
axis([-2 150 -2 150]);
pause(0)
-end

```