



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : SIOD /M2/2020

Mémoire

présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Système d'Information Optimisation Décision**

**Entraînement d'un Contrôleur PID
Par les Algorithmes Génétiques**

**Par :
SADOUKI SADIKA**

ABABSA Tarek

Rapporteur

Remerciement

Tout d'abord, je tiens à remercier Allah, le clément et le miséricordieux de m'avoir donné la force, le courage et la patience pour mener à bien ce modeste travail.

Je tiens à exprimer mes profondes gratitudee et remerciements à mon promoteur monsieur ABABSA TAREK pour sa présence, sa disponibilité à tout instant et surtout ses orientations, qu'il trouve ici le témoignage de mon haute considération et de mon profond respect.

Sans son aide, ce travail n'aurait pas abouti. Ces quelques mots sont le reflet de mes sincères reconnaissances.

Mes remerciements s'adressent également, aux membres de jury, qui me font l'honneur d'évaluer mon modeste travail.

Je remercie également l'ensemble des enseignants du département de l'INFORMATIQUE qui ont contribué à ma formation dans les meilleures conditions.

SADIKA

Je dédie ce travail à tous ceux qui me
sont chers et proches :

à la puissante femme que j'aie jamais vue

« ma mère »

à Mon mari et mes enfants

à mes chers frères et ma sœur

à tous les amis

à tous ceux que j'ai oublié de citer.

sadika

ملخص :

الهدف من هذا العمل هو تحسين معلمات المنظم من النوع النسبي التكاملي المشتق عن طريق الخوارزميات الجينية. تم تطبيق هذا النهج على نظام تنظيم درجة الحرارة. تضبط الخوارزمية الجينية معلمات المنظم عن طريق تقليل التكامل التام للخطأ .

الهدف الرئيسي هو تحسين الخصائص الرئيسية الثلاثة للنظام: السرعة والدقة والاستقرار. تترجم هذه العناصر بسرعة حقيقية قريبة جداً من تلك المطلوبة والتشغيل بالسرعة المحددة دون التذبذبات (تسارع ، تباطؤ ...).

الكلمات المفتاحية: الخوارزمية الجينية، المنظم النسبي التكاملي المشتق .

Résumé :

L'objectif de ce travail est l'optimisation des paramètres d'un contrôleur PID par les algorithmes génétiques. Cette approche est appliquée sur un système de régulation de la température.

L'algorithme génétique ajuste les paramètres du PID en minimisant l'intégrale de l'erreur absolue pondérée par le temps (ITAE) . L'objectif principal est d'améliorer les trois principales caractéristiques d'un système : la rapidité, la précision et la stabilité. Ces éléments se traduisent par une vitesse réelle très proche de celle demandée et un fonctionnement à la vitesse de consigne sans oscillations (accélérations, décélération...).

Mots Clés : Algorithme génétique ; Contrôleur PID.

Abstract:

The objective of this work is the optimization of the parameters of a PID controller by genetic algorithm. This approach is applied on a temperature regulation system.

The Genetic algorithm adjusts PID parameter for the minimum integral of the integral time absolute error (ITAE). The main objective is to improve the three main characteristics of a system: speed, precision and stability. These elements translate with; a real speed very close to that requested and an operation at the set speed without oscillations (acceleration, deceleration...)

Key words: Genetic algorithm; PID controller.

Table de matières

INTRODUCTION GENERALE.....	01
<i>Chapitre I</i>	
L' AUTOMATISME ET L' ASSERVISSEMENT.....	04
I.1. Introduction	05
I.2. Automatismes	06
I.2.1. Définition de l'automatisme	06
I.2.2. Principes de base	06
I.2.2.1. Notion de système, de Boucle Ouverte, de Boucle Fermée	06
I.2.2.2. Nécessité de la boucle fermée	07
I.3. l' Asservissement	08
I.3.1. Définition de l' asservissement	08
I.3.2. Objectif de l' asservissement.....	08
I.3.3. Modélisation d' un système	09
I.3.3.1. Modèle de connaissance	09
I.3.3.2. Modèle comportement	09
I.3.3.3. Modèle intermédiaire	09
I.3.3.4. Modèles utilisés en automatique	09
I.3.3.5. Modèles de calcul et de validation	10
I.3.3.6. Modèles d' état et entrée/sortie	10
I.3.4. Schéma de principe d' un système asservi	10
I.3.4.1. Système de commande en boucle ouverte	10
I.3.4.2. Système de commande en boucle fermée	11
I.3.5. Les caractéristiques d' un système asservi	12
I.4. Conclusion	13

Chapitre II

LE CONTROLEUR PID	14
II.1. Introduction	15
II.2. Contrôleur PID	15
II.2.1 Définition	15
II.2.2 Action proportionnelle (P)	16
II.2.3 Action Intégrale (I)	17
II.2.4 Action Dérivé (D)	17
II.3. Les différentes structures d'un contrôleur PID.....	19
II.3. La forme PID série	19
II.3. La forme PID parallèle	19
II.3. La forme PID standard	20
II.4. Conclusion	21

Chapitre III

LES ALGORITHMES GENETIQUES	22
III.1. Introduction	23
III.2. Définition	24
III.3. Terminologie :.....	24
III.3. 1. Individu/chromosome/séquence.....	24
III.3. 2. Population.....	24
III.3.3. Une génération.....	24
III.3. 4. Environnement.....	24
III.3. 5. Les gènes.....	24
III.3. 6. Fonction d'évaluation (fitness).....	24
III.3. 7. Les chromosomes.....	24

III.4. Le principe des algorithmes génétiques.....	25
III.5. Les Operations D’algorithme Génétique.....	26
III.5. 1. La Sélection.....	26
III.5. 1.1. La Roulette.....	26
III.5. 1.2. Le Tournoi.....	26
III.5. 2. Le Croisement.....	27
III.5. 3. La Mutation.....	27
III.5. 4. Le Codage.....	28
III.5. 4.1.Codage binaire.....	29
III.5. 4.2.Codage réel.....	29
III.5.5 Fonction d’évaluation (fitness)	29
III.6. Conclusion :.....	30

Chapitre VI

LA CONCEPTION	31
IV.1. Introduction	32
IV.2. Description du système	33
IV.2.1. Schéma fonctionnel	33
IV.2.2. Eléments constitutifs du système	34
IV.2.2.1. Thermocouple type K.....	34
IV.2.2.1.1. Avantages dans l’utilisation d’un thermocouple.....	34
IV.2.2.1.2. Désignations correspondants au thermocouple utilisé.....	34
IV.2.2.2. Calculateur.....	35
IV.2.2.3. Une plaque métallique.....	36
IV.2.2.4. Collier chauffant	36
IV.2.2.5. Relais 220V commande à 5V	36

IV.2.2.6. Command On\Off	37
IV.2.2.7. VCC 5v	37
IV.3. Fonctionnement du système.....	37
IV.3. 1. Mesure de température	37
IV.3. 2.chauffage	38
IV.3. 3. Contrôle de température	38
IV.4. Conclusion	39
 Chapitre V	
IMPLEMENTATION ET RESULTATS	40
V.1. Introduction	41
V.2 Adaptation d'AG à l'optimisation du régulateur PID pour la commande de Température.....	41
V.2.1 Le codage	42
V.2.1 Initialisation.....	44
V.2.3 Evaluation de la population.....	44
V.2.4 La Sélection	44
V.2.5 Le croisement	45
V.2.6 La mutation.....	46
V.2.7 Elitisme.....	46
V.3 Environnement de travail	48
V.3.1 Environnement matériel	48
V.3.2 Environnement logiciel	48
V.3.2.1 le langage de programmation.....	48
V.3.2.2 Outil de développement	49
V.4 Développement de l'application.....	49
V.4.1 Implémentation de l'applet.....	49

V.4.1.1 La classe Gene	49
V.4.1.2 La classe Individu	50
V.4.1.3 La classe Population	50
V.4.1.4 La classe PID	51
V.4.2 Structures de données	51
V.5 Résultats	52
V.6 Conclusion.....	55
CONCLUSION GENERALE.....	57

Liste des Figures

Figure I.1 : Schéma d'un système en boucle ouverte	07
Figure I.2: Schéma d'un système en boucle fermée	07
Figure I.3 : Système de commande en boucle ouverte	10
Figure I.4 : Système de commande en boucle fermée	11
Figure I.5: Schéma de principe d'un système asservi	12
Figure II.1 : Modélisation sous Maple de la réponse à un échelon dans un asservissement en position.....	16
Figure II.2 : Modélisation sous Maple de la réponse à un échelon dans un asservissement en vitesse.....	17
Figure II.3 : Modélisation sous Maple de la réponse à un échelon dans un asservissement en vitesse.....	18
Figure II.4 : Le contrôleur PID Série.....	19
Figure II.5 : Le contrôleur PID Parallèle.....	19
Figure II.6 : Le contrôleur PID Standard, Mixte ou Idéal	20
Figure III.1 : organigramme de fonctionnement d'un algorithme génétique	25
Figure III.2 : La roulette	26
Figure III.3 : Le tournoi entre deux individus	27
Figure III.4 : Le croisement	27
Figure III.5 : Illustration de la Mutation	28
Figure III.6 : Les cinq niveaux d'organisation d'un algorithme génétique.....	28
Figure IV.1 : Asservissement PID.....	32
Figure IV.2 : Système de contrôle de température.....	33
Figure IV.3 : Capteur de température (Thermocouple type K).....	34
Figure IV.4 : Microcontrôleur.....	35
Figure IV.5 : Fonctionnement d'un microcontrôleur.....	35
Figure IV.6 : les colliers chauffants.....	36
Figure IV.7 : les Relais.....	36
Figure IV.8 : l'Effet Seebeck.....	37
Figure V.1 : Principe de l'optimisation d'un contrôleur PID par algorithme génétique.....	41
Figure V.2 : représentation binaire d'un chromosome.....	42
Figure V.3 : Représentation d'un individu (codage binaire).....	43
Figure V.4 : Sélection par tournoi.....	45

Figure V.5: logo java.....	48
Figure V.6 : logo éclipse.....	49
Figure V.7 : variation de température chaque 5 seconde.....	52
Figure V.8 : Erreur produite par le système	54

Listes des Acronymes et Symboles

Acronymes

BO : Boucle Ouverte

BF : Boucle Fermée

PID : contrôleur proportionnelintégraldérivé

AGs : Algorithmes Génétiques

Max : maximum

Min : minimum

VCC : Common Collector Voltage (Alimentation Tension Continue)

ITAE : Integral Time Absolute Error (Intégrale de l'Erreur Absolue pondérée par le Temps)

Symboles

P : proportionnel

I : intégral

D : dérivé

e : erreur

K_p : gain proportionnel

K_i : gain Intégral

K_d : gain dérivé

t : le temps

f : fonction

h : fonction

x : variable

i : indice

E_{opt} : ensemble des paramètres optimisés

k_{Popt} : paramètre proportionnel optimisé

k_{Iopt} : paramètre intégral optimisé

k_{Dopt} : paramètre dérivé optimisé

E : espace de recherche

J_{min} : borne minimale de l'espace de recherche

J_{max} : borne maximale de l'espace de recherche

L : longueur

f_{ad} : fonction adaptative

f_{obj} : fonction objective

ΔV : variation de la tension

S : coefficient de Seebeck

ΔT : variation de la température.

C : valeur contrôlé

Liste des Tableaux :

Tableau II.1. Caractéristiques du contrôleur PID	18
Tableau IV.1: caractéristique de thermocouple type K.....	34
Tableau V.1 : Représentation du gène (K_p) de dix individus.....	42
Tableau V.2 : Représentation du gène (K_i) de dix individus.....	42
Tableau V.3 : Représentation du gène (K_d) de dix individus.....	43
Tableau V.4 : Représentation de dix individus (en nombres réels).....	43
Tableau V.5: paramètre de l'AGs	47
Tableau (V.6): la température obtenue	52
Tableau (V.7): Erreur produite par le système	53

Liste des Algorithmes :

Algorithme V.1: Générer_individus_aléatoires.....	44
Algorithme V.2 : Sélection par tournoi binaire.....	45
Algorithme V.3: Croisement.....	45
Algorithme V.4: Mutation.....	46
Algorithme V.5: Optimisation d'un PID par algorithme génétique simple.....	47
Algorithme V.6: calcul de la Commande.....	51

INTRODUCTION GENERALE

L'automatisme fait partie de la vie quotidienne, à pour but de soulager l'homme de tâches dangereuses, répétitives ou fatigantes à accomplir. C'est aussi un moyen de produire plus vite, à toutes heures, des biens ou des services. Il représente l'ensemble des technologies utilisant l'électronique, l'électrotechnique, la mécanique, la télécommunication afin de concevoir des machines ou des processus automatisés qui peuvent fonctionner sans intervention humaine.

Ce là fait appel au système de contrôle ou de commande en boucle fermée qui compare en permanence ce que l'on obtient à ce que l'on souhaite obtenir que ce soit une température, une pression, un débit ou une hygrométrie.

Ce type de contrôle a pour but d'atteindre les performances souhaitées : minimisation de l'erreur, diminution de temps de réponse, stabilité du système.

A fin d'améliorer ces caractéristiques il est nécessaire d'introduire dans la boucle de commande un régulateur. Les régulateurs doivent permettre de réaliser le meilleur compromis entre précision, stabilité et rapidité du système étudié.

La régulation mesure en permanence par biais de sonde ou de capteurs le système à régler puis transmet ces informations au régulateur celui-ci compare cette mesure à la consigne puis suivant son algorithme le régulateur va transmettre ses ordres aux actionneurs (vannes, volets, moteurs, etc.) afin de corriger les perturbations, ce pendant l'importance du régulateur PID reste in affectée à cause de sa structure simple qui peut être facilement comprise et mettre en œuvre.

Malheureusement, il a été difficile de régler le gain de régulateur PID correctement puisque c'est difficile de déterminer ses trois paramètres proportionnels, intégrale et dérivée optimaux ou presque optimaux avec les méthodes de réglages classiques.

Plusieurs approches ont été documentées dans la littérature pour déterminer les paramètres du PID, la méthode de Ziegler et Nichols, l'approche floue réseau de neurones, les algorithmes génétiques ne sont que quelques-uns parmi de nombreuses œuvres.

Les algorithmes génétiques (AG) sont des algorithmes évolutionnistes d'optimisation stochastique inspirés des mécanismes de l'évolution naturelle élaborés par Charles Darwin.

La technique AG est une excellente méthode pour résoudre les paramètres optimaux du régulateur PID, elle peut générer une solution de haute qualité au un temps de calcul plus court.

Ce mémoire est organisé en cinq chapitres comme suit :

Le premier chapitre est divisé en deux parties. La première partie décrit le principe de l'automatisation, notion de système, de Boucle Ouverte et de Boucle Fermée.

La deuxième partie définit l'asservissement et son objectif, la modélisation des systèmes et un peu plus de détails sur les systèmes asservis en boucle ouverte et en boucle fermée.

Le deuxième chapitre est consacré à l'étude du contrôleur PID ainsi qu'à la description des méthodes classiques de réglage de ces gains.

Le troisième chapitre concerne les algorithmes génétiques simples. Nous présentons une description détaillée des algorithmes génétiques simples puis nous rappelons les définitions relatives à leur fonctionnement.

Le quatrième chapitre est réservé pour la conception d'un système de contrôle de température dont l'objectif est d'appliquer l'asservissement de température en boucle fermée par l'insertion d'un régulateur PID.

Le dernier chapitre présente l'implémentation. Nous essayons de transformer le modèle conceptuel établi en des composants logiciels formant le système étudié.

En fin, on termine ce mémoire par une conclusion générale.

CHAPITRE I

L'AUTOMATISME ET L'ASSERVISSEMENT

I.1. Introduction

L'automatisme est une science et une technique qui permet de maîtriser le comportement d'un système (traduit par ses grandeurs de sortie), en agissant de manière adéquate sur ses grandeurs d'entrée.

Un système asservi est un système automatisé qui traite le signal d'écart pour modifier en continu le signal commande afin d'atteindre l'objectif défini par la consigne quels que soient les effets perturbateurs extérieurs.

Les systèmes asservis sont indiqués dans le cas où il faut travailler dans des conditions présentant un caractère aléatoire. La consigne peut être aléatoire (ou de forme imprévisible lors de la conception du système) comme dans le cas du copiage ou du suivi de trajectoire et les perturbations peuvent être aléatoires comme dans le cas d'une régulation de température.

Les systèmes asservis sont indiqués lorsque l'on désire des performances dynamiques élevées. [1]

I.2. Automatisme

L'automatisation consiste à « rendre automatique » les opérations qui exigeaient auparavant l'intervention humaine.

C'est aussi un moyen de produire plus vite, à toutes heures, des biens ou des services.

I.2.1. Définition de l'automatisme

L'automatisation est considérée comme l'étape d'un progrès technique où apparaissent des dispositifs techniques susceptibles de seconder l'homme, non seulement dans ses efforts musculaires, mais également dans son travail intellectuel de surveillance et de contrôle. [2]

Il existe deux domaines d'intervention de l'automatique :

- Dans les systèmes à événements discrets. On parle d'automatisme (séquence d'actions dans le temps). Exemples d'applications : les distributeurs automatiques, les ascenseurs, le montage automatique dans le milieu industriel, les feux de croisement, les passages à niveaux.
- Dans les systèmes continus pour asservir et/ou commander des grandeurs physiques de façon précise et sans aide extérieure. Quelques exemples d'application : l'angle d'une fusée, la vitesse de rotation d'un lecteur CD, la position du bras d'un robot, le pilotage automatique d'un avion.[3]

I.2.2. Principes de base

I.2.2.1. Notion de système, de Boucle Ouverte (BO), de Boucle Fermée (BF)

L'automatique peut s'appliquer à tout ce qui bouge, fonctionne, se transforme. L'objet d'application de l'automatique est appelé système. [3]

Un système se caractérise par ses grandeurs d'entrée et de sortie. Les grandeurs d'entrée sont les grandeurs qui agissent sur le système. Il en existe de deux types :

commandes: celles que l'on peut maîtriser.

perturbations: celles que l'on ne peut pas maîtriser.

Un système est en boucle ouverte (Figure I.1) lorsque la commande est élaborée sans l'aide de la connaissance des grandeurs de sortie : il n'y a pas de feedback (une contre-réaction).

Ce principe nous l'utilisons tous les jours dans la plupart de nos actions. Pour conduire, nous devons regarder la route et sans cesse corriger la direction de la voiture même s'il n'y a pas de virages.

Dans le cas contraire, le système est dit en boucle fermée (Figure I.2). La commande est alors fonction de la consigne (la valeur souhaitée en sortie) et de la sortie. [3]



Figure I.1 : Schéma d'un système en boucle ouverte

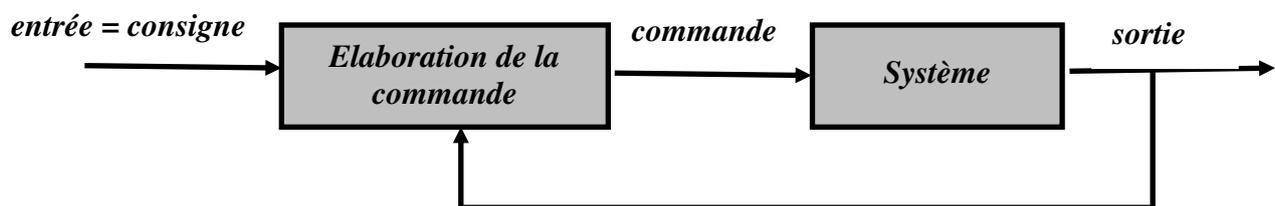


Figure I.2: Schéma d'un système en boucle fermée

I.2.2.2. Nécessité de la boucle fermée

Exceptionnellement, le système de commande peut opérer en boucle ouverte à partir du seul signal de consigne. Mais la boucle fermée (contre réaction) est capable de :

- stabiliser un système instable en BO.
- compenser les perturbations externes.
- compenser les incertitudes internes au processus lui-même.

Un système de commande peut réaliser deux fonctions distinctes :

l'asservissement c'est à dire la poursuite par la sortie d'une consigne variable dans le temps.

la régulation c'est à dire la compensation de l'effet de perturbations variables sur la sortie (la consigne restant fixe).

I.3. l'asservissement

I.3.1. Définition de l'asservissement

En automatique, un asservissement est un algorithme dont l'objet principal est de forcer la sortie ou l'état d'un système à atteindre le plus rapidement possible sa valeur de consigne (la valeur souhaitée en sortie) et de limiter l'écart par rapport à cette dernière, quelles que soient les perturbations externes ou internes. Le principe général est de comparer la consigne et l'état (la sortie) du système de manière à le corriger efficacement. On parle également de système commandé par rétro action négative ou en boucle fermée «feedback».

[4]

I.3.2. Objectif de l'asservissement

Un système dynamique peut être caractérisé par différentes qualités : sa rapidité, le fait d'être plus ou moins amorti. Le but de l'asservissement est de contraindre le système à se comporter d'une manière particulière. Il n'est pas possible physiquement d'obtenir d'un système qu'il réponde de manière instantanée. On peut cependant le contraindre à répondre plus rapidement. On peut aussi limiter son dépassement.

Dans le choix de la loi de commande, il faudra également s'assurer que le système asservi possède un niveau de robustesse suffisant. Par robustesse, on entend la capacité à garder certaines propriétés malgré des variations de l'environnement. Typiquement, en automatique, le comportement du système est connu avec une précision limitée ; de plus, son comportement peu évoluer en fonction de condition extérieure ou de son vieillissement. Il importe que les performances du système asservi ne se dégradent pas trop en présence de ces variations de comportement. [5]

I.3.3. Modélisation d'un système

En automatique, La modélisation des systèmes est une étape nécessaire pour leur analyse et leur commande.

Cette phase se décompose en trois étapes : définir les phénomènes physiques du système à commander, faire le bilan des variables mesurables et identifier l'origine des phénomènes principaux (électrique, mécanique, hydraulique).

L'écriture des équations du modèle peut se présenter sous différentes formes, l'une adaptée à la description des systèmes électromécaniques (moteurs électriques) et l'autre adaptée aux systèmes hydrauliques.

Il est indispensable d'établir *un modèle comportemental* (appelé boîte noire) du système à commander. Ceci nécessite l'écriture *d'un modèle de connaissance* ou de tous modèles intermédiaires.

I.3.3.1. Modèle de connaissance

Les modèles de connaissance sont élaborés à partir des lois de la physique (électricité, mécanique) ou de la chimie. L'objectif étant d'explicitier le fonctionnement d'un système par une relation mathématique. Ces modèles peuvent être assez complexes et comporter de nombreux paramètres à identifier [5].

I.3.3.2. Modèle de comportement

Les modèles de comportement sont des modèles linéaires, dont la validité reste limitée à de petites variations autour d'un point de fonctionnement. Ils se concrétisent par des fonctions de transfert.

I.3.3.3. Modèle intermédiaire

Les modèles intermédiaires sont des modèles hybrides souvent issus d'une simplification ou d'une linéarisation des modèles de connaissances.

I.3.3.4. Modèles utilisés en automatique

En automatique, on utilise le modèle de comportement pour la commande des systèmes. Si ce dernier est complexe, on utilise le modèle intermédiaire issu des simplifications sur le modèle de comportement ou de connaissance.

I.3.3.5. Modèles de calcul et de validation

En commande des systèmes, on cite deux modèles: le modèle de calcul (souvent simplifié), utilisé pour la synthèse des lois de commande et le modèle de validation (simulation) de ces dernières, représentant la plus part des phénomènes physiques du système, où les paramètres et les signaux du modèle sont physiques.

I.3.3.6. Modèles d'état et entrée/sortie

Les systèmes sont représentés par deux modèles qui dépendent de l'espace utilisé:

- **Temporel:** La variable utilisée est le temps «t», variable physique et on obtient le modèle d'état.
- **Complexe:** La variable utilisée est l'opérateur de Laplace «p» (ou «s»), variable abstraite et on obtient le modèle fréquentiel.

I.3.4. Schéma de principe d'un système asservi

Les systèmes de commande se présentent sous deux formes:

I.3.4.1. Système de commande en boucle ouverte

Il est composé seulement d'une chaîne directe et il n'existe pas de retour de la sortie commandée (Figure I.3). Il est sensible aux perturbations externes et est très difficile ou impossible de suivre un signal de consigne variable.

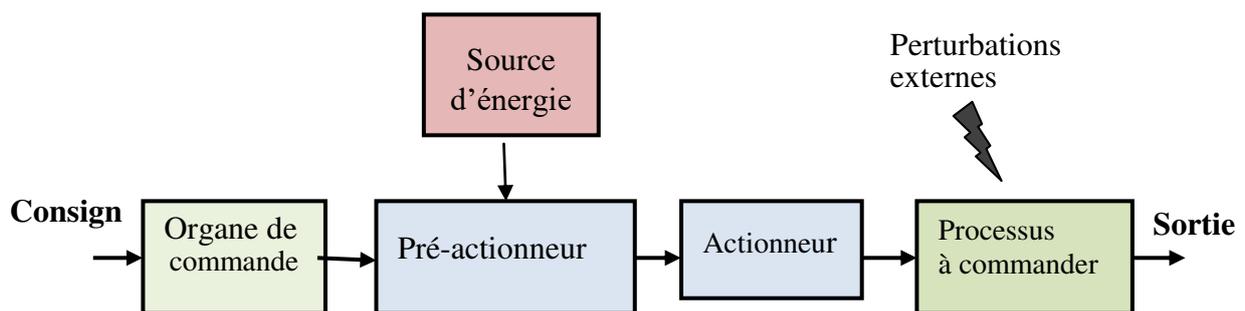


Figure I.3 : Système de commande en boucle ouverte

I.3.4.2. Système de commande en boucle fermée

Il est composé d'une chaîne directe, d'une chaîne de retour et un comparateur de la consigne et de la mesure de la sortie commandée (Figure I.4). Le signal de commande est ajusté en fonction de l'erreur entre la consigne et la mesure. Il permet de compenser les effets des perturbations externes.

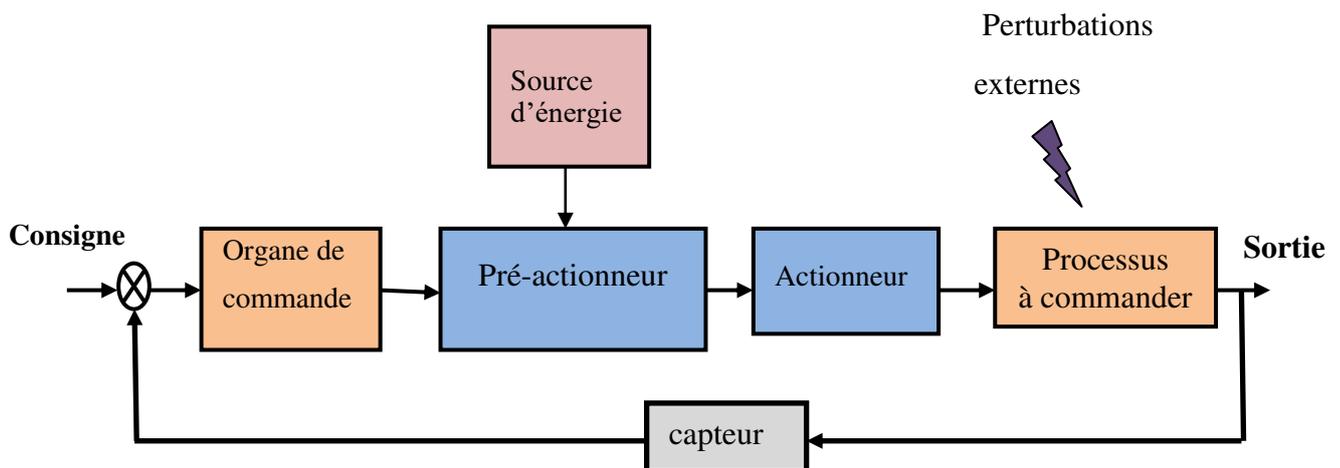


Figure I.4: Système de commande en boucle fermée.

Un système asservi est un système de commande en boucle fermée, c'est -à-dire il existe un retour ou une contre réaction de la sortie commandée (Figure I.5). Cette dernière donne la mesure de la sortie, après adaptation, elle est comparée à la grandeur de consigne et la différence (erreur) est appliquée à l'organe de commande. En fonction de cette erreur, un signal de commande est généré puis, appliqué au pré-actionneur.

Ce dernier est informé par le signal de commande sur la quantité d'énergie à délivrer à l'actionneur par la source.

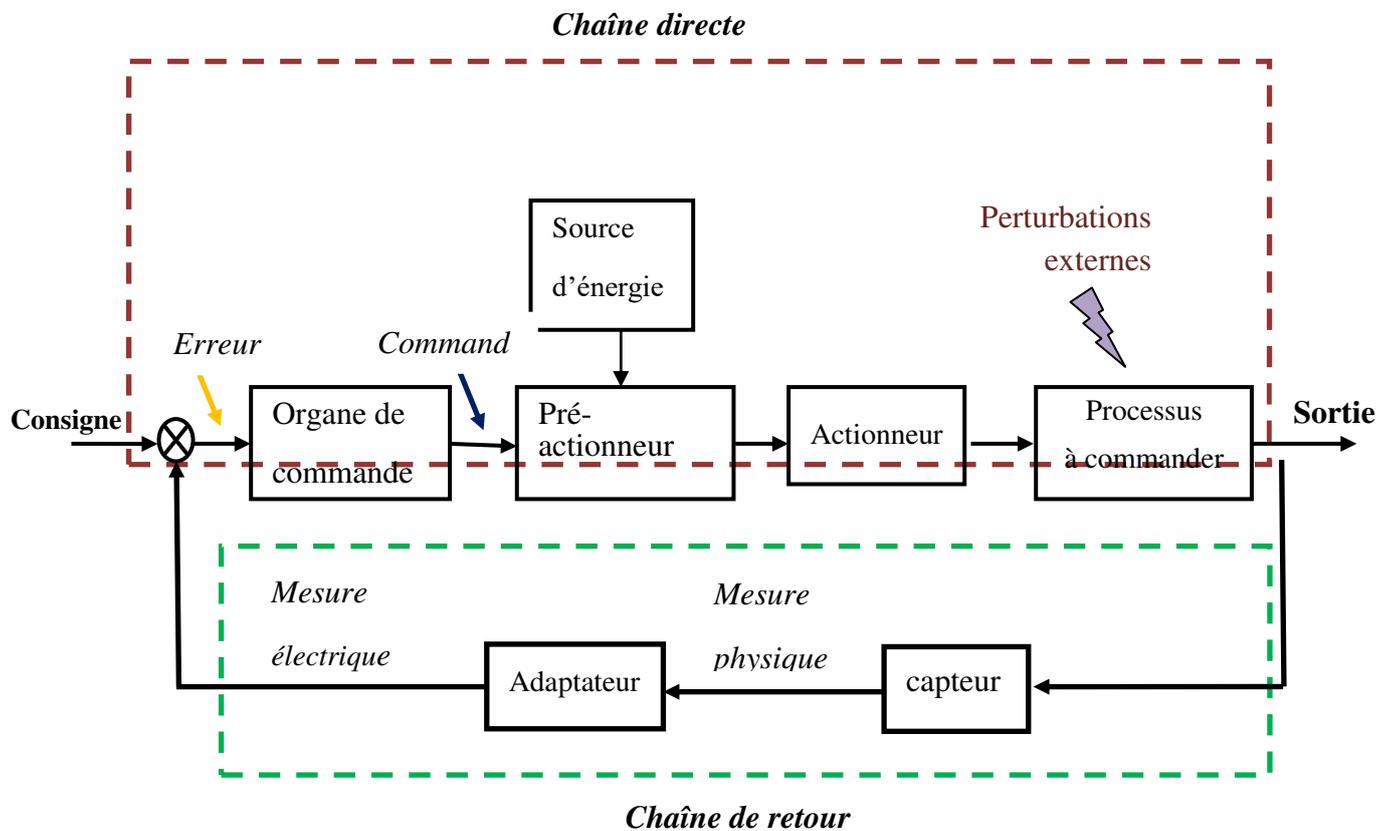


Figure 1.5: Schéma de principe d'un système asservi

I.3.5. Les caractéristiques d'un système asservi

Un système asservi est caractérisé par :

a) **La précision** : C'est la capacité du système à se rapprocher le plus possible de la valeur de consigne.

b) **La rapidité** : C'est la capacité du système à atteindre dans les meilleurs délais son régime stable. La rapidité d'un système est définie par son temps de réponse (plus temps de réponse est petit plus le système est dit rapide).

c) **La stabilité** : Pour une consigne constante la sortie doit tendre vers une constante.

I.4. Conclusion

Dans ce chapitre nous avons présenté la notion de l'automatisme ainsi que celle de l'asservissement, nous avons montré que les systèmes automatiques permettent d'avoir un comportement qui dépend de son milieu extérieur avec lequel il peut interagir certains sont capables de suivre le niveau d'une grandeur physique de consigne, ce sont les systèmes asservis.

L'asservissement a pour but de remplacer l'homme dans diverses tâches pour cela il doit avoir des organes et un comportement à celui d'un homme, c'est-à-dire il doit être capable d'apprécier, de comparer puis d'agir. Le rôle des systèmes asservis est de faire varier une grandeur déterminée suivant une loi (non fixée à l'avance). Une loi imposée par un système de comparaison.

CHAPITRE II

LE CONTROLEUR PID

II.1. Introduction

On s'intéresse dans ce chapitre à l'étude des actions du module PID est dite aussi (correcteur, régulateur, contrôleur), composé par les trois actions de base P, I et D d'où le 'P' correspond au terme proportionnel, 'I' pour le terme intégral et 'D' pour le terme dérivé du module.

Le contrôleur PID était l'outil standard quand le contrôle des processus a émergé dans les années 40. Il est basé sur un mécanisme de rétroaction, largement répandu dans les systèmes de commande industriel ; grâce notamment à sa simple structure et sa stratégie de commande.

Le contrôleur PID est probablement le plus largement utilisé dans le contrôle industriel, car il permet de régler à l'aide de ses trois paramètres les performances (amortissement, temps de réponse) d'une régulation d'un processus modélisé.

Il est relativement robuste par rapport aux variations des paramètres du procédé, quand on n'est pas trop exigeant pour les performances de la boucle fermée par rapport à celles de la boucle ouverte (par exemple, accélération très importante de la réponse ou augmentation très importante de l'amortissement en boucle fermée) [6].

II.2. Contrôleur PID

II.2.1 Définition

Contrôleur PID est un système d'auto régulation en boucle fermée, qui cherche à réduire l'erreur entre la consigne et la mesure. [4]

$$e = \text{consigne} - \text{mesure} \quad (\text{II.1})$$

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

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

II.2.2 Action proportionnelle (P)

L'action est dite proportionnelle lorsque le signal de commande est proportionnel au signal d'erreur. Elle corrige de manière instantanée, donc rapide, tout écart de la grandeur à régler, l'erreur est virtuellement amplifiée d'un certain gain constant qu'il conviendra de déterminer en fonction du système [6].

$$\text{Consigne}(t) = K_p e(t) \quad (\text{II.2})$$

L'idée étant d'augmenter l'effet de l'erreur sur le système afin que celui-ci réagisse plus rapidement aux changements de consignes. Plus la valeur de K_p est grande, plus la réponse l'est aussi. En revanche, la stabilité du système s'en trouve détériorée et dans le cas d'un K_p démesuré le système peut même diverger.

Si l'on prend l'exemple d'une voiture qui dérive, la régulation proportionnelle consiste à contrebraquer rapidement pour rétablir la voiture.

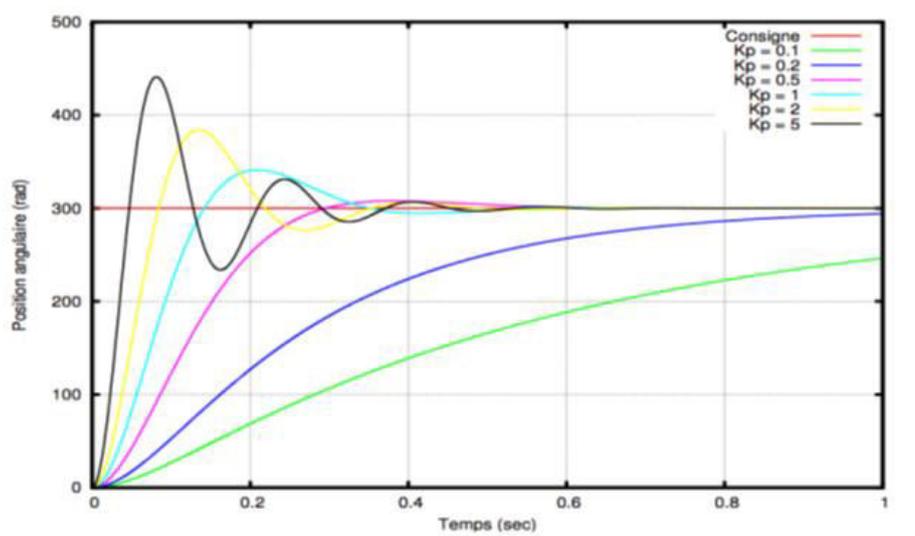


Figure II.1 : Modélisation sous Maple de la réponse à un échelon dans un asservissement en position.

II.2.3. Action Intégrale (I)

Lors d'un simple contrôle proportionnel, il subsiste une erreur statique. Lorsque le système s'approche de sa consigne, l'erreur n'est plus assez grande pour faire avancer un actionneur.

Le terme intégral permet ainsi de compenser l'erreur statique et fournit, par conséquent, un système plus stable en régime permanent. Plus K_i est élevé, plus l'erreur statique est corrigée. L'erreur entre la consigne et la mesure est ici intégrée par rapport au temps et multipliée par une constante qu'il faudra aussi déterminer en fonction du système.

$$\text{Consigne}(t) = K_p e(t) + K_i \int_0^t e(t) dt \quad (\text{II.3})$$

Pour reprendre l'exemple de la voiture qui dérive, le terme intégral consiste à rajouter un petit coup de contre braquage afin de se rétablir correctement.

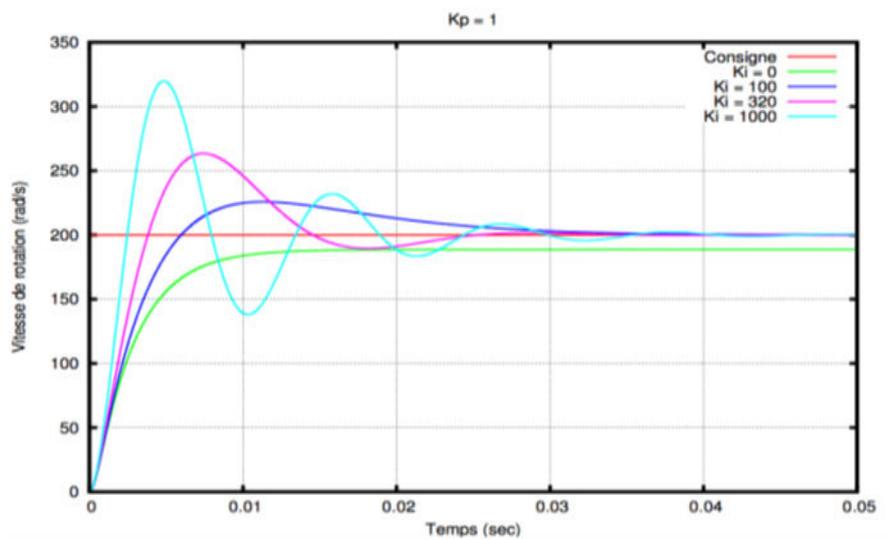


Figure II.2 : Modélisation sous Maple de la réponse à un échelon dans un asservissement en vitesse.

II.2.4. Action Dérivé (D)

Le contrôle PI peut amener à un dépassement de la consigne, ce qui n'est pas toujours très souhaitable.

Le terme dérivé permet de limiter cela. Lorsque le système s'approche de la consigne, ce terme freine le système en appliquant une action dans le sens opposé et permet ainsi une stabilisation plus rapide.

Celui-ci consiste à dériver l'erreur entre la consigne et la mesure par rapport au temps et à le multiplier lui aussi par une constante.

$$\text{Consigne}(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) \quad (\text{II.4})$$

Toujours avec l'exemple de la voiture, le terme dérivé représente le petit contre braquage dans le sens opposé de l'ajustement pour stabiliser la voiture lorsqu'elle s'approche du point qu'elle voulait ré joindre.

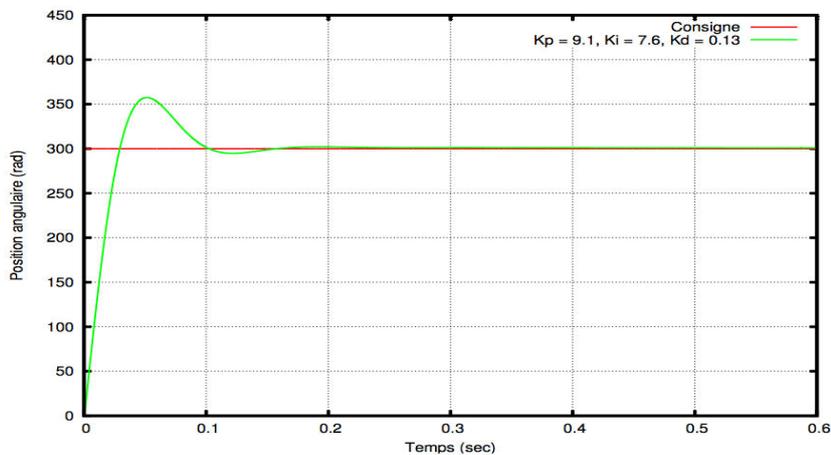


Figure II.3 : Modélisation sous Maple de la réponse à un échelon dans un asservissement en vitesse.

Les effets de chacun des paramètres du contrôleur, K_p , K_i et K_d sur un système en boucle fermée sont résumées dans le tableau ci-dessous :

Intérêt des actions K_p, K_i, K_d				
	Temps de réponse	Dépassement	Temps de stabilisation	Erreur statique
K_p	Diminue	Augmente	Augmente	Diminue
K_i	Diminue	Augmente	Augmente	Éliminer
K_d	---	Diminue	Diminue	Pas de changement

Tableau II.1. Caractéristiques du contrôleur PID

II.3. Les différentes structures d'un contrôleur PID:

Il existe trois formes différentes de mise en œuvre des régulateurs PID :

II.3.1. La forme PID série :

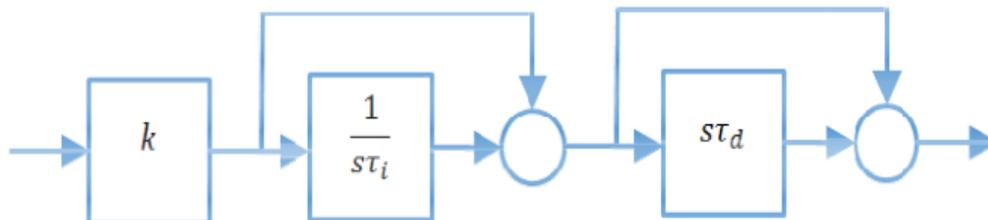


Figure II.4 : Le contrôleur PID Série [6]

Elle est définie par l'équation ci-dessous :

$$C_{\text{série}}(s) = K\left(1 + \frac{1}{s\tau_i}\right)(1 + s\tau_d) \quad (\text{II.5})$$

II.3.2. La forme PID parallèle :

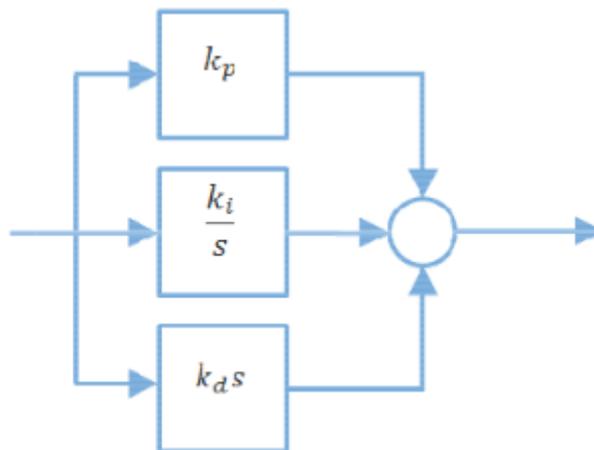


Figure II.5 : Le contrôleur PID Parallèle [6]

Elle est définie par l'équation ci-dessous :

$$C_{\text{parallèle}}(s) = K_p + \frac{K_i}{s} + K_d * s \quad (\text{II.6})$$

II.3.3. La forme PID standard :

Une 3ème forme de PID existe, Celle-ci est appelé "standard" ou "mixte" ou parfois "idéal".

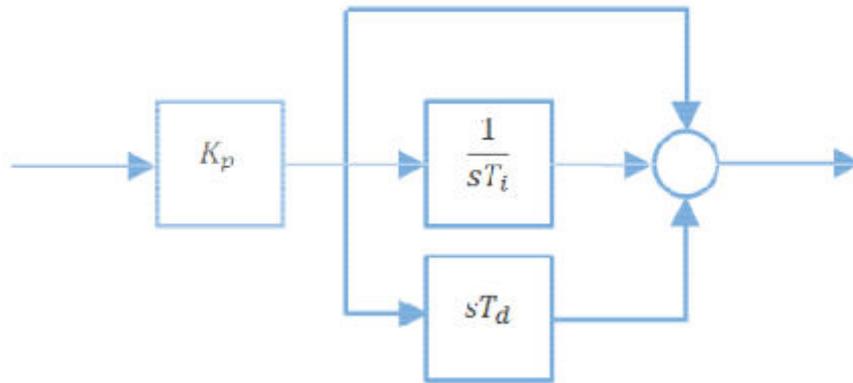


Figure II.6 : Le contrôleur PID Standard, Mixte ou Idéal [6]

Elle est définie par l'équation ci-dessous :

$$C_{\text{standard}}(s) = K_p \left(1 + \frac{1}{sT_i} + sT_d \right) \quad (\text{II.7})$$

II.4. Conclusion

Le fonctionnement du contrôleur PID (algorithme) implique trois paramètres : le proportionnel, les deux valeurs intégrales et dérivées respectivement dénoté I et D. Ces valeurs peuvent être interprétées en termes d'erreur ; P dépend de l'erreur actuelle, I sur l'accumulation des erreurs passées, et D est une prévision de futures erreurs. La somme pondérée de ces trois actions est employée pour ajuster le processus.

En accordant les trois paramètres dans l'algorithme du contrôleur ainsi que des conditions spécifiques, le contrôleur peut fournir une action de commande qui répond à nos besoins.

Cependant il faut très souvent procéder à un réglage fin des coefficients jusqu'à obtenir un système respectant le cahier des charges. Pour cela on utilise les règles suivantes :

- Si K_p augmente alors la montée sera plus rapide mais il y aura plus de dépassement.
- Si K_i augmente alors la montée sera plus rapide et l'erreur statique sera plus faible mais le régime stationnaire sera plus long à atteindre.
- Si K_d augmente alors le dépassement diminuera et le temps d'établissement du régime stationnaire aussi, mais la sensibilité au bruit augmentera.

On rappelle que le rôle d'un contrôleur est de maintenir la grandeur régulée à une valeur de la consigne malgré la présence des perturbations dans le fonctionnement en régulateur ou suivre la variation d'une consigne dans le fonctionnement en asservissement.

Il est possible d'obtenir un comportement désiré en boucle fermée, caractérisant les performances du système.

Il s'agit de proposer des valeurs à donner aux paramètres K_p , K_i , K_d répondant le mieux possible aux spécifications d'un cahier des charges.

Pour effectuer un choix judicieux nous allons adapter un algorithme génétique (AG) simple permettant de trouver les paramètres optimaux d'un PID qui minimisent l'erreur de la réponse indicielle du système.

CHAPITRE III

LES ALGORITHMES GENETIQUES

III.1. Introduction

Les algorithmes génétiques (AGs) sont des algorithmes d'optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique. Ces algorithmes cherchent à simuler le processus de la sélection naturelle dans un environnement défavorable en s'inspirant de la théorie de l'évolution proposée par Charles Darwin [7]. Ils sont des algorithmes itératifs de recherche globale dont le but est d'optimiser une fonction prédéfinie appelée critère ou fonction coût "fitness". Ils permettent de faire évoluer un ensemble initial de solutions vers un ensemble final.

Par analogie avec l'évolution naturelle, les AG's font évoluer un ensemble de solutions candidates, appelé une « population d'individus ».

Un « individu » n'est autre qu'une solution possible du problème à résoudre. Chaque individu de cette population se voit attribuer une fonction appelée fonction d'adaptation (fitness) qui permet de mesurer sa qualité ou son poids; cette fonction d'adaptation peut représenter la fonction objective à optimiser. Ensuite, les meilleurs individus de cette population sont sélectionnés, subissent des croisements et des mutations et une nouvelle population de solutions est produite pour la génération suivante.

Ce processus se poursuit, génération après génération, jusqu'à ce que le critère d'arrêt soit atteint, comme par exemple le nombre maximal de générations [7].

Les algorithmes génétiques sont utilisés dans l'automatique dans le but de résoudre les problèmes d'optimisation dans le domaine de la commande de systèmes.

III.2. Définition

Les algorithmes génétiques sont une méthode d'optimisation basée sur les mécanismes de la sélection naturelle. La solution optimale est cherchée à partir d'une population de solutions en utilisant des processus aléatoires. La recherche de la meilleure solution est effectuée en créant une nouvelle génération de solutions par application successive, à la population courante, de trois opérateurs : la sélection, le croisement et la mutation. Ces opérations sont répétées jusqu'à ce qu'un critère d'arrêt soit atteint [8].

III.3. Terminologie :

Avant de décrire le principe des algorithmes génétiques, il est nécessaire de présenter le vocabulaire que nous allons utiliser tout au long de ce travail.

III.3. 1. Individu/chromosome/séquence : un individu biologique est une forme qui est le produit de l'activité des gènes. Dans le cadre d'un AG traditionnel, l'individu est réduit à un chromosome, on l'appelle indifféremment individu ou chromosome.

III.3. 2. Population : Dans les systèmes naturels, la population est un groupe d'individus. Dans les algorithmes génétiques, la population est l'ensemble des individus ou des chromosomes. Les populations sont également appelées des générations.

III.3.3. Une génération: ensemble d'individus tirés aléatoirement ou sélectionnés parmi la population.

III.3. 4. Environnement : l'espace de recherche.

III.3. 5. Les gènes: contiennent les paramètres permettent d'évaluer une fonction (un ou plusieurs bits dans le cas d'un codage binaire par exemple).

III.3. 6. Fonction d'évaluation (fitness): L'évaluation de l'adaptation de chaque individu à l'environnement est réalisée au moyen d'une fonction d'adaptation (fitness). Cette fonction attribue à chaque individu une valeur qui représente son niveau d'adaptation.

Elle est une fonction du critère à minimiser. En raison de l'analogie avec la théorie de l'évolution (survie des individus les mieux adaptés à leur environnement), l'algorithme génétique est naturellement formulé en terme de maximisation.

III.3. 7. Les chromosomes: sont les éléments à partir desquels sont élaborées les solutions (croisement génétique, mutation) dans le cas le plus simple, l'individu est représenté par un seul chromosome qui contient plusieurs gènes.

III.4. Le principe des algorithmes génétiques :

Un algorithme génétique recherche le ou les extrêmes d'une fonction définie sur un espace de données. Pour l'utiliser, on doit disposer des cinq éléments suivants :

- ✓ Un principe de codage de l'élément de population. Il faut choisir un bon codage qui aide à obtenir l'objectif.
- ✓ Génération de la population initiale avec une taille fixe, formée d'un ensemble fini de solutions.
- ✓ Une fonction à optimiser. Celle-ci retourne une valeur appelée fitness ou fonction d'évaluation de l'individu.
- ✓ Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace d'état.
- ✓ Des paramètres de dimensionnement : taille de la population, nombre total de générations, ou critère d'arrêt, probabilités d'application des opérateurs de croisement et de mutation.

La figure suivante montre l'organigramme de fonctionnement d'un algorithme génétique.

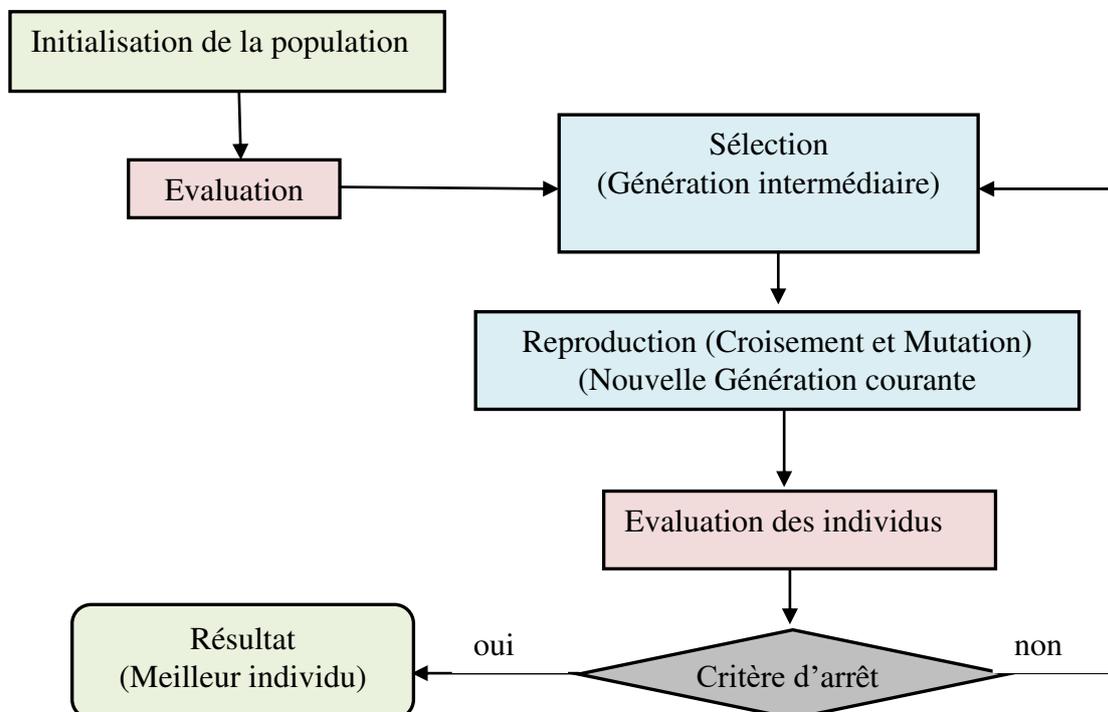


Figure III.1: organigramme de fonctionnement d'un algorithme génétique

III.5. Les Operations D'un algorithme Génétique

III.5.1. La Sélection

La sélection permet d'identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais. On trouve dans la littérature un nombre important de principes de sélection plus ou moins adaptés aux problèmes qu'ils traitent. Les deux principes de sélection suivants sont les plus couramment utilisés :

III.5.1.1. La Roulette

Cette méthode exploite la métaphore d'une roulette de casino. La roue est divisée en autant de secteurs que d'individus dans la population (Figure III.2). La taille de ces secteurs est proportionnelle à l'adaptation de chaque individu. En faisant tourner la roue, l'individu pointé à l'arrêt de la boule est sélectionné. Les individus les mieux adaptés ont donc plus de chance d'être tirés au sort lors du déroulement du jeu [7].

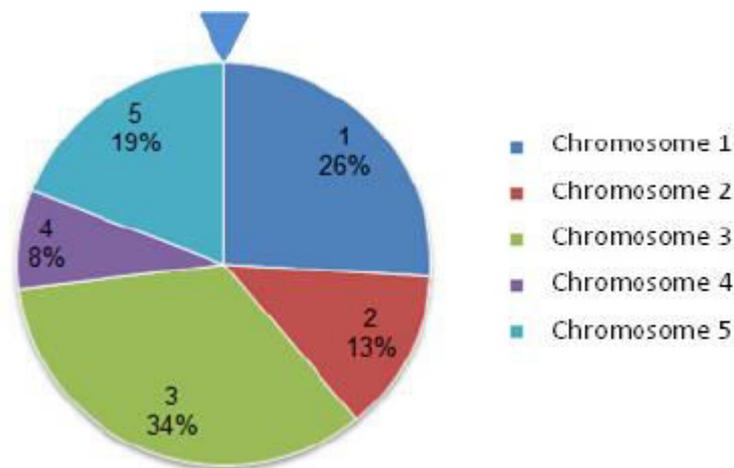


Figure III.2 : La roulette

III.5.1.2. Le Tournoi

Cette méthode ressemble plus à ce qui se passe dans la réalité. Comme son nom l'indique, elle fait s'affronter deux ou plusieurs individus afin que le meilleur gagne (Figure III.3). Plusieurs variantes existent. On peut par exemple faire varier le nombre d'individus qui doivent s'affronter au départ, ou encore permettre ou non que le même individu soit éligible plusieurs fois lors d'un même tournoi.

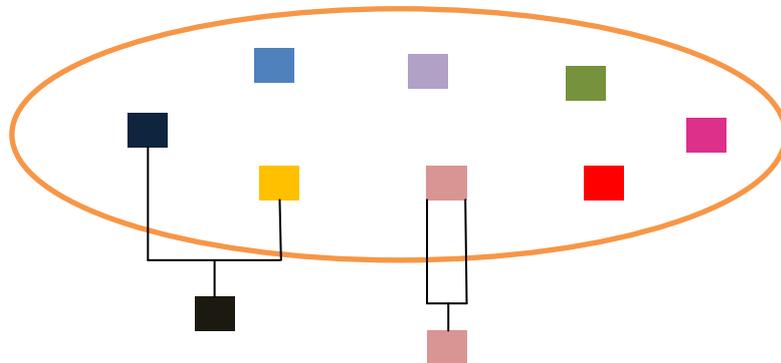


Figure III.3 :Le tournoi entre deux individus

III.5.2. Le Croisement

L'opérateur de croisement a pour objectif de recombinaison les chromosomes d'une paire d'individus sélectionnés (parents), afin de créer une nouvelle paire d'individus (enfants) qui héritent de certaines caractéristiques de leurs parents (Figure III.4). Le croisement est mis en place pour que les nouveaux chromosomes gardent la meilleure partie des chromosomes anciens. Ceci dans le but d'obtenir de meilleurs chromosomes. Le croisement à un site est illustré par figure suivante :

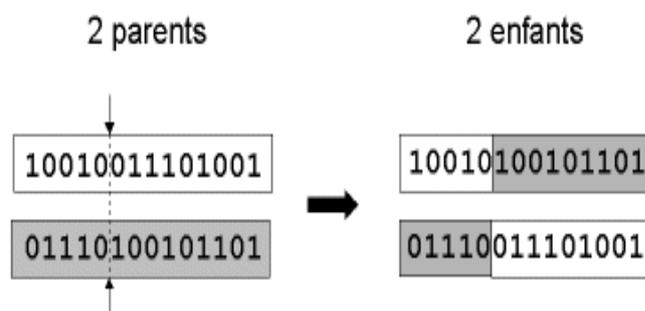


Figure III.4 :Le Croisement

III.5.3. La Mutation

La mutation permet de transformer au hasard le codage d'un individu (Figure III.5) afin d'apporter une certaine diversité dans la population et empêcher que celle-ci converge trop vite vers un seul type d'individu parfait, incapable de sortir d'un minimum local.

La mutation est réalisée en modifiant un gène d'un individu pris au hasard. Dans les Algorithmes Génétiques, la mutation est considérée comme un opérateur secondaire par rapport au croisement.

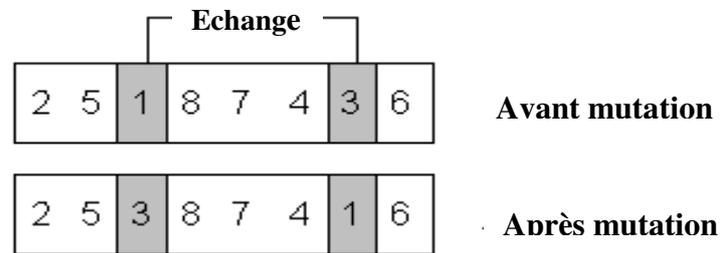


Figure III.5 :Illustration de la Mutation

III.5. 4. Le Codage

Le premier pas dans l'implantation des algorithmes génétiques est de créer une population d'individus initiaux. En effet, les algorithmes génétiques agissent sur une population d'individus, et non pas sur un individu isolé. Par analogie avec la biologie, chaque individu de la population est codé par un chromosome ou génotype.

Une population est donc un ensemble de chromosomes.

Chaque chromosome code un point de l'espace de recherche. L'efficacité de l'algorithme génétique va donc dépendre du choix du codage d'un chromosome [7].

On aboutit à une structure présentant cinq niveaux d'organisation :

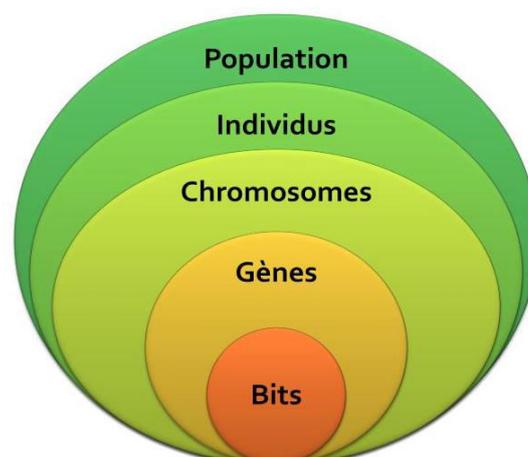


Figure III.6 : Les cinq niveaux d'organisation d'un algorithme génétique.

III.5. 4.1.Codage binaire

Ce type de codage est certainement le plus utilisé car il présente plusieurs avantages. Son principe est de coder la solution selon une chaîne de bits (qui peuvent prendre les valeurs 0 ou 1). Si un gène est représenté par un entier long (par exemple 32 bits), les chromosomes qui sont des suites de gènes sont représentés par des tableaux de gènes et les individus de notre espace de recherche sont représentés par des tableaux de chromosomes.

III.5. 4.2.Codage réel

Dans ce type de codage, chaque chromosome est un vecteur dont les composantes sont les paramètres du processus d'optimisation. Par exemple, si on recherche l'optimum d'une fonction de n variables $f(x_1, x_2, x_3, \dots, x_{n-1}, x_n)$, on peut utiliser tout simplement un chromosome contenant les n variables $(x_1, x_2, x_3, \dots, x_{n-1}, x_n)$, avec ce type de codage, la procédure d'évaluation des chromosomes est plus rapide vu l'absence des étapes de codage et de transcodage (du réel vers le binaire et vice-versa).

III.5.5. Fonction d'évaluation (fitness) :

L'évaluation de l'adaptation de chaque individu à l'environnement est réalisée au moyen d'une fonction d'adaptation (fitness). Cette fonction attribue à chaque individu une valeur qui représente son niveau d'adaptation. La fonction d'adaptation peut affecter directement la qualité des résultats obtenus par l'AG ainsi que le temps d'exécution. Elle est fonction du critère à minimiser. En raison de l'analogie avec la théorie de l'évolution (survie des individus les mieux adaptés à leur environnement), l'algorithme génétique est naturellement formulé en terme de maximisation. Etant donné une fonction réelle à une ou plusieurs variables, le problème d'optimisation sur l'espace de recherche s'écrit de la manière

$$\text{suivante: } \max_{x \in E} f(x)$$

Dans beaucoup de problèmes, l'objectif est exprimé sous forme de minimisation d'une fonction coût: $\min_{x \in E} h(x)$

Le passage du problème de minimisation à un problème de maximisation est obtenu par transformation de la fonction h selon la relation suivante : $f(x) = 1/(1+h(x))$

III.6. Conclusion

Dans ce chapitre, nous avons décrit brièvement le principe des algorithmes génétiques, qui permettent d'aborder la majorité des problèmes d'optimisation en utilisant un formalisme simple. Dans le chapitre suivant de ce travail, nous verrons comment utiliser les algorithmes génétiques pour optimiser au maximum les gains des contrôleurs PID utilisés dans la commande de systèmes.

CHAPITRE IV

LA CONCEPTION

IV.1. Introduction :

La température est une grandeur physique qui caractérise la sensation de chaleur ou de froid laissée par le contact d'un corps. Si la sensation de chaud augmente on dit que la température s'élève et inversement.

La mesure et le contrôle de cette dernière nécessite une régulation pour assurer le bon fonctionnement du procédé selon les critères définis dans un cahier de charge.

Un système de contrôle automatique de la température a la possibilité de surveiller et de contrôler la température dont l'objectif principal est de gérer la température avec une grande souplesse d'une régulation performante qui répond bien aux exigences demandées.

A cet effet, nous avons essayé dans ce chapitre de présenter un modèle de régulation de température et d'implémenter la commande PID de type parallèle, Figure (IV.1) qui permet d'améliorer les performances de réglage de la tension d'alimentation et grâce à l'adaptation de ses paramètres il peut agir efficacement contre les fluctuations de tension, et contre les ondulations de tension.

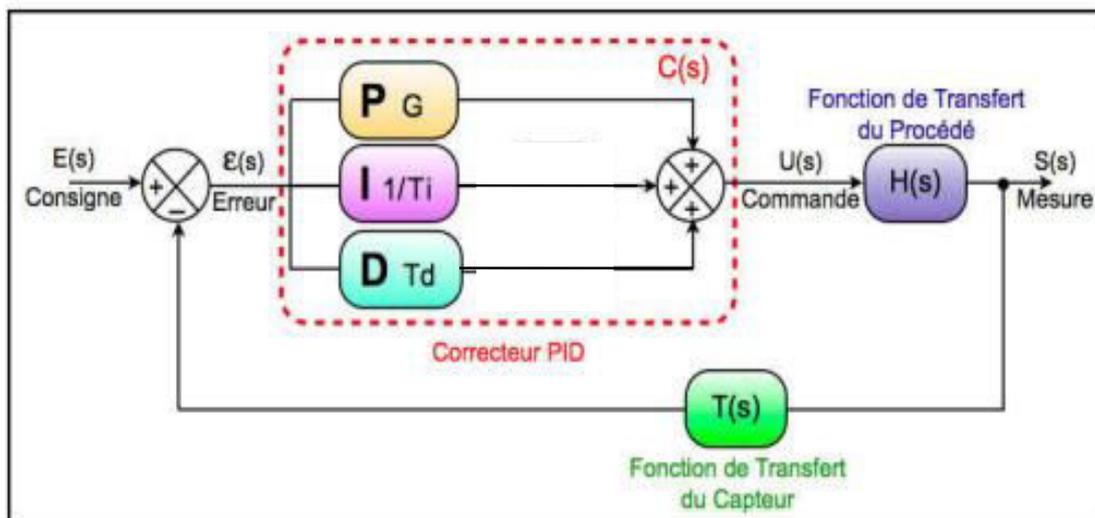


Figure IV.1: Asservissement PID

Dans cette partie, on va présenter un modèle de contrôle de température d'une plaque métallique dont l'objectif est d'appliquer l'asservissement de température en boucle fermée Figure (IV.1) par l'insertion d'un régulateur de température de type PID dans la chaîne d'asservissement.

Le contrôleur PID a été conçu afin de pouvoir fournir une tension de sortie pouvant prendre une valeur quelconque dans une plage bien déterminée, et dont la valeur peut être décidée facilement.

IV.2. Description du système :

Le système dont nous avons proposé permet de chauffer et de contrôler automatiquement la température d'une plaque métallique raccordée par un collier qui offre le chauffage électrique de cette pièce.

Pour surveiller l'évolution de la température de la plaque métallique il était impératif d'installer un capteur de température. Et pour la régulation de chauffe de la plaque à 45°, nous avons utilisé un algorithme exécuté par le microcontrôleur (calculateur) où son entrée est reliée à un capteur de température, et sa sortie reliée à un élément de commande.

IV.2.1. Schéma fonctionnel :

Le schéma de principe représente le synoptique de la régulation de température à réaliser, présenté dans la Figure suivante :

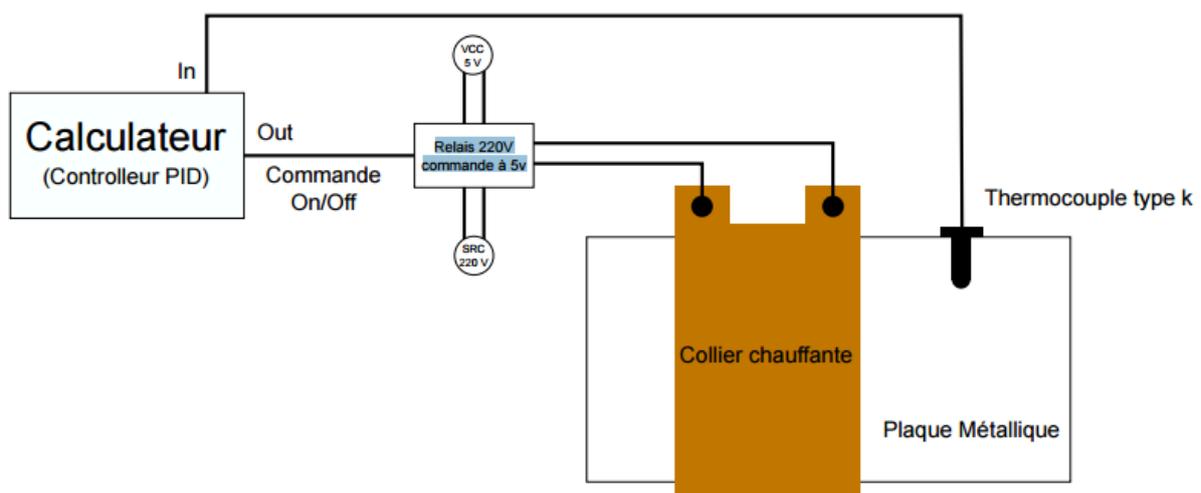


Figure IV.2: Système de contrôle de température

IV.2.2. Eléments constitutifs du système :

Le système proposé comporte les éléments suivants :

IV.2.2.1. Thermocouple type K

C'est un capteur de température qui fonctionne au moyen de deux conducteurs différents, joints à leurs extrémités. A l'intérieur se trouve un circuit électrique formé par deux conducteurs métalliques de matériaux différents soudés ensemble à leurs extrémités. [9] Les thermocouples sont particulièrement durcis et économiques et peuvent opérer sur une large gamme de températures.



Figure (IV.3): Capteur de température (Thermocouple type K)

IV.2.2.1.1. Avantages dans l'utilisation d'un thermocouple:

- ✓ Possibilité de mesurer une large plage de température (de l'ordre de 2000 °C).
- ✓ Possibilité d'utiliser des matériaux résistants aux différentes conditions environnementales.
- ✓ Les thermocouples sont des capteurs de température à faible coût.

IV.2.2.1.2. Désignations correspondants au thermocouple utilisé :

type	Metal A(+)	Metal B(-)	Plage d'utilisation	Coef. Seebeck $\alpha(\mu V/^{\circ}C)$	Erreur standard
K	Chromel	Alumel	40 à +1200°C	39,45 $\mu V/^{\circ}C$ à 0°C	2,2% à 0,75%

Tableau IV.1: caractéristique de thermocouple type K

IV.2.2.2. Calculateur (Un microcontrôleur) : est un circuit programmable capable d'exécuter un programme.[10]

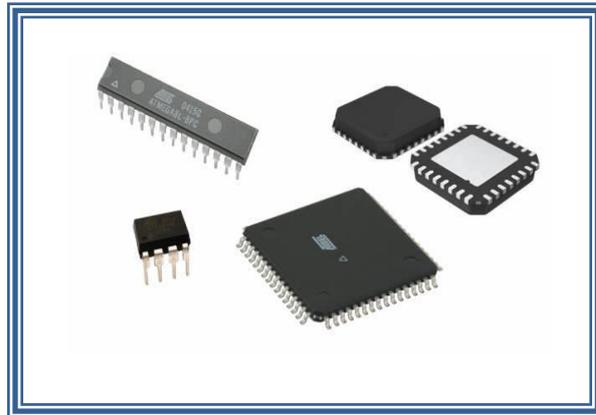


Figure IV.4: Microcontrôleur

Dans notre synoptique le microcontrôleur remplit trois fonctions, à savoir :

- ❖ Acquisition des signaux analogiques (température, consigne),
- ❖ Traitement de l'algorithme de régulation,
- ❖ Commande en puissance d'une plaque métallique..

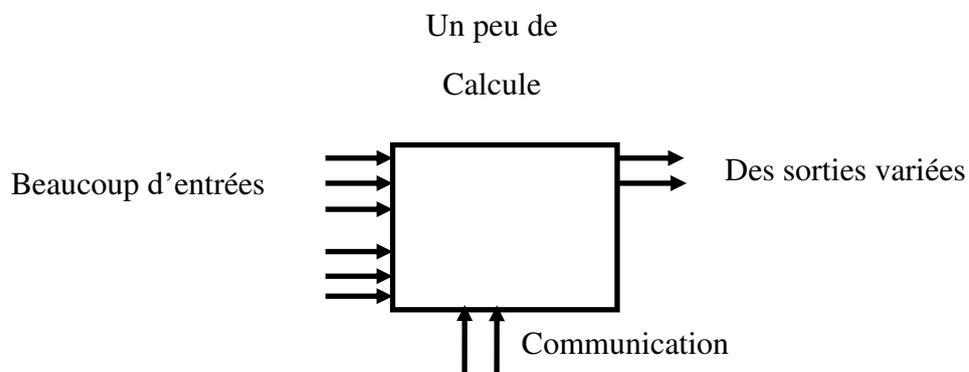


Figure IV.4: Fonctionnement d'un microcontrôleur

IV.2.2.3. Une plaque métallique.

IV.2.2.4. Collier chauffant : sont des éléments, de diamètres et hauteurs variés, destinés au chauffage et au maintien en température de pièces cylindriques.

Les transferts de chaleur entre le collier et la pièce à chauffer, se font principalement par conduction, ou par rayonnement pour les colliers les plus puissants. Ils peuvent chauffer aussi bien les solides, que les liquides ou les gaz.



Figure IV.5: les colliers chauffants

IV.2.2.5. Relais 220V commande à 5V : Un relais est un commutateur électrique qui permet de commander un second circuit utilisant généralement une tension et un courant .Le relais est un mécanisme simple d'interrupteur on/off [11]: il se ferme lorsque l'entrée est à 5V et s'ouvre lorsque l'entrée est à 0V et qui, en contact, supporte 220V.

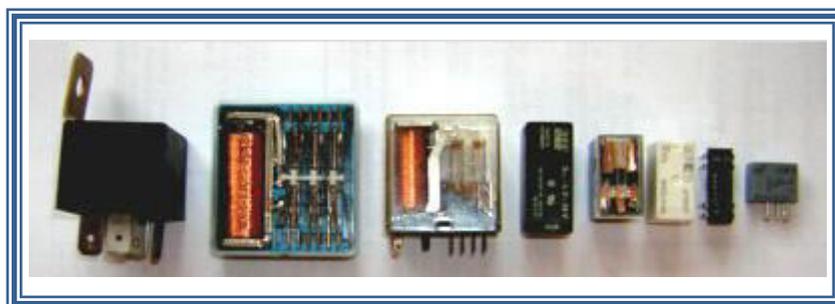


Figure IV.6: les Relais

IV.2.2.6. Command On\Off : Il s'agit globalement d'allumer et d'éteindre le système très rapidement pour obtenir un certain pourcentage de la puissance maximale.

IV.2.2.7. VCC 5v : une notation abrégée anglaise traduisible par Alimentation Tension Continue.

L'unité de mesure de la température est selon l'échelle choisi, soit en degré Celsius noté par °C.

IV.3. Fonctionnement du système :

Notre système peut se décomposé en 3 fonctions principales :

IV.3. 1. Mesure de température :

Le thermocouple se compose de deux fils de métaux différents soudés à l'endroit où la température doit être mesurée. Le point de jonction crée une différence de potentiel (millivolts) entre les fils, lequel augmente en même temps que la température.

Il est créé dès lors que les deux métaux différents entrent en contact, ce qui produit une faible tension en circuit ouvert au point de contact, qui varie en fonction de la température. figure(IV.7)

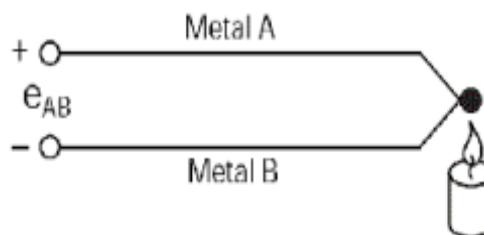


Figure IV.7: l'Effet Seebeck

Cette tension thermoélectrique est connue sous le nom de tension de Seebeck, d'après Thomas Seebeck qui l'a découverte en 1821. La tension n'est pas linéaire en fonction de la température [12]. Cependant, pour de petites variations de température, la tension est approximativement linéaire, soit :

$$\Delta V \approx S \Delta T \quad (IV.1)$$

où ΔV est la variation de la tension, S est le coefficient de Seebeck et ΔT la variation de la température.

La valeur réelle étant contrôlé (C) est détecté par le thermo contrôleur qui, ici, est modélisé comme un simple gain. le capteur produit un signal électrique ; souvent une tension ; qui est ensuite envoyé au calculateur via un convertisseur analogique-numérique.

IV.3. 2.chauffage :

L'unité de mesure de la température est selon l'échelle choisie, soit en degré Celsius noté par °C.

Nous souhaitons commander de la puissance fournie au collier à partir de microcontrôleur, le relais permet de commute cette puissance à la demande.

Le relais ouvre un circuit électrique quand la chaleur atteignait un certain stade (45°) et ferme ce circuit dès que l'on descendait en dessous de cette même température.

IV.3. 3. Contrôle de température :

Le signal original (la température) est récupéré sous forme numérique à l'intérieur de calculateur en divisant la tension représentative grâce au gain du capteur.

Le microcontrôleur lit la température toutes les 05 secondes et la compare à la valeur désirée et fournit une sortie à un élément de commande.

La tension de commande appliquée au relais, permet de l'amorcer ou de le désamorcer autrement dit :

si la valeur de la consigne est supérieur à la valeur de mesure, le dispositif de chauffage sera allumé pour chauffer la plaque (augmentation de la chaleur) .L'élément chauffant est désactivé une fois que la température désirée est atteinte.

IV.4. Conclusion

La sortie du procédé que l'on commande doit évoluer pour suivre la consigne demandée. Il faut donc à tout instant (ou périodiquement en régulation numérique) appliquer, à l'entrée de puissance du procédé, la commande appropriée. Cette commande est calculée par un ensemble de traitements d'informations, le correcteur, qui utilise des opérateurs (sommateurs, gains, intégrateurs, dérivateurs) élaborant la commande à partir du signal d'erreur et des mesures auxiliaires disponibles. Nous allons voir dans le prochain chapitre comment adapter un AG pour ajuster les paramètres du contrôleur PID qui minimise l'erreur du système.

CHAPITRE V

IMPLEMENTATION

ET

RESULTATS

V.1. Introduction

L'implémentation est la phase la plus importante après celle de la conception. Le choix des outils de développement influence énormément sur le coût en temps de programmation, ainsi que sur la flexibilité du produit à réaliser. Cette phase consiste à transformer le modèle conceptuel établi précédemment en des composants logiciels formant notre système. Dans ce chapitre, nous allons commencer par détailler l'adaptation de l'algorithme génétique pour résoudre le problème d'optimisation des paramètres d'un régulateur de température de type PID.

Ensuite la description de l'environnement de travail et une grande partie sera consacrée aux différents composants de l'algorithme ainsi que son principe de fonctionnement.

V.2 Adaptation d' AG à l'optimisation du régulateur PID pour la commande de Température:

L'objectif de cette partie est d'appliquer la stratégie d'optimisation des paramètres du PID par l'utilisation des AG à fin de trouver l'ensemble des paramètres optimisés

$\mathbf{E_{opt}} = [k_p^{(opt)}, k_i^{(opt)}, k_d^{(opt)}]$ de façon à ce que la réponse en vitesse du système en boucle fermée soit stable et la plus robuste possible .(Figure :V.1)

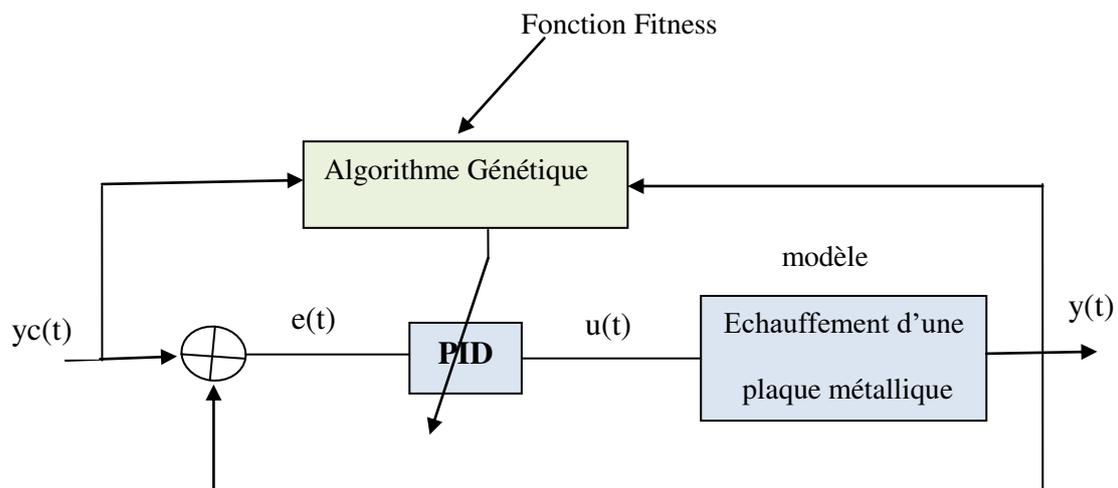


Figure (V.1): Principe de l'optimisation d'un contrôleur PID par algorithme génétique

Dans ce qui suit ,on va détailler les différents composants de l’algorithme génétique adapté à la résolution du notre problème.

V.2.1 Le codage

Nous avons utiliser un codage de type binaire, un chromosome est donc une suite de bits en codage binaire de longueur $l=32$ bits décomposée en trois gènes de la manière suivante :11 bits représentant le paramètre **KP**, 11 bits représentant le paramètre **KI** et 10 bits représentant le paramètre **KD** , comme le montre la figure :

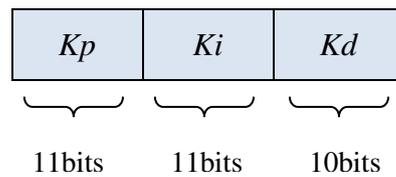


Figure (V.2) : Représentation binaire d’un chromosome

Les tableaux V.1, V.2 et V.3 suivants présentent le codage de dix (10) individus d’une population composée de cent (1000) individus :

1	1	1	0	1	1	1	1	0	1	1
1	1	1	0	1	1	0	1	1	1	1
0	1	1	1	1	0	1	0	0	0	1
1	1	0	1	1	1	1	0	1	1	1
1	1	0	1	1	1	0	1	1	0	1
1	1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	1	0	1	0	1	1
1	0	1	1	0	1	0	1	0	0	0
1	1	1	1	1	0	0	1	0	0	1
1	1	1	1	1	1	0	1	0	1	0

Tableau (V.1) : Représentation du gène (Kp) de dix individus.

1	1	1	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	0	1	1
1	1	1	1	1	1	1	1	1	0	1
1	1	0	1	1	1	1	1	1	1	1
1	1	0	1	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	1	1	0

Tableau (V.2) : Représentation du gène (Ki) de dix individus.

1	1	1	1	1	1	0	1	1	0
0	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	1
1	1	0	1	1	0	0	1	0	1
1	0	1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	0	0	0
1	1	1	1	1	1	1	0	1	1

Tableau (V.3) : Représentation du gène (Kd) de dix individus.

La figure suivante représente un individu (les trois gènes : Kp, Ki et Kd)

1	1	1	0	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1	0	1
1	1	1	1	1	1	1	0	1	1	0

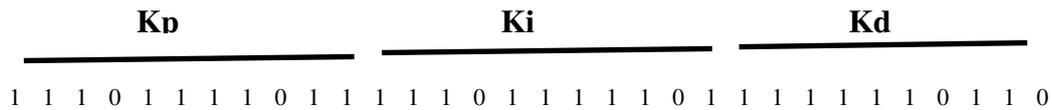


Figure (V.3) : Représentation d'un individu (codage binaire).

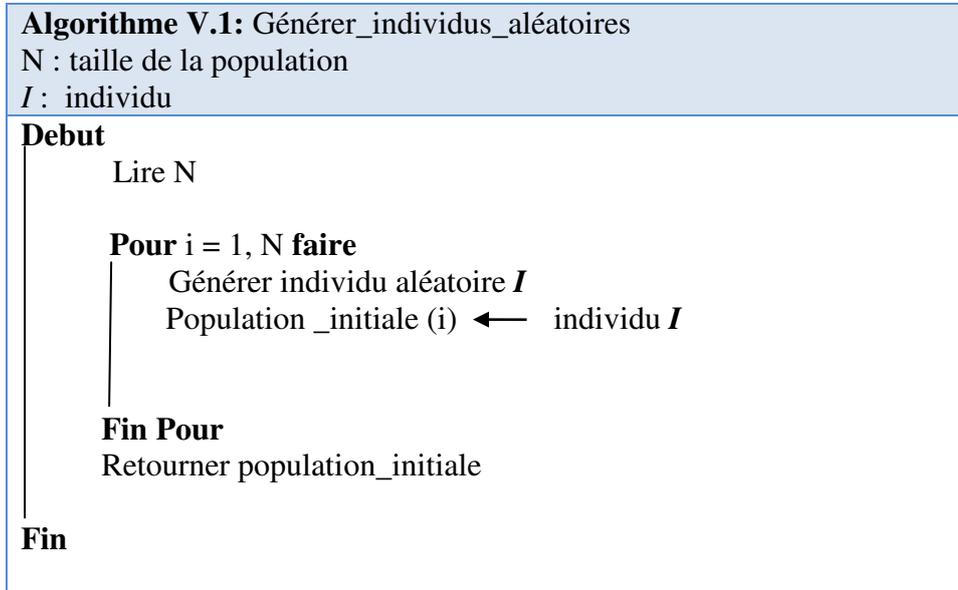
Remarque : la partie décimale de chaque paramètre (gène) est composée de 5 bits.

Kp	Ki	Kd
59.27	63.29	31.22
59.15	63.27	15.31
30.17	63.29	29.31
55.23	55.31	31.31
55.13	52.15	31.31
63.23	63.31	31.29
63.11	59.31	11.50
45.40	63.31	22.10
62.90	63.23	27.80
63.10	47.30	31.27

Tableau (V.4) : Représentation de dix individus (en nombres réels).

Dans le tableau V.4 la première colonne représente le paramètre K_p , la deuxième colonne représente le paramètre K_i et la troisième colonne représente le paramètre K_d .

V.2.2 Initialisation : Lorsqu'on crée une population initiale, les valeurs des paramètres (les allèles des gènes) sont créées aléatoirement avec une distribution uniforme



V.2.3 Evaluation de la population

Chaque individu de la population est évalué par une fonction objectif (fitness) qui représente sa performance. Dans notre projet, la fonction objectif (fitness) représente la somme des erreurs fournies par le système dans le temps t alors nous avons utilisé l'intégrale de l'erreur absolue pondérée par le temps (ITAE) comme critère d'optimisation. La fonction objectif est notée f :

$$f = ITAE = \int |e(t)| dt \quad (V.1)$$

V.2.4 La Sélection

La sélection est de type proportionnel, puisqu'il s'agit du problème de la minimisation, la fonction d'adaptation est l'inverse de la valeur de l'objectif ITAE i.e. $f_{ad} = 1 / f_{obj}$.

Deux individus sont choisis au hasard et on les fait "combattre", celui qui a la fitness la plus faible l'emporte avec une probabilité p comprise entre 0.5 et 1. (Figure :V.4)

Algorithme V.2: Sélection par tournoi binaire
 Une population d'individus : (vecteur d'individus) $E(x_1, x_2, \dots, x_n)$
 N : taille de la population
 $f(f_1, f_2, \dots, f_n)$ (Vecteur d'évaluation
 t : tournoiement ; $t > 1$;

```

Best ← tirage avec remise d'individus de E
Pour  $i=2, N$  faire
  Next ← tirage avec remise d'individus de E
   $r \leftarrow \text{Random}(0,1)$ 
  Si  $f(\text{Next}) > f(\text{Best})$  Alors  $\text{Best} \leftarrow \text{Next}$ 
  Si  $r > 0.5$  Alors return Best
Fin Pour

```

Figure (V.4) : Sélection par tournoi

V.2.5 Le croisement

Le but de ces deux opérateurs est d'améliorer la qualité des résultats obtenus et garantir la diversité de la population, le nombre d'individus sélectionnés pour cette opération doit être $N/2$, où N représente la taille de population, parce que chaque deux individus génèrent deux autres par la recombinaison (croisement) suivant l'algorithme qui suit :

Algorithme V.3: Croisement

T_{pop} : Taille de la population

P_c : probabilité de croisement

On génère une permutation aléatoire $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_{T_{\text{pop}}})$ de $(1, 2, \dots, T_{\text{pop}})$.

pour $i = 1, \dots, (T_{\text{pop}}/2)$ **faire**

On génère μ uniformément dans $[0, 1]$.

si $\mu \leq p_c$ **faire**

On applique le type de croisement choisi avec pour P_1 le $(\sigma_{2i-1})^{\text{ème}}$
 individu de la population et pour P_2 , le $(\sigma_{2i})^{\text{ème}}$

Fin si

Fin pour

V.2.6 La mutation

Après l'opération de croisement une nouvelle génération sera créée de taille N, puis l'opérateur de mutation est intervenu pour modifier les individus comme le montre l'algorithme ci après :

Algorithme V.4: Mutation

Input :a chromosome (i.e offspring) generated by Crossover operation *chromosome*

Output :the mutational chromosome *chromosome*

P_m : probabilité de mutation

L : taille de chromosome

Let n_0 and n_1 to be number of 0 bits and 1 bits in *chromosome*

Let $p_0 = P_m$, $p_1 = P_m * n_0 / n_1$;

for (j =1 to L) {

 Generate a random number r , $0 \leq r \leq 1$;

If (*chromosome*_{1,j} = 1 && $r < p_1$) *chromosome*_{1,j} = 0;

else if (*chromosome*_{1,j} = 0 && $r < p_0$) *chromosome*_{1,j} = 1;

}

Return *chromosome* ;

V.2.7 Elitisme :

Une technique très souvent utilisée avec l'algorithme génétique est l'élitisme.

L'élitisme est la conservation du meilleur individu dans la génération suivante. Après le

croisement, on compare le meilleur individu de la nouvelle génération avec le meilleur

individu de la génération précédente. Le meilleur individu est conservé et l'autre est détruit.

Cette technique permet de s'assurer que la fonction objective augmente (diminue) toujours si

on maximise (minimise) une fonction.

L'algorithme suivant résume les étapes de la simulation :

Algorithme V.5: Optimisation d'un PID par algorithme génétique simple

Donnée : obj=[] ;Nouvelle-population=[] ;max=nombre de génération maximal ;
nbr-ch=taille de "population" ; nbr-gen=0 ;temps de la simulation

Etape 1 : Initialisation

bits=calculerbits (interval,précision) ;
population=créer-pop(bits,nbr-ch) ;

While nbr-gen ≤ max

Etape 2 : Décodage

Etape 3 : Evaluation

For i ← 1 : nbr-ch

construire le contrôleur PID :Gc;
BO=series (G,Gc) ;
BF=feedback (BO,1) ;
Y= step (BF) ;
Erreur = 1-Y;
 f_{obj} =ITAE=sum (abs (temps.erreur)) ;
{ f_{obj} } ∪ ITAE;

End For

Etape 4 : Sélection proportionnelle :

Sélection par tournoi

Etape 5 :Croisement

population=croisement(Nouvelle-population) ;

Etape 6 :Mutation

population=mutation(population) ;
nbr-gen = nbr-gen +1 ;

End While

Le tableau suivant résume les paramètres de l'AG.

Paramètre	valeur
taille de la population(nbr-ch)	1000
précision	0.01
intervalle	[0 , 100]
Probabilité de croisement	0.8
Probabilité de mutation	0.1
nombre de génération (nbr-gen)	50

Tableau (V.5): paramètre de l'AG

V.3 Environnement de travail

L'environnement de travail est constitué par deux parties nommées environnement matériel et environnement logiciel.

V.3.1 Environnement matériel

Le développement de l'environnement matériel est caractérisé par :

1. Système d'exploitation : Windows XP Professionnel.
2. CPU : Pentium M, 1.6 GHz 3. Mémoire : 512 Mo

V.3.2 Environnement logiciel

L'environnement logiciel consiste les composants suivants :

- Java 2 Platform, Enterprise Edition J2EE JDK 1.5
- Outil de développement Eclipse

V.3.2.1 le langage de programmation : Java

Pour implémenter notre système, le langage de programmation java est le mieux adapté. En effet, Le langage Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy, cofondateur de Sun Microsystems. Java a été officiellement présentée le 23 mai 1995 au SunWorld. La société Oracle racheta alors la société Sun en 2009, ce qui explique pourquoi ce langage appartient désormais à Oracle. La particularité et l'intérêt de Java réside dans sa portabilité entre les différents systèmes d'exploitation tels que Unix, Windows, ou MacOS. Un programme développé en langage Java, peut ainsi s'exécuter sur toutes les plateformes, grâce à ses Framework associés visant à garantir cette portabilité [2].



Figure (V.5): logo java

V.3.2.2 Outil de développement : Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et frameworks) mais aussi d'AGL recouvrant modélisation, conception, test, gestion de configuration, reporting [2].



Figure (V.6): logo éclipse

V.4 Développement de l'application

Dans cette partie, nous allons présenter les différentes phases de la réalisation de notre projet en mentionnant des imprimés écrans de notre application.

V.4.1 Implémentation de l'applet

L'implémentation a été développée à partir de plusieurs classes abstraites adaptées des autres classes et qui constituent le fondement logique des algorithmes génétiques ainsi que le contrôleur, soit :

V.4.1.1 La classe Gene : représente l'élément de codage des individus qui composent la population. Cette classe ne comporte qu'une méthode abstraite de la mutation d'un gène. Généralement, la mutation est associée au gène et elle ne représente que la création aléatoire d'un gène.

V.4.1.2 La classe Individu : est représentée par un ensemble de gènes ainsi qu'une valeur d'adaptation qui caractérise l'individu. Les méthodes liées à la valeur d'adaptation de cette classe permettent de lire et d'écrire cette valeur de l'individu. D'autres méthodes permettent d'accéder à un des gènes spécifiques de l'individu, d'effectuer un croisement avec un autre individu, et de réaliser les mutations des gènes de l'individu.

Les méthodes de croisement, de mutation et de calcul de la valeur d'adaptation sont des méthodes définies comme abstraites, car leur implémentation dépend du problème à résoudre. Il faut donc redéfinir les méthodes selon le cas. Par contre, on peut définir une certaine fonctionnalité générale, comme dans le cas de la méthode de mutation suivante :

Pour chaque gène de l'individu

1. On tire au sort un nombre entre 0 et 1.
2. Si ce nombre est plus petit que la probabilité de mutation, alors on fait appel à la méthode de mutation de ce gène.

Si un des gènes a été modifié, alors on recalcule la valeur d'adaptation.

V.4.1.3 La classe Population : permet de créer un ensemble d'individus et de les faire évoluer grâce aux opérateurs génétiques :

- **Croisement :** On répète les étapes suivantes jusqu'à ce que le nombre d'individus soit augmenté dans les proportions fixées par le taux de reproduction. Puis, on tue autant d'individus, parmi les plus mauvais, qu'il est nécessaire pour revenir au nombre d'individus initial de la population.
- **Mutation :** On fait appel à la méthode mutation de chacun des individus en passant en paramètre la probabilité de mutation. Si l'individu est modifié par la mutation, on le range en fonction de sa nouvelle valeur d'adaptation.

V.4.1.4 La classe PID :

Le calcul de Commande se base sur l'algorithme suivant :

Algorithme V.6: calcul de la Commande

Tous les x millisecondes, faire :

```
erreur = consigne - mesure;
somme_erreurs += erreur;
variation_erreur = erreur - erreur_précédente;
commande = Kp * erreur + Ki * somme_erreurs + Kd * variation_erreur;
erreur_précédente = erreur
```

V.4.2 Structures de données

Dans cette section nous décrivons la structure de données que nous utilisons pour présenter les paramètres génétiques de notre application.

- **Gène:** Une représentation d'une variable (d'entrée ou de sortie). Une variable a un nom et un type. La classe Gene présente la mise en œuvre de ce paramètre.
- **Chromosome:** Une liste de gènes. Dans la mise en œuvre du chromosome on utilise la structure List de Java telle que:
List chromosome = new ArrayList ();
- **Individu:** La structure Individu est la représentation d'une opération où on définit les données suivantes:
 - nom :** Nom de l'opération;
 - Chromosome d'entrées:** Une liste définissant le message d'entrée de l'opération
 - Chromosome de sorties:** Une liste définissant le message de sortie de l'opération
 - double val :** Un double qui détermine la valeur d'adaptation de l'opération avec l'opération cible;
- **Population:** Une collection d'individus, appelée aussi génération, représente l'élément de codage des individus qui composent la population. Cette classe ne comporte qu'une méthode abstraite de la mutation d'un gène.

V.5 Résultats

Dans le tableau suivant, la première colonne représente le temps de mesure et la deuxième colonne représente la température de la plaque métallique détectée par le thermocouple dans chaque (05) secondes.

Temps (s)	Température de la plaque
0	30,00
5	30,00
10	30,00
15	30,00
20	30,00
25	30,00
30	30,00
35	30,00
40	30,00
45	30,00
50	30,50
55	30,50
60	30,50
65	31,00
70	31,00
75	31,00
80	31,50
85	31,50
90	32,00
95	32,00
100	32,50
105	32,50
110	33,00
115	33,00
120	33,50
125	34,00
130	34,00
135	34,00
140	34,00
145	34,00
150	34,00
155	35,00
160	35,00
165	35,00
170	35,00
175	36,00
180	37,00
185	37,00
190	38,00
195	38,00
200	39,00
205	39,00
210	40,00
215	41,00
220	42,00
225	42,50
230	42,50
235	43,00
240	43,00
245	43,00
250	43,00
255	43,50
260	43,50
265	43,50
270	44,00
275	44,00
280	44,00
285	44,50
290	44,50
295	44,50
300	44,50
305	44,50
310	45,00
315	45,00
320	45,00
325	45,00
330	45,00
335	44,50
340	44,50
345	44,50
350	45,00
355	45,00
360	45,00
365	45,00
370	45,50
375	45,50
380	45,00
385	45,00
390	45,00
395	45,00
400	45,00
405	45,00

Tableau (V.6): la température obtenue

La figure suivante représente la courbe de température mesurée de la plaque métallique chaque 05 seconde :

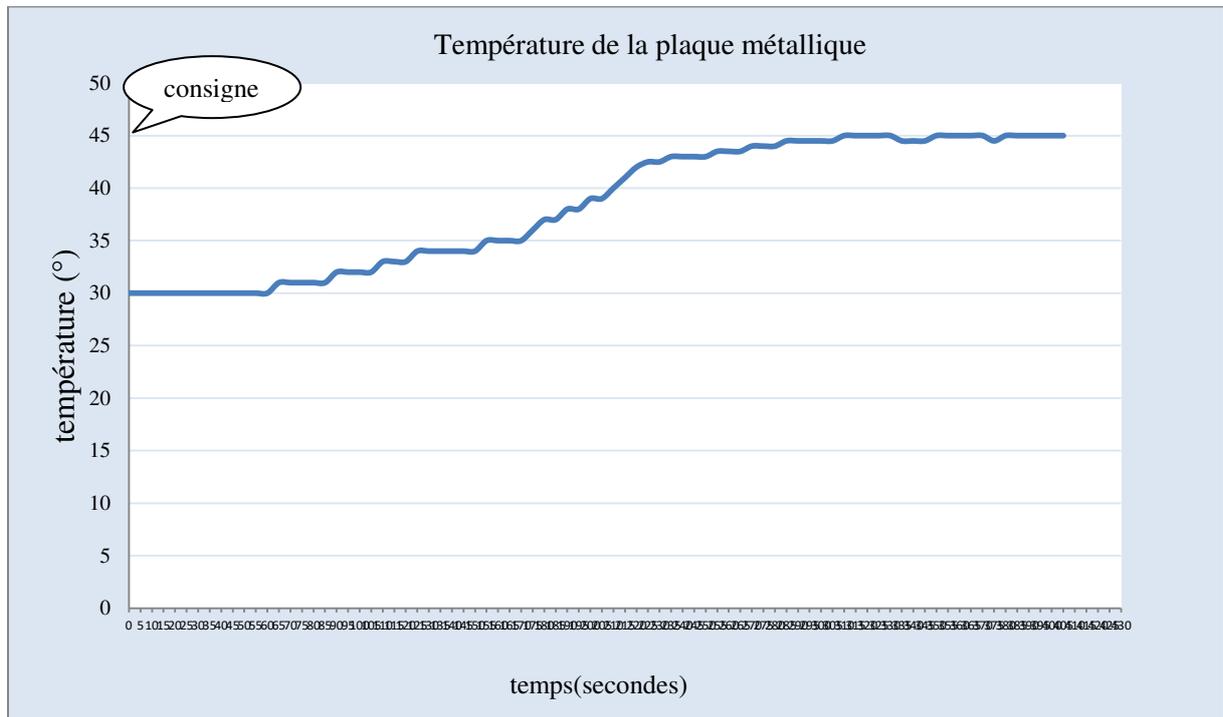


Figure (V.7) : variation de température

N°génération	Erreur		
0	5,00	25	0,35
1	4,39	26	0,35
2	3,97	27	0,35
3	3,03	28	0,35
4	2,50	29	0,35
5	2,29	30	0,35
6	1,80	31	0,35
7	1,76	32	0,35
8	1,66	33	0,35
9	1,66	34	0,31
10	1,66	35	0,31
11	1,66	36	0,31
12	1,66	37	0,31
13	1,66	38	0,31
14	1,07	39	0,31
15	0,50	40	0,31
16	0,47	41	0,31
17	0,44	42	0,30
18	0,44	43	0,30
19	0,44	44	0,30
20	0,44	45	0,30
21	0,35	46	0,30
22	0,35	47	0,30
23	0,35	48	0,30
24	0,35	49	0,19
		50	0,18

Tableau (V.7): l'Erreur produite par le système

Le tableau (V.7) représente l'erreur produite par le système de meilleur individu dans chaque itération.

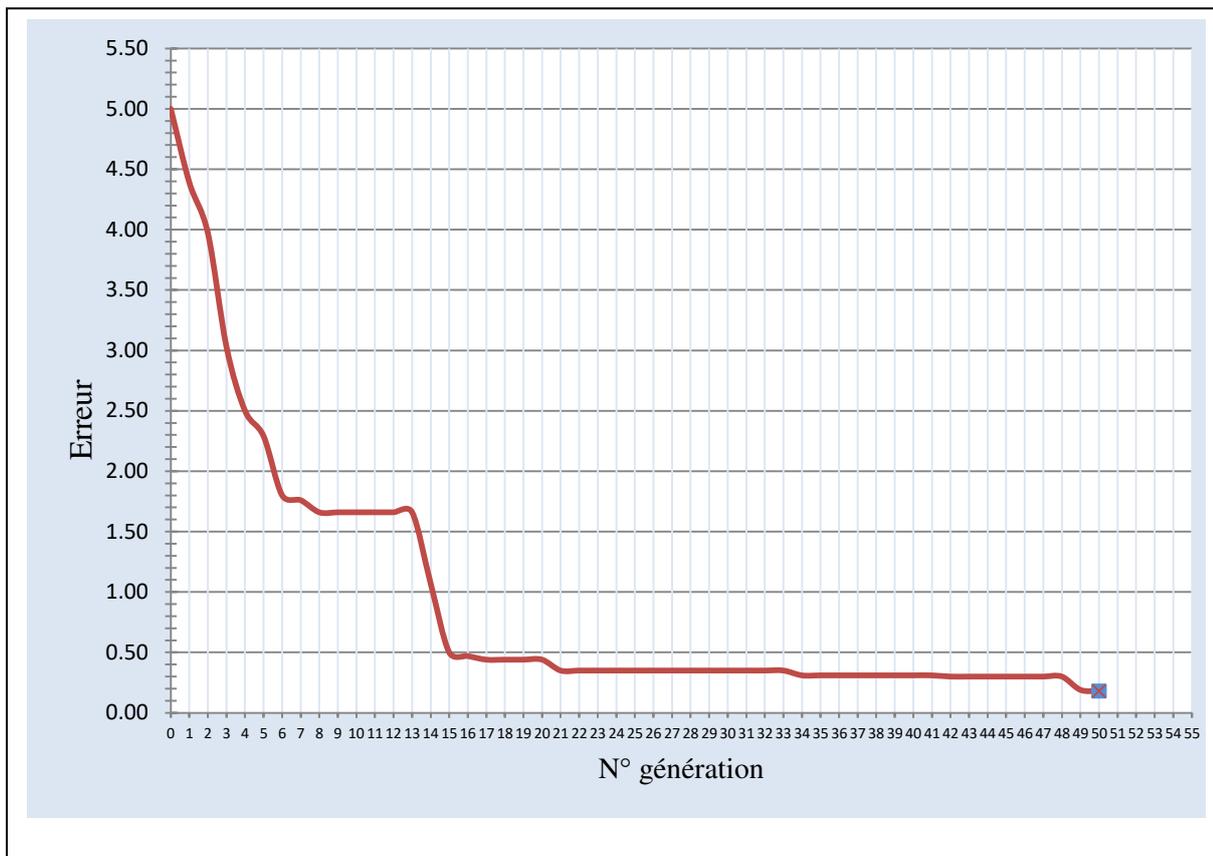


Figure (V.8) : Erreur produite par le système

D'après la figure V.8, on remarque que la courbe de l'erreur de meilleur individu en fonction de génération est en état décroissant, la valeur d'erreur en autre terme du meilleur individu se diminue d'une itération à une autre

Après 50 itérations \longrightarrow **Erreur = 0.18**

Pour cette minimum valeur d'erreur et une consigne égale à 45° , on a obtenu les trois paramètres optimaux du contrôleur PID :

$$\left\{ \begin{array}{l} K_{p(\text{opt})} = 1.05 \\ K_{i(\text{opt})} = 0.66 \\ K_{d(\text{opt})} = 6.34 \end{array} \right.$$

V.6 Conclusion

Dans ce chapitre, nous avons utilisé un algorithme génétique simple pour rechercher les trois paramètres (K_d , K_p et K_i) qui donnent une réponse optimale en sens d'un critère de performance temporelle exprimé par ITAE. au début nous avons commencé par présenter l'environnement de travail, ensuite nous avons décrit les composants essentiels de l'algorithme génétique choisi pour la résolution du problème posé. En deuxième temps, nous avons présenté les résultats numériques obtenus par notre algorithme. Les résultats obtenus à travers un exemple de commande d'un système de contrôle de température sont très satisfaisants

CONCLUSION GENERALE

Pour notre projet de fin d'étude l'objectif été de contrôler un système de température à l'aide d'un PID. Nous avons cherché à atteindre une consigne de température (la poursuite ou l'asservissement) et d'éliminer des perturbations pour que la valeur souhaitée reste constante. Lorsque le contrôleur de température PID est réglé de façon optimale, il permet de minimiser l'écart par rapport au point de consigne et offre une réponse rapide aux perturbations ou changements de valeurs de consigne, avec un dépassement minimal.

A cet effet nous avons essayé de déterminer les paramètres optimaux du régulateur à l'intérieur du domaine de stabilité qui garantissent des bonnes performances en boucle fermée. Les algorithmes génétiques, qui sont connu comme des outils puissants pour l'optimisation globale, ont été proposés pour répondre à ce besoin et pour minimiser certains critères de performance comme le critère ITAE.

Cependant, ce système reste à améliorer, et donc nous proposons comme travail futur :

- Améliorer les résultats par la mise en œuvre d'une ou plusieurs heuristiques pour générer la population initiale selon le problème étudié pour générer des solutions faisables.

- Améliorer la valeur des paramètres du PID à partir des autres algorithmes d'optimisation comme la logique floue ou les colonies de Fourmies artificiel.

Références Bibliographiques

- [1] Cours préparation aux agrégations internes de mécanique et génie mécanique (Asservissements):
réalisé par F.Binet 2005.
- [2] Encyclopédia Universalis.
- [3] Cours de Systèmes Asservis : J.Baillou, J.P.Chemla, B. Gasnier,.LethiecqPolytech'Tours.
- [4] Asservissement par PID : Daniel Ross, Etienne Deguine, Michael Camus 30 mars 2010.
- [5] Asservissement des systèmes linéaires à temps continu :Edouard Laroche
(laroche@unistra.fr).Housseem Halalchi (Housseem.Halalchi@lsiit.u-strasbg.fr) Université
de Strasbourg.
- [6] O.Aidan, « handbook of PI and PID controller tuning rules 1 », published by Imperial College
Press,2006.
- [7] Goldberg, D.E. Genetic algorithms in search, optimization, and machine learning.
Addison-Wesley,1989.
- [8] Y.Chaudhari, « Design and implementation of intelligent controller for a continuous
stirred tank reactor system using genetic algorithm », International Journal of Advances in
Engineering and Technology,India, pp.325-335, Mars, 2013.
- [9] www.resistanceelectrique.com/produits/controle-de-la-temperature/thermocouples/
- [10] cours de microcontrôleurs : Christophe Durand Année universitaire 2009-2010,
Université Joseph Fourier Polytech' Grenoble.
- [11] Cours pour l'apprentissage des bases de l'électronique et de la programmation sur
Arduino , <https://arduino.education>
- [12] National Instruments ;Tutorial :Mesure des températures par les thermocouples