

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE DU 20 AOUT 1955 SKIKDA

Faculté de Technologie
Département : Pétrochimie



Mémoire

En vue de l'obtention du diplôme de Master.

Filière : Industrie pétrochimique

Spécialité : Automatisation et contrôle en industrie pétrochimique

Thème :

**Commande intelligente d'un système industriel au
laboratoire (FESTO MPS PA)**

Par :

- *NABET Nassim*
- *DORBI Tarek*

Encadreur :

Dr. BENDIB Riad

Soutenu publiquement le : 10 / 07 / 2023 devant le jury composé de :

Président

METATLA Hassina

U-Skikda

Examineur

KERBOUA Adlen

U-Skikda

Promotion : 2022/2023

Remerciements

Ce travail est le résultat d'un dur labeur et de beaucoup de sacrifices : nos remerciements vont d'abord à ALLAH, créateur de l'univers qui nous a doté d'intelligence, et nous a maintenu en santé mentale et physique pour mener à bien cette année d'étude, à qui nous profondément et sincèrement reconnaissons.

Nous tenons à remercier toutes les personnes commençons par nos parents, Ma familles par leurs sacrifices pour nous fournir les meilleures conditions de travail, ainsi mes amis qui ont contribué au succès de notre travail et qui nous ont aidé lors de la rédaction de ce mémoire.

Je vous remercier, mon Encadreur Monsieur BENDIB Riad, qui fut le premier à nous faire découvrir le sujet qui a guidé notre mémoire pour avoir accepté de nous guider par sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre réflexion.

Nous remercions également tous les membres du jury pour nous avoir honorés par leur présence et pour avoir accepté d'évaluer notre travail.

Dédicaces

NABET NASSIM

Je dédie ce modeste travail à mon père et ma belle belle mère et Ma sœur

Nour el houda, Ma tante Louiza

Et mon frère Hemza

Tous mes oncles et toute la famille

Mes amis et ma deuxième famille

Et toute ma promotion

Qu' est toujours trouvé même dans les moments les plus sombres de ma vie et
elle se fait de se souvenir d'allumer la lumière .

DORBI Tarek

Je dédie ce modeste travaille à mon père et ma belle belle mère , mes frères
et mes sœurs

Tous mes oncles et toute la famille

Mes amis et ma deuxième famille

Et toute ma promotion

Qu' est toujours trouvé même dans les moments les plus sombres de ma vie et
elle se fait de se souvenir d'allumer la lumière .

Résumé (Français) :

Ce mémoire de recherche se concentre sur l'utilisation des algorithmes génétiques, de la logique floue et du glissement flou pour améliorer le contrôle d'un système complexe, en l'occurrence le système contrôlé FESTO. Les algorithmes génétiques permettent d'optimiser les paramètres de contrôle, tandis que la logique floue et le glissement flou permettent de modéliser l'incertitude et d'ajuster les gains de contrôle de manière adaptative. L'étude présente les fondements théoriques de ces approches, décrit la méthodologie de recherche et analyse les résultats obtenus. Les résultats démontrent que l'intégration de ces techniques améliore l'adaptabilité et les performances du système de contrôle FESTO dans des conditions complexes et incertaines. Cette recherche ouvre de nouvelles perspectives pour l'amélioration du contrôle des systèmes complexes.

ملخص (عربي) :

تركز هذه الدراسة على استخدام خوارزميات الجينات المطابقة والمنطق الغامض والانزلاق الغامض لتحسين التحكم في نظام معقد محدد، وهو نظام FESTO. تُسمح خوارزميات الجينات المطابقة بتحسين معاملات التحكم، بينما يُمكن للمنطق الغامض والانزلاق الغامض نمذجة عدم اليقين وتعديل معاملات التحكم بشكل مُتكيف. توفر الدراسة الأسس النظرية لهذه الأساليب، وتصف منهجية البحث وتحليل النتائج المُتحققة. تُظهر النتائج أن دمج هذه التقنيات يُحسن قابلية التكيف وأداء نظام التحكم FESTO في ظروف معقدة وغير مُؤكدة. تفتح هذه الدراسة آفاقاً جديدة لتحسين التحكم في الأنظمة المعقدة.

Abstract (English):

This research paper focuses on the utilization of genetic algorithms, fuzzy logic, and fuzzy sliding for enhancing the control of a complex system, specifically the FESTO controlled system. Genetic algorithms are employed to optimize control parameters, while fuzzy logic and fuzzy sliding enable the modeling of uncertainty and adaptive adjustment of control gains. The study presents the theoretical foundations of these approaches, describes the research methodology, and analyzes the obtained results. The findings demonstrate that the integration of these techniques enhances the adaptability and performance of the FESTO control system under complex and uncertain conditions. This research opens up new perspectives for improving control in complex systems.

Sommaire

Introduction générale.....	1
Chapitre I : L'algorithme génétique	
I.1.Introduction.....	3
I.2.Principes généraux.....	3
I.3.Les algorithmes génétiques : mécanisme de fonctionnement.....	5
I.3.1. Codage des variables.....	5
I.3.1.1. Codage binaire.....	6
I.3.1.2. Le codage réel.....	6
I.3.2. Génération aléatoire de la population initiale.....	7
I.4. Les opérateurs génétiques.....	8
I.4.1. La sélection.....	8
I.4.2. Le croisement dans la génétique algorithme :.....	9
I.4.3. La mutation.....	11
I.4.4. Élitisme (facultatif).....	12
I.4.5. Réplication.....	12
I.5.La fonction de cout.....	12
I.6. L'évolution dans la génétique algorithme.....	13
I.7. Avantages et limitations.....	14
I.7.1. Les avantages.....	14
I.7.2. Les limitations.....	15
I.8.Conclusion.....	15
Chapitre II : La logique flou	
II.1. Introduction.....	17
II.2. Généralités sur la logique floue.....	17
II.2.1.Historique.....	17
II.2.2 Quelques domaines d'applications.....	17

II.2.3. Les avantages de la logique floue	18
II.2.4. Les inconvénients de la logique floue.....	19
II.3. La différence entre la logique floue et la logique booléenne	20
II.4. Les fonctions d'appartenance	21
II.5. Les opérateurs dans la logique floue	23
II.6. La logique floue et l'intelligence artificielle	25
II.6.1. Les étapes de traitement flou	26
II.6.1.1. La fuzzification	26
II.6.1.2. Le degré d'activation.....	27
II.6.1.3. L'implication.....	27
II.6.1.4. L'agrégation	28
II.6.1.5. La défuzzification.....	29
II.7. conclusion.....	30
Chapitre III : Le mode glissant	
III.1. Introduction	31
III.2. Historique	31
III.3. Principe de la Commande par mode glissant :	32
III.3.1. Mode de convergence (MC) :	32
III.3.2. Mode de glissement (MG) :	32
III.3.3. Mode de régime permanent (MRP) :	32
III.4. Conception de la commande par mode de glissant	33
III.4.1. Le choix de la surface	33
III.4.2. Les conditions de convergence	34
III.4.3. La loi de commande :	34
III.5. Formalisme classique du régime glissant.....	35
III.5.1. Régime glissant idéal :	35
III.5.2. Régime glissant réel :	36

III.5.3. Phénomène de broutement :	37
III.5.4. L'élimination du phénomène de broutement.....	38
III.5.5. Commande par régime glissant d'ordre supérieur.....	38
III.6. Avantages et limitations de SMC.....	39
III.6.1. Avantages :	39
III.6.2. Limitations.....	39
III.7. Conclusion.....	39
Chapitre IV : Simulation et résultats	
IV.1. Introduction.....	39
IV.2. Description du processus (Poste de travail compact FESTO MPS PA)	40
IV.3. Contrôle de système avec la génétique algorithmique	42
IV.3.1. L'optimisation d'un contrôleur PI en utilisant la GA.....	42
IV.3.2. Les paramètres de la génétique algorithmique	43
IV.3.3. Résultats et discussion :.....	43
IV.4. Contrôle de système par la logique floue.....	47
IV.4.1. Réglage du contrôleur PI avec la logique floue	47
IV.4.1.1. Règles floues pour le réglage de kp	47
IV.4.1.2. Règles floues pour le réglage de ki	48
IV.4.2. Simulation de logique floue-PI.....	48
IV.4.2.1. Création du système d'inférence flou (SIF),.....	48
IV.4.2.2 Fuzzification des variables d'entrées et de sorties	50
IV.4.2.3. Les règles de SIF :	51
IV.4.2.4. La défuzzification :	52
IV.4.2.5 Test du système d'inférence flou (SIF)	53
IV.5. Le mode glissant flou.....	55
IV.5.1. Contrôle de mode glissant flou basé sur un contrôle équivalent.....	55
IV.5.2. Conception du contrôle flou	56

IV.5.3. Application sur système :	57
IV.5.4. Résultats:	60
IV.5.4.1. Le mode glissant traditionnel.....	60
IV.5.4.2. Le mode glissant flou.....	63
IV.6. Conclusion	65

Liste des figures

Chapitre I

Figure I. 1.Principe générale des algorithmes génétiques	4
Figure I. 2. Illustration de codage des variables dans la génétique algorithme.....	6
Figure I. 3. Codage réel multidimensionnels	7
Figure I. 4. La sélection en génétiques algorithmes	9
Figure I. 5. Exemple d'un opérateur de mutation	12
Figure I. 6. Illustration d'une évolution dans la génétique algorithme	14

Chapitre II

Figure II. 1. Schéma d'une fonction triangulaire	21
Figure II. 2. Schéma d'une fonction trapézoïdale	22
Figure II. 3. Schéma d'une fonction gaussienne	22
Figure II. 4. Représentation de hauteur, noyau et support	23
Figure II. 5. Représentation d'un opérateur « NON ».....	24
Figure II. 6. Représentation d'un opérateur « ET »	24
Figure II. 7. Représentation d'un opérateur « OU».....	25
Figure II. 8. Processus de traitement flou.....	26
Figure II. 9. Processus de fuzzification	26
Figure II. 10. Activation des règles	27
Figure II. 11. L'implication.....	28
Figure II. 12.L'agrégation	29
Figure II. 13 La défuzzification.....	29

Chapitre III

Figure III. 1. Les différents modes pour la trajectoire dans le plan de phase	33
Figure III. 2. régime idéal.....	36
Figure III. 3. Régime glissant réel.....	37

Chapitre IV

Figure IV. 1. Le système FESTO	40
Figure IV. 2. Contrôle de niveau au système FESTO	41
Figure IV. 3. Schéma du contrôle pi	42
Figure IV. 4. Résultats d'optimisation de k_p et k_i	44
Figure IV. 5. Evolution des paramètres pendant les différents générations.....	44

Figure IV. 6. Les différentes réponses du système pendant la première génération.....	45
Figure IV. 7. Les réponses du système en 2ème génération	45
Figure IV. 8. Les réponses de système en 10 ème génération	46
Figure IV. 9. Les meilleurs réponse et fonction d'objectif de chaque génération	46
Figure IV. 10. Création de système d'inférence flou	50
Figure IV. 11. Fonctions d'appartenance de e et e'	51
Figure IV. 12. Fonctions d'appartenance de kp et ki	51
Figure IV. 13. Optimisation des valeurs de kp et ki.....	52
Figure IV. 14. Le signal de l'entrée.....	53
Figure IV. 15. Développement de kp et ki au cours du temps	54
Figure IV. 16. La réponse du système contrôlé.....	54
Figure IV. 17. Le modèle Simulink de mode glissant flou	58
Figure IV. 18 La perturbation en forme d'une fonction gaussienne	59
Figure IV. 19 Fonction d'appartenance de s(t)	59
Figure IV. 20. Fonction d'appartenance de μ	60
Figure IV. 21. La réponse de système contrôlé.....	61
Figure IV. 22. La commande u(t).....	61
Figure IV. 23. L'erreur entre le signal réel et souhaité	62
Figure IV. 24. La dérivée de l'erreur	62
Figure IV. 25. La surface de glissement.....	63
Figure IV. 26. La réponse de système contrôlé.....	64
Figure IV. 27. L'entrée de commande u(t)	64
Figure IV. 28. La variation de mu	65

Liste des tableaux

Chapitre I

Tableau I. 1. Le croisement en un point	9
Tableau I. 2. Le croisement en deux points.....	10
Tableau I. 3. Le croisement uniforme	10

Chapitre IV

Tableau IV. 1. Les paramètres de la GA	43
Tableau IV. 2. Règles floues d'optimiser le k_p	48
Tableau IV. 3. Règles floues pour optimiser le k_i	48

Introduction générale

Le contrôle des systèmes complexes est un défi majeur dans de nombreux domaines tels que l'ingénierie, l'automatisation industrielle et la robotique. Les systèmes réels présentent souvent des caractéristiques complexes, non linéaires et soumis à des perturbations ou à des incertitudes. Dans le but d'obtenir des performances optimales et une réponse adaptative, différentes approches de contrôle ont été développées et étudiées.

Dans cette étude, nous nous intéressons à l'utilisation des algorithmes génétiques, de la logique floue et du glissement flou pour le contrôle des systèmes complexes. Ces approches offrent des techniques puissantes pour modéliser et résoudre des problèmes de contrôle qui peuvent être difficiles à aborder avec des méthodes traditionnelles.

L'objectif de cette étude est d'explorer l'application et l'intégration des algorithmes génétiques, de la logique floue et du glissement flou dans le domaine du contrôle. Nous chercherons à démontrer comment ces approches peuvent être combinées pour résoudre des problèmes de contrôle complexes et améliorer les performances des systèmes dans des environnements incertains.

Pour atteindre cet objectif, Dans le premier chapitre de ce mémoire, nous aborderons le sujet de l'algorithme génétique. Nous explorerons les principes fondamentaux des algorithmes génétiques, leur fonctionnement et leur utilisation dans le contexte du contrôle des systèmes. En mettant l'accent sur l'optimisation des paramètres de contrôle.

Dans le deuxième chapitre, nous nous concentrerons sur la logique floue. Nous présenterons les principes fondamentaux de la logique floue, y compris les concepts de variables linguistiques, de fonctions d'appartenance et de règles floues. Nous examinerons comment la logique floue peut être utilisée pour modéliser les relations complexes entre les entrées et les sorties d'un système de contrôle.

Dans le troisième chapitre, nous explorerons le concept du mode glissant. Nous étudierons les fondements théoriques du mode glissant, y compris la notion de surface de glissement, les lois de commande et les conditions de stabilité. Nous analyserons comment le mode glissant peut être utilisé pour obtenir un contrôle précis et robuste, en réduisant l'impact des perturbations et des incertitudes.

Enfin, dans le quatrième chapitre, nous présenterons notre système contrôlé et les résultats de simulation obtenus. Nous décrirons le système étudié, ses caractéristiques et les objectifs de

contrôle spécifiques que nous cherchons à atteindre. Nous détaillerons les méthodes de simulation utilisées pour évaluer les performances du système avec l'application des algorithmes génétiques, de la logique floue, du mode glissant et mode glissant flou. Les résultats obtenus seront analysés et discutés pour évaluer l'efficacité des différentes approches et mettre en évidence les avantages de chacune dans le contexte de notre système contrôlé.

En intégrant ces différents chapitres, cette étude vise à fournir une compréhension approfondie des concepts clés de l'algorithme génétique, de la logique floue et du mode glissant, ainsi que de leur application dans le domaine du contrôle des systèmes. Les résultats obtenus à travers la simulation et l'évaluation du système contrôlé fourniront des informations précieuses sur les performances et les avantages des approches utilisées, ouvrant ainsi la voie à des améliorations et à des applications futures dans des domaines variés.

En résumé, cette étude vise à enrichir notre compréhension des techniques de contrôle basées sur les algorithmes génétiques, la logique floue et le glissement flou. En explorant leur potentiel et en les combinant de manière synergique, nous espérons ouvrir de nouvelles perspectives pour la résolution de problèmes de contrôle complexes et l'amélioration des performances des systèmes.

Chapitre I

L'algorithme génétique

I.1.Introduction

Le premier chapitre de ce mémoire se concentre sur l'algorithme génétique, une technique d'optimisation inspirée des principes de l'évolution biologique. L'algorithme génétique est largement utilisé pour résoudre des problèmes complexes et trouver des solutions efficaces dans de nombreux domaines, y compris le contrôle et l'optimisation des systèmes.

L'objectif de ce chapitre est de fournir une introduction approfondie à l'algorithme génétique, en expliquant ses principes fondamentaux, son fonctionnement et ses composants clés. Nous examinerons également comment l'algorithme génétique peut être appliqué au domaine du contrôle, en utilisant ses capacités d'optimisation pour améliorer les performances des systèmes de contrôle.

Nous allons présenter les concepts de base de l'algorithme génétique, tels que la représentation des individus, les opérateurs génétiques (crossover et mutation) et la sélection des individus. Nous aborderons ensuite les différentes étapes de l'algorithme génétique, y compris l'initialisation de la population, l'évaluation des générations, la reproduction et la mise à jour de la population.

I.2.Principes généraux

Les algorithmes génétiques sont des méthodes d'optimisation qui se basent sur des techniques inspirées de la génétique et de l'évolution naturelle, telles que les croisements, les mutations, et la sélection. Ils sont utilisés pour trouver les extrémités d'une fonction dans un espace de données. Pour les utiliser, il est nécessaire de disposer de cinq éléments fondamentaux.

Le premier élément est un système de codage pour la population. Cette étape associe une structure de données à chacun des points de l'espace d'état, et est généralement réalisée après une phase de modélisation mathématique du problème traité. Le choix du codage des données est crucial pour le succès de l'algorithme. Les codages binaires ont été très populaires à l'origine, mais les codages réels sont maintenant couramment utilisés pour optimiser des problèmes avec des variables continues dans des domaines applicatifs spécifiques.

Le deuxième élément est un mécanisme pour générer la population initiale. Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global. Si l'on ne connaît rien du

problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.

Le troisième élément est une fonction d'évaluation, également appelée fitness ou fonction d'évaluation de l'individu. Cette fonction prend ses valeurs dans R^+ et est utilisée pour sélectionner et reproduire les meilleurs individus de la population.

Le quatrième élément est un ensemble d'opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace d'état. L'opérateur de croisement recompose les gènes d'individus existants dans la population, tandis que l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'état.

Le cinquième et dernier élément est un ensemble de paramètres de dimensionnement, tels que la taille de la population, le nombre total de générations ou critère d'arrêt, ainsi que les probabilités d'application des opérateurs de croisement et de mutation. Ces paramètres sont ajustés en fonction du problème spécifique à résoudre.

Le principe général du fonctionnement d'un algorithme génétique est représenté sur la (figure I.1)

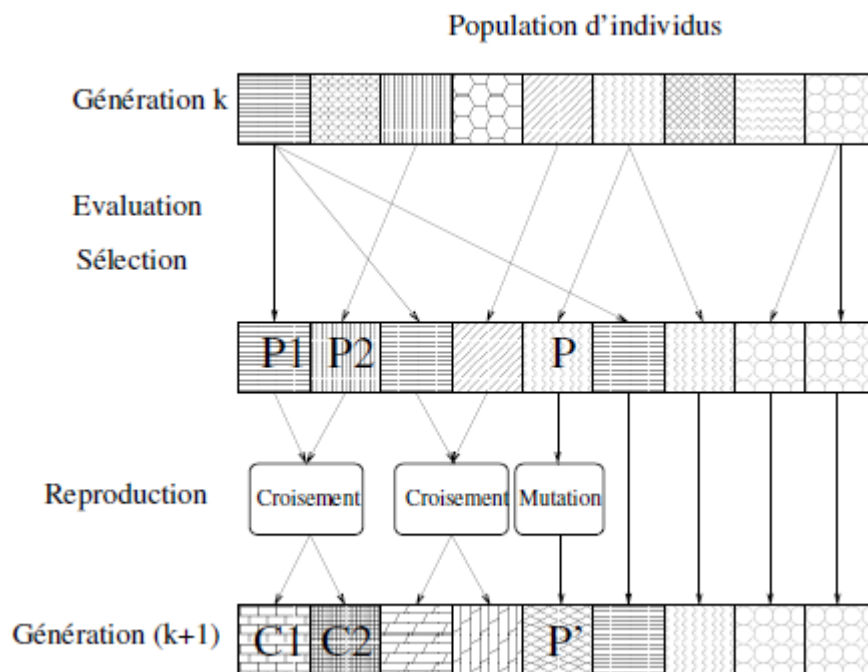


Figure I. 1.Principe générale des algorithmes génétiques

Le processus démarre par la génération aléatoire d'une population d'individus. Pour passer d'une génération k à la génération k + 1, trois opérations sont répétées pour chaque élément

de la population k . Tout d'abord, des couples de parents $P1$ et $P2$ sont sélectionnés en fonction de leur adaptation. Ensuite, l'opérateur de croisement leur est appliqué avec une probabilité P_c , généralement autour de 0.6, ce qui engendre des couples d'enfants $C1$ et $C2$. D'autres éléments P sont sélectionnés en fonction de leur adaptation, et l'opérateur de mutation leur est appliqué avec une probabilité P_m , qui est généralement très inférieure à P_c . Cela engendre des individus mutés $P0$. Enfin, les enfants ($C1$, $C2$) et les individus mutés $P0$ sont évalués avant d'être insérés dans la nouvelle population. La Figure 1.1 illustre le cas où les enfants et les individus mutés remplacent les parents. Différents critères d'arrêt de l'algorithme peuvent être choisis, tels que la fixation a priori du nombre de générations à exécuter lorsque l'on doit trouver une solution dans un temps limité, ou l'arrêt de l'algorithme lorsque la population n'évolue plus suffisamment rapidement.

– L'algorithme peut être arrêté lorsque la population n'évolue plus ou plus suffisamment rapidement.[1]

I.3. Les algorithmes génétiques : mécanisme de fonctionnement

Le processus de l'algorithme consiste à faire évoluer une population de points dans l'espace à partir d'un tirage initial, dans le but de trouver un ou plusieurs optimums d'un problème d'optimisation. Ce processus se déroule en générations successives, avec une population constante notée N , qui contient des individus à chaque génération. Pour faire évoluer la population de la génération k à la génération $k+1$, plusieurs opérations sont effectuées. Tout d'abord, une sélection est effectuée pour reproduire les individus les plus performants. Ensuite, un croisement est effectué entre les paires d'individus sélectionnés, avec une probabilité P (généralement autour de 0,7), pour produire de nouveaux individus. Un opérateur de mutation est également appliqué à une certaine proportion de la population (probabilité P , généralement très inférieure à P). Les nouveaux individus sont évalués et intégrés à la population de la génération suivante, avant de répéter les opérations de sélection, de croisement et de mutation pour compléter la nouvelle population. L'algorithme peut être arrêté après un nombre fixé de générations ou lorsque la population n'évolue plus assez rapidement.[2]

I.3.1. Codage des variables

La première étape consiste à formuler et à encoder correctement le problème d'optimisation. Chaque paramètre de dispositif x_i est associé à un gène. Un ensemble de gènes est appelé chromosome. Chaque dispositif est représenté par un individu possédant un génotype

composé d'un ou plusieurs chromosomes. Un ensemble d'individus est appelé population que nous allons faire évoluer. Le terme génotype est utilisé pour décrire la représentation binaire d'un individu, tandis que le terme phénotype est utilisé pour décrire la valeur réelle correspondante dans l'espace de recherche.[2]

I.3.1.1. Codage binaire

Ce passage décrit les trois principaux types de codage utilisés en algorithme génétique : binaire, gray et réel. Les auteurs ont parfois fait le parallèle avec la biologie pour expliquer le concept. En informatique, le codage binaire est le plus couramment utilisé. Dans ce type de codage, chaque gène est un entier long représenté par 2^N bits. Un chromosome est un tableau de gènes, et un individu est un tableau de chromosomes. La population est un tableau d'individus, et cette structure complexe est à la base du comportement des algorithmes génétiques. L'un des avantages du codage binaire est qu'il permet de représenter facilement toutes sortes d'objets, tels que des réels, des entiers, des valeurs booléennes et des chaînes de caractères. Pour passer d'une représentation à l'autre, il suffit d'utiliser des fonctions de codage et de décodage appropriées.[2]

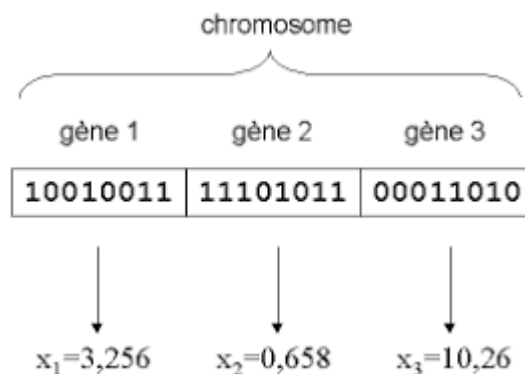


Figure I. 2. Illustration de codage des variables dans la génétique algorithmme

I.3.1.2. Le codage réel

Le codage réel est une autre méthode couramment utilisée pour la représentation des solutions dans les algorithmes génétiques. Contrairement au codage binaire, qui représente les solutions sous forme de chaînes binaires, le codage réel utilise des nombres réels pour représenter les variables de décision. Dans cette méthode, chaque individu est représenté par un vecteur de nombres réels qui correspond aux valeurs des différentes variables de décision.

Ce type de codage peut être particulièrement efficace pour les problèmes d'optimisation continue, car il permet une représentation directe des variables réelles. Cependant, le codage réel peut également être plus coûteux en termes de temps de calcul que le codage binaire, en raison du traitement des nombres réels. Cependant, le codage réel peut présenter quelques inconvénients.

- Suivant le problème, la résolution de l'algorithme peut être coûteuse en temps.
- Le croisement et la mutation peuvent être inadaptés (création d'individus n'appartenant pas à l'espace de recherche).[3]

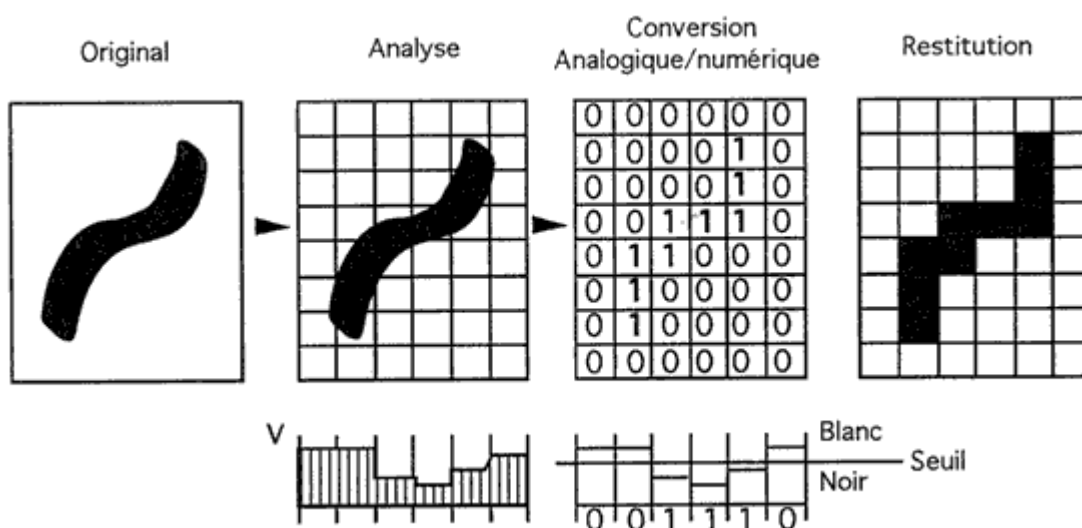


Figure I. 3. Codage réel multidimensionnels

I.3.2. Génération aléatoire de la population initiale

Le choix de la population initiale est crucial pour la rapidité et la qualité de convergence de l'algorithme d'optimisation. Si l'emplacement de la solution optimale est inconnu, il est recommandé de générer une population initiale de manière aléatoire en tirant uniformément dans chaque domaine associé à l'espace d'état, tout en respectant les contraintes du problème. Si des informations sont disponibles sur le problème, il est préférable de générer les individus dans un sous-domaine spécifique pour accélérer la convergence. Si la gestion des contraintes n'est pas possible directement, les contraintes sont souvent incluses dans le critère

d'optimisation sous forme de pénalités pour encourager la recherche de solutions qui respectent les contraintes.

I.4. Les opérateurs génétiques

La reproduction est le processus qui permet de construire une population $k + 1$ à partir d'une population k . Ce processus est constitué par l'utilisation de l'opération de sélection, de l'opération de croisement ou/et de l'opération de mutation.

I.4.1. La sélection

Il existe deux types de sélection qui permettent de déterminer quels individus sont plus enclins à obtenir les meilleurs résultats dans les algorithmes évolutionnaires. Le premier type est appelé la sélection pour la reproduction, qui permet de choisir les individus qui participent à une reproduction, comme le croisement ou la mutation. Généralement, cette opération choisit les individus les plus forts, c'est-à-dire ceux ayant les meilleurs scores d'adaptation,

pour produire les enfants les plus performants. Le deuxième type est appelé la sélection pour le remplacement, qui choisit les individus les plus faibles pour être remplacés par les nouveaux. Il existe plusieurs techniques de sélection, telles que :

- La sélection par rang, qui choisit toujours les individus possédant les meilleurs scores.
- La sélection par tournoi, qui utilise la probabilité de sélection proportionnelle à l'adaptation sur des paires d'individus, puis choisit parmi ces paires celui qui a le meilleur score d'adaptation.
- La sélection uniforme, qui choisit aléatoirement sans faire intervenir la valeur d'adaptation.

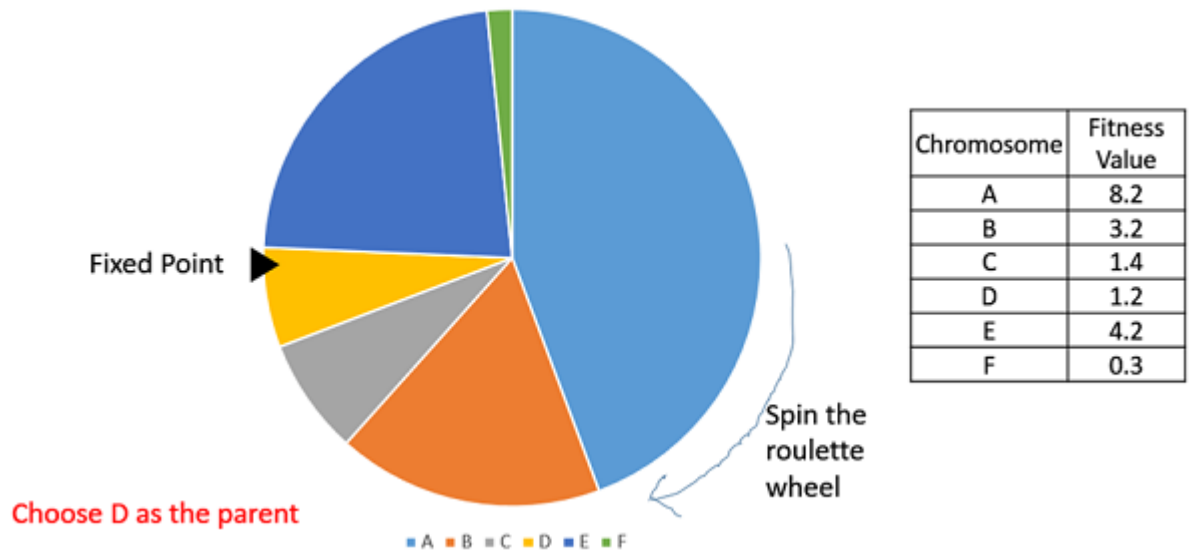


Figure I. 4. La sélection en génétiques algorithmmes

I.4.2. Le croisement dans la génétique algorithmme :

Le croisement est l’opération principale dans les algorithmmes génétiques (AG), qui permet de créer de nouveaux individus à partir des meilleurs individus de la génération précédente. L’idée est de combiner les caractéristiques des parents pour créer des descendants plus performants. Les croisements peuvent être effectués de différentes manières, chacune ayant ses avantages et ses inconvénients.

- Le croisement en un point (ou simple point) est la méthode la plus simple et la plus couramment utilisée. Le principe est de choisir un point de coupe au hasard dans le génome de chaque parent, puis de combiner les parties avant et après le point de coupe pour créer un nouvel individu. Cela produit deux descendants à partir de deux parents.

Parent 1	1 0 0 1 0 1 1 0 1
Parent 2	0 1 0 1 1 1 0 1 1
Position Aléatoire	↑
Enfant 1	1 0 0 1 1 1 0 1 1
Enfant 2	0 1 0 1 0 1 1 0 1

Tableau I. 1. Le croisement en un point

- Le croisement en deux points est une extension du croisement en un point. Dans cette méthode, deux points de coupe sont choisis au hasard dans le génome de chaque parent, puis les parties entre ces deux points sont échangées pour créer deux nouveaux descendants.


Parent 1	1 0 0 1 0 1 1 0 1
Parent 2	0 1 0 1 1 1 0 1 1
Position Aléatoire	
Enfant 1	1 0 0 1 1 1 0 1 1
Enfant 2	0 1 0 1 0 1 1 0 1

Tableau I. 2. Le croisement en deux points

- Le croisement uniforme est une méthode qui permet de combiner les caractéristiques de deux parents de manière aléatoire. Pour chaque position du génome, un descendant est choisi au hasard soit parmi le parent 1, soit parmi le parent 2. Cette méthode peut être utile pour éviter que l’algorithme ne converge vers un minimum local.

Parent 1	1 0 0 1 0 1 1 0 1
Parent 2	0 1 0 1 1 1 0 1 1
Chaîne Aléatoire de bits	0 1 1 0 1 0 1 0 0
Enfant 1	1 1 0 1 1 1 0 0 1
Enfant 2	0 0 0 1 0 1 1 1 1

Tableau I. 3. Le croisement uniforme

- Le croisement en cascade est une méthode qui permet de combiner les caractéristiques de trois parents. Cette méthode est utilisée pour les problèmes multi-

objectifs, où plusieurs objectifs contradictoires doivent être satisfaits. Le principe est de prendre le meilleur trait d'un parent pour chaque objectif et de l'attribuer au descendant.

- Le croisement arithmétique : Cette méthode utilise une formule mathématique pour combiner les gènes des parents et créer un chromosome enfant.

Le choix de la méthode de croisement dépend du problème spécifique à résoudre et des caractéristiques des individus dans la population. Le croisement est souvent combiné avec d'autres opérations telles que la mutation pour améliorer la diversité et l'efficacité de l'algorithme génétique. [4]

I.4.3. La mutation

La mutation est l'une des principales opérations utilisées dans les algorithmes génétiques pour introduire de la variabilité dans la population de solutions. La mutation consiste à changer aléatoirement un ou plusieurs gènes d'un chromosome, ce qui peut donner naissance à une nouvelle solution qui n'existait pas auparavant.

Dans un algorithme génétique, la mutation est souvent utilisée en conjonction avec le croisement pour éviter que la population ne converge trop rapidement vers une solution locale optimale. La mutation permet d'explorer de nouvelles régions de l'espace de recherche qui ne seraient pas accessibles par le croisement seul. Il existe plusieurs méthodes de mutation dans les algorithmes génétiques, notamment :

1. La mutation par inversion : Cette méthode consiste à inverser l'ordre des gènes dans une partie du chromosome.
2. La mutation par décalage : Cette méthode consiste à décaler une partie du chromosome d'un certain nombre de positions.
3. La mutation par permutation : Cette méthode consiste à permuter l'ordre des gènes dans une partie du chromosome.
4. La mutation aléatoire : Cette méthode consiste à choisir aléatoirement une nouvelle valeur pour un ou plusieurs gènes dans le chromosome.

Le choix de la méthode de mutation dépend du problème spécifique à résoudre et des caractéristiques des individus dans la population. Le taux de mutation doit également être ajusté pour éviter que la population ne converge trop rapidement vers une solution locale optimale. En général, un taux de mutation faible est utilisé pour les problèmes où les

solutions sont proches l'une d'une autres, tandis qu'un taux de mutation élevé est utilisé pour les problèmes où les solutions sont très différentes l'une d'une autres.

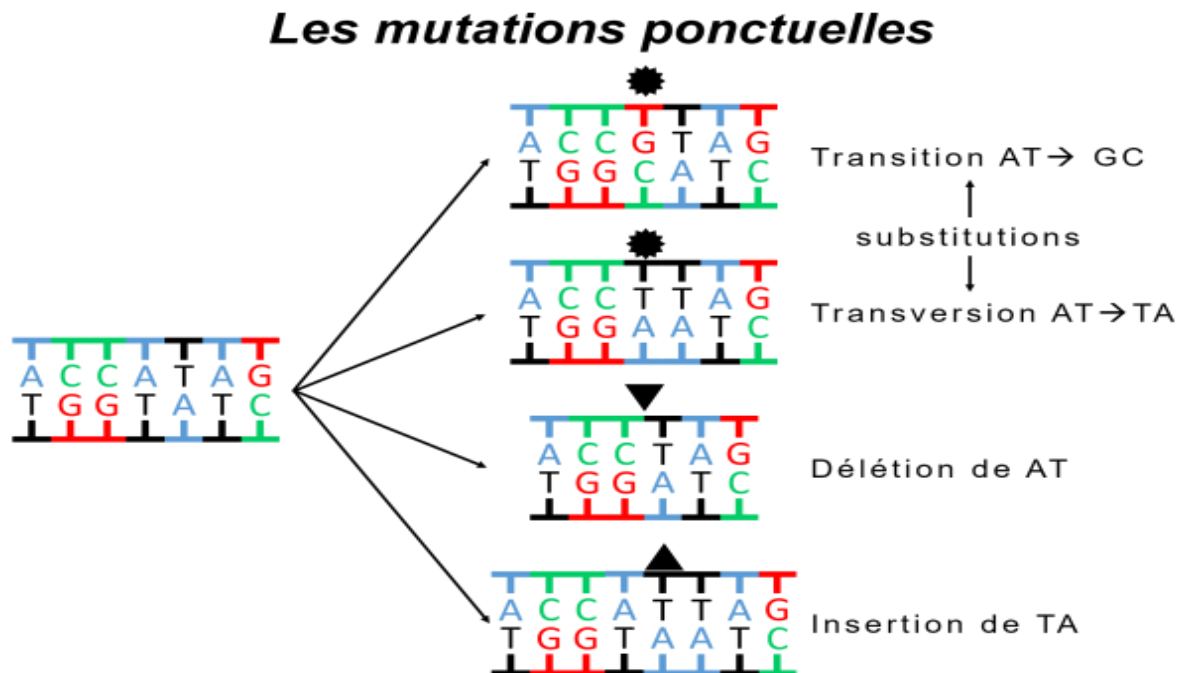


Figure I. 5. Exemple d'un opérateur de mutation

I.4.4. Élitisme (facultatif) :

Un nombre défini d'individus les plus en forme avec les meilleures performances sont transmis directement à la génération suivante.

I.4.5. Réplication :

Un individu est sélectionné pour passer à la génération suivante.

I.5. La fonction de coût

La fonction de coût, également connue sous le nom de fonction d'objectif ou de fonction d'évaluation, est un élément essentiel dans les problèmes d'optimisation. Elle mesure la performance ou la qualité d'une solution candidate par rapport à un ensemble de critères prédéfinis.

Dans le contexte de l'optimisation des systèmes de contrôle, la fonction de coût est utilisée pour évaluer la performance du système en fonction de différents objectifs tels que la précision, la stabilité, la réactivité ou la consommation d'énergie. Elle permet de quantifier la

distance entre la réponse réelle du système et la réponse souhaitée, en tenant compte des spécifications et des contraintes spécifiques.

La fonction de coût peut prendre différentes formes, selon le type de problème et les objectifs spécifiques. Elle peut être définie comme une somme pondérée des écarts entre la sortie réelle du système et la référence souhaitée, ou comme une fonction non linéaire qui attribue des pénalités en fonction de la violation de certaines contraintes.

La fonction de coût est une mesure quantitative de la performance d'un système de contrôle par rapport à des critères spécifiques. Elle attribue une valeur numérique à chaque solution candidate, permettant ainsi de comparer différentes solutions et de les classer en fonction de leur performance.

La valeur de la fonction de coût dépend des objectifs et des critères de performance définis pour le système de contrôle. Par exemple, dans le cas d'un contrôle de précision, la valeur de la fonction de coût peut représenter l'écart moyen entre la sortie réelle et la référence souhaitée. Dans le cas d'un contrôle de stabilité, la valeur de la fonction de coût peut mesurer la marge de stabilité du système.

L'objectif de l'optimisation est de minimiser ou de maximiser la valeur de la fonction de coût, selon les critères spécifiés. En ajustant les paramètres de contrôle, on cherche à trouver la combinaison optimale qui minimise la fonction de coût et maximise la performance du système.

Il est important de noter que la valeur de la fonction de coût est relative et dépendante du contexte. Elle n'a pas de signification absolue en dehors du problème d'optimisation spécifique. Son rôle principal est de guider le processus d'optimisation en fournissant une mesure objective de la performance du système de contrôle.

I.6. L'évolution dans la génétique algorithmme

Pour les opérations de réplication, de croisement et de mutation, les individus sont sélectionnés au hasard passer à la génération suivante, la probabilité de sélection augmentant avec la forme physique. Les opérations génétiques sont illustrées pour l'exemple de contrôle PID dans la figure suivante.

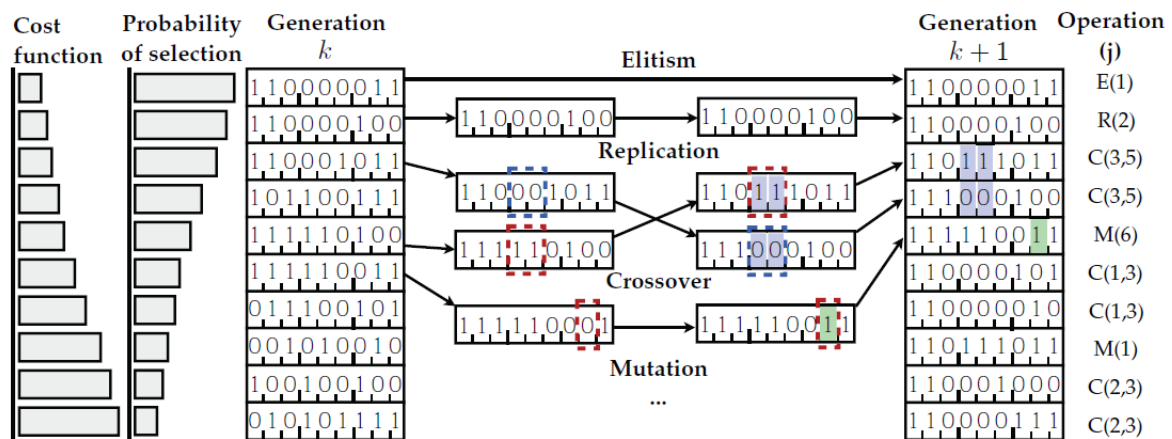


Figure I. 6. Illustration d'une évolution dans la génétique algorithmme

Les générations évoluent jusqu'à ce que la forme physique des individus supérieurs converge ou d'autres arrêts critères sont remplis.[5]

I.7. Avantages et limitations

I.7.1. Les avantages

Les algorithmes génétiques présentent plusieurs avantages dans le domaine de l'optimisation. Voici quelques-uns des avantages clés :

- Exploration de l'espace des solutions : Les algorithmes génétiques permettent d'explorer de manière exhaustive l'espace des solutions potentielles. Grâce à leur approche basée sur la sélection, la recombinaison et la mutation, ils peuvent trouver des solutions optimales dans des problèmes complexes où la recherche exhaustive n'est pas possible.
- Adaptabilité : Les algorithmes génétiques sont capables de s'adapter à des environnements changeants. Ils peuvent ajuster les paramètres des solutions en fonction des conditions variables, ce qui les rend particulièrement utiles dans les problèmes dynamiques où les conditions évoluent au fil du temps.
- Parallélisme et évolutivité : Les algorithmes génétiques sont parallélisables, ce qui signifie qu'ils peuvent être exécutés simultanément sur plusieurs processeurs ou machines, accélérant ainsi le processus d'optimisation. De plus, ils sont évolutifs, ce qui signifie qu'ils peuvent être appliqués avec succès à des problèmes de taille croissante sans perdre en efficacité.

- Recherche globale : Les algorithmes génétiques ont la capacité de rechercher des solutions globales plutôt que de rester bloqués dans des optima locaux. Leur capacité à explorer de manière exhaustive l'espace des solutions leur permet de trouver des solutions de haute qualité qui peuvent être situées dans des régions éloignées de l'espace de recherche.[16]

I.7.2. Les limitations

1. Le temps de calcul : par rapport à d'autres métaheuristiques, ils nécessitent de nombreux calculs, en particulier au niveau de la fonction d'évaluation.
2. Ils sont le plus souvent difficiles à mettre en œuvre : des paramètres comme la taille de la population ou le taux de mutation sont parfois difficiles à déterminer. Or le succès de l'évolution en dépend et plusieurs essais sont donc nécessaires, ce qui limite encore l'efficacité de l'algorithme. En outre, choisir une bonne fonction d'évaluation est aussi critique. Celle-ci doit prendre en compte les bons paramètres du problème. Elle doit donc être choisie avec soin.
3. Il faut aussi noter l'impossibilité d'être assuré, même après un nombre important de générations, que la solution trouvée soit la meilleure. On peut seulement être sûr que l'on s'est approché de la solution optimale (pour les paramètres et la fonction d'évaluation choisie), sans la certitude de l'avoir atteinte.
4. Un autre problème important est celui des optima locaux. En effet, lorsqu'une population évolue, il se peut que certains individus qui à un instant occupent une place importante au sein de cette population deviennent majoritaires. À ce moment, il se peut que la population converge vers cet individu et s'écarte ainsi d'individus plus intéressants mais trop éloignés de l'individu vers lequel on converge. Pour vaincre ce problème, il existe différentes méthodes comme l'ajout de quelques individus générés aléatoirement à chaque génération, des méthodes de sélection différentes de la méthode classique.

I.8. Conclusion

En conclusion de ce premier chapitre dédié à l'algorithme génétique, nous avons exploré les principes fondamentaux et le fonctionnement de cette technique d'optimisation puissante. L'algorithme génétique offre une approche basée sur l'évolution pour résoudre des problèmes complexes, en trouvant des solutions efficaces et adaptatives.

En comprenant les concepts de base de l'algorithme génétique, nous sommes mieux préparés à explorer son utilisation dans le domaine du contrôle. L'algorithme génétique peut être appliqué pour optimiser les paramètres des systèmes de contrôle, améliorer les performances et répondre aux exigences spécifiques.

Cependant, il est important de noter que l'algorithme génétique présente des limites et des considérations à prendre en compte lors de sa mise en œuvre.

Chapitre II

La logique floue

II.1. Introduction

Le deuxième chapitre de ce mémoire se concentre sur la logique floue, une approche mathématique permettant de modéliser et de traiter l'incertitude et la subjectivité dans les systèmes de contrôle. La logique floue offre une méthode puissante pour gérer des informations vagues ou imprécises, en permettant une prise de décision plus flexible et adaptative.

L'objectif de ce chapitre est de fournir une introduction approfondie à la logique floue, en expliquant ses principes fondamentaux, ses opérations et ses applications dans le domaine du contrôle.

Nous commencerons par présenter les bases de la logique floue, notamment la notion de variables linguistiques, les ensembles flous et les opérations de logique floue telles que la composition et l'inférence. Nous aborderons ensuite les techniques d'identification des fonctions d'appartenance, la création de règles floues et la défuzzification.

II.2. Généralités sur la logique floue

II.2.1. Historique

Les fondements de la logique floue ont émergé avant les années 1940, lorsque les premières approches du concept d'incertitude ont été développées par des chercheurs américains. Cependant, ce n'est qu'en 1965 que le concept de sous-ensemble flou a été introduit par L. Zadeh, un ingénieur en automatique renommé et professeur à l'université de Berkeley en Californie. Son objectif était de modéliser les phénomènes sous une forme floue afin de surmonter les limitations des modèles classiques basés sur des équations différentielles, qui étaient confrontés à des incertitudes. En 1974, Mamdani a mis en pratique la théorie de Zadeh en utilisant une chaudière à vapeur, un équipement reconnu pour sa complexité, et a ainsi introduit la commande floue dans la régulation d'un processus industriel. En Europe, de nombreuses applications ont émergé pour des systèmes parfois très complexes, comme la régulation des fours de cimenterie réalisée par la société F.L. Smidt-Fuller. Aujourd'hui, une large gamme de nouveaux produits arborent l'étiquette "produit flou" ou "fuzzy" [6].

II.2.2 Quelques domaines d'applications

La logique floue, ou la théorie des ensembles flous, trouve de nombreuses applications dans de nombreux domaines. Voici quelques exemples d'utilisation de la logique floue :

- ✚ Contrôle de la qualité : la logique floue peut être utilisée pour évaluer la qualité des produits en fonction de critères tels que la couleur, la texture, la taille, etc.
- ✚ Systèmes de contrôle de la température : la logique floue peut être utilisée pour contrôler la température des bâtiments, des usines, des serres, etc.
- ✚ Conception de systèmes d'alarme : la logique floue peut être utilisée pour détecter les anomalies dans les systèmes tels que les alarmes incendie, les systèmes de surveillance des intrus, etc.
- ✚ Systèmes de recommandation : la logique floue peut être utilisée pour recommander des produits ou des services en fonction de préférences d'un utilisateur, de son historique d'achats, etc.
- ✚ Systèmes de navigation : la logique floue peut être utilisée pour la navigation des véhicules autonomes, des drones, des robots, etc.
- ✚ Traitement de l'image : la logique floue peut être utilisée pour améliorer la qualité d'image en fonction de critères tels que la netteté, la luminosité, le contraste, etc.
- ✚ Systèmes de contrôle de vitesse : la logique floue peut être utilisée pour réguler la vitesse des machines, des véhicules, etc.

Ces exemples ne sont pas exhaustifs et la logique floue trouve de nombreuses autres applications dans différents domaines.[7]

II.2.3. Les avantages de la logique floue

La logique floue présente plusieurs avantages par rapport à la logique classique, notamment:

- La logique floue permet de traiter des informations imprécises et incertaines : Contrairement à la logique classique qui ne peut traiter que des informations binaires (vrai ou faux), la logique floue permet de traiter des informations imprécises et incertaines. Cela la rend particulièrement utile pour les problèmes qui impliquent des concepts tels que "chaud", "grand", "beaucoup", etc.
- La logique floue est plus flexible : La logique floue est capable de s'adapter à des environnements changeants. Elle permet de prendre en compte des variations mineures des conditions d'un problème, ce qui la rend plus flexible que la logique classique.

- La logique floue est plus facile à comprendre : La logique floue utilise des concepts linguistiques plutôt que des valeurs numériques, ce qui la rend plus facile à comprendre pour les utilisateurs non spécialisés. Les ensembles flous permettent de représenter des concepts de manière plus intuitive que les équations mathématiques.
- La logique floue est plus robuste : La logique floue est capable de fonctionner même lorsque certaines données sont manquantes ou incomplètes. Elle peut également fonctionner même lorsque les données sont bruyantes ou incertaines.
- La logique floue peut être combinée avec d'autres méthodes : La logique floue peut être combinée avec d'autres méthodes telles que les réseaux de neurones ou les algorithmes génétiques pour améliorer leur performance. Cette combinaison de méthodes est appelée "intelligence artificielle hybride".

La logique floue présente de nombreux avantages par rapport à la logique classique, notamment pour traiter des informations imprécises et incertaines, et pour s'adapter à des environnements changeants. Elle est également plus facile à comprendre pour les utilisateurs non spécialisés et peut être combinée avec d'autres méthodes pour améliorer les performances.

II.2.4. Les inconvénients de la logique floue

Bien que la logique floue présente de nombreux avantages, elle présente également certains inconvénients. En voici quelques-uns :

- La complexité des systèmes flous : Les systèmes flous peuvent devenir très complexes en raison de la nécessité de définir des ensembles flous, des règles floues et des fonctions d'appartenance. Cela peut rendre la mise en œuvre de la logique floue plus difficile et plus coûteuse.
- Le risque de sur-apprentissage : Dans certains cas, la logique floue peut sur-apprendre les données d'entraînement, ce qui peut entraîner une mauvaise performance sur les données de test ou de nouvelles données. Il est donc important de contrôler la complexité du modèle et d'utiliser des méthodes de régularisation pour éviter le sur-apprentissage.

- Les ensembles flous ne sont pas toujours interprétables : Bien que les ensembles flous permettent de représenter des concepts de manière plus intuitive que les équations mathématiques, ils peuvent parfois être difficiles à interpréter. Par exemple, les fonctions d'appartenance floues peuvent être définies de manière subjective et peuvent varier d'un utilisateur à l'autre.
- La difficulté de choisir les paramètres : Le choix des paramètres, tels que le nombre d'ensembles flous ou le degré de flou, peut avoir un impact significatif sur la performance du système flou. Il peut être difficile de trouver les paramètres optimaux, en particulier pour des problèmes complexes.
- La nécessité de données d'entraînement : Comme pour toute méthode d'apprentissage automatique, la logique floue nécessite des données d'entraînement pour construire un modèle. Si les données ne sont pas représentatives ou si elles sont bruyantes, le modèle flou peut ne pas fonctionner correctement.

La logique floue présente certains inconvénients, tels que la complexité des systèmes flous, le risque de sur-apprentissage, la difficulté de choisir les paramètres et la nécessité de données d'entraînement. Il est important de comprendre ces inconvénients pour pouvoir utiliser la logique floue de manière efficace.

II.3. La différence entre la logique floue et la logique booléenne

La principale différence entre la logique floue et la logique booléenne réside dans la façon dont elles traitent les incertitudes. La logique booléenne traite les variables comme étant vraies ou fausses (1 ou 0), tandis que la logique floue permet des valeurs de vérité partielles, représentant ainsi une certaine incertitude.

Par exemple, dans la logique booléenne, si la variable A est vraie et la variable B est fausse, alors la proposition "A et B" est fausse. En revanche, dans la logique floue, les variables peuvent avoir une valeur de vérité partielle, par exemple, A peut avoir une valeur de vérité de 0,9 et B une valeur de vérité de 0,1. Dans ce cas, la proposition "A et B" peut avoir une valeur de vérité intermédiaire, reflétant l'incertitude associée à ces variables.

En outre, la logique floue peut être utilisée pour modéliser des relations non linéaires et des systèmes dynamiques, tandis que la logique booléenne est plus adaptée pour des problèmes de décision binaires.

En résumé, la logique floue diffère de la logique booléenne en permettant des valeurs de vérité partielles, ce qui lui permet de traiter les incertitudes et de modéliser des relations non linéaires et des systèmes dynamiques.[8]

II.4. Les fonctions d'appartenance

La fonction d'appartenance est utilisée pour décrire l'évolution du degré de vérité d'une variable linguistique. Le degré d'appartenance à un ensemble flou est représenté par un nombre compris entre 0 et 1. Une valeur précise de la fonction d'appartenance, liée à une valeur de la variable, est notée μ et appelée "facteur d'appartenance". La fonction d'appartenance qui décrit un sous-ensemble flou possède quatre propriétés distinctes [9]:

Le type: Il s'agit de la forme du nombre flou, qui peut être triangulaire, trapézoïdale, gaussienne ou sigmoïdale.

Parmi les différents types de fonctions d'appartenance, on retrouve:

- **les fonctions triangulaires :**

$$\mu(x) = \max \left[\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right]$$

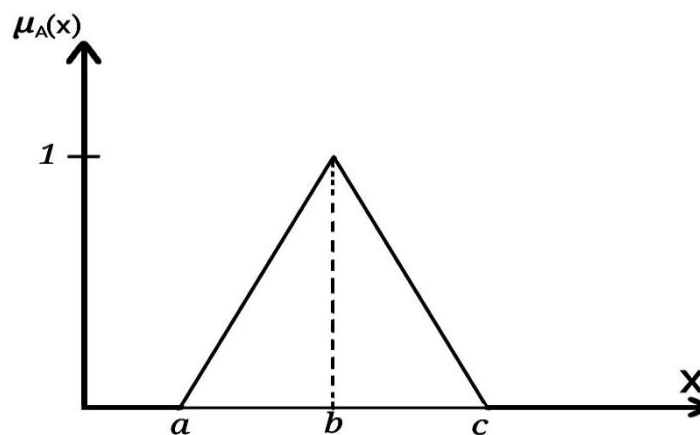


Figure II. 1. Schéma d'une fonction triangulaire

- **les fonctions trapézoïdales**

$$\mu(x) = \max \left[\min \left(\frac{x-a}{b-a}, 1, \frac{c-x}{c-b} \right), 0 \right]$$

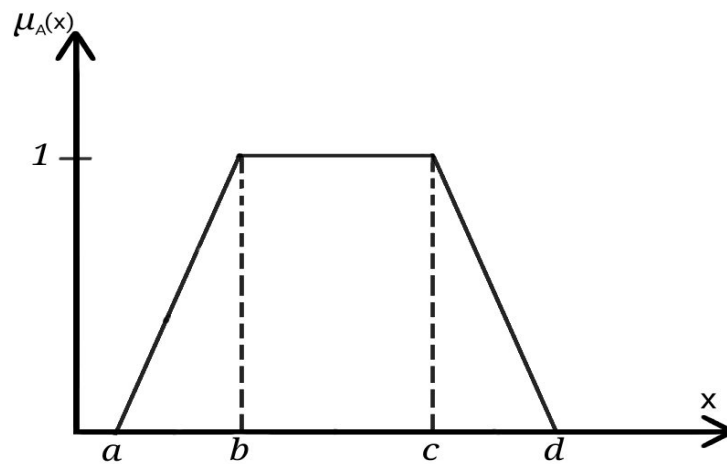


Figure II. 2. Schéma d'une fonction trapézoïdale

- les fonctions gaussiennes

$$\mu(x) = \exp\left(-\frac{(x - m)^2}{2\delta}\right)$$

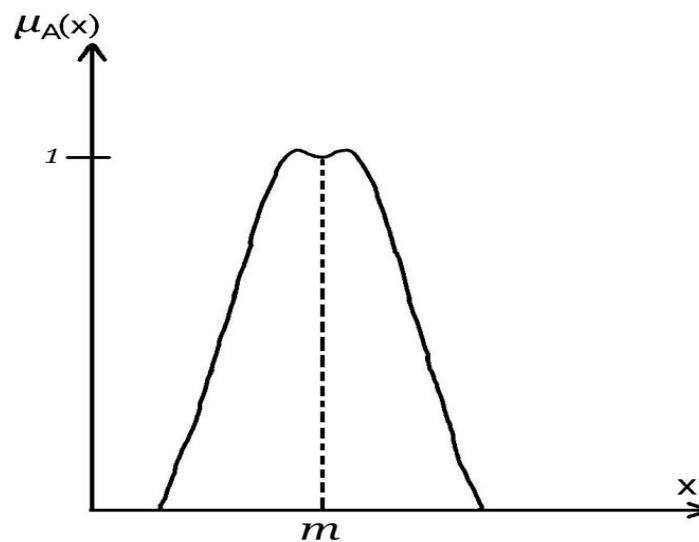


Figure II. 3. Schéma d'une fonction gaussienne

La hauteur: Elle est présentée par la fonction suivante:

$$H(A) = \text{Sup}(A)_{x \in X}(\mu_a(x))$$

La hauteur d'un ensemble flou normalisé est égale à 1.

Le noyau: il est présenté par la fonction suivante:

$$N(a) = \{x | \mu_a(x) = 1\}$$

Le noyau (ou le cœur) d'un ensemble flou est l'ensemble des éléments qui appartiennent totalement à A. Pour les fonctions de type triangulaire, le noyau est un singleton qui est également appelé valeur modale.

Le support: Il est représenté par :

$$S(A) = \{x | \mu_a(x) \neq 0\}$$

Il s'agit de l'ensemble des éléments qui ont un degré d'appartenance non nul à A, c'est-à-dire ceux qui sont partiellement inclus dans A.

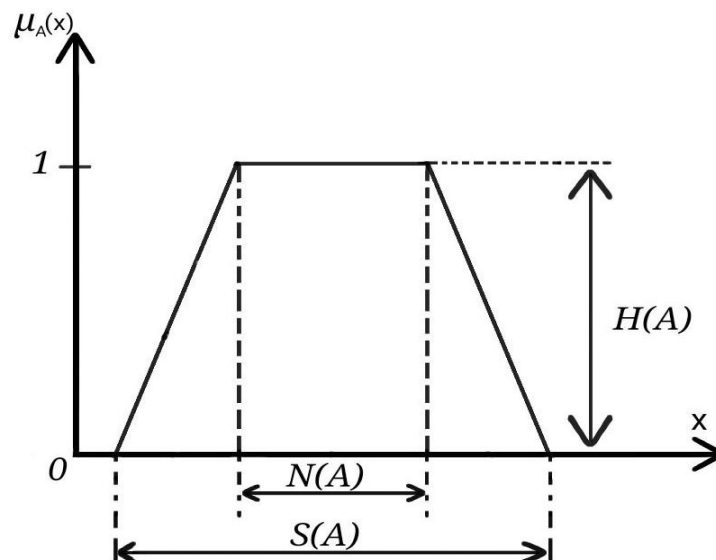


Figure II. 4. Représentation de hauteur, noyau et support

II.5. Les opérateurs dans la logique floue

Dans la logique floue, on définit des opérateurs similaires à ceux de la théorie des ensembles classiques. On a ainsi l'opérateur d'intersection, l'opérateur d'union et l'opérateur de complémentaire pour les ensembles flous, représentés respectivement par les opérateurs "et", "ou" et "non", qui sont les plus couramment utilisés.

Les figures (II.5), (II.6) et (II.7) illustrent la mise en œuvre des opérateurs "et", "ou" et "non" pour des ensembles flous a et b, utilisant des fonctions d'appartenance triangulaires, et les comparent aux opérateurs booléens correspondants.

Soit $\mu_a(\mathbf{x})$ et $\mu_b(\mathbf{x})$ les degrés d'appartenance de l'élément x dans l'univers de discours X (représenté par $x \in X$) [10].

• **Opérateur “NON”**

Selon la théorie des ensembles, le complémentaire d'un ensemble A , noté \bar{A} , est défini par les éléments de x qui n'appartiennent pas à l'ensemble A . Dans le cas de la logique floue, cette définition peut être exprimée par les fonctions d'appartenance de la manière suivante :

$$\mu_{\bar{A}}(\mathbf{x}) = 1 - \mu_A(\mathbf{x})$$

La figure (I-4) illustre cette relation.

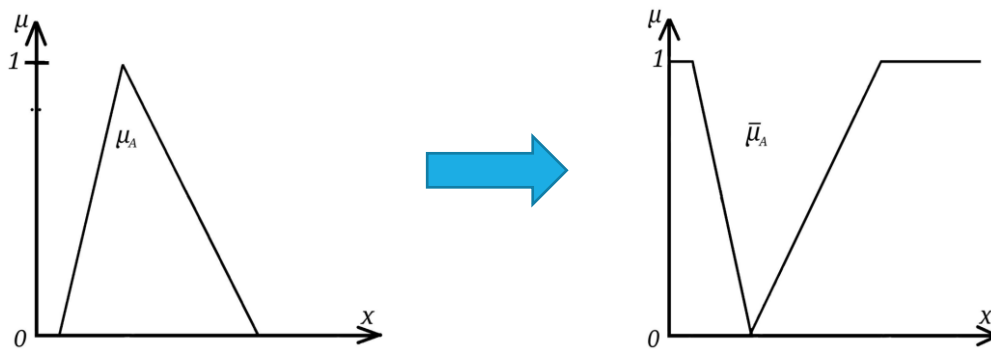


Figure II. 5. Représentation d'un opérateur « NON ».

• **Opérateur “ET”**

L'opérateur “et” correspond à l'intersection de deux ensembles a et b , noté $a \cap b$. Dans la logique floue, cet opérateur est généralement réalisé en utilisant la formulation du minimum appliquée aux fonctions d'appartenance $\mu_a(\mathbf{x})$ et $\mu_b(\mathbf{x})$ des deux ensembles a et b :

$$\mu_C(\mathbf{x}) = \min[\mu_a(\mathbf{x}) , \mu_b(\mathbf{x})]$$

On parle alors de l'opérateur minimum. Cette opération est représentée dans la figure (I-5).

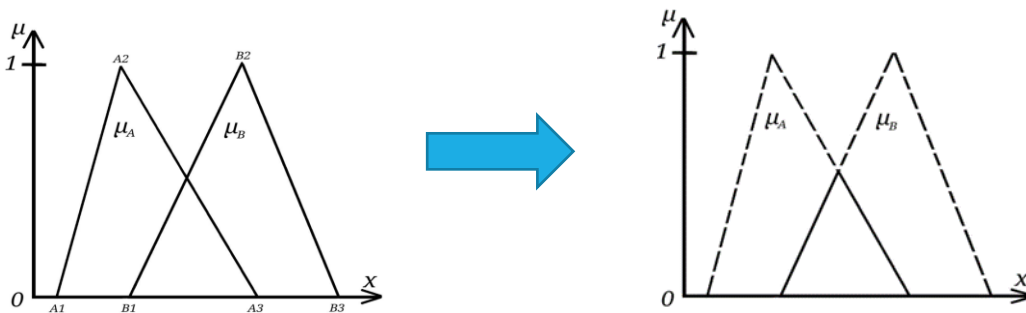


Figure II. 6. Représentation d'un opérateur « ET »

• **Opérateur “OU”**

L'opérateur "ou" correspond à l'union de deux ensembles **a** et **b**, noté **a** \cup **b**. Dans la logique floue, cet opérateur est généralement réalisé en utilisant la formulation du maximum appliquée aux fonctions d'appartenance $\mu_a(\mathbf{x})$ et $\mu_b(\mathbf{x})$ des deux ensembles **a** et **b** :

$$\mu_c(x) = \max[\mu_a(x), \mu_b(x)]$$

Cette opération est représentée dans la figure (II-7).

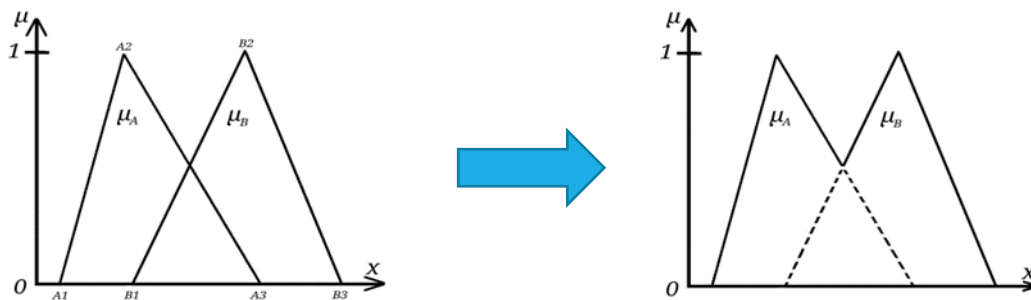


Figure II. 7. Représentation d'un opérateur « OU »

II.6. La logique floue et l'intelligence artificielle

Ces opérateurs permettent de manipuler et de combiner les ensembles flous de manière similaire aux opérations logiques classiques, mais en tenant compte du degré de vérité flou.

La logique floue est un domaine de l'intelligence artificielle qui vise à formaliser et à mettre en œuvre le raisonnement humain. L'outil le plus couramment utilisé dans les applications de logique floue est la base de règles floues. Une base de règles floues est constituée de règles qui sont généralement utilisées en parallèle, mais peuvent également être enchaînées dans certaines applications.

Une règle floue est composée de deux parties :

A. Prédicat :

Le prédicat, également appelé prémisse ou condition, est une combinaison de propositions reliées par des opérateurs logiques tels que ET, OU, NON. Par exemple, dans la règle "Si température élevée ET pression forte, ALORS ventilation forte ET soupape grande ouverte", les propositions "température élevée" et "pression forte" sont combinées par l'opérateur ET pour former le prédicat de la règle.

B. Conclusion :

La conclusion d'une règle floue est une combinaison de propositions liées par des opérateurs logiques ET. Dans l'exemple précédent, "ventilation forte" et "soupape grande ouverte" sont la conclusion de la règle.

Le traitement flou d'une règle comprend trois étapes fonctionnelles, comme illustré dans la Figure II.8 :

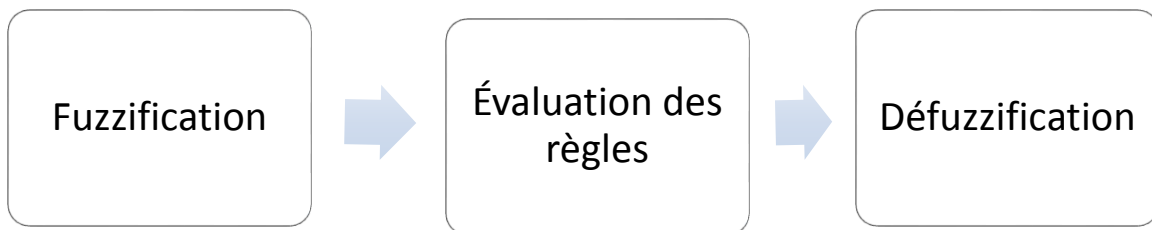


Figure II. 8. Processus de traitement flou

II.6.1. Les étapes de traitement flou

Ces étapes fonctionnelles permettent de traiter les informations floues et de prendre des décisions basées sur des règles floues, ce qui est une caractéristique importante de la logique floue dans le domaine de l'intelligence artificielle.

II.6.1.1. La fuzzification

Elle est la première étape du mécanisme d'inférence de type Mamdani. Elle consiste à évaluer les fonctions d'appartenance utilisées dans les prédicats des règles. Cela implique de déterminer le degré d'appartenance de chaque variable d'entrée à ses ensembles flous correspondants. La (figure II.9). illustre ce processus de fuzzification.

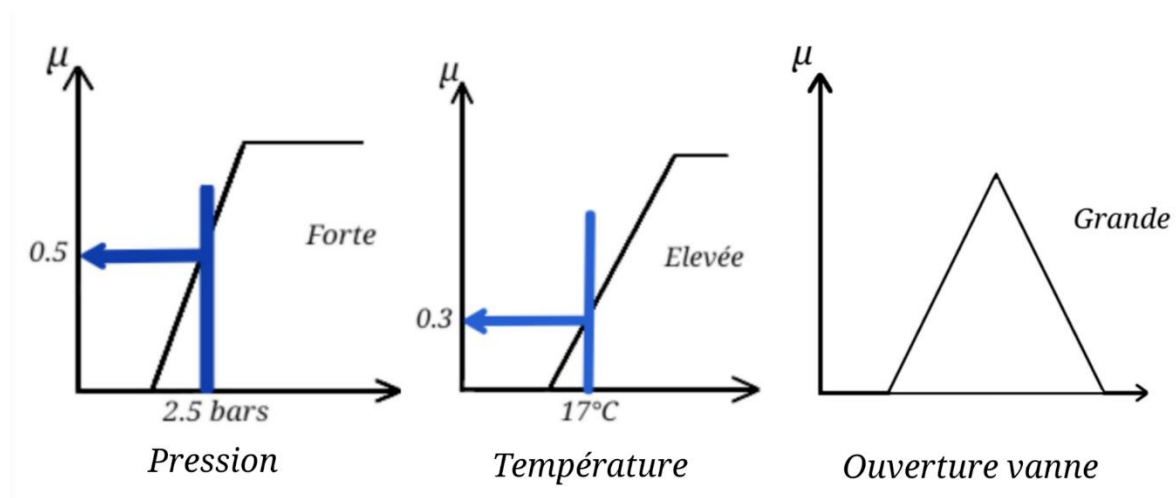


Figure II. 9. Processus de fuzzification

II.6.1.2. Le degré d'activation

Le degré d'activation d'une règle est ensuite évalué dans l'étape suivante. Cela se fait en combinant logiquement les propositions du prédicat de la règle, généralement par un opérateur "ET". Le degré d'activation est calculé en prenant le minimum des degrés de vérité des propositions. La (figure II.10) représente cette étape de détermination du degré d'activation des règles.

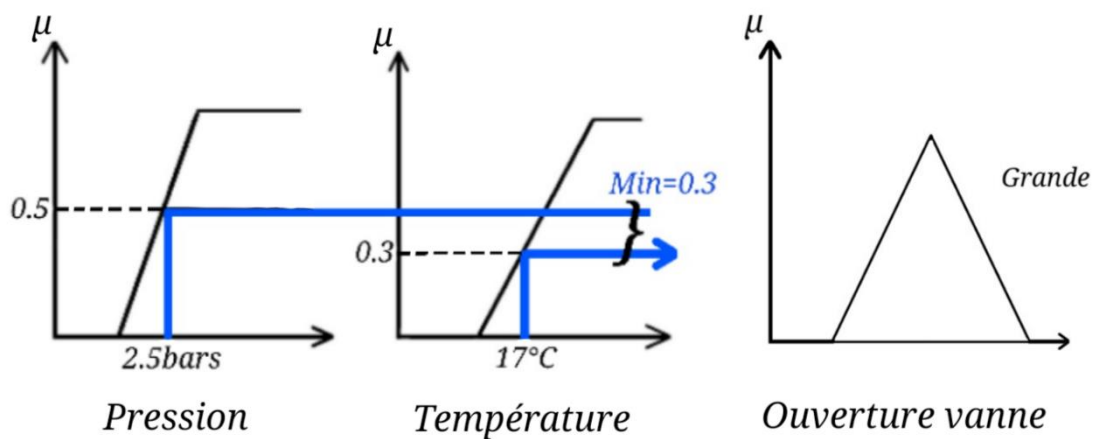


Figure II. 10. Activation des règles

II.6.1.3. L'implication

Une fois que le degré d'activation de chaque règle est connu, l'étape d'implication est réalisée. L'implication consiste à déterminer la conclusion de chaque règle en utilisant un opérateur d'implication, souvent le "minimum". L'ensemble flou de conclusion est construit en prenant le minimum entre le degré d'activation et la fonction d'appartenance de la conclusion. Cela permet de restreindre la fonction d'appartenance de la conclusion (figure II.11).

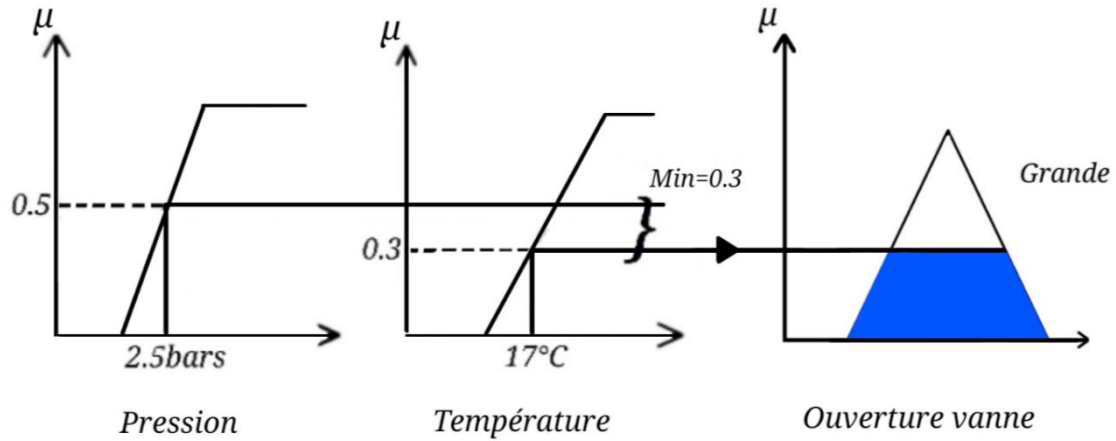


Figure II. 11. L'implication

II.6.1.4. L'agrégation

Dans l'étape d'agrégation, les ensembles flous obtenus par chaque règle concernant la même sortie sont agrégés pour former l'ensemble flou global de sortie. Cela se fait généralement en utilisant un opérateur "OU" logique, et en prenant le maximum entre les fonctions d'appartenance résultantes pour chaque règle. La (figure II.12) présente l'agrégation des règles dans un exemple à deux règles.

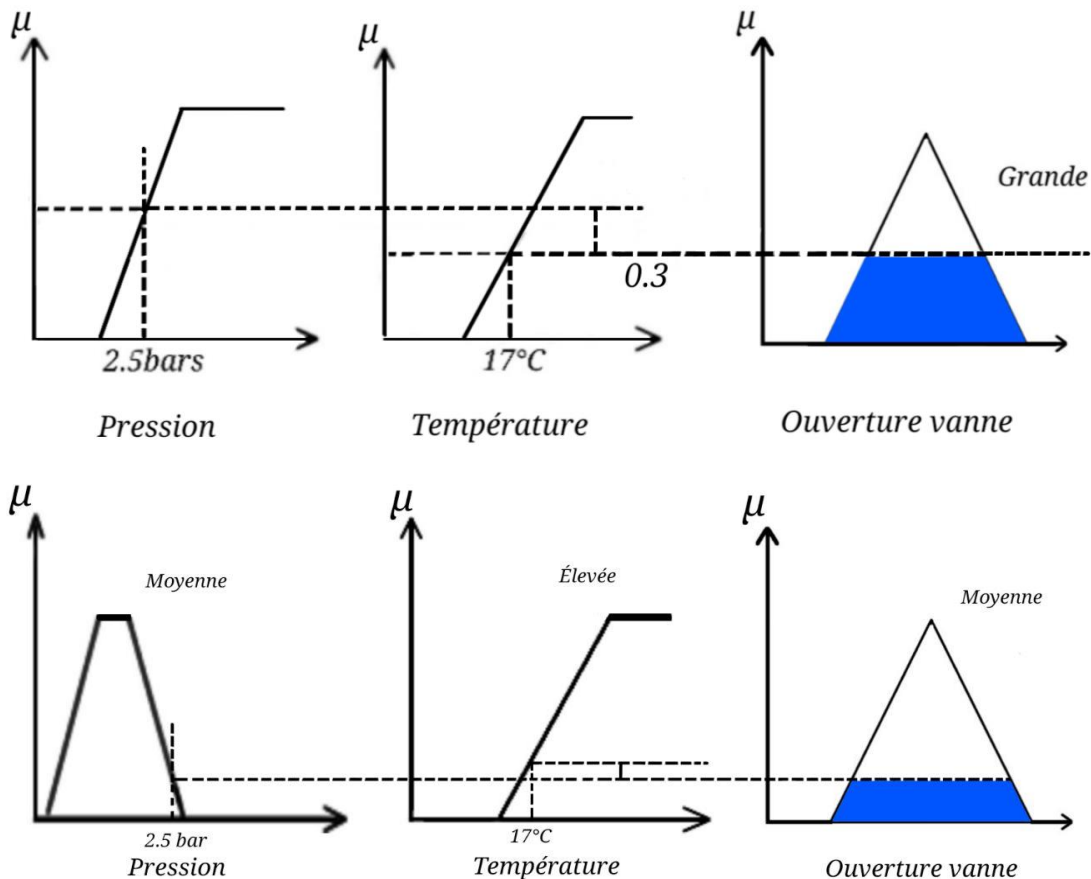




Figure II. 12. L'agrégation

II.6.1.5. La défuzzification

La défuzzification est réalisée pour obtenir une valeur précise à partir de l'ensemble flou de sortie. Différentes méthodes de défuzzification peuvent être utilisées, telles que le centre de gravité de la surface, le bissecteur de la surface, la moyenne des maximas, etc. La méthode la plus courante est celle du "centre de gravité", qui calcule le point central de l'ensemble flou. La (figure II.13) illustre la défuzzification par le centre de gravité.

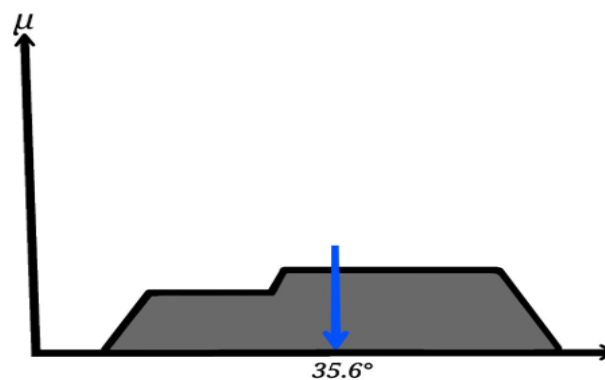


Figure II. 13 La défuzzification

En résumé, le mécanisme d'inférence de type Mamdani comprend les étapes de fuzzification, de degré d'activation, d'implication, d'agrégation et de défuzzification. Ces étapes permettent de transformer les informations floues en une valeur précise pour la sortie du système.

II.7.conclusion

En conclusion de ce deuxième chapitre dédié à la logique floue, nous avons exploré les principes fondamentaux et les applications de cette approche de modélisation et de prise de décision. La logique floue offre une méthode puissante pour traiter l'incertitude et la subjectivité dans les systèmes de contrôle, permettant une prise de décision plus flexible et adaptative.

La logique floue trouve de nombreuses applications dans le domaine du contrôle, allant de la modélisation de systèmes à la conception de régulateurs flous et à la prise de décision adaptative. Cependant, il est important de noter que la logique floue présente des limites et des considérations à prendre en compte lors de sa mise en œuvre.

Chapitre III

Le mode glissant

III.1. Introduction

Le troisième chapitre de ce mémoire se concentre sur le mode glissant (sliding mode), une technique de commande robuste largement utilisée pour assurer la stabilité et la performance des systèmes de contrôle face à des perturbations et des incertitudes. Le mode glissant offre une approche dynamique et adaptative, permettant de maintenir le système dans un “glissement” continu vers une trajectoire souhaitée.

L’objectif de ce chapitre est de fournir une introduction approfondie au mode glissant, en expliquant ses principes fondamentaux, ses mécanismes et ses applications dans le domaine du contrôle.

Nous commencerons par présenter les bases du mode glissant, notamment les concepts de surface de glissement, les lois de commande et les mécanismes de commutation. Nous aborderons ensuite les différents régimes du mode glissant, enfin on va discuter sur le phénomène de broutement.

III.2. Historique

Les systèmes de contrôle à structure variable (VSCS) avec commande par mode glissant ont une histoire riche, qui trouve ses origines dans les travaux révolutionnaires d'Emel'yanov et Barbashin en Russie au début des années 1960. À l'origine, ces idées sont restées principalement confinées en Russie jusqu'au milieu des années 1970, lorsque Itkis a publié un livre complet sur le sujet en anglais (1976), et Utkin a présenté un article de synthèse (1977). La dissémination internationale des concepts VSCS a ouvert la voie à leur application généralisée dans divers domaines.

Depuis lors, les concepts de commande par mode glissant ont été largement utilisés dans la conception de régulateurs robustes, de systèmes de référence de modèle, de schémas adaptatifs, de systèmes de poursuite, d'observateurs d'état et de schémas de détection de défauts. L'adaptabilité et l'efficacité des VSCS ont permis leur mise en œuvre réussie dans divers domaines. Les applications notables comprennent le contrôle automatique de vol, le contrôle précis des moteurs électriques, l'optimisation des processus chimiques, l'amélioration de la stabilité des hélicoptères, les avancées dans les systèmes spatiaux et le développement de systèmes robotiques agiles.[22]

En fusionnant les premières contributions d'Emel'yanov et Barbashin avec les avancées ultérieures présentées par Itkis et Utkin, les VSCS ont évolué en une méthodologie de

contrôle polyvalente qui continue d'inspirer et de stimuler des solutions innovantes dans le domaine de l'ingénierie des systèmes de contrôle.

Essentiellement, le contrôle en mode glissant utilise des lois de commande à réaction discontinue pour amener l'état du système à atteindre, puis à rester sur une surface spécifiée dans l'espace des états (appelée surface de glissement ou de commutation). Les dynamiques du système, lorsqu'elles sont confinées à la surface de glissement, sont décrites comme un mouvement de glissement idéal et représentent le comportement du système contrôlé.[23]

III.3. Principe de la Commande par mode glissant :

III.3.1. Mode de convergence (MC) :

Ce mode implique le déplacement de la variable à régler à partir de n'importe quel point initial dans le plan de phase, avec une tendance à atteindre la surface de commutation $S(x)=0$. Il se caractérise par une loi de commande spécifique et un critère de convergence.

III.3.2 Mode de glissement (MG) :

Ce mode survient lorsque la variable d'état atteint la surface de glissement et tend vers l'origine du plan de phase. La dynamique de ce mode est déterminée par l'établissement de la surface de glissement $S(x)=0$.

III.3.3. Mode de régime permanent (MRP) :

Ce mode est ajouté pour étudier la réponse du système autour de son point d'équilibre, qui correspond à l'origine du plan de phase. Il est caractérisé par la qualité et les performances de la commande [11].

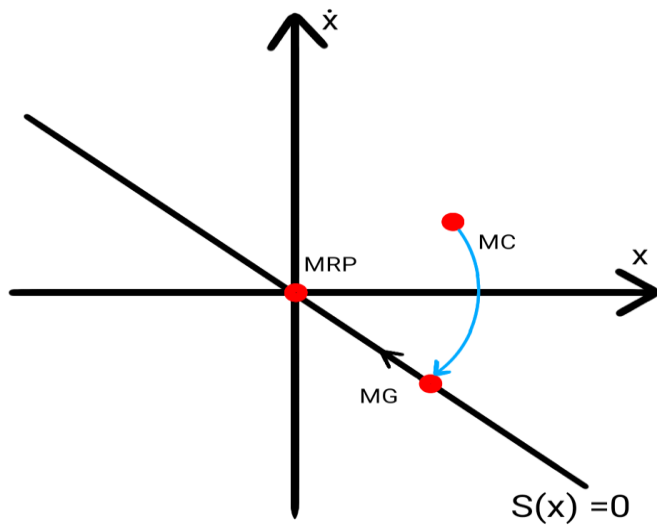


Figure III. 1. Les différents modes pour la trajectoire dans le plan de phase

III.4. Conception de la commande par mode de glissant

La conception de la commande peut être effectuée en trois étapes principales très dépendantes l'une de l'autre :

- Choix de la surface ;
- L'établissement des conditions d'existence ;
- Détermination de la loi de commande

III.4.1. Le choix de la surface

Le choix de la surface de glissement dépend du nombre et de la forme des fonctions nécessaires, qui sont déterminés par l'application et l'objectif visé, le vecteur de surface δ a la même dimension que le vecteur de commande U .

La surface de glissement est une fonction scalaire sur laquelle la variable à réguler glisse et tend vers l'origine du plan de phase. Sa forme non linéaire dépend de l'erreur entre la variable à réguler x et sa référence, et est donnée par l'équation suivante :

$$S(x) = e(x) * \left[\lambda + \left(\frac{\partial(x)}{\partial t} \right) \right]^{r-1}$$

Où :

- $E(x)$ représente l'écart entre la variable à réguler et sa référence.
- λ est une constante positive.
- r est le degré relatif, indiquant le nombre de fois qu'il faut dériver la surface pour obtenir la commande.

L'expression de la surface est une équation différentielle dont la seule solution est $e(x) = 0$. L'objectif de la commande est de maintenir la surface à zéro.

III.4.2. Les conditions de convergence

Nous avons identifié deux conditions de convergence issues de la littérature, qui permettent aux dynamiques du système de converger vers les surfaces de glissement. Ces conditions déterminent le mode de convergence de l'état du système.

- **Fonction directe de commutation :**

Cette condition vise à donner à la surface une dynamique convergente vers zéro. Elle est définie par l'équation suivante :

$$S(x).S'(x) = 0$$

- **Fonction de Lyapunov :**

La fonction de Lyapunov est une fonction scalaire définie positive $V(x) > 0$. La loi de commande doit garantir la décroissance de cette fonction. L'idée est de choisir une fonction scalaire $\delta(x)$ qui assure l'attraction de la variable à contrôler vers sa valeur de référence, et de concevoir une commande u telle que le carré de la surface corresponde à une fonction de Lyapunov. La fonction de Lyapunov peut être définie comme suit :

$$V(x) = \frac{1}{2} S^T(x).S(x)$$

La dérivée de cette fonction peut être exprimée comme suit :

$$V'(x) = S(x).S^T(x)$$

Il est clair que pour vérifier la décroissance de $v(x)$, il suffit de vérifier l'inégalité $V'(x) < 0$.

III.4.3. La loi de commande :

La loi de commande par mode glissant vise à orienter les trajectoires du système vers une surface de glissement égale à zéro en premier lieu, puis à maintenir ces trajectoires sur cette surface en deuxième lieu. Cette commande est composée de deux éléments. Tout d'abord, une composante discontinue, notée u_{sw} , permet de ramener les trajectoires vers la surface de glissement et assure la robustesse face aux perturbations extrêmes. Ensuite, une seconde

composante continue, appelée commande équivalente u_{eq} , permet le maintien et le glissement le long de la surface de glissement.

Il convient de noter que la commande équivalente est déterminée en utilisant les conditions d'invariance suivantes de la surface de glissement.

$$\begin{cases} s = 0 \\ s = \frac{\partial s}{\partial x} = \frac{\partial s}{\partial x} \frac{\partial x}{\partial t} = 0 \end{cases}$$

La commande par mode glissant est choisie selon l'expression suivante :

$$u = u_{sw} + u_{eq}$$

Les termes utilisés sont définis comme suit :

$$u_{eq} = - \left(\frac{\partial s}{\partial t} g(x) \right)^{-1} \cdot \frac{\partial s}{\partial x} f(x)$$

$$u_{sw} = -k \left(\frac{\partial s}{\partial t} g(x) \right)^{-1} \text{sgn}(s)$$

Ici, $k \geq D + \eta$ représente une constante de conception, ρ est une variable, et $\text{sgn}(\rho)$ est une fonction de signe définie comme suit :

$$\text{sgn}(\rho) = \begin{cases} 1 & \text{si } s > 0 \\ 0 & \text{si } s = 0 \\ -1 & \text{si } s < 0 \end{cases}$$

III.5. Formalisme classique du régime glissant

III.5.1. Régime glissant idéal :

Dans la théorie, l'organe de commutation est considéré comme idéal, insensible au bruit, et la trajectoire en régime glissant correspond parfaitement à l'équation $S(x) = 0$. Ainsi, on parle de régime glissant idéal.

Définition :

Le régime glissant idéal se caractérise par une oscillation de fréquence infinie et d'amplitude nulle. Le point représentatif de l'évolution du système glisse parfaitement sur l'hypersurface de commutation $S(x)$.

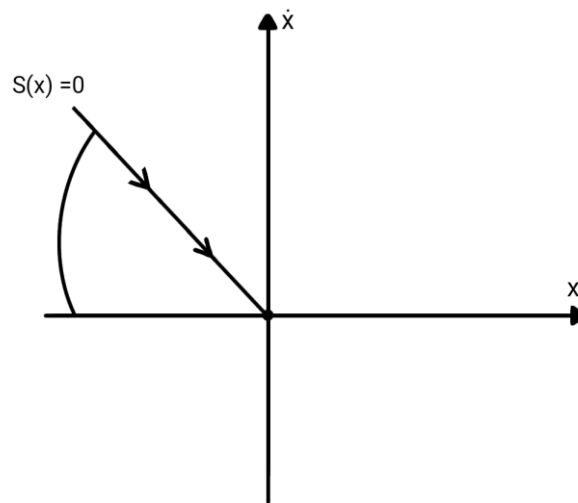


Figure III. 2. Régime idéal

III.5.2. Régime glissant réel :

En pratique, l'organe de commutation est réalisé à partir de relais qui présentent des imperfections, tels que des retards de commutation. La trajectoire de phase en régime glissant reste proche de la surface de commutation, ce qui entraîne des oscillations indésirables réduisant la précision du système, tout en maintenant néanmoins sa stabilité.

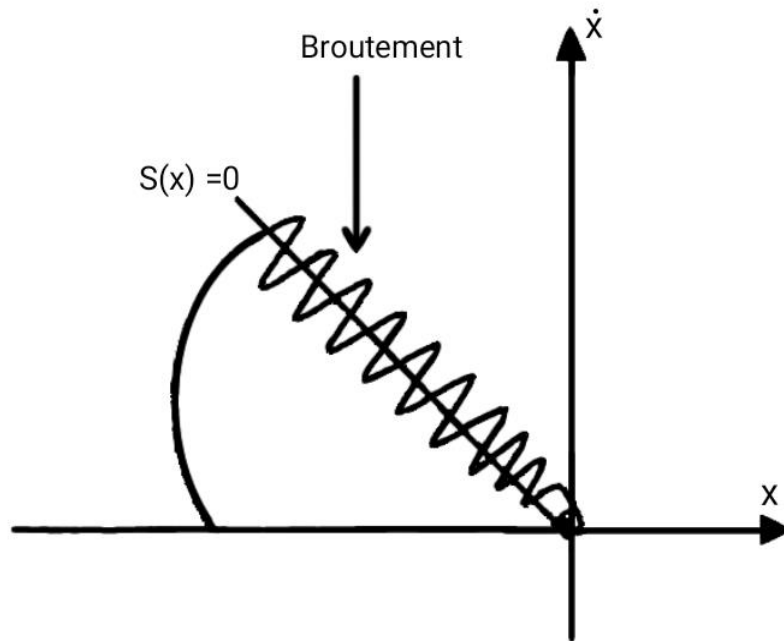


Figure III. 3. Régime glissant réel

III.5.3. Phénomène de broutement :

Un régime glissant idéal nécessite une commande capable de commuter à une fréquence infinie. Cependant, dans une utilisation pratique, seule une commutation à une fréquence finie est réalisable. Par conséquent, pendant le régime glissant, les discontinuités appliquées à la commande peuvent entraîner un phénomène de broutement appelé réticence (ou "chattering" en anglais). Ce phénomène se manifeste par des oscillations importantes des trajectoires du système autour de la surface de glissement.

Les principales causes de ce phénomène sont les retards de commutation au niveau de la commande et la présence de dynamiques parasites en série avec les systèmes commandés. Cela constitue un inconvénient significatif car, même s'il est possible de filtrer les oscillations en sortie du processus, elles peuvent exciter des modes à haute fréquence qui n'ont pas été pris en compte dans le modèle du système. Cette situation peut être si préjudiciable que l'utilisation d'une commande par mode glissant d'ordre un peut être déconseillée dans certaines applications, car elle risque de dégrader les performances et même conduire à l'instabilité.

La réticence entraîne également des contraintes mécaniques importantes au niveau des actionneurs, ce qui peut entraîner une usure rapide, ainsi que des pertes d'énergie non négligeables au niveau des circuits électriques de puissance.

III.5.4. L'élimination du phénomène de broutement

Dans le but de réduire ou d'éliminer le phénomène de broutement, plusieurs solutions ont été proposées. Parmi celles-ci, on trouve :

- ✚ L'introduction d'autres approximations non linéaires du terme de commande discontinue, par exemple, l'approximation fractionnaire définie comme : équation.
- ✚ Le remplacement de la couche limite par un secteur de glissement.
- ✚ L'utilisation de contrôleurs SM dynamiques.
- ✚ L'utilisation de contrôleurs SM flous.
- ✚ L'utilisation de contrôleurs SM d'ordre supérieur, tels que des contrôleurs du deuxième ordre ou d'ordre supérieur.

Ces approches alternatives visent à atténuer les effets indésirables du broutement et à améliorer les performances du système de commande. [3]

III.5.5. Commande par régime glissant d'ordre supérieur

Pendant le fonctionnement en mode glissant, l'utilisation de commandes discontinues peut causer des problèmes de broutement connus sous le nom de chattering. Ce phénomène se manifeste par des oscillations importantes des trajectoires du système autour de la surface de glissement. Pour remédier à ce problème tout en conservant les avantages principaux du mode glissant d'ordre un, tels que la convergence en temps fini et la robustesse, plusieurs solutions ont été proposées. L'une des stratégies les plus intéressantes consiste à utiliser des modes de glissement d'ordre supérieur. Cette méthode implique le calcul des dérivées successives de la surface de glissement le long d'une trajectoire d'un système discontinu, ce qui aboutit à une discontinuité d'un certain ordre. Les régimes glissants d'ordre supérieur peuvent alors être classifiés en fonction du nombre de dérivées successives de la surface s de glissement, tel que r l'ordre de glissement est déterminé par l'équation suivante :

$$S = s' = s'' = s''' = \dots = s^{(r-1)}$$

Les lois de commande basées sur cette méthode sont caractérisées par des commandes discontinues agissant sur les dérivées d'ordre supérieur de la variable de glissement, ce qui constitue un inconvénient majeur des algorithmes d'ordre supérieur car le nombre d'informations nécessaires sur l'entrée augmente avec l'ordre du régime glissant [14]

III.6. Avantages et limitations de SMC

Le mode glissant est une technique de commande non linéaire utilisée pour contrôler des systèmes dynamiques présentant des perturbations ou des incertitudes. Voici quelques avantages et limitations associés au mode glissant :

III.6.1. Avantages :

- ✓ Robustesse : Le mode glissant est connu pour sa robustesse face aux perturbations et aux incertitudes. Il est capable de maintenir le système en mode glissant, ce qui garantit une réponse rapide et précise malgré les perturbations.
- ✓ Conception simple : Le mode glissant est facile à concevoir et à mettre en œuvre. Il est basé sur un concept simple de la projection du système sur une surface de glissement, ce qui permet d'obtenir un contrôleur robuste et efficace.
- ✓ Stabilité : Le mode glissant est capable d'assurer la stabilité du système en toutes circonstances. Il est capable de maintenir le système en mode glissant, ce qui garantit une réponse rapide et précise même en présence de perturbations importantes.

III.6.2. Limitations

- ❖ Broutement : Le principal inconvénient du mode glissant est le phénomène de broutement, qui se produit lorsque le contrôleur passe constamment d'un côté à l'autre de la surface de glissement. Cela peut provoquer des oscillations indésirables et affecter la performance du système.
- ❖ Sensibilité aux paramètres : Le mode glissant est sensible aux variations des paramètres du système. Une petite variation des paramètres peut entraîner une grande variation de la réponse du système, ce qui peut affecter sa performance.
- ❖ Effet de commutation : Le mode glissant implique des commutations fréquentes entre différentes lois de commande, ce qui peut causer des problèmes de bruit et de vibrations dans le système.[15]

III.7. Conclusion

En conclusion de ce troisième chapitre dédié au mode glissant, nous avons exploré les principes fondamentaux et les applications de cette technique de commande robuste.

Le mode glissant trouve de nombreuses applications dans le domaine du contrôle, offrant une solution efficace pour la régulation de systèmes linéaires et non linéaires, la compensation des perturbations et la rejet des incertitudes. Cependant, il est important de noter que le mode

glissant présente des limites et des considérations à prendre en compte lors de sa mise en œuvre. Une attention particulière doit être portée à la conception de la surface de glissement et à la commutation des lois de commande.

Chapitre IV

Simulation et résultats

IV.1. Introduction

Dans ce quatrième chapitre, nous nous concentrons sur l'application des techniques d'optimisation génétique, floue et glissante à notre système contrôlé, le système FESTO. FESTO est un système complexe largement utilisé dans le domaine de l'automatisation industrielle, et notre objectif est d'améliorer ses performances de contrôle en exploitant les avantages de ces trois approches avancées.

Tout d'abord, nous explorerons l'application de l'optimisation génétique au contrôle de notre système FESTO. Les algorithmes génétiques nous permettent d'optimiser les paramètres du contrôleur PI (Proportionnel Intégral), en cherchant les meilleures combinaisons pour obtenir une réponse précise et stable du système. Nous examinerons les différentes étapes de l'optimisation génétique appliquée au contrôleur PI de FESTO et les résultats obtenus.

Ensuite, nous aborderons l'application de la logique floue au contrôleur PI de FESTO. La logique floue offre une approche flexible pour prendre des décisions basées sur des ensembles flous et des règles linguistiques. Nous décrirons comment nous avons utilisé la logique floue pour définir les fonctions d'appartenance des variables d'entrée et de sortie du contrôleur PI, permettant ainsi d'adapter le contrôle en fonction des conditions changeantes du système.

Enfin, nous étudierons l'application du mode glissant flou. Le mode glissant flou assure une stabilité robuste et une résistance aux perturbations en maintenant le système sur une trajectoire souhaitée. Nous expliquerons comment nous avons intégré le mode glissant flou pour un système FESTO, permettant ainsi d'obtenir une réponse plus rapide et plus précise du système face aux perturbations.

Cette application conjointe des techniques d'optimisation génétique, floue et glissante au système contrôlé FESTO vise à améliorer ses performances de contrôle, en garantissant une réponse adaptative, stable et résiliente face aux variations et aux perturbations.

IV.2. Description du processus (Poste de travail compact FESTO MPS PA)

Le poste de travail compact FESTO MPS PA, illustré dans la Figure 3, est un support de laboratoire utilisé pour concevoir différents types de régulateurs PID. Comme on peut le voir sur la photo, le poste est composé d'une pompe, de tuyaux, de deux réservoirs, de différents types de capteurs, d'actionneurs et d'un contrôleur PLC S7-300 (Siemens Aktiengesellschaft, Nuremberg, Allemagne) permettant de tester différents systèmes de régulation. À l'aide d'un algorithme de serveur OPC, le PLC peut communiquer avec le logiciel MATLAB, dans lequel les différents algorithmes de régulation sont conçus. Le système est équipé d'un outil graphique implémenté à l'aide du logiciel WINCC, permettant de superviser le processus et d'indiquer les différentes mesures.



Figure IV. 1. Le système FESTO

Le poste de travail FESTO MPS PA Compact permet de réaliser des systèmes de régulation pour des paramètres tels que le niveau, la pression, la température et le débit. Dans notre cas, la boucle de régulation de niveau est utilisée. La boucle considérée est présentée dans la Figure IV.2. Le processus de régulation de niveau est réalisé par la pompe P101, qui transfère le liquide du réservoir B101 au réservoir B102, et par le capteur à ultrasons situé en haut du réservoir B102. Le signal de mesure transmet la valeur du liquide dans le réservoir B102, et la

correction est réalisée par un signal d'actionneur qui correspond à la tension appliquée à la pompe. Comme le montre la Figure 4.1., le processus admet une perturbation représentée par l'ouverture/fermeture manuelle de la vanne V102 ou de la vanne V110.[18]

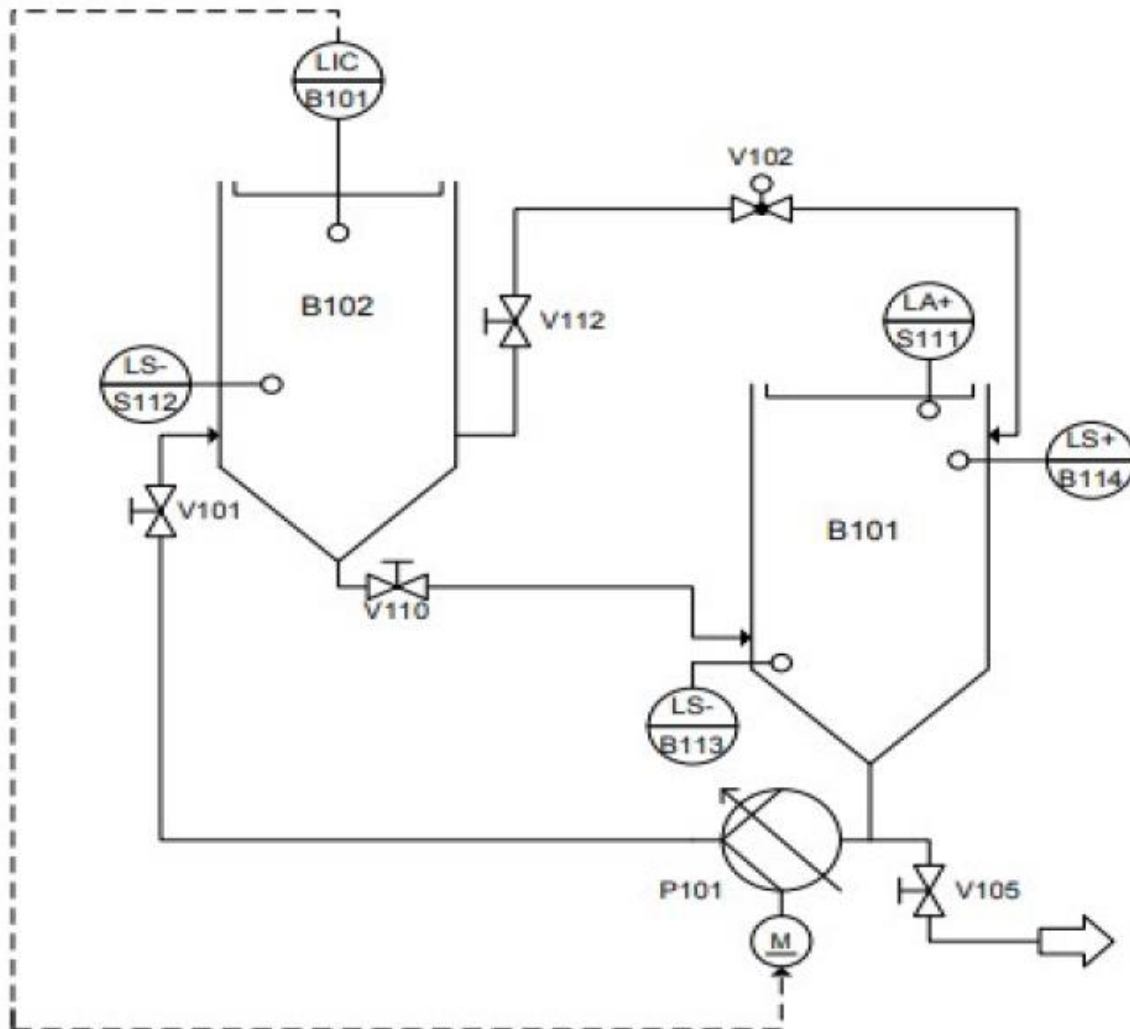


Figure IV. 2. Contrôle de niveau au système FESTO

Le poste de travail FESTO MPS PA Compact offre une plateforme complète pour la conception, le test et la mise en œuvre de systèmes de régulation, avec la capacité de s'intégrer au logiciel MATLAB et d'utiliser des outils graphiques pour la supervision et l'affichage des mesures.

Le meilleur modèle qui décrit la relation entre la tension de la pompe et le débit de sortie est celui qui a été identifié, de sorte qu'il fournit de meilleures valeurs de meilleure adaptation pour les tests d'entraînement et de validation. La fonction de transfert donnée par l'équation est le meilleur modèle pour représenter cette relation.[18]

$$\frac{Y(z)}{U(z)} = \frac{0.013402 z^{-1} - 0.013391 z^{-2}}{1 - 1.970562 z^{-1} + 0.970593 z^{-2}}$$

IV.3. Contrôle de système avec la génétique algorithmme

IV.3.1. L'optimisation d'un contrôleur PI en utilisant la GA

Dans cette expérience, nous utiliserons l'algorithme génétique pour ajuster un régulateur proportionnel-intégral (PI). Cependant, il convient de noter que cela n'est qu'une simple démonstration d'algorithmes évolutifs, et qu'il n'est pas recommandé d'utiliser une méthode aussi complexe pour ajuster un régulateur PI en pratique, car il existe des techniques beaucoup plus simples.

Le contrôle PI est l'une des architectures de contrôle les plus simples et les plus largement utilisées dans les systèmes de contrôle industriels, notamment pour le contrôle de la position et de la vitesse des moteurs, pour l'ajustement de divers sous-systèmes dans une automobile, et pour le contrôle de la pression et de la température dans les machines à espresso modernes, pour ne citer que quelques-unes des innombrables applications. Comme son nom l'indique, le contrôle PI combine de manière additive deux termes pour former le signal d'activation, sur la base du signal d'erreur et de son intégrale dans le temps. Un schéma du contrôle PI est montré dans la figure IV.3.

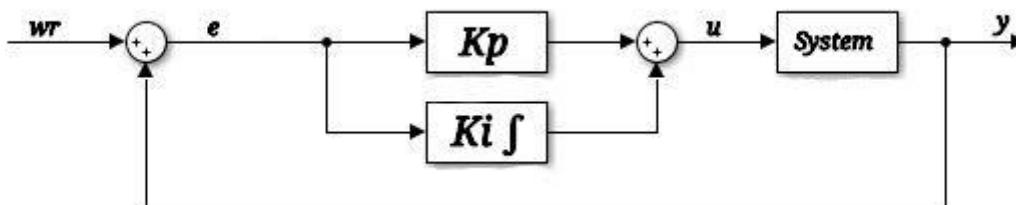


Figure IV. 3. Schéma du contrôle pi

Il était possible de réduire l'erreur de suivi de référence en augmentant le gain de contrôle proportionnel K_P dans la loi de commande $u = -K_P (w_r - y)$. Cependant, l'augmentation du gain peut éventuellement causer de l'instabilité dans certains systèmes et ne permettra pas d'éliminer complètement l'erreur de suivi en régime permanent.

L'ajout d'un terme de contrôle intégral, $K_I \int_0^T (w_r - y)$, est utile pour éliminer l'erreur de suivi de référence en régime permanent tout en soulageant le travail requis par le terme proportionnel.

Il existe des règles formelles pour choisir les gains PI pour différentes spécifications de conception, telles qu'une réponse rapide et une faible oscillation. Dans cet exemple, nous explorons l'utilisation d'un algorithme génétique pour trouver des gains PI efficaces afin de minimiser une fonction de coût. Nous utilisons une fonction de coût LQR :

$$J = [Q(w_r - y)^2 + Ru^2] dt$$

Avec $Q = 1$ et $R = 0,001$ pour une réponse en échelon $w_r = 1$. La fonction de coût est pour minimiser l'erreur entre set-point et la réponse de système.

IV.3.2. Les paramètres de la génétique algorithme

Nous avons choisi les paramètres (présentés dans le tableau.4.1.) pour lancer notre simulation Matlab sur le système contrôlé.

Nombre de population	100
Nombre max des générations	15
Mutation	0.2
Croissement	Dans un point
Pourcentage	70%
Réplication	20%
Sélection	10%

Tableau IV. 1. Les paramètres de la GA

IV.3.3. Résultats et discussion :

- ✓ Après l'exécution de code matlab on obtient les paramètres k_p et k_d en appelons x dans la fenêtre des commande et la valeur de fonction de cout .

```
>> x
x =
    26.1235    8.0405
>> fval
fval =
    0.3992
```

Figure IV. 4. Résultats d'optimisation de k_p et k_i

- ✓ L'évolution de la fonction de coût à travers différentes générations est présentée dans la figure IV.5.

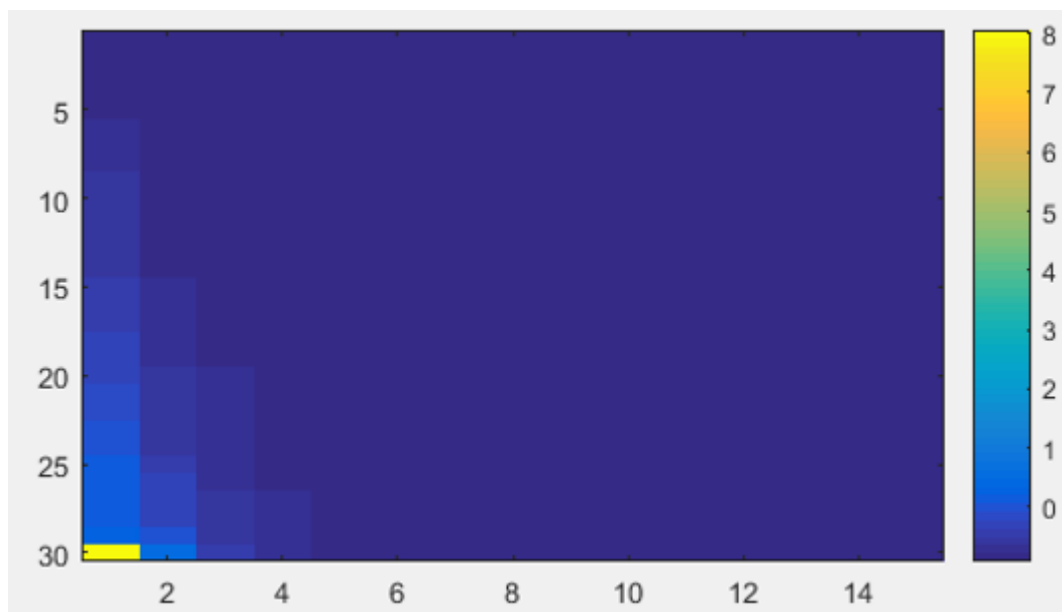


Figure IV. 5. Evolution des paramètres pendant les différents générations

Au fur et à mesure que les générations progressent, la fonction de coût diminue régulièrement. Au fur et à mesure que l'algorithme génétique progresse, les gains PI commencent à se regrouper autour de la solution optimale.

- ✓ L'évolution de la réponse indicielle du système à travers différentes générations est présentée dans les figures (IV.6.), (IV.7) et (IV.8.)

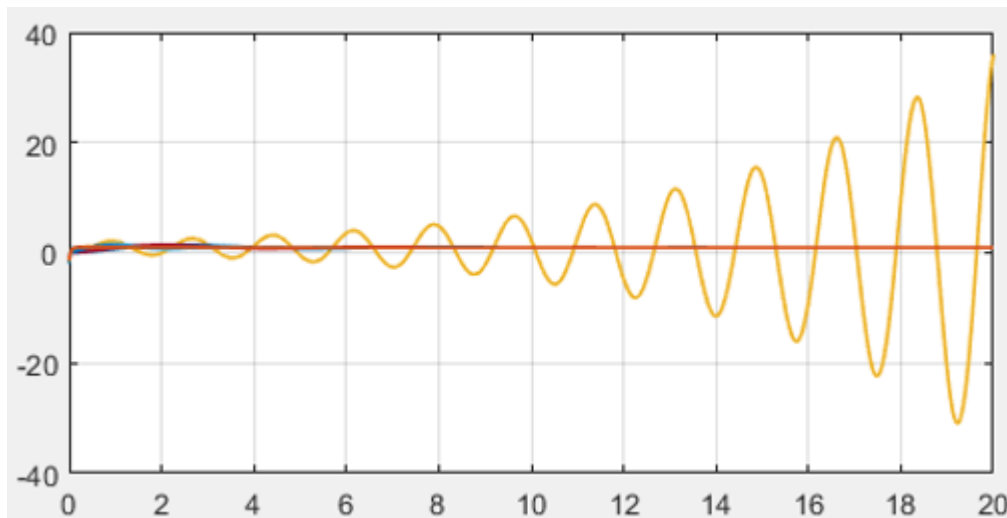


Figure IV. 6. Les différentes réponses indicielles du système pendant la première génération

La figure IV.6. Montre la sortie en réponse indicielles aux contrôleurs PI de la première génération. Il est clair à partir de ce graphique que de nombreux contrôleurs échouent à stabiliser le système, entraînant de grandes déviations dans y .

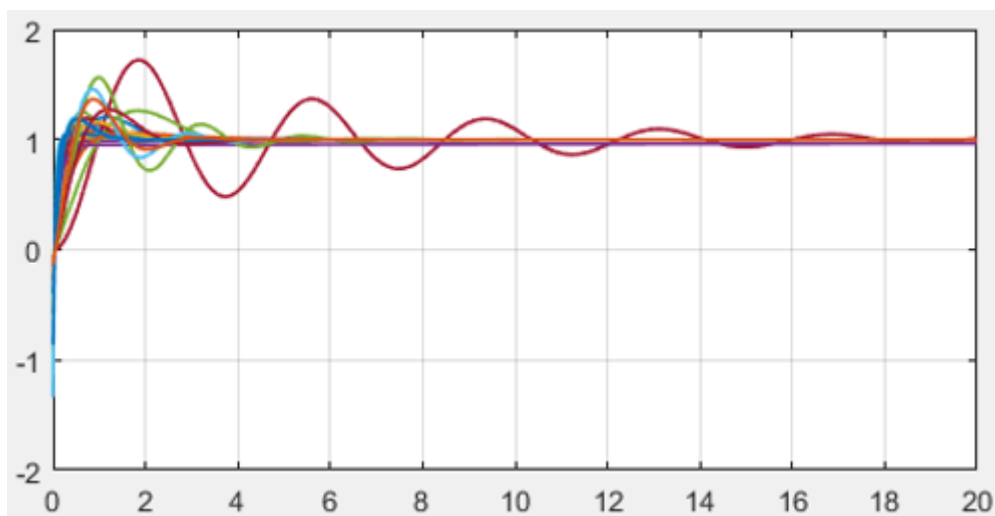


Figure IV. 7. Les réponses du système en 2ème génération

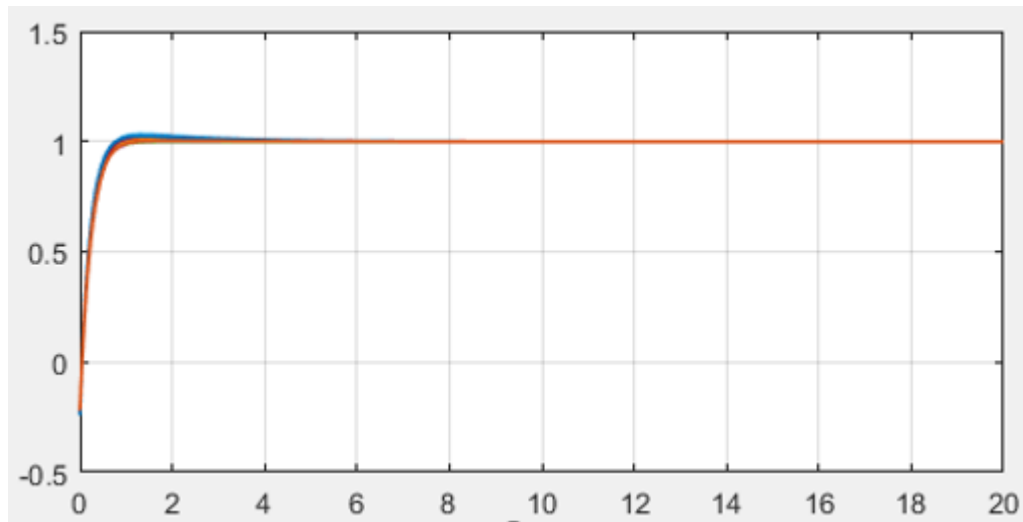


Figure IV. 8. Les réponses indicielles de système en 10^{ème} génération

En revanche, la figure IV.7 montre la sortie en réponse indicielle aux contrôleurs PI de la 2^{ème} génération. Et la figure IV.8 montre la sortie de la 10^{ème} génération. Dans l'ensemble, ces contrôleurs sont plus efficaces pour produire une réponse de marche stable. L'amélioration de réponse indicielle de système vers la stabilité au cours de des générations et l'augmentation de rapidité de système.

- ✓ Les meilleurs contrôleurs de chaque génération sont présentés dans la figure IV.9.

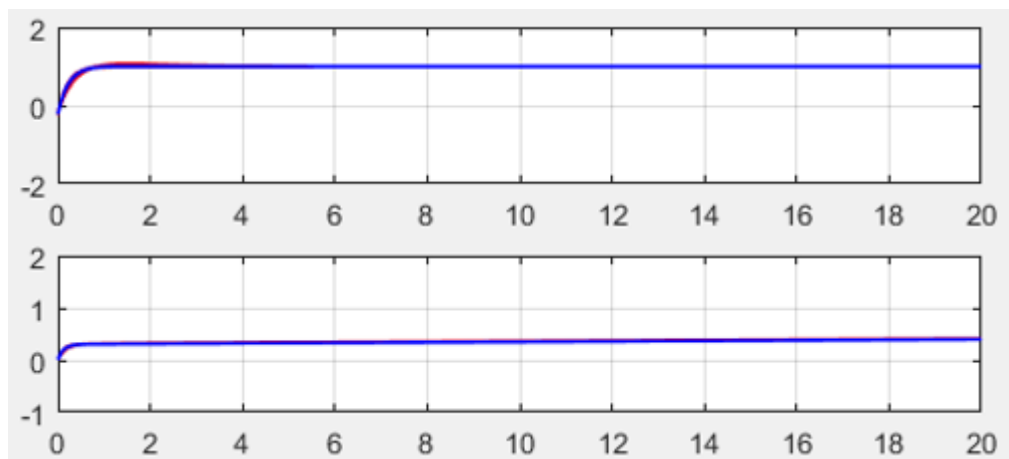


Figure IV. 9. Les meilleurs réponse et fonction d'objectif de chaque génération

La réponse indicielle et la fonction de cout pour les meilleurs contrôleurs de chaque génération sont présentés dans la figure IV.9. Sur ce graphique, les contrôleurs des premières générations sont plus rouges, tandis que les contrôleurs des générations ultérieures sont plus bleus. À mesure que l'algorithme génétique progresse, le contrôleur est capable de minimiser les oscillations de sortie et d'atteindre un temps de montée rapide (il arrive au signal idéale dans 2 secondes) .

IV.4. Contrôle de système par la logique floue

IV.4.1. Réglage du contrôleur PI avec la logique floue

L'algorithme PI discret est donné par :

$$u(k) = k_p e(k) + k_i T \sum_{j=0}^k e(j)$$

Où T est le temps d'échantillonnage, $e(k)$ est l'erreur à l'instant k , alors que $k_p > 0$, $k_i > 0$.

L'erreur $e(k)$ est définie comme :

$$e(k) = y_d(k) - y(k)$$

Où $y_d(k)$ est la valeur désirée.

Dans le contrôleur PI (IV.3), le réglage de k_p et k_i est une question importante. Pour concevoir les règles floues, nous considérons la plage de e et ec comme étant de 0 à 1, et nous définissons les ensembles flous de e et ec comme N, O, P, qui représentent respectivement négatif, zéro et positif.

IV.4.1.1. Règles floues pour le réglage de k_p

Le principe du réglage de k_p est le suivant : lorsque $e(k)$ est positif (e est P), nous devons augmenter Δk_p , donc Δk_p doit être positif ; lorsque $e(k)$ est négatif (e est N) et qu'un dépassement apparaît, nous devons diminuer k_p , donc Δk_p doit être négatif.

Lorsque l'erreur est proche de zéro (e est Z), nous avons trois conditions : si ec est N, la valeur du dépassement a tendance à augmenter, Δk_p doit être négatif ; si ec est Z, pour réduire l'erreur statique, Δk_p doit être positif ; si ec est P, l'erreur aura tendance à augmenter, nous devons donc augmenter k_p et Δk_p doit être positif. Ces règles sont données dans le Tableau IV.2.[19]

Ec	N	Z	P
E			
N	N	N	N

Z	N	P	P
P	P	P	P

Tableau IV. 2. Règles floues d'optimiser le k_p

IV.4.1.2. Règles floues pour le réglage de k_i

En utilisant la tactique de séparation de l'intégrale pour régler k_i , lorsque l'erreur est très faible, nous choisissons une grande valeur de Δk_i , sinon nous choisissons une très petite valeur de Δk_i . Ces règles sont données dans le Tableau IV.3.

E_c	N	Z	P
e			
N	Z	Z	Z
Z	P	P	P
P	Z	Z	Z

Tableau IV. 3. Règles floues pour optimiser le k_i

Nous pouvons régler k_p et k_i en ligne de la manière suivante :

$$k_p = k_{p0} + \Delta k_p; k_i = k_{i0} + \Delta k_i$$

IV.4.2. Simulation de logique floue-PI

Nous définissons la plage de e , ec , k_p et k_i en fonction du signal idéal, de l'erreur initiale et de notre expérience. Pour le système flou "a", si nous utilisons la commande "plotmf", nous pouvons obtenir les MF de e , ec , k_p et k_i .

La mise en œuvre d'un système d'inférence flou (SIF) pour le diagnostic sous Matlab peut être réalisée en suivant les étapes schématisées dans la figure (II.8).[19]

IV.4.2.1. Création du système d'inférence flou (SIF),

Dans cette étape de création du système d'inférence flou (SIF), on commence par choisir le type de système que l'on souhaite implémenter, soit le type Mamdani ou Sugeno.

La méthode de Mamdani, développée en 1975, est largement utilisée dans la théorie du contrôle de procédés. La méthode de Sugeno, quant à elle, diffère de la méthode de Mamdani

en ce qui concerne la définition de la variable de sortie et, par conséquent, les méthodes de défuzzification.

La fuzzification des variables d'entrée reste inchangée pour les deux méthodes. La variable de sortie peut prendre soit une valeur constante (singleton) indépendante des valeurs d'entrée, soit une combinaison linéaire de ces valeurs. De plus, chaque méthode présente ses propres caractéristiques et avantages. Pour la méthode de Mamdani :

- Elle est intuitive.
- Elle est largement utilisée.
- Elle simule le raisonnement humain.

D'autre part, la méthode de Sugeno présente les avantages suivants :

- Elle est très efficace du point de vue du calcul.
- Elle fonctionne bien avec les techniques linéaires, l'optimisation et les techniques adaptatives.
- Elle privilégie l'analyse mathématique.

Dans notre étude, nous avons choisi d'utiliser un système d'inférence flou (SIF) de type Mamdani, bien qu'il soit possible d'utiliser le système de type Sugeno.

La création d'un système d'inférence flou (SIF) de diagnostic à l'aide des commandes Matlab se fait en utilisant la commande "newfis", qui accepte jusqu'à 7 arguments. La syntaxe générale avec les 7 arguments est la suivante :

```
Sys_flou = newfis('nom_syst', 'type', 'ET_method', 'OU_method', 'imp_method',  
'agg_method', 'defuzz_method');
```

- ✓ Name='fuzzpi'
- ✓ Type='mamdani'/Version=2.0
- ✓ ET_Method='min'
- ✓ OU_Method='max'
- ✓ Imp_Method='min'
- ✓ Agg_Method='max'
- ✓ Defuzz_Method='centroid'

La figure IV.10. Présente un schéma du plan de création d'un SIF.

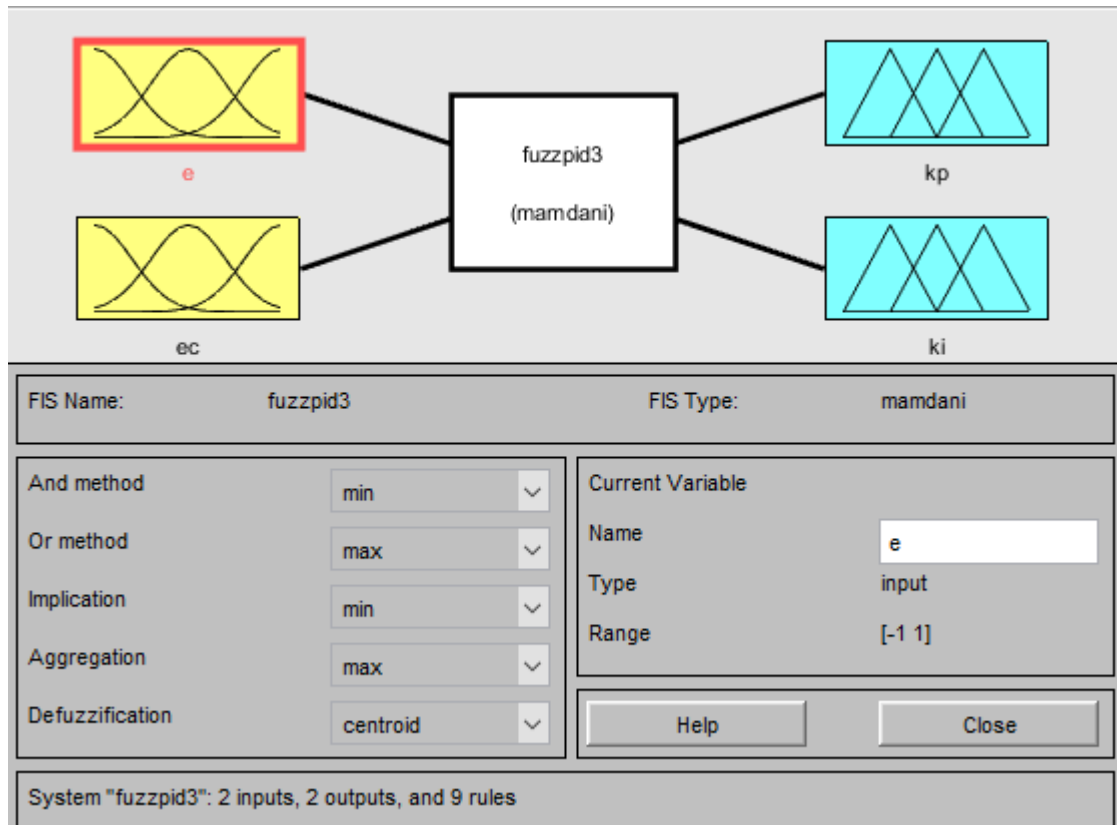


Figure IV. 10. Création de système d'inférence flou

IV.4.2.2 Fuzzification des variables d'entrées et de sorties

La fuzzification des variables d'entrée et de sortie est effectuée en utilisant les commandes “addvar” et “addmf” dans Matlab.

La définition des variables d'entrée et de sortie se fait avec la commande “addvar” :

- ❖ `Sys_flou = addvar('nom_syst', 'type', 'nom', 'intervalle');`
- ❖ “nom_syst” : nom du système flou,
- ❖ “type” : variable d'entrée ('input') ou de sortie ('output'),
- ❖ “nom” : nom de la variable auquel les règles floues se référeront,
- ❖ “intervalle” : intervalle de définition de la variable.

Après avoir défini les différentes variables d'entrée et de sortie avec la commande “addvar”, les différentes fonctions d'appartenance associées à ces variables sont spécifiées avec la commande “addmf”. Après l'exécution de programme on obtient les figures suivantes :

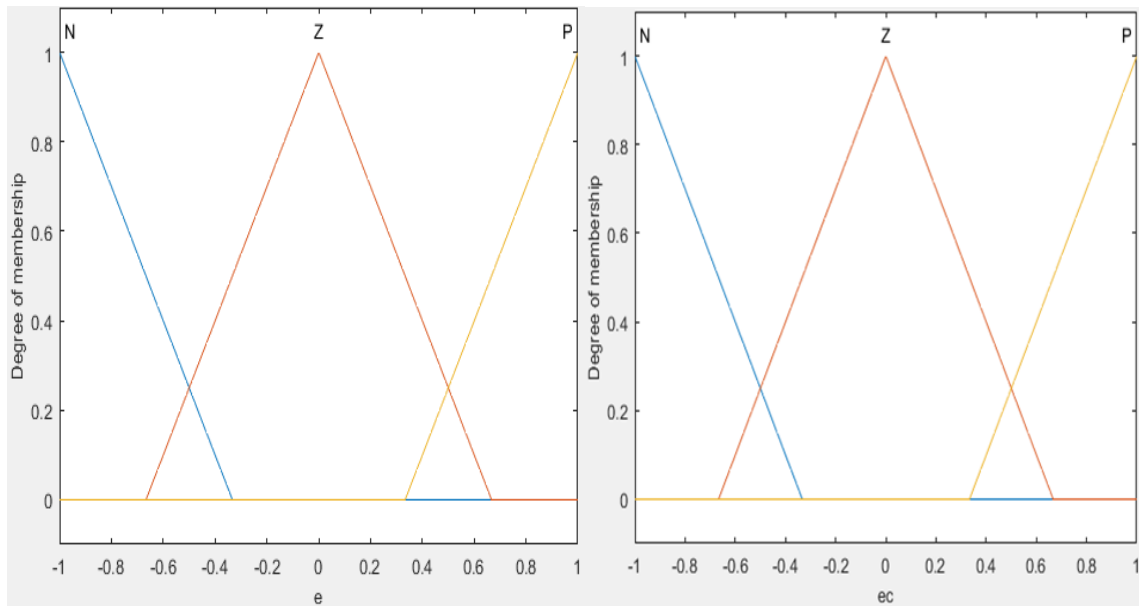


Figure IV. 11. Fonctions d'appartenance de e et e'

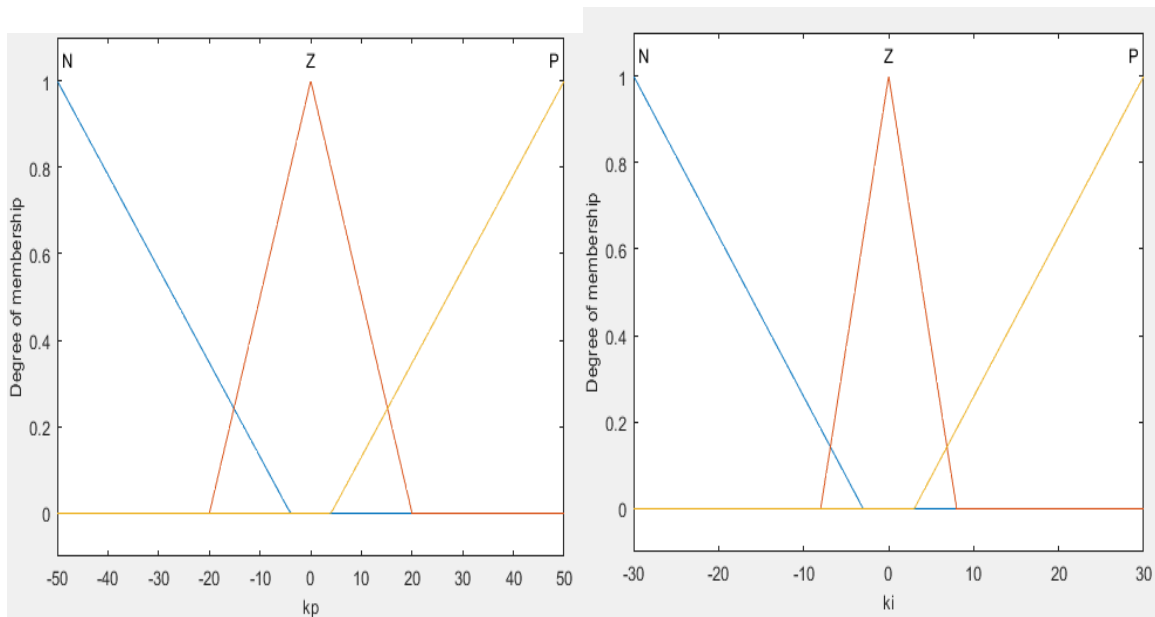


Figure IV. 12. Fonctions d'appartenance de k_p et k_i

IV.4.2.3. Les règles de SIF :

En utilisant la commande “showrule(a)”, nous pouvons obtenir 9 règles floues :

- Si (e est N) et (ec est N), alors (k_p est N) et (k_i est Z) (1)
- Si (e est N) et (ec est Z), alors (k_p est N) et (k_i est Z) (1)
- Si (e est N) et (ec est P), alors (k_p est N) et (k_i est Z) (1)
- Si (e est Z) et (ec est N), alors (k_p est N) et (k_i est P) (1)
- Si (e est Z) et (ec est Z), alors (k_p est P) et (k_i est P) (1)

- f) Si (e est Z) et (ec est P), alors (k_p est P) et (k_i est P) (1)
- g) Si (e est P) et (ec est N), alors (k_p est P) et (k_i est Z) (1)
- h) Si (e est P) et (ec est Z), alors (k_p est P) et (k_i est Z) (1)
- i) Si (e est P) et (ec est P), alors (k_p est P) et (k_i est Z) (1)

On déclare ces règles dans Matlab est on l'exécute.[17]

IV.4.2.4. La défuzzification :

Dans l'étape 3.4.4, la défuzzification est réalisée pour obtenir une valeur réelle de sortie à partir des surfaces obtenues lors de l'étape d'inférence floue. La méthode la plus couramment utilisée pour la défuzzification est celle du centre de gravité.

Pour observer la défuzzification selon la méthode choisie, tu peux utiliser la commande "ruleview" pour afficher la fenêtre Rule Viewer :

On obtient les valeurs finales de k_p et k_i comme il est présenté dans la figure IV.13

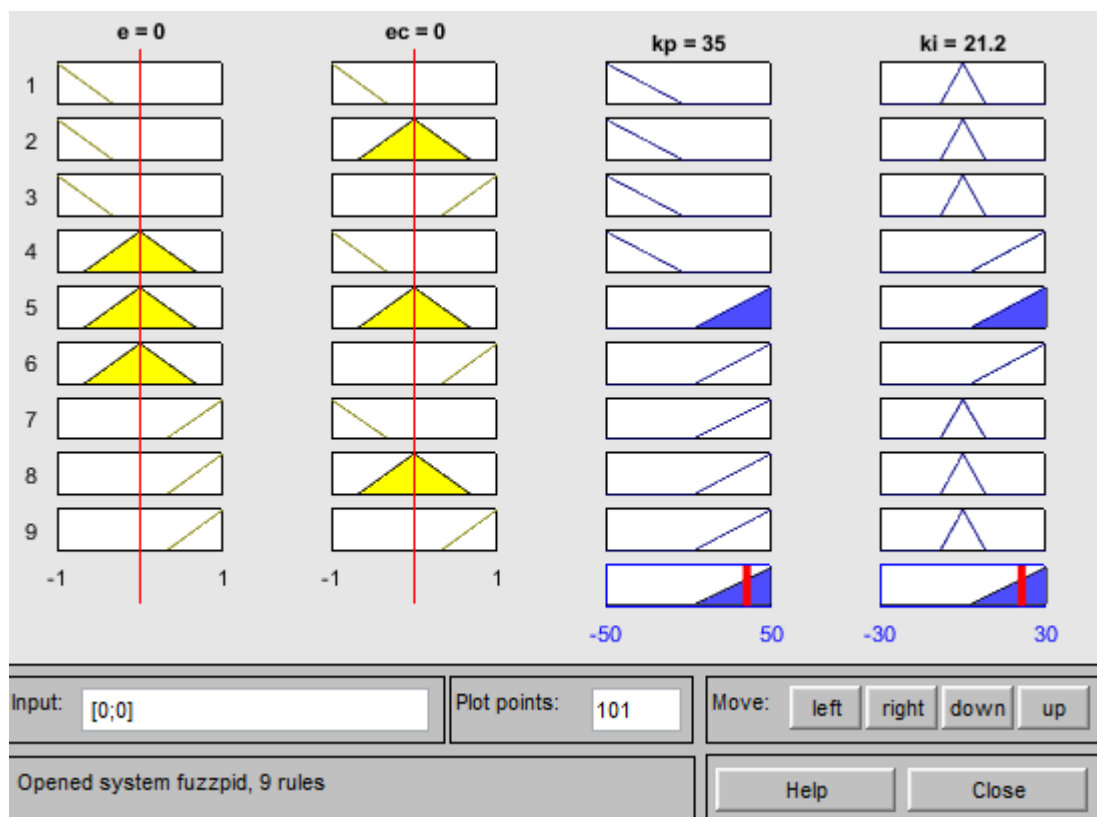


Figure IV. 13. Optimisation des valeurs de k_p et k_i

IV.4.2.5 Test du système d'inférence flou (SIF)

Le test du système d'inférence flou (SIF) peut être obtenu par la commande evalfis. Si la matrice représentant le système d'inférence flou (SIF) n'est pas présente dans l'espace de travail, sa lecture se fait par la commande readfis. [17]

Après l'exécution on obtient :

1. Le signal de l'entrée présenté dans la figure (IV.14.), qu'il se varie vers une valeur constante qui permet de la stabilité de système à une valeur constante.

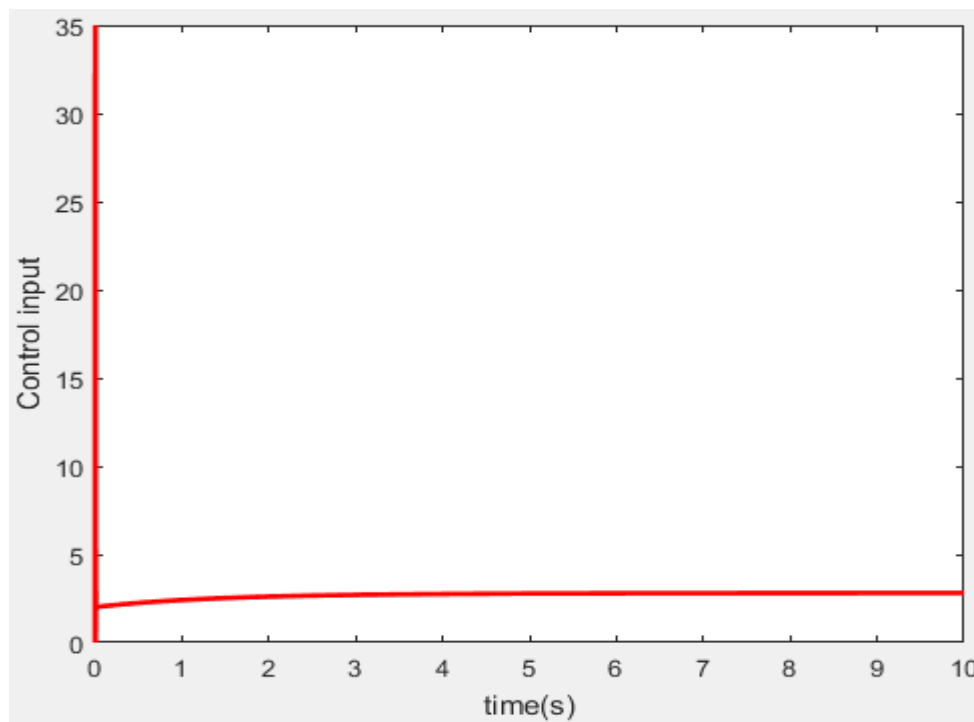


Figure IV. 14. Le signal de l'entrée

2. Le meilleur contrôleur est présenté dans la figure IV.15. Le contrôleur PI tend vers une valeur optimisée d'après l'erreur et la variation de l'erreur pour garantir la stabilité de système.

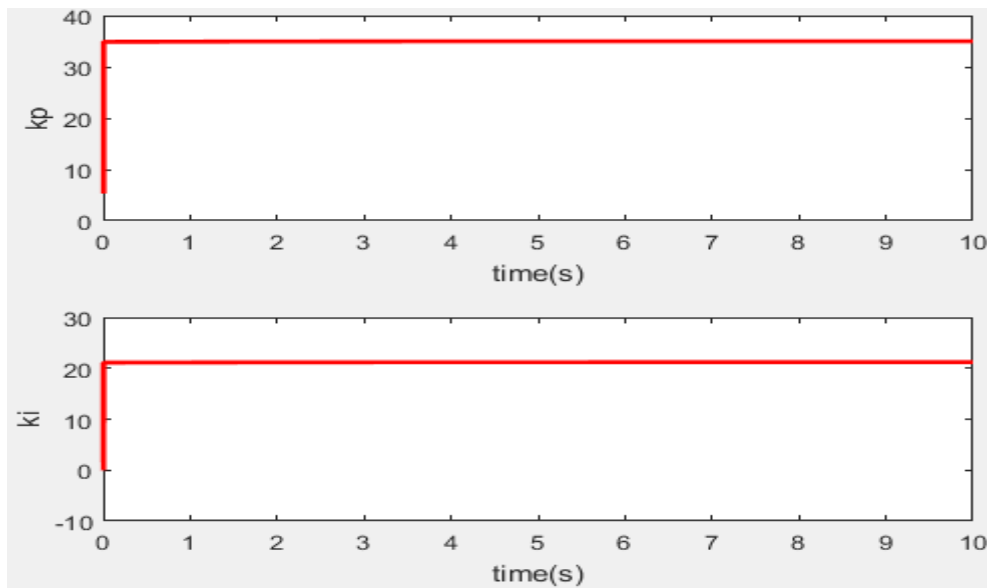


Figure IV. 15. Variation de k_p et k_i au cours du temps

3. La réponse indicielle de système contrôlé est présentée dans la figure IV.16. Elle tend vers une valeur idéale au cours de 5 secondes avec l'absence des oscillations

On voit la rapidité de système est efficace. Avec l'absence des oscillations et de dépassement.

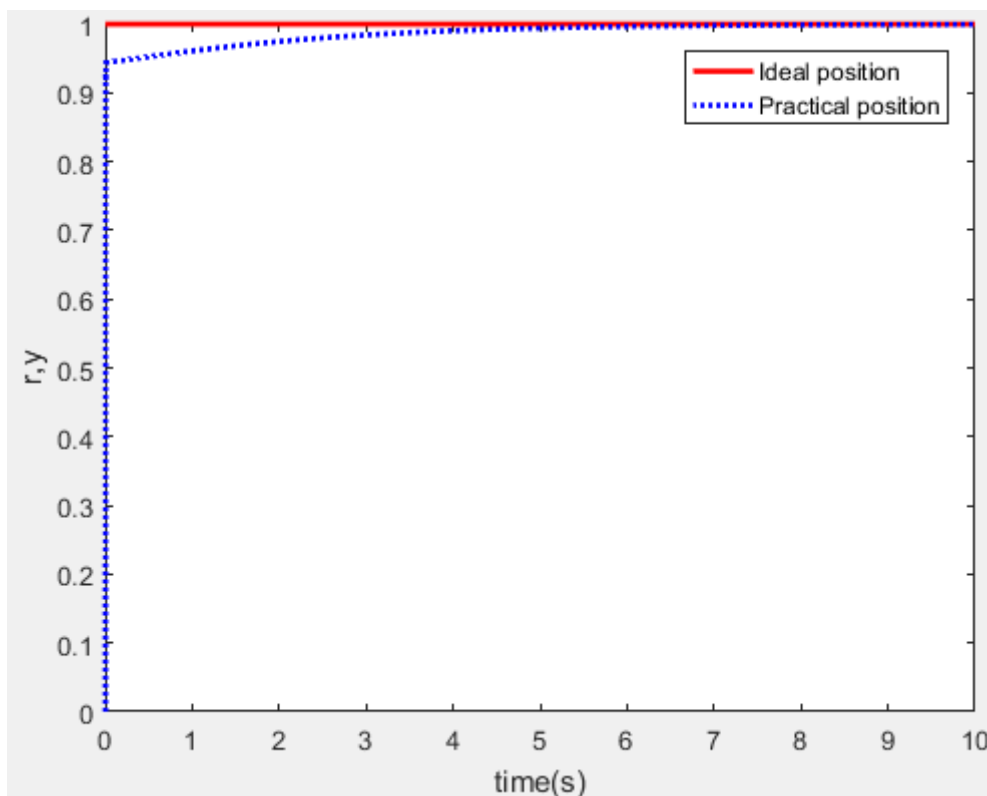


Figure IV. 16. La réponse indicielle du système contrôlé

IV.5. Le mode glissant flou

La commande par mode glissant a démontré son efficacité à travers diverses études théoriques, principalement dans les domaines de la robotique et de la commande des moteurs électriques. Son avantage clé réside dans sa robustesse face aux perturbations et aux incertitudes du modèle. Cependant, ces performances sont souvent accompagnées d'inconvénients tels que le phénomène de chattering ou broutement causé par la partie discontinue de la commande, pouvant avoir un impact néfaste sur les actionneurs. De plus, le système est constamment soumis à une commande élevée pour assurer sa convergence vers l'état désiré, ce qui n'est pas souhaitable.

Pour remédier à ces problèmes, plusieurs solutions ont été proposées. Parmi celles-ci, la commande par mode glissant à bande limite consiste à remplacer la fonction de commutation par une fonction de saturation. Toutefois, cette approche n'est qu'un cas particulier de la commande par mode glissant flou. Ainsi, il est intéressant d'utiliser une commande qui combine la logique floue et le mode glissant afin d'obtenir une commande à la fois robuste et régulière.

Certains chercheurs ont cherché à établir une similitude entre la commande floue non linéaire et la commande par mode glissant, en combinant les deux approches. Cela implique d'utiliser la commande floue pour sa rapidité et sa facilité de mise en œuvre, et le régime glissant pour ses fondements théoriques rassurants en termes de stabilité et de robustesse. Cette combinaison, qui tire parti de l'invariance de la commande à régime glissant face aux incertitudes et aux perturbations, ainsi que de la rapidité et de la précision de la commande floue dans le suivi des trajectoires, permet de surmonter les problèmes de broutement associés à la commande par mode glissant, tout en réduisant le nombre de règles floues nécessaires.

Dans cette partie, nous examinerons un contrôleur flou dont le fonctionnement reflète le principe de la commande par mode glissant pour assurer la stabilité et la robustesse du système. Ensuite, nous étudierons un contrôleur combinant le mode glissant et la logique floue.[20]

IV.5.1. Contrôle de mode glissant flou basé sur un contrôle équivalent

Le contrôle équivalent de mode glissant a été décrit. La loi de commande se compose d'un contrôle équivalent et d'un contrôle de commutation u . Les états du système sont maintenus

sur la surface de glissement par un contrôle équivalent, et les états du système atteignent la surface de glissement par un contrôle de commutation. En utilisant des règles floues, le système flou est établi en fonction du contrôle équivalent et du contrôle de commutation. Le contrôle de mode glissant basé sur un contrôle équivalent utilise le contrôle de commutation à gain de commutation pour réduire le phénomène de bavardage et garantir la stabilité de Lyapunov. Un petit contrôle de commutation à gain de commutation est adopté lorsque les amplitudes des perturbations sont faibles, et un grand contrôle de commutation à gain de commutation est adopté lorsque les amplitudes des perturbations sont élevées. Cette fonction peut être réalisée par un système flou avec des règles floues.

IV.5.2. Conception du contrôle flou

Considérez le système suivant avec des perturbations et des incertitudes :

$$x'' = f(x, t) + g(x, t)u(t) + d(t)$$

$X(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T = [x(t), \dot{x}(t), \dots, x^{(n-1)}(t)]^T$ Le vecteur d'état.

$X_d(t) = [x_{1d}(t), x_{2d}(t), \dots, x_{nd}(t)]^T = [x_d(t), \dot{x}_d(t), \dots, x_d^{(n-1)}(t)]^T$ Le vecteur d'état désiré.

$f(X, t)$ et $b(X, t)$ sont des fonctions du vecteur d'état, $u(t)$ est la commande et $d(t)$ la perturbation considérée comme étant bornée : $|d(t)| \leq D$.

Soit l'erreur de suivi $e = x_d - x$, et soit la fonction de commutation :

$$S = ce + e'$$

Selon la théorie du contrôle de mode glissant, un régulateur de mode glissant se compose d'un contrôle équivalent de mode glissant et d'un contrôle de commutation. La loi de commande est la suivante :

- Si $s(t)$ est N, alors u est P (4.1)
- Si $s(t)$ est Z, alors u est Z (4.2)
- Si $s(t)$ est P, alors u est P (4.3)

Où les ensembles flous Z, N et P désignent respectivement "zéro", "négatif" et "positif".

La règle floue (4.2) indique que le régulateur flou est le contrôle équivalent lorsque la fonction de commutation est égale à zéro. De même, les règles floues (4.1) et (4.3) indiquent que le régulateur flou est le contrôle équivalent $u + ux$ lorsque la fonction de commutation n'est pas égale à zéro.

La sortie du système de déduction floue est l'appartenance. Avec une méthode de floue, le contrôle flou est conçu comme suit :

$$u = u_{eq} + \mu \cdot u_s$$

Lorsque $\mu = 1$, $u = u_{eq} + u_s$. Par conséquent, la loi de commande est le contrôle équivalent de mode glissant traditionnel. Lorsque $\mu \neq 1$, le phénomène de bavardage peut être réduit par la variation de l'appartenance de μ . [21]

IV.5.3. Application sur système :

Le système FESTO est présenté avec l'équation différentielle suivante :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = k_1 x_1 + k_2 x_2 + r u(t) + d(t) \end{cases}$$

Avec : $k_1 = -29,85$

$k_2 = -31,47$

$r = 1$

L'approche de la commande par mode glissant flou pour la poursuite de trajectoire est de trouver une loi de commande telle que, étant donnée une trajectoire désirée $x_d(t)$, l'erreur de poursuite $x(t) - x_d(t)$ tend vers zéro malgré la présence des perturbations externes.

Le vecteur d'erreur de poursuite est défini par :

$$E(t) = X(t) - X_d(t) = [e(t), \dot{e}(t), \dots, e^{(n-1)}(t)]^T$$

$$E(t) = [e_1(t), \dots, e_{n-1}(t), e_n(t)]^T$$

Et l'erreur de poursuite :

$$e(t) = x(t) - x_d(t)$$

Donc la dérivée est :

$$\dot{e}(t) = \dot{x}(t) - \dot{x}_d(t)$$

On définit une fonction linéaire S qui représente la surface de glissement :

$$S(X, t) = c_1 e_1 + c_2 e_2 + \dots + c_{n-1} e_{n-1} + e_n$$

Alors l'équation de surface de notre système est

$$S(X, t) = c e_1 + e_2$$

Avec $e_1 = x_1(t) - x_{1d}(t)$ et $e_2(t) = x_2(t) - x_{2d}(t)$

L'expression de la commande devient :

$$u = u_1 = u_{1eq} - K_1 Sat(S_1 b_1(X) / \Phi_1)$$

avec

$$u_{1eq} = \frac{-c_1 e_2 - f_1(X) + \dot{x}_{2d}}{b_1(X)}$$

$X = [x_1, x_2, x_3]$, x_1 et x_2 représentent respectivement la position et la vitesse, u est l'entrée de contrôle.

Le modèle Simulink est présenté dans la figure IV.18

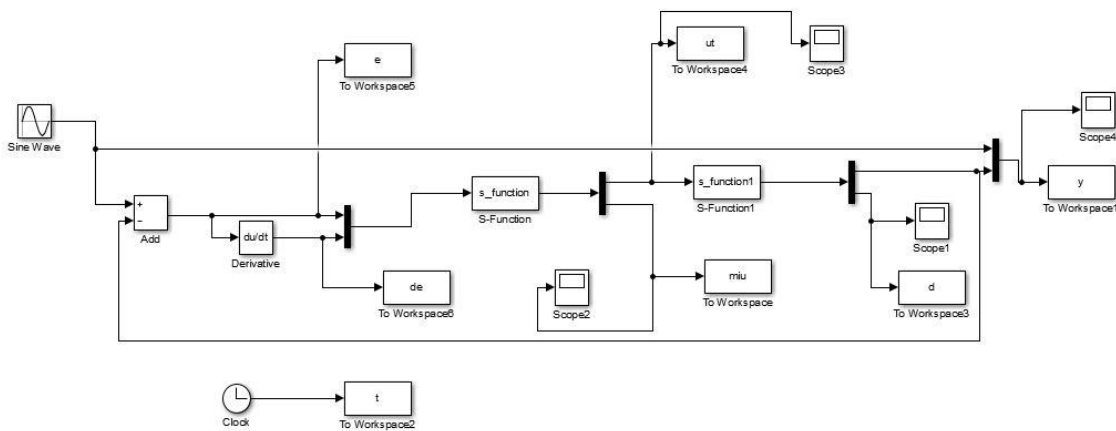


Figure IV. 17. Le modèle Simulink de mode glissant flou

On ajoute une perturbation pour tester l'efficacité de notre méthode de contrôler le système FESTO.

Considérons la perturbation $d(t)$ sous la forme d'une fonction de Gauss suivante :

$$D(t) = 5 \exp\left(-\frac{(t - c_i)^2}{2b_i^2}\right)$$

Où $b_i = 0,50$, $c_i = 5,0$ et $S = 0,15$. La borne supérieure de la perturbation est $D = \max(d(t)) + 1 = 5,15$. La perturbation $d(t)$ est présentée dans la Figure IV.17. La trajectoire de position souhaitée est $x = \sin(t)$.

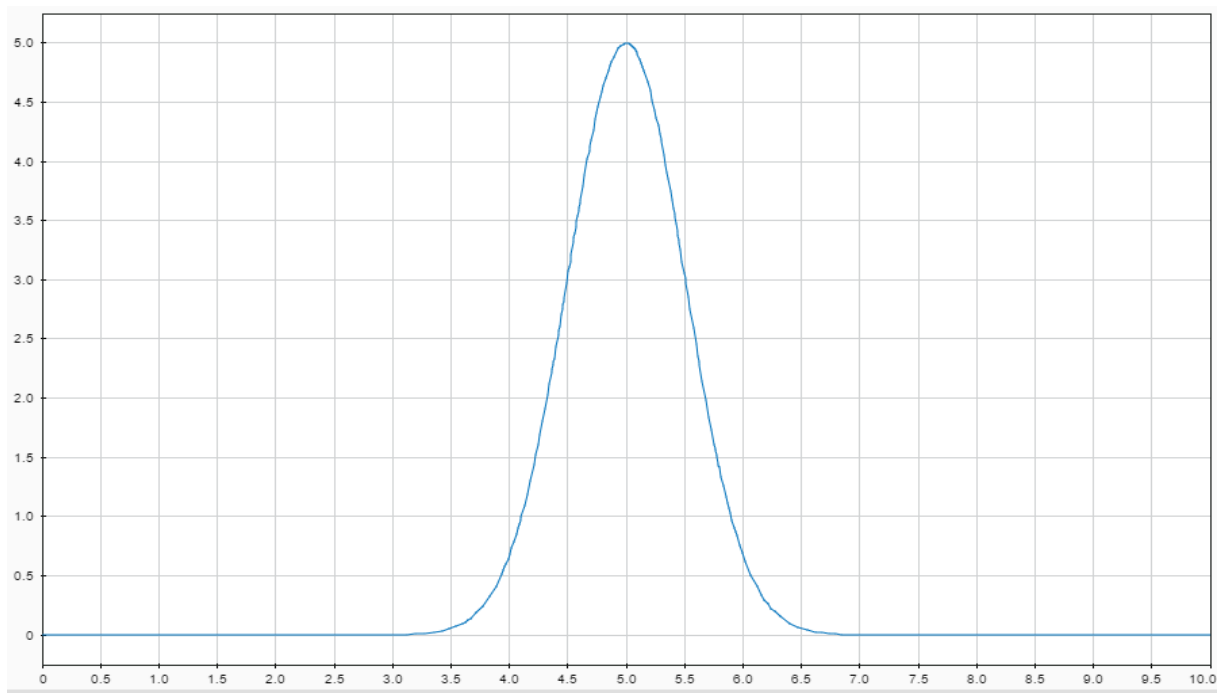


Figure IV.18 La perturbation en forme d'une fonction gaussienne

Le système flou est établi dans la fonction_s et la bibliothèque de règles est maintenue en cours d'exécution grâce à la commande de persistance. Les fonctions d'appartenance des entrées et des sorties du système flou sont présentées dans les figures 9.2 et 9.3 respectivement.

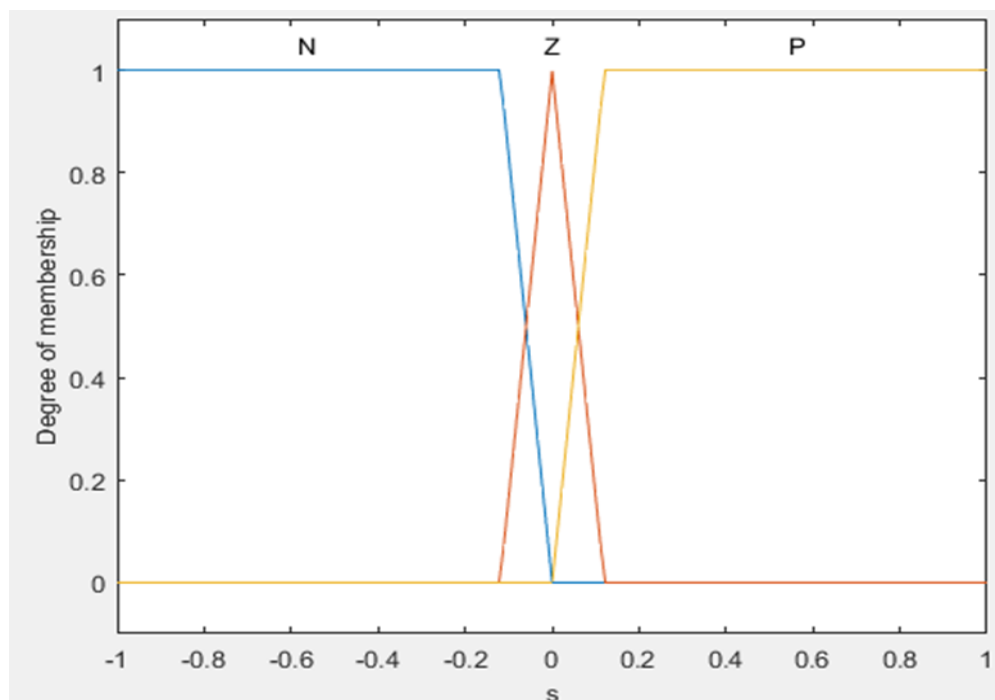


Figure IV.19 Fonction d'appartenance de $s(t)$

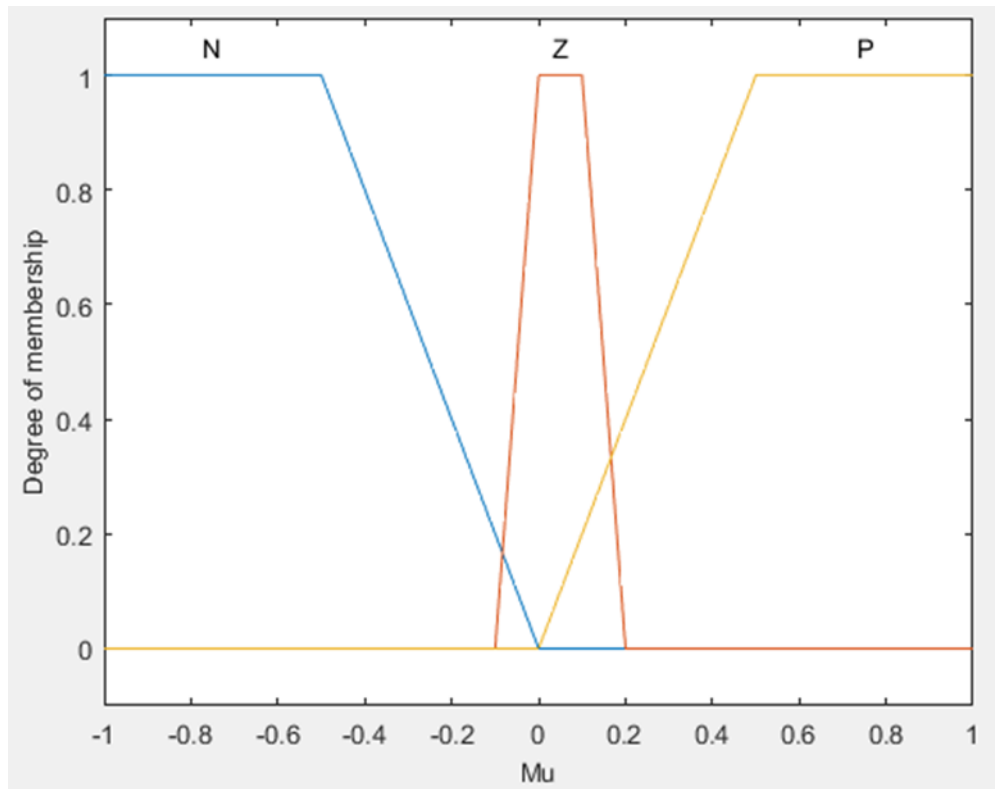


Figure IV. 20. Fonction d'appartenance de μ .

La règle floue est conçue comme suit :

- (1) Si (s est N) alors (Mu est P)
- (2) Si (s est Z) alors (Mu est Z)
- (3) Si (s est P) alors (Mu est P)[20]

IV.5.4. Résultats:

IV.5.4.1. Le mode glissant traditionnel

Pour $\mu = 1$

On donne le système une entrée signosuidale et on exécute le programme

- ✓ On obtient une sortie signusoidale très qui tends vers le signal idéal dans façon rapide et continue en cas stable malgré la perturbation sitée . elle est présenté dans la figure IV.20.

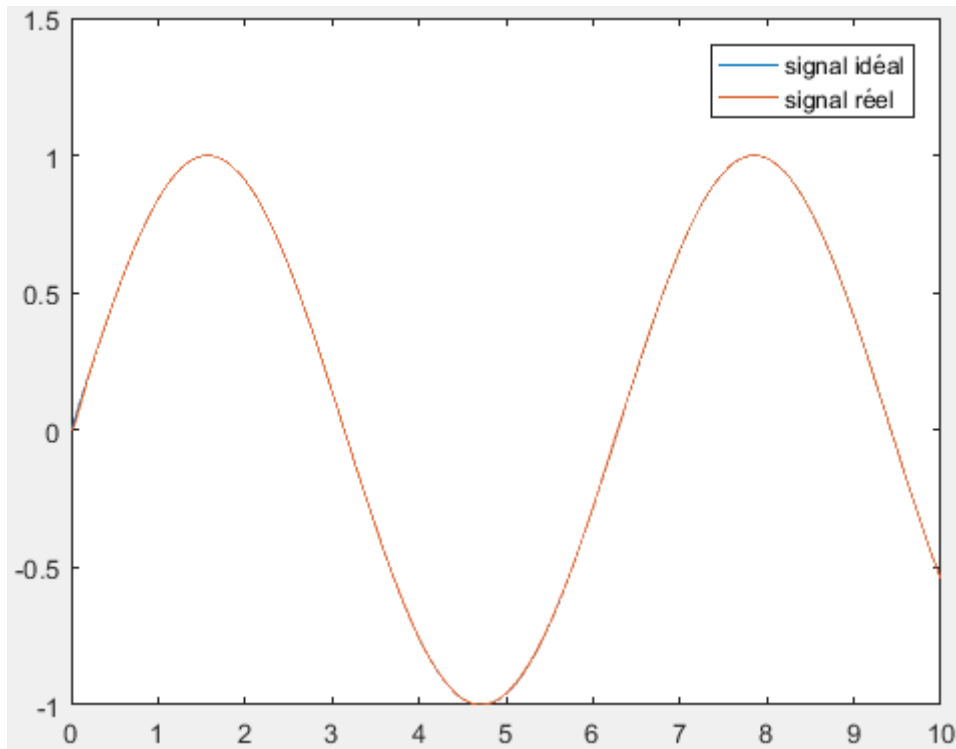


Figure IV. 21. La réponse fréquentielle de système contrôlé

- ✓ Avec une commande $u(t)$ présentée dans la figure IV.22. Elle implique la stabilité de système mais on observe le phénomène de brouttement.

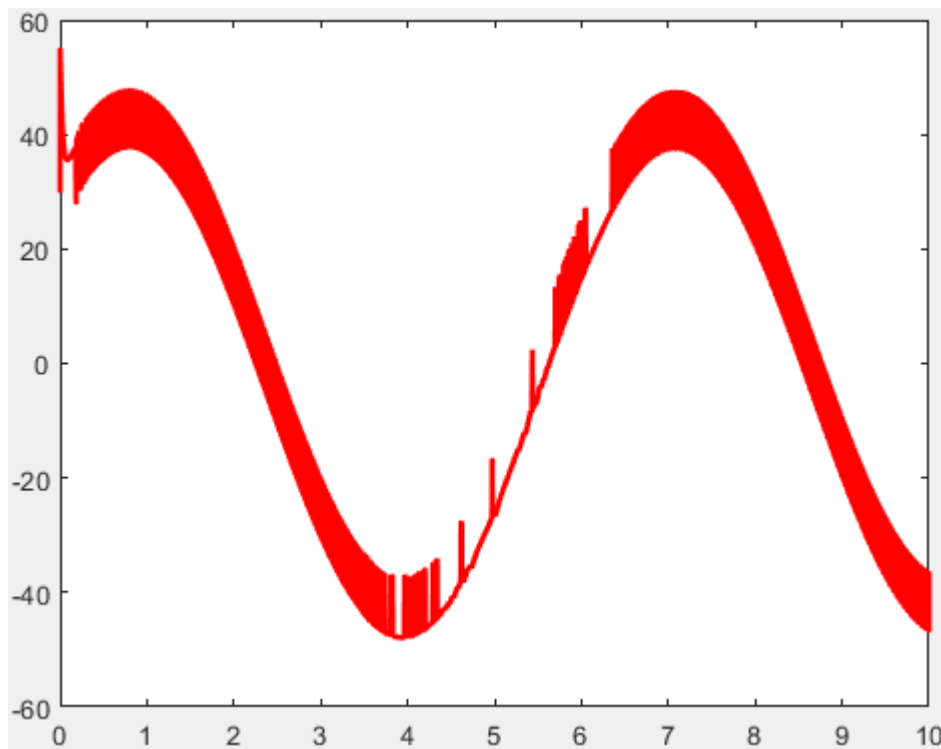


Figure IV. 22. La commande $u(t)$

- ✓ L'erreur entre la trajectoire souhaitée et réelle tends vers une valeur de zéro ,il est présenté dans la figure IV.23. Cela indique la bonne performance de controle pour obtenir une performance précise et fiable.

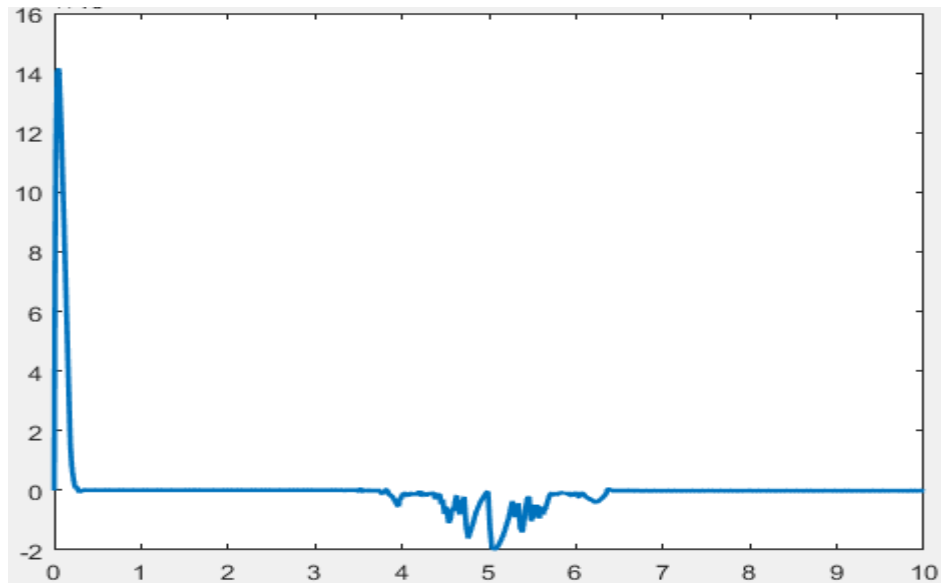


Figure IV. 23. L'erreur entre le signal réel et souhaité

- ✓ Ainsi que la dérivation de l'erreur est présenté dans la figure IV.24. Cela peut indiquer la stagnation de l'erreur.

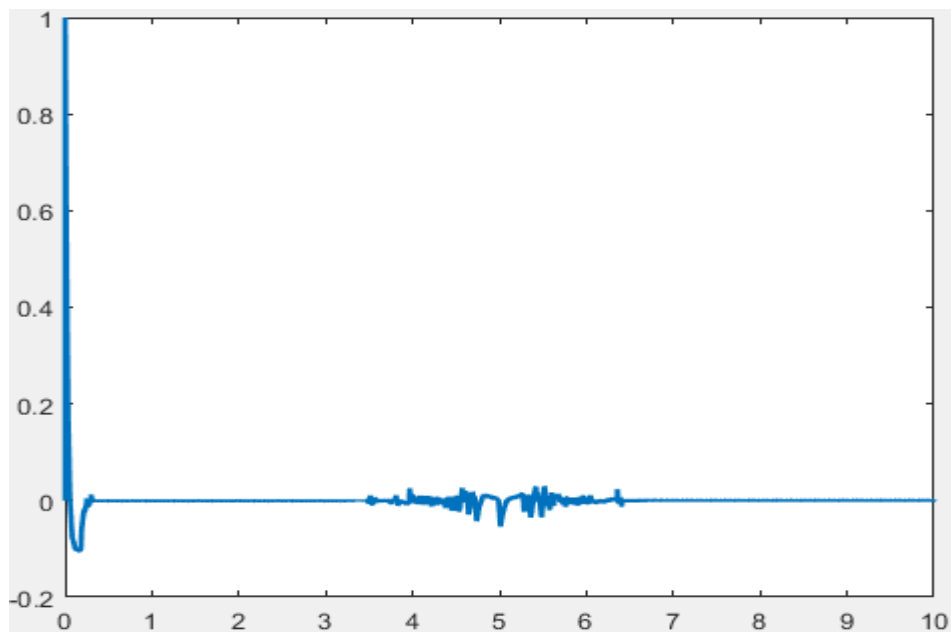


Figure IV. 24. La dérivée de l'erreur

- ✓ La surface de glissement est alors présenté dans la figure IV.25.

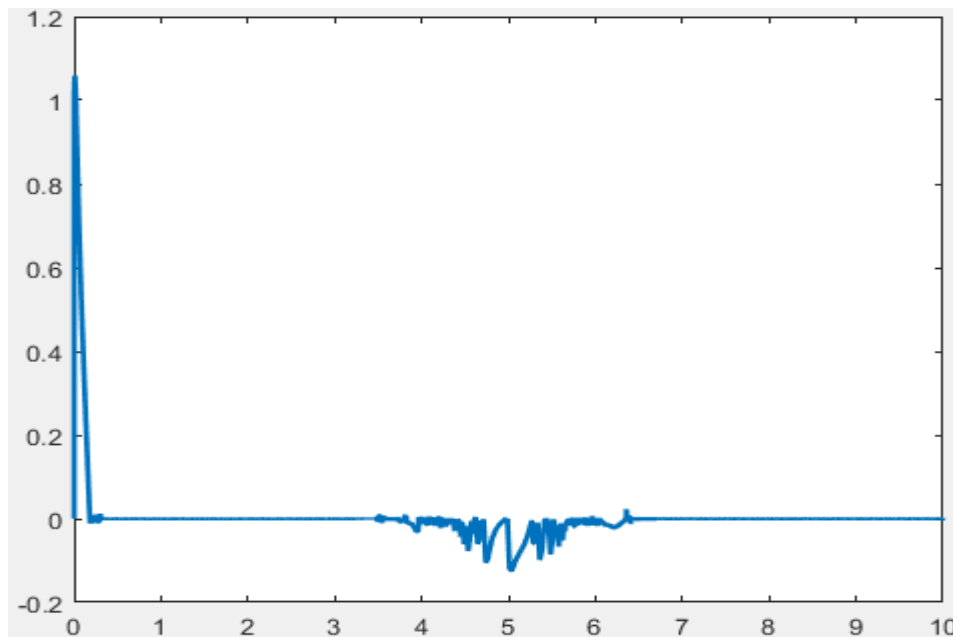


Figure IV. 25. La surface de glissement

Le mode glissant nous permet d'obtenir une réponse de contrôle efficace même en présence des perturbations, en ajustant les actions de contrôle pour minimiser l'erreur entre la sortie réelle et souhaitée. Cela garantit une meilleure robustesse et une adaptation aux variations du système.

IV.5.4.2. Le mode glissant flou

Pour $\mu \neq 1$

On obtient :

- Une sortie très proche au réponse idéale avec une rapidité remarquée .

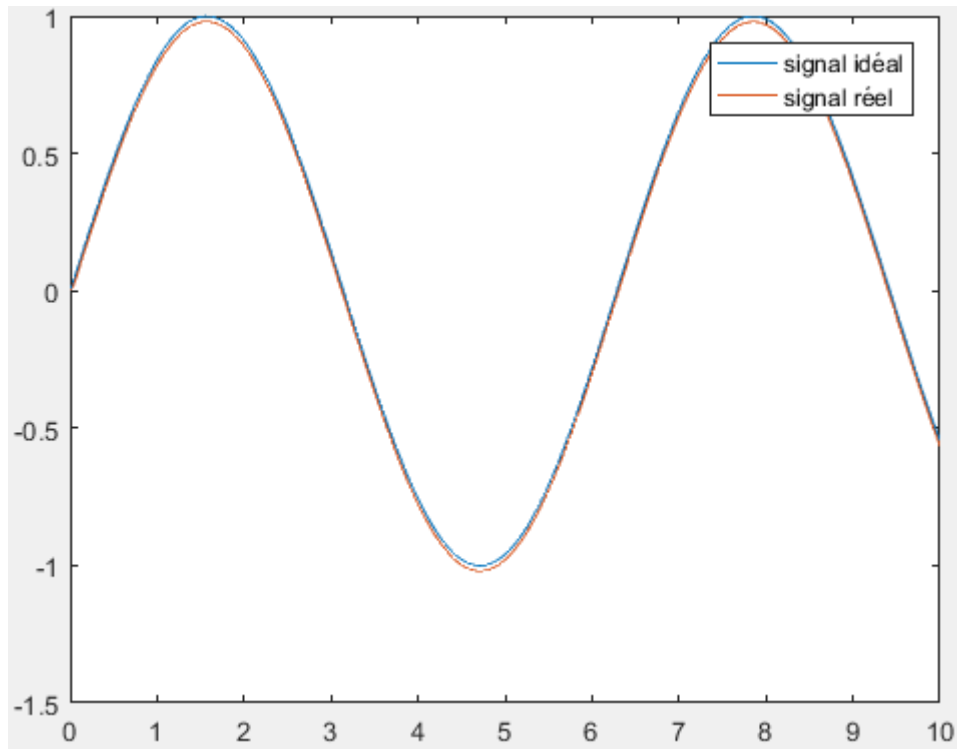


Figure IV. 26. La réponse fréquentielle de système contrôlé

- Une commande $u(t)$ qui assure la stabilité et l'efficacité de système avec l'absence des oscillations qui sont au mode glissant classique alors notre mode de control à éliminer les broutements. La commande est présentée dans la figure IV.27.

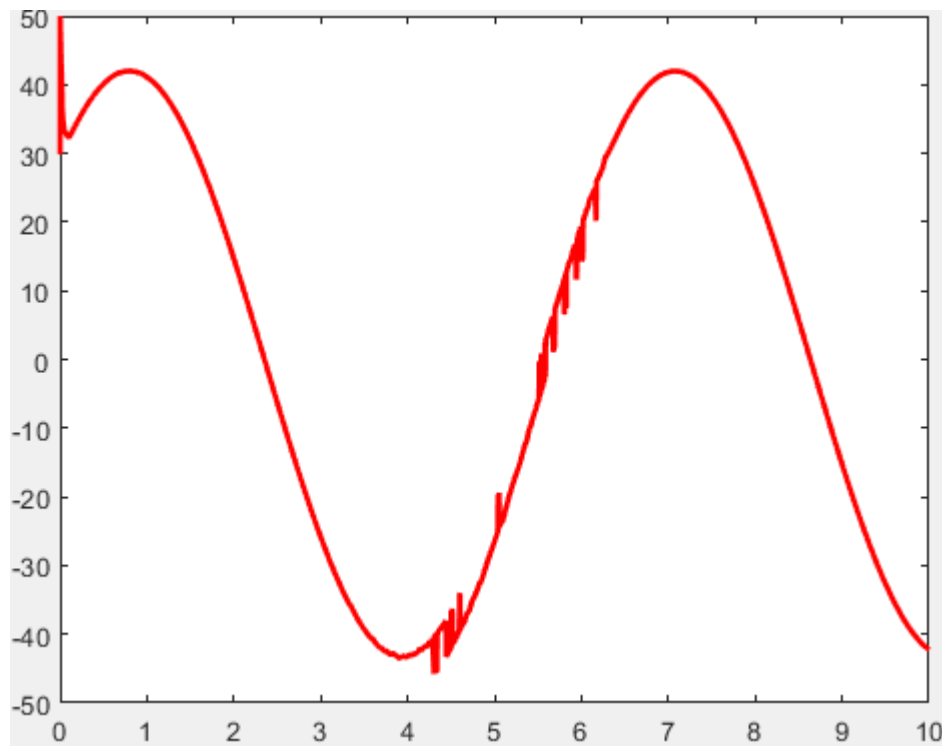


Figure IV. 27. L'entrée de commande $u(t)$

- La variation de coefficient μ déduit par la logique floue est présentée dans la figure IV.28. Le coefficient tends vers -1 en cas de perturbation. Cela signifie que la contribution de la commande de commutation est devenue très importante devant la commande équivalente.

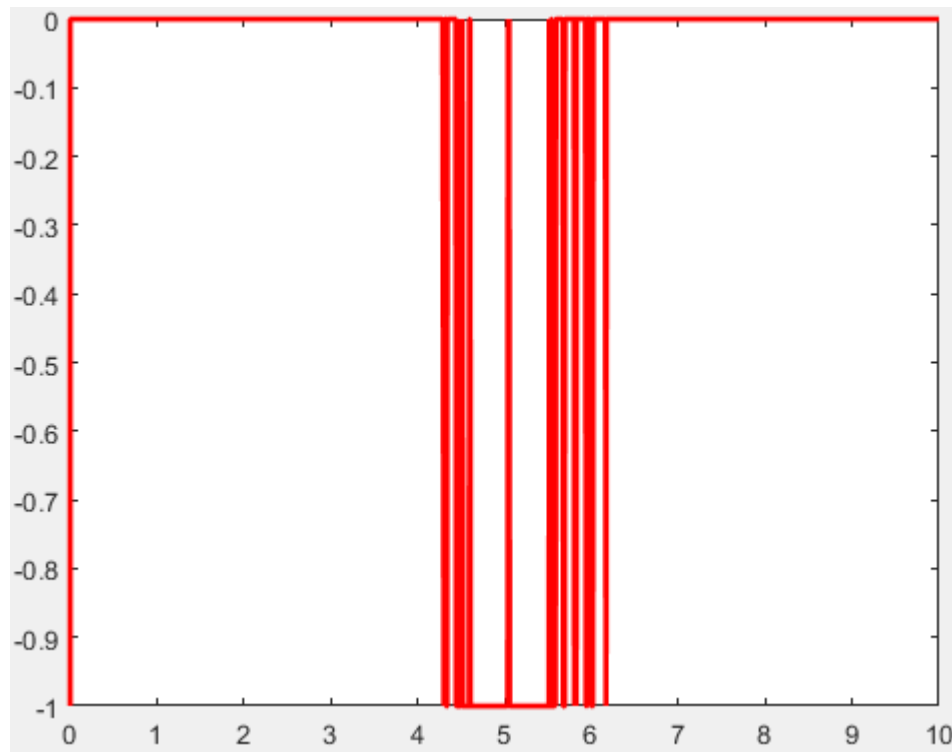


Figure IV. 28. La variation de μ

Le mode glissant floue nous permet d'obtenir une réponse de contrôle plus douce et adaptative grâce à l'utilisation des règles floues qui ajuste les actions de contrôle en fonction des conditions spécifiques de système. Cela peut conduire à une meilleure réduction de broutement et à une amélioration des performances globales du système.

IV.6. Conclusion

En conclusion de ce quatrième chapitre, nous avons exploré l'application des techniques d'optimisation génétique, floue et glissante au système contrôlé FESTO. L'optimisation génétique a permis d'optimiser les paramètres du contrôleur PI, offrant ainsi une réponse précise et stable du système.

L'application de la logique floue a permis d'adapter le contrôle en fonction des conditions changeantes du système, en utilisant des ensembles flous et des règles linguistiques pour la

prise de décision. Enfin, l'application du mode glissant a renforcé la stabilité et la résistance aux perturbations du contrôleur PI de FESTO.

Ces applications combinées des techniques d'optimisation génétique, floue et glissante ont permis d'améliorer les performances de contrôle de FESTO, offrant une réponse adaptative et précise pour notre système.

Conclulsion générale

Notre étude a exploré l'application des algorithmes génétiques, de la logique floue et du glissement flou dans le domaine du contrôle. Nous avons examiné comment ces approches peuvent être combinées pour résoudre des problèmes de contrôle complexes et améliorer les performances des systèmes.

L'utilisation des algorithmes génétiques dans le processus d'optimisation du contrôle a démontré leur efficacité pour trouver des solutions optimales dans des espaces de recherche complexes. En combinant ces algorithmes avec la logique floue, nous avons pu modéliser et représenter des connaissances incertaines de manière plus précise, permettant ainsi une prise de décision plus adaptative et flexible.

Le glissement flou (fuzzy sliding) s'est révélé être une approche prometteuse pour résoudre les problèmes de contrôle en présence de perturbations et d'incertitudes. En utilisant des règles floues pour ajuster dynamiquement les gains de contrôle, nous avons pu maintenir la stabilité et améliorer les performances du système même dans des conditions changeantes.

Notre étude a également souligné l'importance de la modélisation et de la sélection appropriées des paramètres pour obtenir des résultats optimaux. L'ajustement des paramètres des algorithmes génétiques, des systèmes de logique floue et des mécanismes de glissement flou doit être soigneusement réalisé pour éviter les problèmes tels que le surapprentissage ou la sous-optimisation.

En conclusion, l'intégration des algorithmes génétiques, de la logique floue et du glissement flou offre des perspectives intéressantes pour résoudre des problèmes de contrôle complexes. Ces approches permettent d'améliorer l'adaptabilité, la flexibilité et les performances des systèmes de contrôle dans des environnements incertains. Cependant, des recherches supplémentaires sont nécessaires pour affiner et optimiser ces approches, ainsi que pour les appliquer à une gamme plus large de problèmes de contrôle.

En somme, notre travail de recherche a apporté des contributions significatives à la compréhension et à l'application des techniques de contrôle basées sur les algorithmes génétiques, la logique floue et le glissement flou. Nous espérons que cette étude servira de base pour de futures recherches dans le domaine et aidera à améliorer les performances des systèmes de contrôle dans diverses applications industrielles et scientifiques.

Les références

- [1] Nicholas DURAND. Algorithmes génétiques et autres outils d'optimisation appliqués à la gestion du trafic aérien. Optimisation et contrôle [math OC]. INPT, 2004.
- [2] Nawel TEJ, Samir NEJIM, Naceur benhadj Braiek. Algorithmes génétiques appliqués à l'optimisation des lois de commande. Presses académiques Francophones. 2004. P(14.20)
- [3] Z. Michalewicz. Genetic algorithms + data structures = evolution programs (3^{ème} éd.). Springer-Verlag, 1996.
- [4] T. Bäck, D. B. Fogel, and Z. Michalewicz. Handbook of Evolutionary Computation. Institute of Physics Publishing, Bristol, UK, 1997.
- [5] Steven L. Brunton, J Nathan Kutz. Data Driven Science and Engineering. Cambridge University Press, 2019. P(256.263)
- [6] François CHEVRIE et François GUELY. La logique floue .Cahier technique Schneider .n° 191 Edition mars 1998.
- [7] Li, X., Li, Y., & Ren, J. An overview of fuzzy logic control. International Journal of Control, Automation and Systems, 2019. P(776-789).
- [8] A. Tchamkerten, A. Benoit. "Fuzzy Logic and Boolean Logic: Basic Differences and Applications" (2019)
- [9] BENZENATI .S et RABIAI .M. Commande floue d'un moteur à courant continu à excitation séparée, Mémoire de fin d'étude, Université Abderrahmane Mira Bejaia, 2013.
- [10] G.J. Klir et B. Yuan. Fuzzy Sets and Fuzzy Logic Theory and Applications, selected paper (Vol.6). World scientific, 1996.
- [11] M. SAHRAOUI. "Commandes non linéaires de la machine asynchrone," Université de M'sila, 2013.
- [12] T.Hicham. "Contrôle par mode glissant: Observation et estimation paramétrique d'une machine à induction avec défauts Présenté," Université Mohamed Khider – Biskra, 2016.
- [13] B. Przemysław , Ignaciuk .Andrzej. Congestion Control in Data Transmission Networks : sliding mode and other designs. Springer science and Business Media, 2012.

- [14] Tomàs SALGADO JIMENEZ. Contribution à la commande d'un robot sous-marin Autonome de type torpille, Thèse de doctorat. Université Mont pellier II, 2004.
- [15] Khalil, H. K. Performances recovery under output feedback sampled-data stabilization of a class of non linear systems. IEEE Transactions on Automatic Control, 2004.
- [16] Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing company, Reading, MA, 1989.
- [17] HARROUCHE Fateh. Application de la logique floue aux machines tournantes, Université Ferhat Abbas-Sétif, 2019.
- [18] BENDIB Riad, El-ARKAM Mechhoud, KASSAH Maroua, KATTOUCHE Wiam, ZENNIR Yousef. Design of type 2 Fuzzy Logic Controller for FESTO Process Workstation. Engineering Proceeding (Vol 29 ,No 6), 2023.
- [19] Jinkun Liu . Intelligent Control Design ans MATLAB Simulation ,Published with Beihang University Press, Beijing,2018. P(49-56).
- [20] Amieur Toufik. Commande des Systèmes Non linéaires par mode glissant floue. Université Mohamed Kheider -Beskra, 2009. P(57.70).
- [21] Jinkun Liu, Xinhua Wang . Advanced Sliding Mode Control For Mechanical systems. National University Of Singapore, 2012. P(234-236).
- [22] C.Edward, S.K.Spurgeon. Sliding mode control: Theory and applications, CRC Press. 1998.
- [23] C.Vecchio. Sliding mode control: theoretical developement and applications to Uncertain mechanical systems, Universetà degli studi di pavia. 2008.