



People's Democratic Republic of Algeria



Ministry of Higher Education and Scientific Research

University 20 out 1955 –SKIKDA-

Faculty of Technology

Department of Petrochemical

Final thesis: For Obtaining the Master Degree

Option : Automation And Control In Petrochemical Industry

Title :

**Optimization Of Control Laws Using
Artificial Intelligence Techniques**

❖ **Submitted by:**

- Rechache Chaima
- Belkolli Manar

❖ **Membre of the jury :**

SUPERVISOR	PRESIDENT	EXAMINER
Dr.R.BENDIB	Pr.L.ZIGHED	Dr.H.METATLA

University year :2023-2024

Thanks

It is frequently stated that the trip is as essential as the goal. During these years of mastery, he was able to comprehend the meaning of this seemingly easy words. In truth, this voyage was not without difficulties and raised many issues. Her responses will take many hours to complete. We would want to thank God Almighty for giving us trust at the completion of our job. And for allowing us to get here, despite all the sacrifices we've made and all the challenges we've faced over the years. We would like to express our heartfelt gratitude to **Dr.R.bendib**, and especially **Dr.Djari Hamid**, I would like to express my sincere gratitude and appreciation to my esteemed professor, for his continuous support and valuable guidance throughout the preparation of this thesis. Thank you very much, and I wish you continued success and prosperity.for his unwavering support, vast experience, and essential help. We would also want to thank all of the professors who contributed to our education during our studies, particularly all of my teachers at the Department of Petrochemistry, for their tremendous efforts, which enabled us to finish this task. We also express our gratitude to the jury members, who have honored us by accepting and judging our work, as well as adding to their observations and scientific critique. Finally, we'd want to thank everyone who helped us finish this project, whether directly or indirectly

Dedication

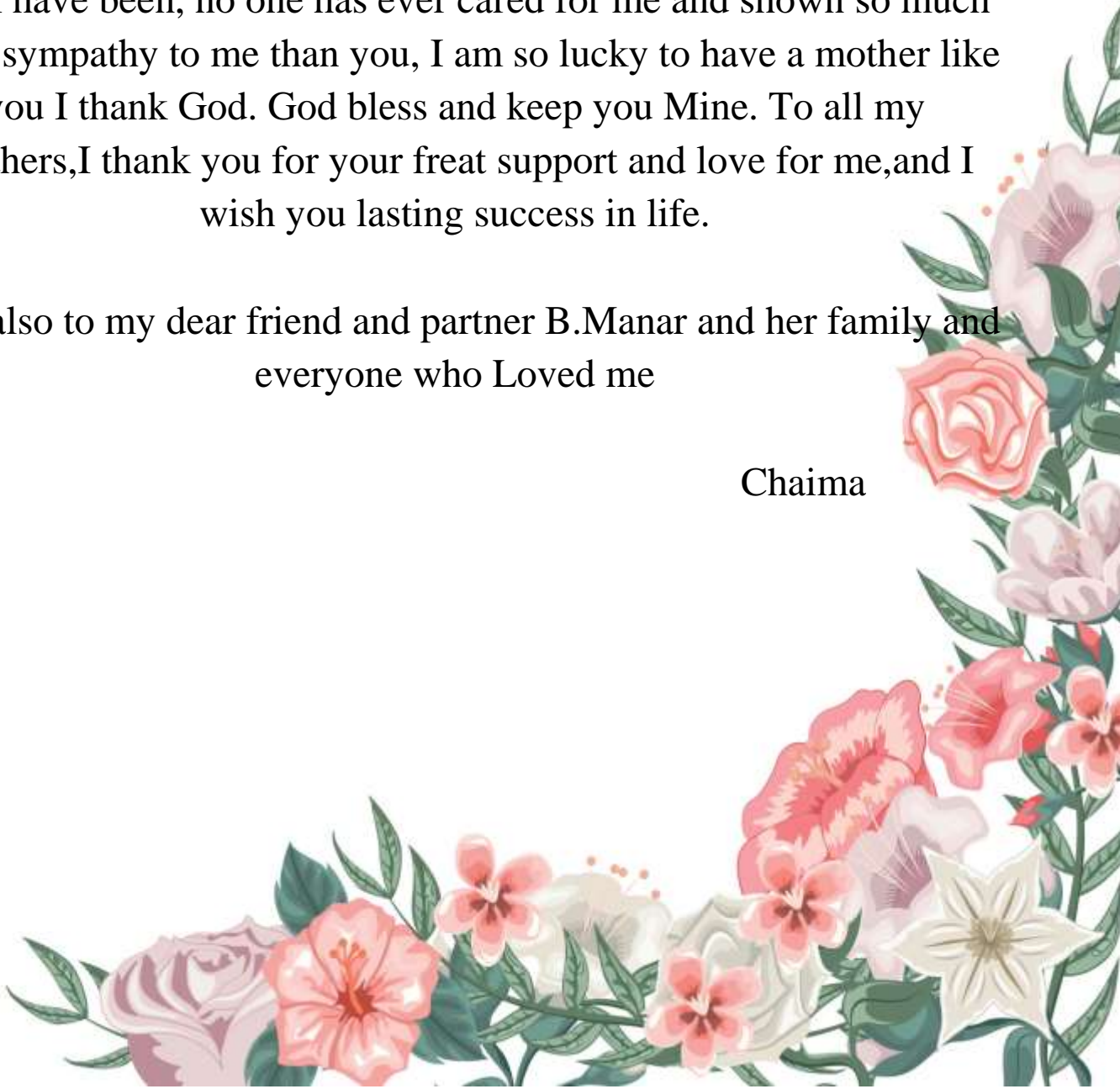
To my dear parents

I owe you what I am today thanks to your love, your patience and your countless sacrifices. May this modest work be for you a small compensation and gratitude for the incredible things you have done for me. May God, the Almighty, preserve you and grant you health and long life. so that I can in turn fill you.

And especially to the most important person in life, my dear mother, I thank you for your support, patience and sacrifices in all that I have been, no one has ever cared for me and shown so much more sympathy to me than you, I am so lucky to have a mother like you I thank God. God bless and keep you Mine. To all my brothers,I thank you for your great support and love for me,and I wish you lasting success in life.

And also to my dear friend and partner B.Manar and her family and everyone who Loved me

Chaima



Dedication

To my parents

and especially to the most important person in life, my dear mother, I thank you for your support, patience and sacrifices in all that I have been, no one has ever cared for me and shown so much more sympathy to me than you, I am so lucky to have a mother like you I thank God. God bless and keep you Mine.

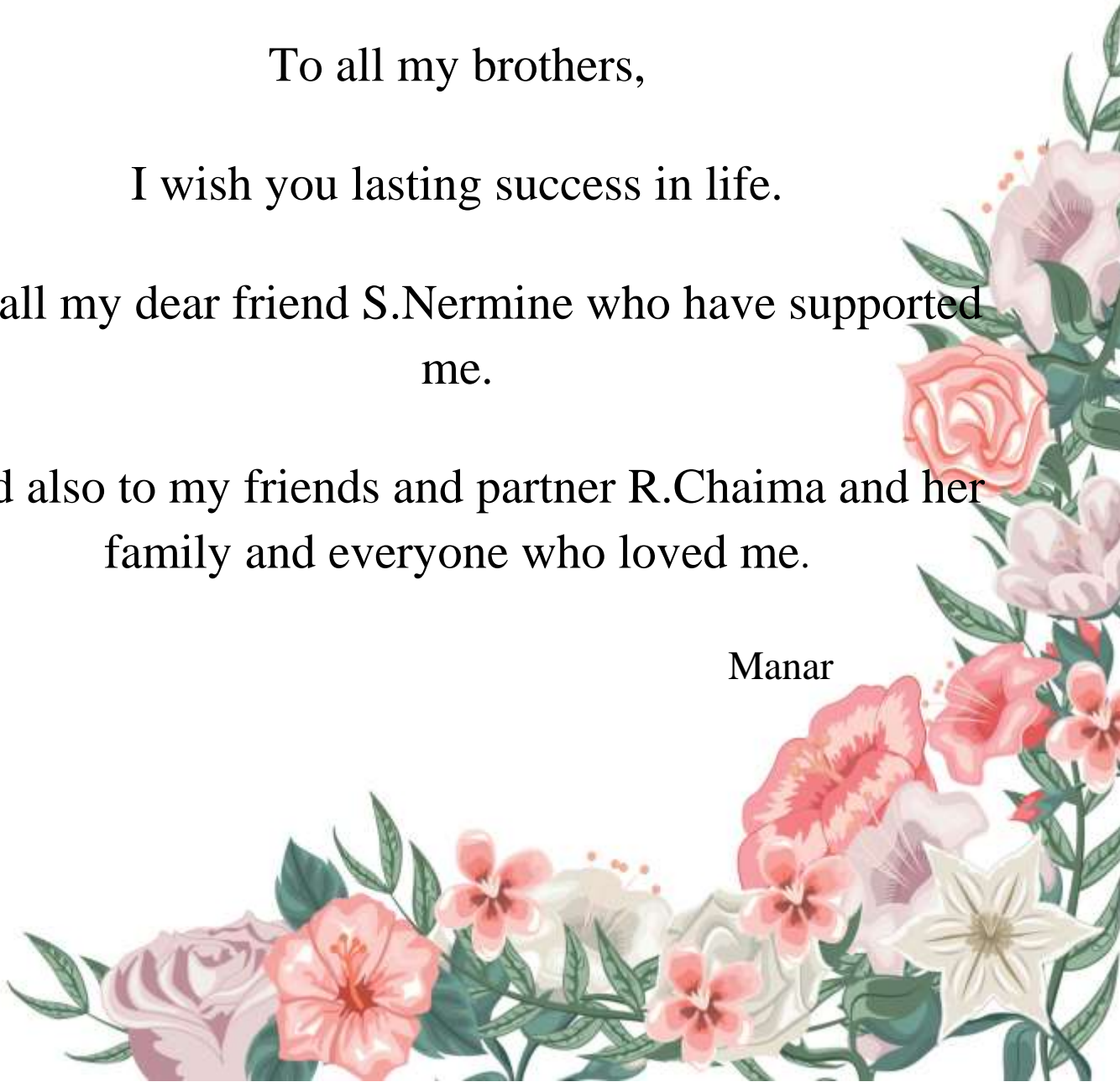
To all my brothers,

I wish you lasting success in life.

To all my dear friend S.Nermine who have supported me.

And also to my friends and partner R.Chaima and her family and everyone who loved me.

Manar



ملخص

تعتبر الأتمتة الصناعية مجالاً يتطلب دقة عالية في التحكم لضمان كفاءة وفعالية العمليات. تقنية التحكم بالوضع المنزلق (SMC) هي إحدى التقنيات الفعالة في هذا السياق، لكنها تحتاج إلى تحسين مستمر لمواكبة متطلبات الأنظمة الحديثة. يمكن استخدام تقنيات الذكاء الاصطناعي مثل خوارزميات تحسين الجسيمات (PSO) والخوارزميات الجينية (GA) لتعزيز أداء SMC. تتيح هذه التقنيات استكشاف مساحات البحث الواسعة وإيجاد الحلول المثلى، مما يسهم في تحسين الكفاءة والدقة في الأنظمة الصناعية. تهدف الدراسة إلى التركيز على استخدام هذه الخوارزميات لتعزيز أداء SMC، مما يوفر تحسينات كبيرة ويقدم رؤى حول أفضل التطبيقات لكل تقنية.

الكلمات المفتاحية: تحسين، فيستوم ب س ب ا، تحسين سرب الجسيمات، خوارزمية الجينية، ميتاهوريستيك. التحكم بالوضع المنزلق، محاكاة، ماتلاب

Resumé

L'automatisation industrielle exige une précision de contrôle élevée pour assurer l'efficacité et l'efficience des processus. La technique de contrôle par mode glissant (SMC) est efficace dans ce contexte, mais elle nécessite des améliorations continues pour répondre aux exigences des systèmes modernes. Les techniques d'intelligence artificielle, telles que les algorithmes d'optimisation par essaim de particules (PSO) et les algorithmes génétiques (GA), peuvent être utilisées pour améliorer les performances du SMC. Ces techniques permettent d'explorer de vastes espaces de recherche et de trouver des solutions optimales, contribuant ainsi à améliorer l'efficacité et la précision des systèmes industriels. L'étude se concentre sur l'utilisation de ces algorithmes pour renforcer les performances du SMC, offrant ainsi des améliorations significatives et des perspectives sur les meilleures applications pour chaque technique.

Les mots clés : Optimisation, Festo MPS PA , PSO ,GA , Métaheuristique , Régulateur de mode glissant, Simulation , MATLAB.

Abstract

Industrial automation requires high control precision to ensure the efficiency and effectiveness of processes. The sliding mode control (SMC) technique is effective in this context, but it needs continuous improvement to meet the demands of modern systems. Artificial intelligence techniques, such as particle swarm optimization (PSO) and genetic algorithms (GA), can be used to enhance the performance of SMC. These techniques allow for the exploration of vast search spaces and the finding of optimal solutions, thereby contributing to improved efficiency and accuracy in industrial systems. The study focuses on the use of these algorithms to enhance SMC performance, providing significant improvements and insights into the best applications for each technique.

Key words: Optimization, Festo MPS PA, PSO, GA, Metaheuristic , Sliding mode control , Simulation , MATLAB.

List of abbreviation

MHOA: Meta-heuristic Optimization Algorithms

GA: Genetic Algorithm

PSO: Particle Swarm Optimization

BCO: Bee Colony Optimization

BFOA: Bacterial Foraging Optimization Algorithm

ACO: Ant Colony Optimization

SA: Simulated Annealing

HS: Harmony Search

DE: Differential Evolution

MH: Meta-heuristic

ABC: Artificial Bee Colony

BA: Bat Algorithm

BMO: Bird Mating Optimizer

FA: Firefly Algorithm

GWO: Grey Wolf Optimization

SMC: Sliding Mode Control

CM: Convergence Mode

SMS: Sliding Mode surface

SSM: Steady State Mode

DC: Direct Current

Festo MPS PA: Festo Modular Production System for Process Automation

PLC: Programmable Logic Controller

HMI: Human-Machine Interface

I/O: Input/Output

AI:Artificial intelligence

Lists of figures

Figure I-1 : The particle swarm optimization	11
Figure I-2 : Initialization of genetic algorithm	13
Figure I-3 :The crossover process (Recombination)	14
Figure I-4 : The mutation process	15
Figure I-5 :The pack hierarchy	16
Figure I-6:Prey tracing map	17
Figure II-1:The different modes of operation	22
Figure II-2:the different modes for trajectories in a phase plane	22
Figure II-3: trajectory of the state with respect to the sliding surface	24
Figure II-4:Ideal sliding regime	26
Figure II-5 :chattering phenomenon.....	28
Figure IV.1 : sign function.....	37
Figure IV.2 :Simulation of The SMC of Festo MPS	38
Figure IV.3:Results of optimization by PSO with sat and sign function.....	40
Figure IV.4 :Results of optimization by GA with sat and sign function.....	42

List of Tables

Table I.1 :Comparison between three techniques.....	20
Table III.1 .Technical data of festo MPS PA.....	34
Table IV.1 .The general performances of system optimized by PSO and GA algorithms....	44

SUMMARY:

<i>General introduction</i>	<i>1</i>
-----------------------------------	----------

Chapter I : Metaheuristic

I.1.Introduction	4
I.2.Metaheuristic Algorithms	4
I.2.1.Definition:	4
I.2.2.Characteristics of metaheuristics.....	5
I.2.3.Application of metaheuristic algorithm:	5
A) Engineering:	5
B) Operations Research:	6
C) Computer Science	6
E) Healthcare:	6
F) Energy:	6
G) Telecommunications:	6
H) Biology and Bioinformatics:	7
I) Manufacturing:.....	7
J) Transportation:.....	7
I.2.4 Advantages and Disadvantages of metaheuristics	7
I.2.5.Metaheuristic techniques.....	8
A) ant colony optimization (ACO):	8
B) artificial bee colony (ABC):	8
D) Bat algorithm (BA) :.....	9
E) Bird mating optimizer (BMO):	9
F) The firefly algorithm (FA):	9
I.2.5.particle swarm optimization (PSO).....	10
I.2.5.1.Definition:	10
I.2. 5.2.Mathematical model of PSO	11
I.2.5.3.Advantages and Disadvantages of PSO	12
I.2.6.Genetic Algorithm(GA)	13
I.2.6.1. Definition:	13
I.2.6.2.Work of genetic algorithm.....	13
A) Initialization.....	13
B) Fitness Assignment.....	13
C) Selection:	14
E) Termination:.....	15
I.2.6.3.Advantages and disadvantages of GA.....	15
I.2.7.Grey wolf optimization	15
I.2.7.1.Definition:	15
I.2.7.2.Mathematical model of GWO	16
I.2.7.3.Avantages and disadvantages	18
I.3.Comparaison between three technique :	18
I.4.Conclusion :	20

Chapter II : Sliding mode control

II.1.Introduction	21
II.2.Definition:	21
II.3.Modes of Operation.....	21
II.4.Principle of sliding mode control	22
A. Convergence mode (CM):	22
B. Sliding mode (SM):.....	22
C. Steady state mode (SSM):.....	22
II.5.Design of Sliding mode control	23
II.5.1.Choice of sliding surface	23
II.5.2.Convergence conditions.....	24
II.5.2.1.Direct switching function	24
II.5.2.2.Function of LYAPPUNOV	24
II.5.3.The law of control.....	25
II.6.Classic formalism of the sliding regime.....	25
II.6.1. Ideal sliding regime	25
II.6.2.Real sliding regime	26
II.7.Applications of Sliding Mode Control (SMC).....	26
A)Robotics:.....	26
B) Automotive Systems	26
C) Aerospace	27
D) Power Systems.....	27
E) Renewable Energy Systems.....	27
F) Process Control	27
II.8.Chattering phenomenon	27
II.9.Eliminate the chattering phenomenon	28
II.11.Conclusion.....	29

Chapter III:Description of the process

III.1.Introduction	30
II.2.Description of festo MPS PA	30
III.3.Components of festo MPS PA	30
III.3.1.Hardware components.....	30
A) Base Unit	30
B) Pneumatic Components	30
III.3.1.1.Sensors	30
A) Pressure Sensors	30
B) Flow Sensors.....	30
C) Level Sensors.....	31
D) Temperature Sensors	31
E) Proximity Sensors	31
III.3.1.2.Actuators	31
A) Pneumatic Actuators.....	31
B) Electric Actuators	31
C) Valves	31
III.3.1.3.Control Systems.....	31
A)Programmable Logic Controllers (PLCs).....	31
B) Human-Machine Interfaces (HMIs):	31

C) Communication Modules:	31
D) I/O Modules:.....	31
III.3.2. Software Components :	31
A) Power Supply Units:.....	31
III.3.3. Safety Components:.....	32
III.4.Applications:	32
III.4.1. Educational :	32
A) Training Technical Education:	32
B) Hands-on Learning:	32
C) Curriculum Integration:	32
III.4.2. Industrial Training :.....	32
A) Workforce Development:	32
B) Skill Enhancement:	32
C) Safety Training:	32
III.4.3. Research and Development :	32
A) Process Optimization:.....	32
B) Innovation Testing:	33
III.4.4. Industrial Simulation:	33
A) Process Simulation:	33
B) System Integration Testing:	33
III.4.5.Product Development :	33
A) System Testing:	33
B) Quality Assurance:.....	33
III.4.6. Industry-Specific Applications :.....	33
A) Chemical and Pharmaceutical Industries:	33
B) Food and Beverage Industry:	33
C) Automotive Industry:.....	33
III.5.Technical Data	33
III.6.Conclusion :	34

Chapter IV : Simulation and results

IV.1.introduction	36
IV.2.Application	36
IV.2.1.Transfer function and state space representation	36
A) Determine the equivalent control U_{eq}	37
IV.3.Simulation of The SMC of Festo MPS PA.....	38
IV.3.1.Simulation diagrams	38
IV.3.2.Optimization by PSO	39
IV.3.3.Optimization by GA	44
IV.3.4.Discussion	43
A) Comparaision between sign and sat functions.....	43
B)Comparaision between PSO and GA algorithms.....	43
IV.4.Conclusion	44
General conclusion.....	46
References.....	47

General introduction

In the world of automation and industrial control, artificial intelligence techniques play a crucial role in enhancing the performance and efficiency of control systems and The optimization process plays a crucial role in solving a wide range of complex problems in applied mathematics and engineering sciences. It involves finding the best possible solution from a set of alternatives while considering the problem's constraints and objectives. Optimization can aim to maximize profits or quality and minimize time, cost, or risk. Traditional optimization methods often face limitations such as sensitivity to initial estimates and a tendency to converge on local optima. Additionally, as the dimensionality of the optimization problem increases, classical methods struggle to find the global optimum due to the exponential growth of the search space. To address these challenges, metaheuristic algorithms have emerged as effective solutions. This thesis, titled "**Optimization of control laws using the artificial intelligence**" aims to explore and improve the precise performance of control systems using advanced artificial intelligence methods. The research focuses on applying optimization algorithms such as Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), and Genetic Algorithm (GA) to neural network structures and sliding mode control systems. The main objective is to study how to effectively improve control laws using these advanced techniques, enhancing the efficiency and accuracy of control systems in complex dynamic environments.

The thesis is structured into four chapters:

The first chapter : provides a detailed explanation of the various metaheuristic techniques used in optimization. It includes Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), and Genetic Algorithm (GA). The theoretical basis of each algorithm is discussed, along with how they work and examples of their applications in different fields.

The second chapter: This chapter focuses on explaining Sliding Mode Control (SMC) as a powerful control method used to handle nonlinear dynamic systems. The basic principles of this technique are explained, along with how to design control laws using it. Examples of SMC applications in various systems are provided,

The third chapter :describes the Festo MPS PA system as an application model for the study. It includes an explanation of the system's components, functions, and how it is used in simulations and practical experiments.

and **the fourth chapter** :presents the simulation results obtained using the optimized control techniques developed in the previous chapters. A detailed analysis of the results is provided, with a comparison of performance before and after applying the improvements using artificial intelligence techniques. Various performance indicators are used to evaluate the effectiveness and efficiency of the new control laws, along with a discussion of the results and recommendations for further improving control system performance in the future..

This work aims to enrich the understanding of how to improve control techniques using modern methods, enabling engineers and researchers to implement innovative solutions in industrial automation. This research contributes to achieving higher and better performance in complex dynamic systems by demonstrating how advanced artificial intelligence techniques can enhance the accuracy and efficiency of control systems in industrial environments.

Chapter I

Metaheuristic

I.1.Introduction

The term metaheuristic describes higher level heuristics that are proposed for the solution a wide range of optimization problems.

Recently, many metaheuristics algorithms are successfully being applied for solving intractable problems. The appeal of using these algorithms for solving complex problems is that they obtain the best optimal solutions even for very large problem sizes in small amounts of time. The optimization problems that attracted the attention of metaheuristic approaches have a large variance, ranging from single to multiobjective, continuous to discrete, constrained to unconstrained. Solving these problems is not a straightforward task due to their complex behavior. Exact algorithms are mostly non-polynomial and, although providing best solutions usually have impractical execution times and/or computational requirements for large data sizes. Metaheuristic algorithms provide a practical and elegant solution to many such problems and are designed to achieve approximate optimal solutions in practical execution times for NP-Hard optimization problems (Neumann & Witt, 2010).

I.2.Metaheuristic Algorithms

I.2.1.Definition:

Metaheuristic is well known as an efficient method for hard optimization problems, that is, the problems that cannot be solved optimally using deterministic approach within a reasonable time limit. Metaheuristic methods work for three main purposes: for fast solving of problem, for solving large problems, and for making a more robust algorithm. These methods are also simple to design as well as flexible and easy to implement. In general, metaheuristic algorithms use the combination of rules and randomization to duplicate the phenomena of nature. The biological system imitations of metaheuristic algorithm, for instance, are evolution strategy, GA, and DE. Phenomena of ethology for examples are particle swarm optimization (PSO), bee colony optimization (BCO), bacterial foraging optimization algorithms (BFOA), and ant colony optimization (ACO). Phenomena of physics are SA, microcanonical annealing, and threshold accepting method. Another form of metaheuristic is inspired by music phenomena, such as HS algorithm.

A metaheuristic (MH) can be described as:

$$\mathbf{MH} = (\mathbf{O}, \mathbf{A}, \mathbf{Rc}, \mathbf{Ri}, \mathbf{Ro})$$

where, \mathbf{O} is a set of metaheuristic methodologies (i.e. metaheuristic, adaptive, automotive, trial-and-error, cognitive, etc.); \mathbf{A} is a set of generic algorithms (e.g., genetic algorithm, particle swarm optimization, evolutionary algorithm, ant colony optimization, etc.); $\mathbf{Rc} = \mathbf{O} \times \mathbf{A}$, is a set of internal relations; $\mathbf{Ri} \subseteq \mathbf{A}' \times \mathbf{A}$, $\mathbf{A}' \wedge \mathbf{A}$, is a set of input relations; where \mathbf{C}' is a set of external concepts and \mathbf{c} is concept environment. For convenience, $\mathbf{Ri} = \mathbf{A}' \times \mathbf{A}$ may be simply denoted as $\mathbf{Ri} = \mathbf{C}' \times \mathbf{c}$. $\mathbf{Ro} \subseteq \mathbf{c} \times \mathbf{C}'$ is a set of output relations

I.2.2.Characteristics of metaheuristics

These are properties that characterize most metaheuristics:

- Metaheuristics are strategies that guide the search process.
- The goal is to efficiently explore the search space in order to find near-optimal solutions.
- Techniques which constitute metaheuristic algorithms range from simple local search procedures to complex learning processes.
- Metaheuristic algorithms are approximate and usually non-deterministic.
- Metaheuristics are not problem-specific.
- Metaheuristics sample a subset of solutions which is otherwise too large to be completely enumerated or otherwise explored.
- Metaheuristics may make relatively few assumptions about the optimization problem being solved and so may be usable for a variety of problems.
- Many metaheuristics implement some form of stochastic optimization, so that the solution found is dependent on the set of random variables generated in combinatorial optimization, by searching over a large set of feasible solutions, metaheuristics can often find good solutions with less computational effort than optimization algorithms, iterative methods, or simple heuristics. As such, they are useful approaches for optimization problems.

I.2.3.Application of metaheuristic algorithm:

Metaheuristic algorithms have a broad range of applications in various fields due to their ability to solve complex optimization problems.

Here are some specific applications across different domains:

A) Engineering:

Control Systems:

Optimizing the parameters of controllers like PID, SMC, and adaptive controllers to improve performance, robustness, and stability.

Design Optimization:

Structural optimization, aerodynamic design, and thermal system design where multiple conflicting objectives must be balanced.

B) Operations Research:**Scheduling:**

Solving complex scheduling problems in manufacturing, transportation, and logistics.

Routing:

Finding optimal routes for vehicles in logistics and transportation networks (e.g., the Vehicle Routing Problem).

C) Computer Science:**Software Engineering:**

Automatic code optimization, test case generation, and software project scheduling.

Machine Learning:

Feature selection, hyperparameter tuning, and optimization of neural network architectures.

D) Finance:**Portfolio Optimization:**

Selecting the best mix of assets to maximize returns and minimize risk.

Risk Management:

Optimizing risk assessment models and strategies for financial markets.

E) Healthcare:

Medical Diagnosis: Optimizing parameters of diagnostic models for better accuracy.

Treatment Planning:

Personalized treatment plans and scheduling for patient care.

F) Energy:**Power Systems:**

Optimal power flow, load forecasting, and grid management.

Renewable Energy:

Placement and optimization of renewable energy sources like wind turbines and solar panels.

G) Telecommunications:**Network Design:**

Optimizing the layout and configuration of telecommunication networks for better performance.

Resource Allocation:

Efficient allocation of bandwidth and other resources in wireless networks.

H) Biology and Bioinformatics:**Protein Folding:**

Predicting the three-dimensional structure of proteins. Genetic Sequencing: Aligning sequences and finding motifs in genetic data.

I) Manufacturing:**Process Optimization:**

Improving the efficiency and quality of manufacturing processes.

Inventory Management:

Optimizing inventory levels and order schedules to reduce costs

J) Transportation:**Traffic Management:**

Optimizing traffic signal timings and route planning to reduce congestion.

Airport Operations:

Scheduling flights, gates, and ground services for efficiency.

I.2.4 Advantages and Disadvantages of metaheuristics

Metaheuristic algorithms are a family of general-purpose approximate algorithms. They are usually iterative procedures that guide a subordinate search heuristic, intelligently combining different concepts to properly explore and exploit the search space. Some advantages and disadvantages are as follows:

❖ Advantages

- General-purpose algorithms .
- Great success in practice.
- Easily deployable .
- Easily parallelizable.

❖ Disadvantages

- They are approximate algorithms, not exact .
- They are non-deterministic (probabilistic) The factors that may be interesting for the use of metaheuristic algorithms are the following.
- When there is no exact method of resolution, or it requires a lot of computation time and memory (inefficient).

- When the optimal solution is not needed, and a good quality one is enough.

I.2.5. Metaheuristic techniques

A) ant colony optimization (ACO):

The search agents are called as ants that mimic the foraging behavior of the real ants. This comes under swarm intelligence. A colony of ants is maintained and considered to be the population in this algorithm. Each ant searches food near its nest in a random manner. It evaluates the quality and quantity of the food that is found. Based on this evaluation, it deposits pheromone when it returns to its nest with the food. This pheromone trails provides indirect communication with other ants to find the shortest path between its nest and food source. The value associated with each path is taken in combination to evaluate the objective function. Subsequently, the optimal value is found after required iterations with newly formed ant colonies. A different objective function called square error percentage is used. An ACO algorithm is used for feature subset selection which helps in removing the input attributes that produces noise. A variant of ACO for continuous optimization is incorporated with classical gradient based methods to train.

B) artificial bee colony (ABC):

Is a metaheuristic algorithm proposed by Karaboga in 2005. It is one of the most cited new generation metaheuristics in literature (Basturk, 2006; Karaboga, 2005; Karaboga, Gorkemli, Ozturk, & Karaboga, 2014). The artificial bee colony (ABC) algorithm is developed based on the intelligent foraging behavior of the natural bee colony.

The population of ABC consists of three types of artificial bees, namely the employed bees which occurs half of the bee population that share the information of their food source at the dancing area by a waggle dance, second is the onlooker bees that wait in the hive for the food source information from employed bees and third one is the scout which search for the new food source. Each employed bees are associated with one food source and it becomes scout bee once its food source gets exhausted. The high quality food source selected by artificial bees represents the good optimized solution to the network.

C) The bacterial foraging optimization algorithm (BFOA):

Describes the chemotactic behavior of the *Escherichia coli* bacteria that live in our intestines. The group of bacteria forms the population and each bacterium acts as the search agent. This foraging process is dictated by its control system using the steps such as chemotaxis, swarming, reproduction, elimination and disposal. The member of a bacterium which gives an optimal error during training is considered as optimum solution. The position of the bacterium

with highest fitness is considered as the set of optimum values and discussed in terms of accuracy.

D) Bat algorithm (BA) :

BA metaheuristic is first proposed by Yang (2010c). Bats use echolocation (i.e., a type of sonar) to avoid obstacles, detect prey, and locate their nests in the dark.

A bat emits a sound and follows the echo that reflects from the objects in the environment. Bats can also detect the difference between food/prey and barriers by using echolocation. The motivation of BA is that this echolocation talent of bats can be formalized as a means to find an optimal solution in an objective function.

E) Bird mating optimizer (BMO):

The bird mating optimizer (BMO) is inspired by the mating behavior of the birds. The population of the birds is called society which consists of four kinds of birds namely monogamous, polygynous, polyandrous and promiscuous. Each bird is the search agent that has the predefined number of genes. The birds take part at the mating process during the breeding season to brood with superior genes. The quality of each brood is calculated using the objective function or fitness function which specifies the mating strategy. Based on the quality, the birds are sorted and divided into four categories. Then the mating process is carried out. The quality of the brood is compared with that of the bird and if better, the brood will replace the bird. The BMO is used to find the weights for training ANN. Each bird is a solution to the problem. The number of genes in a bird is based on the problem variables and decides the d-dimension in which breeding happens. The quality of bird depends on its gene and if better than its brood, it causes brood to be abandoned from society. If otherwise, the contrary is followed. This is repeated for maximum generations. Finally the brood or bird with the good quality of genes is kept as the optimal solution of weights for training ANN.

F) The firefly algorithm (FA):

Is proposed by Yang (2010a), The firefly optimization algorithm (FA) is based on the characteristics of flashing light of the fireflies produced by bioluminescence process to attract the mating partners and the potential prey. The swarm of fireflies constitutes the population where each firefly is the search agent. The light intensity decreases as the distance between the fireflies increases. The flashing light is associated with the objective function. The three idealized rules describing the FA is given as:

1. all fireflies are attracted towards each other as they are considered to be unisex.

2. attractiveness is proportional to their brightness and the movement of one firefly towards other in the d-dimensional space based on this.

3. the brightness of a firefly is affected by the landscape of the objective function.

The new solution is evaluated and the light intensity is updated at each time to find the attractiveness. The firefly that yields the best solution is considered to be the most optimal.

❖ There are also many other techniques, however In this chapter we will focus on three techniques:

particle swarm optimization (PSO), genetic algorithm (GA), and grey wolf optimization (GWO).

I.2.5. particle swarm optimization (PSO)

I.2.5.1. Definition:

The Particle Swarm Optimization (**PSO**) algorithm is a computational technique inspired by the collective behavior of natural organisms, such as birds or fish, that move together to achieve a common goal. In PSO, a group of particles (representing potential solutions) navigates through a problem's solution space to find the best possible solution. Each particle adjusts its position based on its own best-known solution (personal best) and the best solution discovered by the entire group (global best). This collaborative movement enables particles to converge toward optimal solutions over iterations. PSO is widely used for optimization problems in various fields, leveraging the power of collective intelligence to explore complex solution spaces and find optimal outcomes efficiently.

Particles: Each particle represents a potential solution.

Position: Represents the current state (setpoints or controller parameters).

Velocity: Adjusts the particle's movement through the solution space.

Personal Best (pBest): Best position achieved by the particle.

Global Best (gBest): Best position achieved by any particle in the swarm

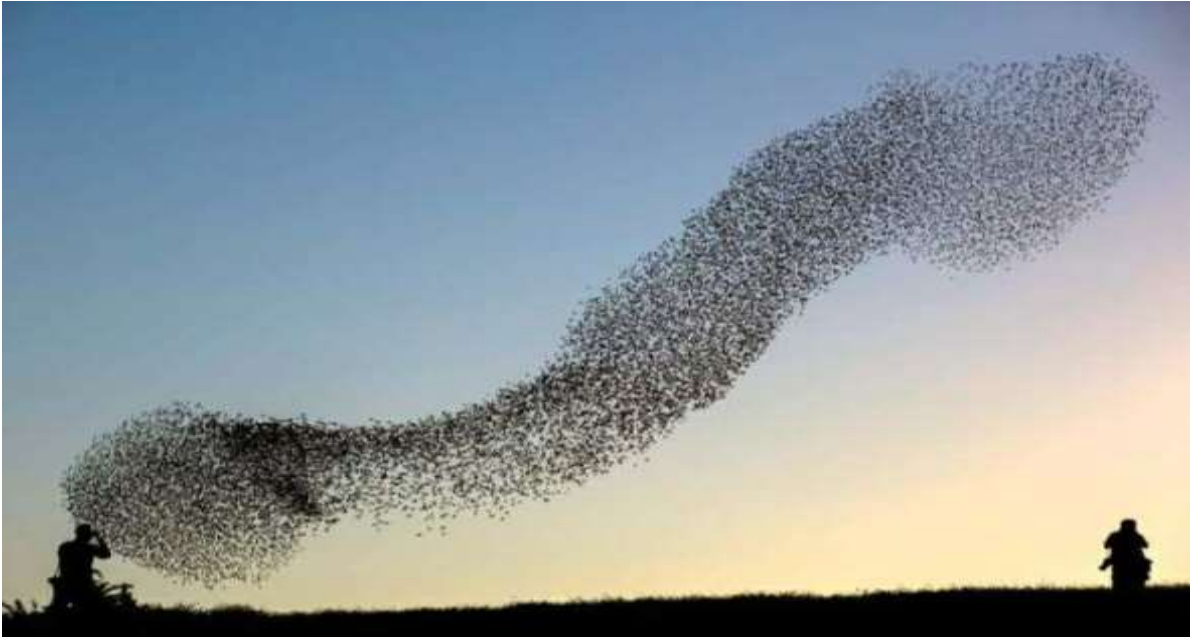


Figure I-1 : The particle swarm optimization

I.2.5.2. Mathematical model of PSO

Putting it in a mathematical framework, let $A \subset \mathbb{R}^n$, be the search space and $f: A \rightarrow Y \subseteq \mathbb{R}$ be the objective function. In order to keep descriptions as simple as possible, we assume that A is also the feasible space of the problem at hand, i.e. there are no further explicit constraints posed on the candidate solutions. Also, note that no additional assumptions are required regarding the form of the objective function and search space. As mentioned in the previous chapter, PSO is a population-based algorithm, i.e. it exploits a population of potential solutions to probe the search space concurrently. The population is called the swarm and its individuals are called the particles; a notation retained by nomenclature used for similar models in social sciences and particle physics.

The swarm is defined as a set:

$$S = \{x_1, x_2, x_3, \dots, x_N\}$$

of N particles (candidate solutions), defined as:

$$x = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iN})^T \in A, i = 1, 2, \dots, N.$$

Indices are arbitrarily assigned to particles, while N is a user-defined parameter of the algorithm. The objective function $f(x)$ is assumed to be available for all points in A . Thus, each particle has a unique function value $f = f(x) \in Y$.

The particles are assumed to move within the search space A iteratively. This is possible by adjusting their position using a proper position shift called velocity and denoted as:

$$v = (v_{i1}, v_{i2}, \dots, v_{iN})^T, i = 1, 2, \dots, N.$$

▪ Velocity is also adapted iteratively to render particles capable of potentially visiting any region of A. If t denotes the iteration counter, then the current position of the i -th particle and its velocity will be henceforth denoted as $x_i(t)$ and $v_i(t)$, respectively.

▪ Velocity is updated based on information obtained in previous steps of the algorithm. This is implemented in terms of a memory, where each particle can store the best position it has ever visited during its search. For this purpose, besides the swarm S which contains the current positions of the particles, PSO maintains also a memory set:

$$P = \{p_1, p_2, \dots, p_N\}.$$

which contains the best positions:

$$p = (p_{i1}, p_{i2}, \dots, p_{iN}) \quad T \in A, i = 1, 2, \dots, N.$$

ever visited by each particle, These positions are defined as:

$$p_i(t) = \arg \min f_i(t), \text{ where } t \text{ stands for the iteration counter.}$$

PSO is based on simulation models of social behavior. thus, an information exchange mechanism shall exist to allow particles to mutually communicate their experience. The algorithm approximates the global minimizer with the best position ever visited by all particles, Therefore, it is a reasonable choice to share this crucial information.

Let g be the index of the best position with the lowest function value in P at a given iteration t , i.e.

$$p_g(t) = \arg \min f(p_i(t)).$$

Then, the early version of PSO is defined by the following equations (Eberhart & Kennedy, 1995; Eberhart et al., 1996; Kennedy & Eberhart, 1995)

$$v_{ij}(t+1) = v_{ij}(t) + c_1 R_1 (p_{ij}(t) - x_{ij}(t)) + c_2 R_2 (p_{jg}(t) - x_{ij}(t))$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1)$$

I.2.5.3. Advantages and Disadvantages of PSO

❖ Advantages

- ✓ Insensitive to scaling of design variables.
- ✓ Derivative free.
- ✓ Very few algorithm parameters.
- ✓ Very efficient global search algorithm.
- ✓ Easily parallelized for concurrent processing.

❖ Disadvantages

- ✓ Slow convergence in the refined search stage (Weak local search ability).

I.2.6. Genetic Algorithm (GA)

I.2.6.1. Definition:

A genetic algorithm (GA) is an adaptive heuristic search algorithm inspired by "**Darwin's theory of evolution in Nature**". It is used to solve optimization problems it is one of the important algorithms as it helps solve complex problems that would take a long time to solve. The genetic algorithm is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions.

I.2.6.2. Work of genetic algorithm

The genetic algorithm works on the evolutionary generational cycle to generate high-quality solutions. These algorithms use different operations that either enhance or replace the population to give an improved fit solution. It basically involves five phases to solve the complex optimization problems, which are given as below:

A) Initialization:

The process of a genetic algorithm starts by generating the set of individuals, which is called population. Here each individual is the solution for the given problem. An individual contains or is characterized by a set of parameters called Genes. Genes are combined into a string and generate chromosomes, which is the solution to the problem. One of the most popular techniques for initialization is the use of random binary strings.

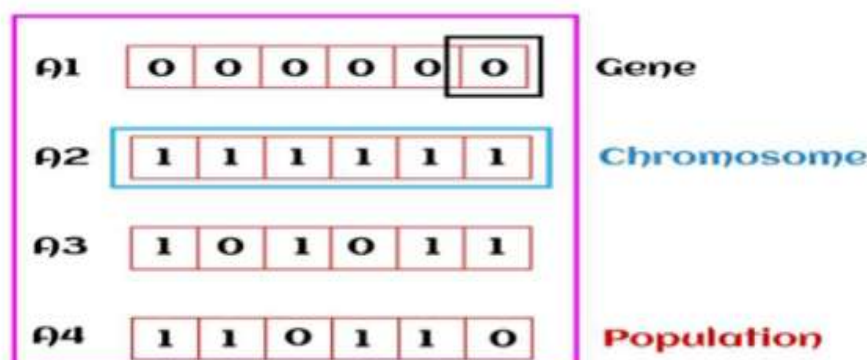


Figure I-2 : Initialization of genetic algorithm

B) Fitness Assignment

Fitness function is used to determine how fit an individual is. It means the ability of an individual to compete with other individuals. In every iteration, individuals are evaluated based on their fitness function.

The fitness function provides a fitness score to each individual, this score further determines the probability of being selected for reproduction. The high the fitness score, the more chances of getting selected for reproduction.

C) Selection:

The selection phase involves the selection of individuals for the reproduction of offspring. All the selected individuals are then arranged in a pair of two to increase reproduction, then these individuals transfer their genes to the next generation. There are three types of Selection methods available, which are:

.Roulette wheel selection. .Tournament selection .Rank-based selection.

D) Reproduction:

After the selection process, the creation of a child occurs in the reproduction step. In this step, the genetic algorithm uses two variation operators that are applied to the parent population, the two operators involved in the reproduction phase are given below:

▪ Crossover:

The crossover plays a most significant role in the reproduction phase of the genetic algorithm. In this process, a crossover point is selected at random within the genes. Then the crossover operator swaps genetic information of two parents from the current generation to produce a new individual representing the offspring. The genes of parents are exchanged among themselves until the crossover point is met. These newly generated offspring are added to the population. This process is also called Recombination or crossover. Types of crossover styles available:

.One point crossover. .Two-point crossover. .Livery crossover .Inheritable Algorithms crossover.

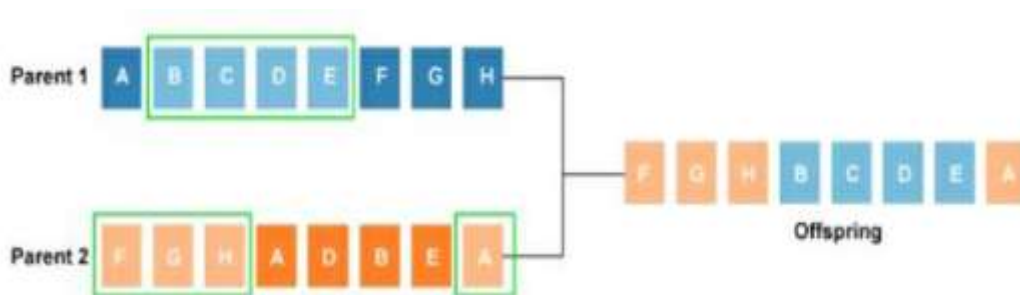


Figure I-3 :The crossover process (Recombination)

▪ Mutation :

The mutation operator inserts random genes in the offspring (**new child**) to maintain the diversity in the population. It can be done by flipping some bits in the chromosomes. Mutation helps in solving the issue of premature convergence and enhances diversification. The below image shows the mutation process:

- Types of mutation styles available:

.Flip bit mutation .

.Gaussian mutation.

Exchange/Swap mutation.



Figure I-4 : The mutation process

E) Termination:

After the reproduction phase, a stopping criterion is applied as a base for termination. The algorithm terminates after the threshold fitness solution is reached. It will identify the final solution as the best solution in the population.

I.2.6.3. Advantages and disadvantages of GA

❖ Advantages

- The parallel capabilities of genetic algorithms are best.
- It helps in optimizing various problems such as discrete functions, multi objective problems, and continuous functions.
- It provides a solution for a problem that improves over time. -A genetic algorithm does not need derivative information.

❖ Disadvantages

- Genetic algorithms are not efficient algorithms for solving simple problems.
- It does not guarantee the quality of the final solution to a problem.
- Repetitive calculation of fitness values may generate some computational challenges.

I.2.7. Grey wolf optimization

I.2.7.1. Definition:

GWO is one of the recently proposed swarm intelligence based algorithms, which is developed by Mirjalili et al in 2014. The GWO algorithm is inspired by grey wolves in nature that searching for the optimal way for hunting preys.

GWO algorithm applies the same mechanism in nature, where it follows the pack hierarchy for organizing the different roles in the wolves pack.



Figure I-5 :The pack hierarchy

In GWO, pack's members are divided into four groups based on the type of the wolf's role that help in advancing the hunting process. The four groups are alpha, beta, delta and omega, where :

- a) **Alpha α** wolf is considered the dominant wolf in the pack and his/her orders should be followed by the pack members.
- b) **Beta β** are subordinate wolves, which help the alpha in decision-making and are considered as the best candidate to be the alpha.
- c) **Delta δ** wolves have to submit to the alpha and beta, but they dominate the omega. There are different categories of delta-like Scouts, Sentinels, Elders, Hunters, Caretakers etc.
- d) **Omega ω** wolves are considered as the scapegoat in the pack, are we the least important individuals in the pack and are only allowed to eat at last. In addition to the social hierarchy of wolves, group hunting is another interesting social behavior of grey wolves.

According to Muro et al. the main phases of grey wolf hunting are as follows:

- Tracking, chasing, and approaching the prey.
- Pursuing, encircling, and harassing the prey until it stops moving.
- Attack towards the prey.

I.2.7.2. Mathematical model of GWO

Mathematical model During the GWO operation, the positions of wolf α , wolf β , and wolf δ are continuously updated at each iteration, whose mathematical model is described as:

$$\mathbf{D} = |\mathbf{C} \cdot \mathbf{X}_p(t) - \mathbf{X}(t)| \omega \quad (1)$$

$$\mathbf{X}(t + 1) = |\mathbf{X}_p(t) - \mathbf{A} \cdot \mathbf{D}| \quad (2)$$

Equation (1) is the distance between the grey wolf and the prey it is the number of current iterations.

$X_p(t)$ and $X(t)$ are the prey's locations and the grey wolf's location at t iterations, respectively. Equation (2) is the formula for updating the location of the grey wolf, A and C are the coefficient vectors, which are calculated by the following equation:

$$A = 2a \cdot r_1 - a \quad (3)$$

$$C = 2 \cdot r_2 \quad (4)$$

r_1, r_2 are random vectors between $[0, 1]$, and the primary role is to increase the randomness of the grey wolf movement. a represents the convergence factor, which will decay linearly from 2 to 0 as the algorithm progresses, and the linear relationship defines GWO :

$$a = 2 - 2t/T_{max} \quad (5)$$

Where t is the current number of iterations and T_{max} is the maximum number of iterations of the algorithm. Predating in abstract space and accurately identifying the location of prey is impossible. GWO simulated hunting behavior. Based on the fitness value, wolf α , wolf β , and wolf δ were selected to find the prey using the relationship between the three positions and guide the other wolves to move toward the prey, as in Figure (I.6).

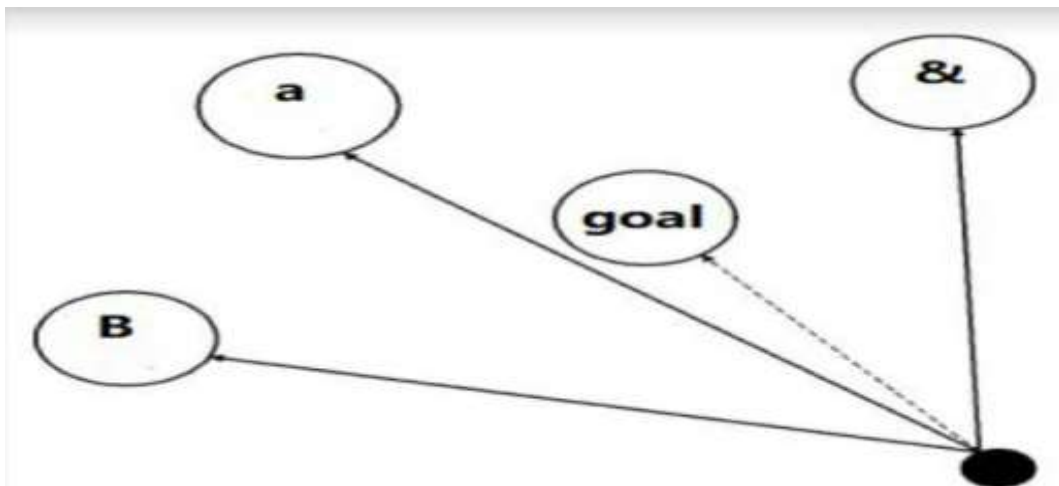


Figure I-6. Prey tracing map

By iterating several times until the location of the prey is reached, the mathematical model is as follows:

$$\begin{cases} D\alpha = |C_1 \cdot X\alpha - X| \\ D\beta = |C_2 \cdot X\beta - X| \\ D\delta = |C_3 \cdot X\delta - X| \end{cases} \quad (6)$$

$$\begin{cases} X_1 = |X\alpha - A_1 \cdot D\alpha| \\ X_2 = |X\beta - A_2 \cdot D\beta| \\ X_3 = |X\delta - A_3 \cdot D\delta| \end{cases} \quad (7)$$

$$X(t+1) = (X_1 + X_2 + X_3) / 3 \quad (8)$$

Where:

D_{α} is the distance between wolf pack w and a wolf α .

D_{β} is the distance between wolf pack w and β wolf.

D_{δ} is the distance between wolf pack w and wolf δ .

The Equation (7) presents the location of the new generation of wolves after the update.

I.2.7.3. Advantages and disadvantages

❖ Advantages

- fewer parameters.
- simple principles.
- implementing easily.

❖ Disadvantages

- slow convergence speed.
- low solution accuracy.
- easy to fall into the local optimum.

I.3. Comparaison between three technique :

comparison of Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Grey Wolf Optimizer (GWO) presented in a table format:

Feature	Partical swarm optimization (PSO)	Genetic algorithm (GA)	Grey wolf optimization (GWO)
Inspiration	Social behavior of birds flocking or fish schooling.	Natural selection and genetics.	Social hierarchy and hunting behavior of grey wolves.
Population	Swarm (particles).	Population (individuals).	Pack (wolves).
Search mechanism	Particles adjust positions based on personal and global best experiences.	Individuals evolve through selection, crossover, and mutation.	Wolves adjust positions based on alpha, beta, and delta wolves.
Exploration vs Exploitation	Balances both through velocity and	Balances both through genetic	Balances both through social

	position updates.	operators.	hierarchy and position updates.
Convergence speed	Generally fast but can get stuck in local minima.	Moderate; depends on genetic diversity	Fast and efficient in avoiding local minima
Parameter tuning	Requires setting of inertia weight, cognitive, and social coefficients	Requires setting of population size, crossover rate, and mutation rate	Less parameter tuning needed, primarily the number of wolves and coefficients
Diversity maintenance	Moderate, can converge prematurely	High, maintains diversity through crossover and mutation	Good, maintains diversity through dynamic social hierarchy
Complexity	Simple to implement and understand.	More complex due to multiple genetic operators.	Simple and intuitive based on social behavior.
Memory usage	Uses memory to store best solutions	No memory, relies on current population	Uses memory to store positions of alpha, beta, and delta wolves
Applications	Widely used in continuous optimization problems	Broadly applicable in discrete and continuous problems	Effective in continuous and discrete optimization, especially with constrained problems
Advantages	Easy to implement, few parameters, good convergence	Flexible, maintains genetic diversity, suitable for various problems	Simple structure, few parameters, good balance between exploration and exploitation
Disadvantages	Can get stuck in local optima, sensitive to parameter settings	Can be computationally expensive, requires	May require fine-tuning for specific problems, can

		careful parameter tuning	converge prematurely
--	--	-----------------------------	----------------------

Table I.1 :Comparison between three techniques

I.4.Conclusion :

In conclusion, metaheuristic algorithms provide a versatile and powerful approach to solving complex optimization problems that are beyond the reach of traditional methods due to their high computational cost and complexity. This chapter will delve into the core principles and methodologies of these algorithms, with a focus on Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), highlighting their effectiveness in efficiently exploring large and intricate search spaces.

Chapter II :Sliding mode control

II.1. Introduction

In the formulation of any practical control problem, there will always be a discrepancy between the actual plant and its mathematical model used for the controller design. These discrepancies (or **mismatches**) arise from unknown external disturbances, plant parameters, and parasitic/unmodeled dynamics. Designing control laws that provide the desired performance to the closed-loop system in the presence of these disturbances/uncertainties is a very challenging task for a control engineer. This has led to intense interest in the development of the so-called robust control methods which are supposed to solve this problem. One particular approach to robust controller design is the so-called sliding mode control technique. Control systems play a critical role in ensuring the efficient and safe operation of many devices, machines, and processes. One of the most robust and effective control techniques that has gained popularity in recent years is sliding mode control (SMC). The objective of this chapter is to provide an in-depth introduction to sliding mode, by explaining its fundamental principles, mechanisms and applications in the field of control.

II.2. Definition:

Sliding Mode Control Sliding mode control (SMC) is a feedback control technique that uses a discontinuous control law to drive the system state to a sliding surface, where the dynamics of the system are simplified. The control law is designed to maintain the system state on the sliding surface, which is typically defined as a hyperplane in the state space. The sliding surface is defined by a switching function, which is a continuous function that changes sign when the system state crosses the surface. The control law is discontinuous because it switches between two modes of operation depending on the sign of the switching function.

II.3. Modes of Operation

The two modes of operation are **the sliding mode** and **the reaching mode**. In sliding mode, the control law is designed to keep the system state on the sliding surface. This is achieved by applying a high control effort that drives the system towards the sliding surface. In the reaching mode, the control law is designed to bring the system state to the sliding surface in a finite time. This is achieved by applying a lower control effort that drives the system towards the sliding surface but does not keep it on the surface. The goal of SMC is to design the switching function and the control law such that the system state remains on the sliding surface for all time. This ensures that the system is robust to uncertainties, disturbances, and nonlinearities that may affect the system dynamics.

II.4. Principle of sliding mode control

A. Convergence mode (CM):

This mode involves moving the variable to be adjusted from any initial point in the phase plane, with a tendency to reach the switching surface $S(x)=0$. He characterized by a specific control law and a convergence criterion.

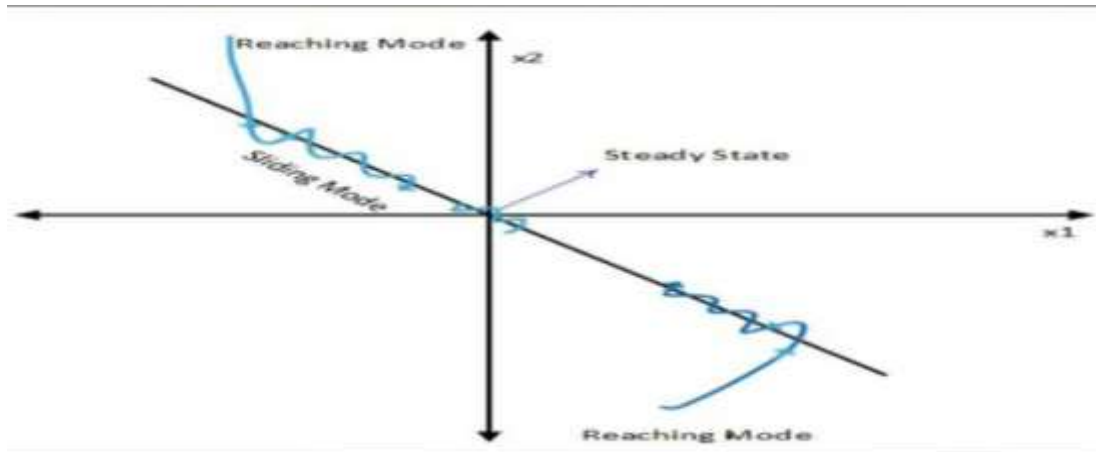


Figure II-1 : The different modes of operation

B. Sliding mode (SM):

This mode occurs when the state variable reaches the sliding surface and tends towards the origin of the phase plane. The dynamics of this mode is determined by the establishment of the surface of slip $S(x)=0$.

C. Steady state mode (SSM):

This mode is added to study the response of the system around its equilibrium point, which corresponds to the origin of the phase plane. It is characterized by the quality and performance of the command.

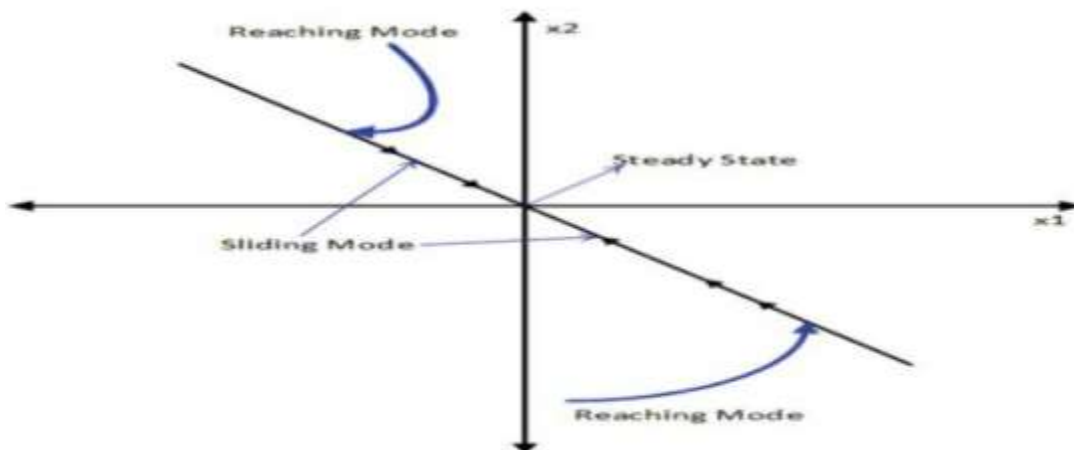


Figure II-2 : the different modes for trajectories in a phase plane

II.5. Design of Sliding mode control

The design of regulators using sliding modes takes care of the problems of stability and desired performance in a systematic way. Implementing this ordering method mainly requires three steps:

- **The choice of surface.**
- **Establishing the conditions for the existence of convergence.**
- **Determination of the control law.**

II.5.1. Choice of sliding surface

The choice of the sliding surface depends on the number and shape of the required functions, which are determined by the application and the intended objective. The surface vector δ has the same dimension as the control vector U . The sliding surface is a scalar function on which the variable to be controlled slides and tends towards the origin of the phase plane. Its nonlinear form depends on the error between the variable to be controlled x and its reference, and is given by the following equation:

$$\begin{cases} \dot{x}(t) = f(x, t) + g(x, t)v(t) \\ y = C^T x, \quad y \in R^m \end{cases}$$

"It's necessary to choose m sliding surfaces for a vector y of dimension m . Regarding the shape of the surface, two possibilities present themselves, either in the phase plane or in the state space. In the latter case, we find the method known as "state reaction switching law". This method uses the concepts of state counter reaction adjustment to synthesize the switching law. Its major disadvantage lies in the fact that it has a slow transient response and is very difficult to design. In the case of phase space processing, the switching function is a scalar function, such that the variable to be adjusted slides on this to reach the origin of the phase plane. Thus, the surface $S(x)$ represents the desired dynamic behavior of the system. Professor J. Slotine proposes a form of general equation to determine the sliding surface which ensures the convergence of a variable towards its desired value."

$$S(x) = \left(\frac{\partial}{\partial x} + \lambda_x \right)^{r-1} e(x)$$

With :

$e(x)$: The deviation of the variable from regulated.

$$e(x) = X_{ref} - X$$

λ : A positive constant that interprets the bandwidth of the desired control.

r : Relative degree, equal to the number of times it shunts the output to bring up the command.

Pour $r=1$, $S(x)=e(x)$
 Pour $r=2$, $S(x)=\lambda x e(x)+e'(x)$
 Pour $r=3$, $S(x)=\lambda_x^2 e(x)+2\lambda_x e'(x)+e''(x)$.

In other words, the difficulty comes down to a problem of trajectory continuation whose objective is to preserve $S(x)$ to zero. This is equivalent to an exact linearization of the difference respecting the convergence condition. The exact linearization of the gap aims to deforce the dynamics of the gap in order to force the dynamics of the gap (reference – output) to be a dynamic of an autonomous linear system of order "r".

II.5.2. Convergence conditions

We have identified two convergence conditions from the literature, which allow dynamics of the system to converge towards the sliding surfaces. These conditions determine the mode of convergence of the state of the system.

II.5.2.1. Direct switching function

This involves giving the surface a dynamic converging towards zero. It is given by:

$$S'(x) \cdot S(x) < 0$$

II.5.2.2. Function of LYAPUNOV

This involves formulating a positive scalar function $V(x) > 0$ for the system state variables, and to choose the switching law which will decrease this function $V(x) < 0$, the idea is to choose a scalar function $S(x)$ to guarantee the attraction of the variable to be controlled towards its reference value and to design a command V such that the square of the surface corresponds to a Lyapunov function. This function is generally used to guarantee the stability of nonlinear systems.

By defining the Lyapunov function by:

$$V(x) = 1/2 \cdot S^2(x)$$

And its derived by:

$$\dot{V}(x) = S(x) \cdot S'(x)$$

For the Lyapunov function to decrease, it is enough to ensure that its derivative is negative.

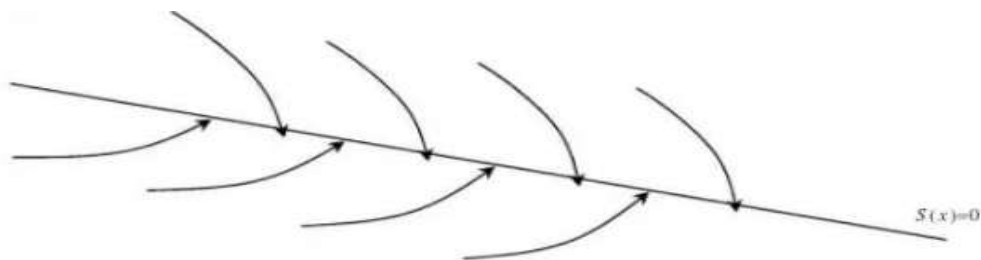


Figure II-3 : trajectory of the state with respect to the sliding surface

This is checked if $S(x)\dot{S}(x) < 0$. This equation shows that the square of the distance to the surface, measured by $S^2(x)$, decreases all the time, forcing the trajectory of the system to move towards the surface on both sides Figure (II.6). This condition assumes an ideal sliding regime

II.5.3. The law of control

The sliding mode control law aims to orient the trajectories of the system towards a sliding surface equal to zero in the first place, then to maintain these trajectories on this surface secondly. This command is made up of two elements.

First of all, a discontinuous component, denoted u_{sw} , makes it possible to bring the trajectories towards the surface of sliding and ensures robustness in the face of extreme disturbances. Then, a second continuous component, called equivalent control u_{eq} , allows holding and sliding along the sliding surface. It should be noted that the equivalent order is determined using the conditions following invariance of the sliding surface.

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

Control by sliding mode is chosen according to the following expression:

$$U(t) = U_n + U_{eq}$$

The terms used are defined as follows:

$$U_{eq} = - \left(\frac{\partial S}{\partial t} g(x) \right)^{-1} \frac{\partial S}{\partial t} f(x)$$

$$U_n = -K \left(\frac{\partial S}{\partial t} g(x) \right)^{-1} \text{sign}(S(x))$$

Here, $k \geq D + \eta$ represents a design constant, ρ is a variable, and $sg(\rho)$ is a sign function defined as follows:

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

II.6. Classic formalism of the sliding regime

II.6.1. Ideal sliding regime

In theory, the switching element is considered ideal, insensitive to noise, and the sliding regime trajectory corresponds perfectly to the equation $S(x) = 0$. We thus speak of ideal sliding regime.

- The ideal sliding regime is characterized by an oscillation of infinite frequency and amplitude nothing. The point representative of the evolution of the system slides perfectly on the hypersurface of switching of $S(x)$.

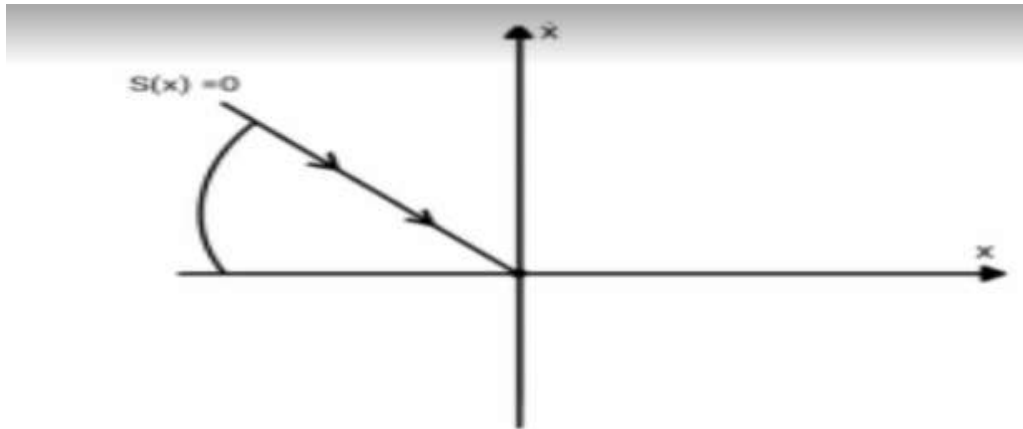


Figure II-4: Ideal sliding regime

II.6.2. Real sliding regime

In practice, the switching member is made from relays which have imperfections, such as switching delays. The phase trajectory in sliding regime remains close to the switching surface, resulting in unwanted oscillations reducing the precision of the system, while nevertheless maintaining its stability.

II.7. Applications of Sliding Mode Control (SMC)

A) Robotics:

Manipulator Control:

SMC is used to control robotic arms and manipulators, ensuring precise tracking and positioning despite external disturbances and model uncertainties.

Mobile Robots

It helps in path following and stabilization of mobile robots.

B) Automotive Systems

Vehicle Dynamics Control

SMC is used for stability control, traction control, and anti-lock braking systems.

Engine Control

It helps in managing fuel injection, air-fuel ratio, and other parameters to optimize engine performance.

C) Aerospace

Flight Control Systems:

SMC is applied to control the attitude, altitude, and trajectory of aircraft, providing robustness against aerodynamic uncertainties and external disturbances.

Satellite Attitude Control

It is used for maintaining the orientation of satellites in space.

D) Power Systems

Electric Drives

SMC is employed in controlling the speed and position of electric motors.

Power Converters

It is used in controlling the output of power converters to achieve desired voltage and current levels.

E) Renewable Energy Systems

Wind Turbines

SMC helps in maximizing power extraction and stabilizing the operation under varying wind conditions.

Photovoltaic Systems

It optimizes the performance of solar panels by ensuring maximum power point tracking.

F) Process Control

Chemical Processes

SMC is used to regulate temperature, pressure, and other variables in chemical reactors.

Manufacturing

It ensures precision and robustness in various manufacturing processes, including CNC machining and additive manufacturing

II.8. Chattering phenomenon

In the design of sliding-mode controllers for realistic applications, it is essential to determine a proper sliding surface so that the output deviations can be reduced to an acceptable level. The sliding-mode control, despite the advantages of simplicity and robustness, generally suffers from the problem, called chattering, which is a very high-frequency oscillation of the sliding variable around the sliding manifold. Chattering is undesirable for the real systems and actuator since it may lead to actuator breakdown and unreasonably large control signal. In practice, the occurrence of time delays in actuators and in many industrial processes, such as transportation lag, time delay, physical limitations and dead time cannot switch at

an infinite frequency along the sliding surface as demanded by the assumption of sliding mode control algorithms.

Many approaches have been projected to overcome chattering phenomenon.

It may be prevailed over by smoothing out the control discontinuity. Frequently preferred approaches are to substitute the control by a saturation function, hyperbolic functions, sigmoid functions, and hysteresis saturation functions.

II.9. Eliminate the chattering phenomenon

In order to reduce or eliminate the chattering phenomenon, several solutions have been proposed. Among these, we find:

- The introduction of other nonlinear approximations of the control term discontinuous, for example, the fractional approximation defined as: equation.
- The replacement of the boundary layer by a slippery sector.
- The use of dynamic SM controllers.
- The use of fuzzy SM controllers.
- The use of higher order SM controllers, such as second order controllers order or higher order. These alternative approaches aim to mitigate the undesirable effects of chattering and improve control system performance.

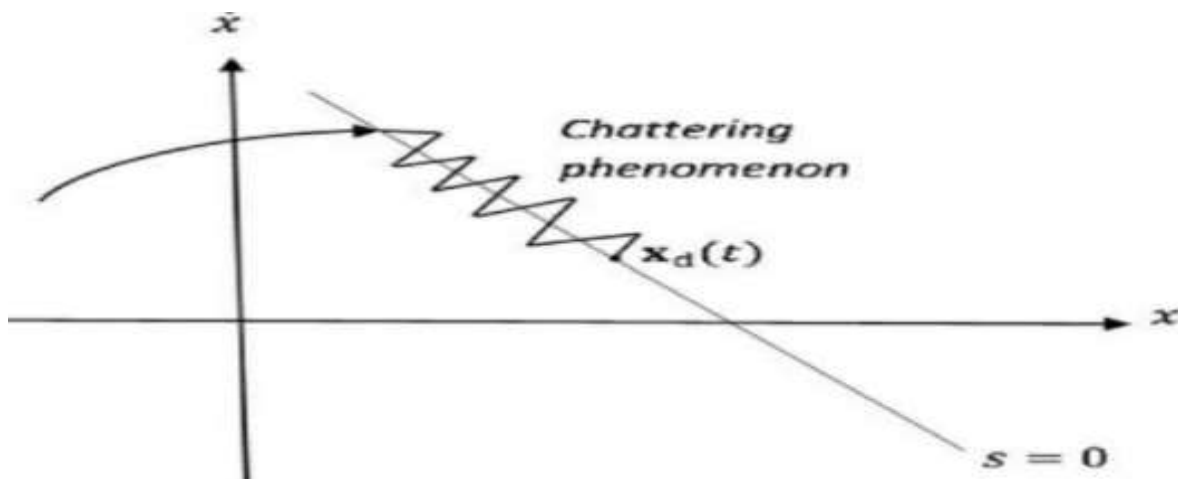


Figure II-5 :chattering phenomenon

II.10. Advantages and disadvantages of sliding mode control

❖ Advantages

- Low sensitivity to plant parameter uncertainty.
- Greatly reduced-order modeling of plant dynamics.
- Finite time convergence (due discontinuous control law)

❖ Disadvantages

- Chattering due to implementation imprecisions
- Over-focus on matched uncertainties (i.e; uncertainties that enter the control channel)

II.11. Conclusion

Sliding mode control is a powerful control technique that can handle uncertainties, disturbances, and nonlinearities in a system. In this chapter, we have discussed the concept of SMC, as well as the techniques used in it, also its applications, advantages and disadvantages.

Chapter III:

Description of the process

III.1.Introduction

The Festo Modular Production System for Process Automation (**Festo MPS PA**) is an advanced educational and training platform designed to simulate real-world industrial process automation systems. It allows users to learn, experiment, and train on various aspects of process automation using standard components that simulate functions found in actual industrial environments.

In this chapter, we will discuss the concept, components, and method of operation of Festo MPS PA.

II.2.Description of festo MPS PA

The system is composed of various modular units such as tanks, pumps, valves, sensors, actuators, and control components that can be arranged and connected to create specific process automation scenarios. All components are connected using appropriate wiring and piping. Sensors are placed at critical points to monitor variables like pressure, temperature, and flow. Actuators like pumps and valves are installed to control the flow and processing of materials.

III.3.Components of festo MPS PA

Festo's Modular Production System (MPS) for Process Automation (PA) is designed to simulate and teach industrial process automation in an educational or training setting. The components of a typical Festo MPS PA station can be categorized into hardware, sensors, actuators, and control systems. Here are the key components:

III.3.1.Hardware components

A) Base Unit

- Framework and mounting plates.
- Modular panels for component integration.

B) Pneumatic Components

- Air supply units (compressors, air preparation units).
- Pneumatic cylinders and actuators.

III.3.1.1.Sensors

A) Pressure Sensors

- Measure pressure in pneumatic and hydraulic systems.

B) Flow Sensors

- Measure the flow rate of liquids and gases.

C) Level Sensors

- Detect the level of liquids in tanks.

D) Temperature Sensors

- Monitor the temperature of fluids and components.

E) Proximity Sensors

- Detect the presence or absence of objects.

III.3.1.2. Actuators**A) Pneumatic Actuators**

- Control valves and cylinders for automation.

B) Electric Actuators

- Motors and solenoids for precise control.

C) Valves

- Solenoid valves for controlling fluid flow.
- Ball valves, butterfly valves, etc.

III.3.1.3. Control Systems**A) Programmable Logic Controllers (PLCs)**

- Central control units for automation processes.
- Festo-specific PLCs or general-purpose PLCs.

B) Human-Machine Interfaces (HMIs):

- Touch panels or display units for operator interaction.

C) Communication Modules:

- Ethernet, PROFIBUS, PROFINET, etc., for network communication.

D) I/O Modules:

- Interface units for connecting sensors and actuators to the PLC.

III.3.2. Software Components :

- Simulation and programming software for developing and testing automation sequences (e.g., Festo's FluidSIM, CIROS).

A) Power Supply Units:

- Provide necessary electrical power to all components.

III.3.3. Safety Components:

- Emergency stop buttons, safety relays, and guards.

These components collectively create a flexible and comprehensive platform for simulating and understanding industrial process automation. Each module can be configured to replicate real-world industrial processes, providing hands-on experience in controlling and optimizing automated systems.

III.4.Applications:

III.4.1. Educational :

A) Training Technical Education:

Used in vocational schools, technical colleges, and universities to teach students about industrial automation, pneumatics, hydraulics, and control systems.

B) Hands-on Learning:

Provides practical experience in setting up, programming, and troubleshooting automation systems.

C) Curriculum Integration:

Integrated into engineering, mechatronics, and industrial technology courses to give students real-world experience with industrial processes.

III.4.2. Industrial Training :

A) Workforce Development:

Used by companies to train employees on the latest automation technologies and processes.

B) Skill Enhancement:

Helps workers upgrade their skills in operating, maintaining, and troubleshooting modern automated systems.

C) Safety Training:

Simulates real-world scenarios to train employees in safe operating practices and emergency response.

III.4.3. Research and Development :

A) Process Optimization:

Used in R&D labs to develop and test new automation processes and technologies.

B) Innovation Testing:

Provides a controlled environment to experiment with innovative ideas in process automation before deploying them in actual industrial settings.

III.4.4. Industrial Simulation:**A) Process Simulation:**

Allows for the simulation of complex industrial processes for planning, testing, and optimization purposes.

B) System Integration Testing:

Tests the integration of various components and subsystems in a controlled environment to ensure compatibility and performance.

III.4.5. Product Development :**A) System Testing:**

Used by manufacturers to test and validate the performance of new automation products and systems.

B) Quality Assurance:

Ensures that new products meet industry standards and perform reliably under various conditions.

III.4.6. Industry-Specific Applications :**A) Chemical and Pharmaceutical Industries:**

Simulates processes like mixing, filling, and packaging of chemicals and pharmaceuticals.

B) Food and Beverage Industry:

Models processes such as sorting, filling, and packaging of food and beverages.

C) Automotive Industry:

Simulates assembly lines, part sorting, and quality control processes.

III.5. Technical Data:

Parameter	Value
max. operating pressure piping system	50 kPa (0.5 bar)
power supply for station	24 V DC
profile plate dimensions	700 x 700 x 32 mm
Weight	700 x 700 x 907 mm
(volumetric) flow rate of the pump	~5 l/min
tank volume	10 l max.

flexible piping system	DN10 (\emptyset , 15mm)	
digital inputs	7	
digital outputs	5	
analog inputs	4	
analog outputs	2	
amount of tanks	3	
output range control element	pump (0...24 VDC)	voltage 0...10 V
	2/2W-proportional valve	voltage 0...10 V
	heating element 230 VAC (power 1000 W)	On/Off (control relay 24 VDC)
working range closed-loop system for level control	0...10 l mm	
measuring range level sensor	0...9 l	
signal range level sensor current	4...20 mA	
working range closed-loop system for flow rate control	0...7 l/min	
measuring range flow sensor	0,3...9,0 l/min	
signal range flow sensor	frequency 0...1200 Hz	
working range closed-loop system for pressure control	0...30 kPa(0...300 mbar)	
measuring range pressure sensor	0...10 kPa (0...100 mbar)	
signal range pressure sensor voltage	0...10 V	
working range closed-loop system for temperature control	0...60° C	
measuring range temperature sensor	-50° C...+150° C	
signal range temperature sensor	resistance PT100	

Table III.1 : Technical data of festo MPS PA

III.6. Conclusion :

In conclusion, the Festo MPS PA system stands out as an advanced educational and training tool in the field of industrial automation. Through our review of the system's various components, whether hardware, software, or safety components, we have gained an understanding of the complex infrastructure that makes this system an effective model for automation applications. We also highlighted the practical applications of this system, enhancing our understanding of how it can be used to simulate and improve industrial processes. The technical data provided offers vital information that contributes to the optimal use of the system. The knowledge acquired from this chapter represents an important step

towards developing automation skills and understanding its advanced technologies, opening new horizons for trainees and professionals in this field.

Chapter IV:

Simulation and results

IV.1.introduction

This chapter explores the simulation of the Festo MPS PA system using MATLAB and optimization techniques such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), for Sliding Mode Control (SMC). Festo MPS PA systems emulate real industrial processes, making them ideal for studying industrial automation control. SMC is a robust control technique for handling system uncertainties. The chapter aims to apply PSO and GA, to tune SMC parameters and compare their effectiveness. MATLAB's powerful simulation and optimization capabilities are used to implement these algorithms. The chapter covers the system components, algorithm principles, MATLAB implementation, simulation results, and discusses findings and future work.

IV.2.Application

The system dynamics FESTO MPS PA can be described by differential equations or transfer functions, which are converted to state-space representation for SMC design :

- Assume the system has a transfer function ($G(z)$) with a sample time $T_s=0.1$ s:

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

IV.2.1.Transfer function and state space representation

❖ Calculation of matrices:

- We convert this discret function to continue function then to state space as follows:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

$\mathbf{x}(t)$: The state vector

$\mathbf{u}(t)$: The control input

$\mathbf{y}(t)$: The system output

- The matrices A, B, C and D were obtained using MATLAB ,These matrices represents the dynamic state of the Festo MPS PA system after conversion to state-space representation.
- the value of matrix are as follows

$$A = \begin{bmatrix} -0.2958 & -0.0031 \\ 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C=[0.1360 \quad 0.0012]$$

$$D=[0]$$

IV.2.2.Design of sliding mode control

sliding mode control is designed according to the following expression:

$$U(t)=U_{eq}(t)+U_n(t)$$

A) Determine the equivalent control U_{eq}

- To maintain the desired state, we first calculate the equivalent control U_{eq} , this is done by ensuring that the sliding surface ($s(t)=0$).
- The sliding surface ($s(t)$) is defined as:

$$s(t)=\lambda e(t)+\dot{e}(t) \quad / \quad e(t)=(y_r(t)-y(t)) \quad \text{and} \quad y(t)=Cx(t)$$

λ : positive parameter ($\lambda > 0$)

$y_r(t)$: reference parameter

$y(t)$: output

$$s(t)=\lambda(y_r(t)-y(t))+(\dot{y}_r(t)-\dot{y}(t)) \quad \longrightarrow \quad =\lambda(y_r(t)-Cx(t))+(\dot{y}_r(t)-C\dot{x}(t))$$

To maintain $s(t)=0$, we solve for the equivalent control U_{eq} :

$$s'(t)=\lambda\dot{e}(t) = \lambda(\dot{y}_r(t)-C\dot{x}(t)) \quad [\text{Assuming } (\ddot{y}_r(t)=0)]$$

- we replace x :

$$s'(t)=\lambda(y_r'(t)-CAx(t)-CBu(t))=0$$

So :

$$U_{eq}=(CB)^{-1}(y_r'(t)-CAx(t))$$

B) the switching control U_n

$$U_n=K(CB)^{-1}\text{sign}(s(t))$$

$$\text{Sign}(s)= \begin{cases} 1 & \text{if } s > 0 \\ 0 & \text{if } s = 0 \\ -1 & \text{if } s < 0 \end{cases}$$

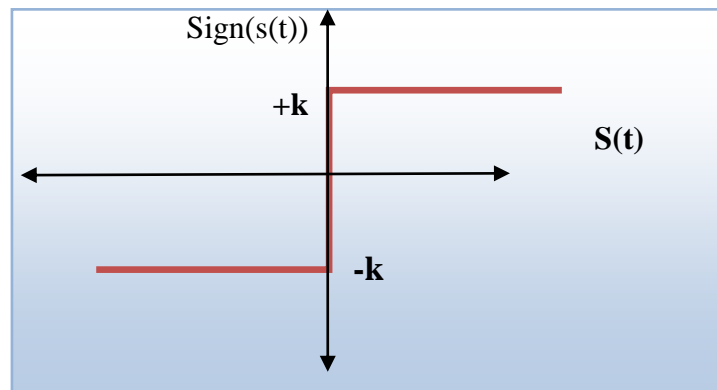


Figure IV.1 : sign function

we implement sliding mode control (SMC) to manage the Festo MPS PA using MATLAB 2015, the model was constructed in simulink ,where the simulation settings were configured using ode1 (Euler) With a sample time $T_s=0.01$ s , a stop time =10 and a simulation time range $t=[0;5]$ s.

The primary objective was to enhance the system's performance by minimizing the error .to quantify this we employed a fitness function $G(t)=\int (abs (e(t)))^2 dt$ where $e(t)$ represents the error . This integral of the squared error over time served as the key of performance metric for our optimization efforts. To achieve this optimization, we employed two advanced artificial intelligence techniques:

IV.3.2.Optimization by PSO

- For using the PSO algorithm we choose :
 - Size of The swarm $n=30$
 - Maximum number of iterations =10
 - The individual wich optimized is: $[k, \lambda]^T$
 - Range of each parameter is: $K=[0.2;5]$,and $\lambda=[0.01;10]$.
 - The results are obtained as follows :

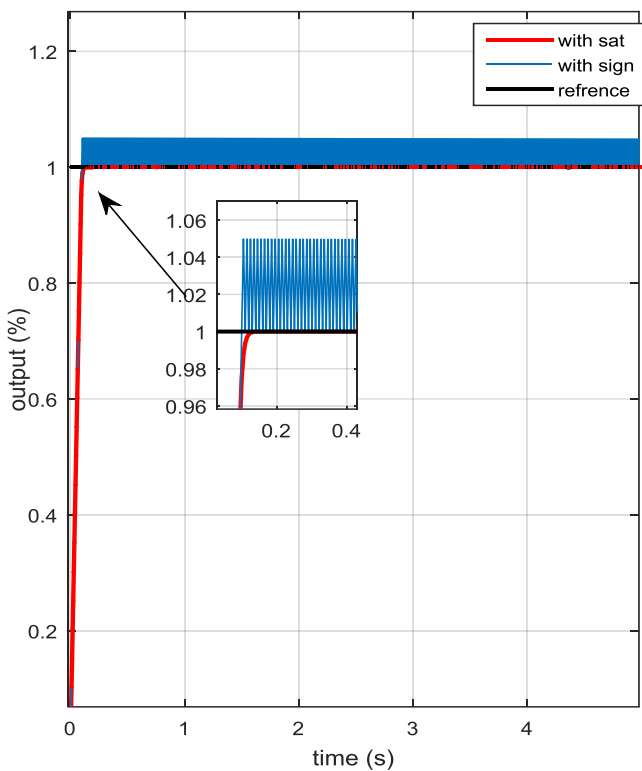


Figure A

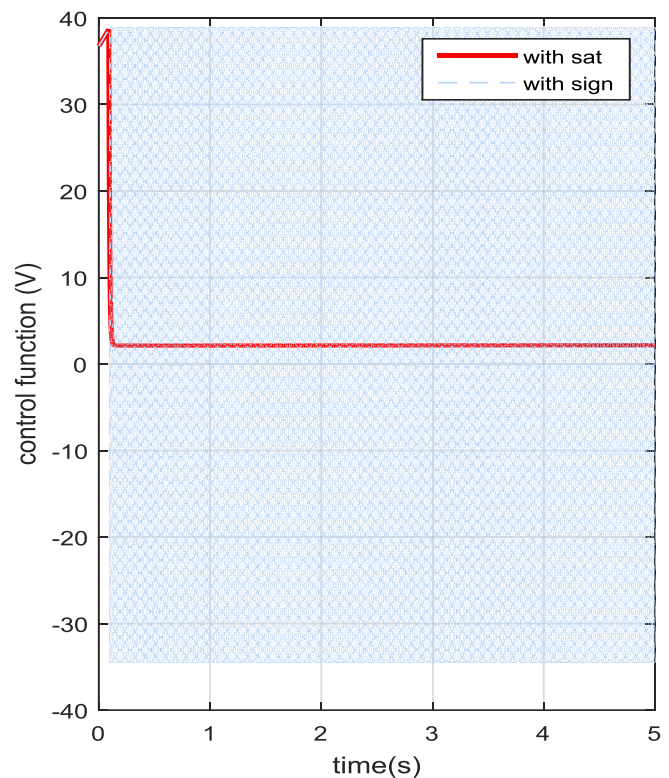


Figure B

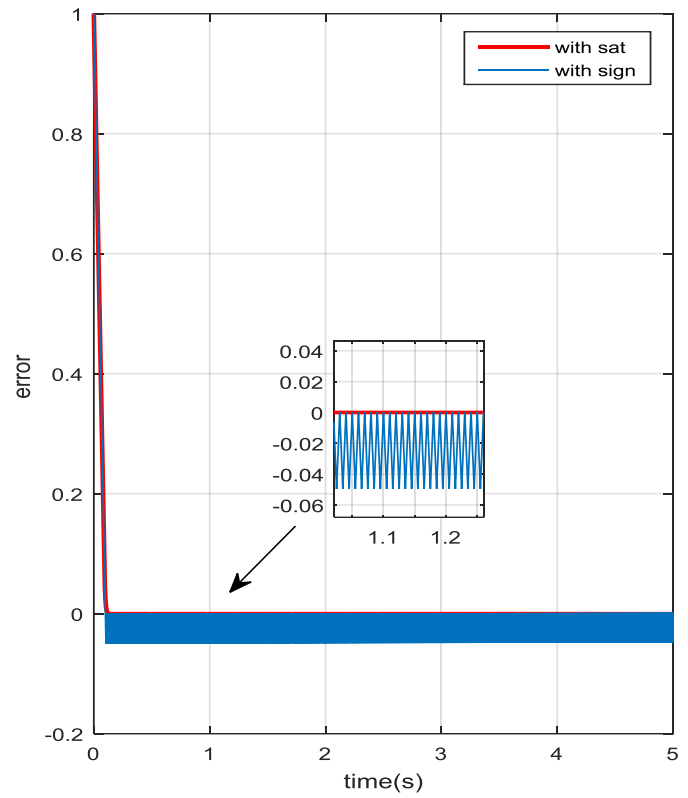
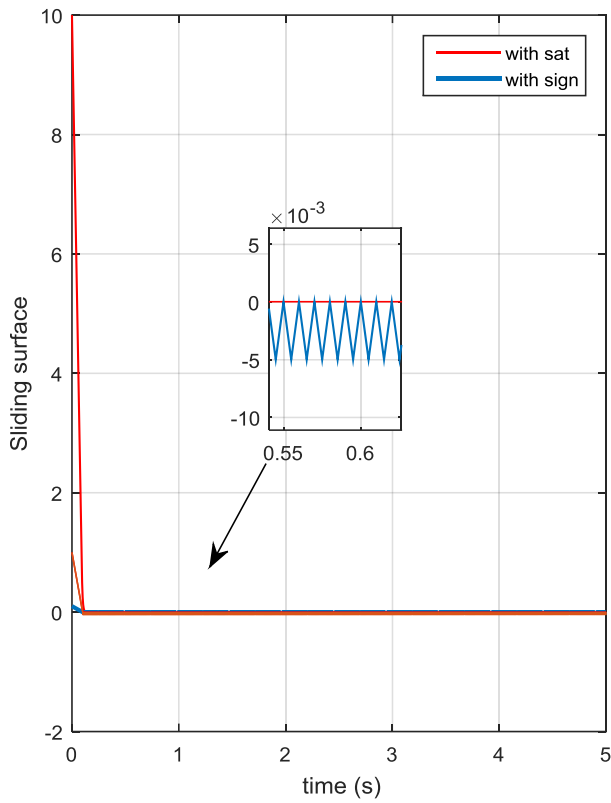


Figure C

Figure D

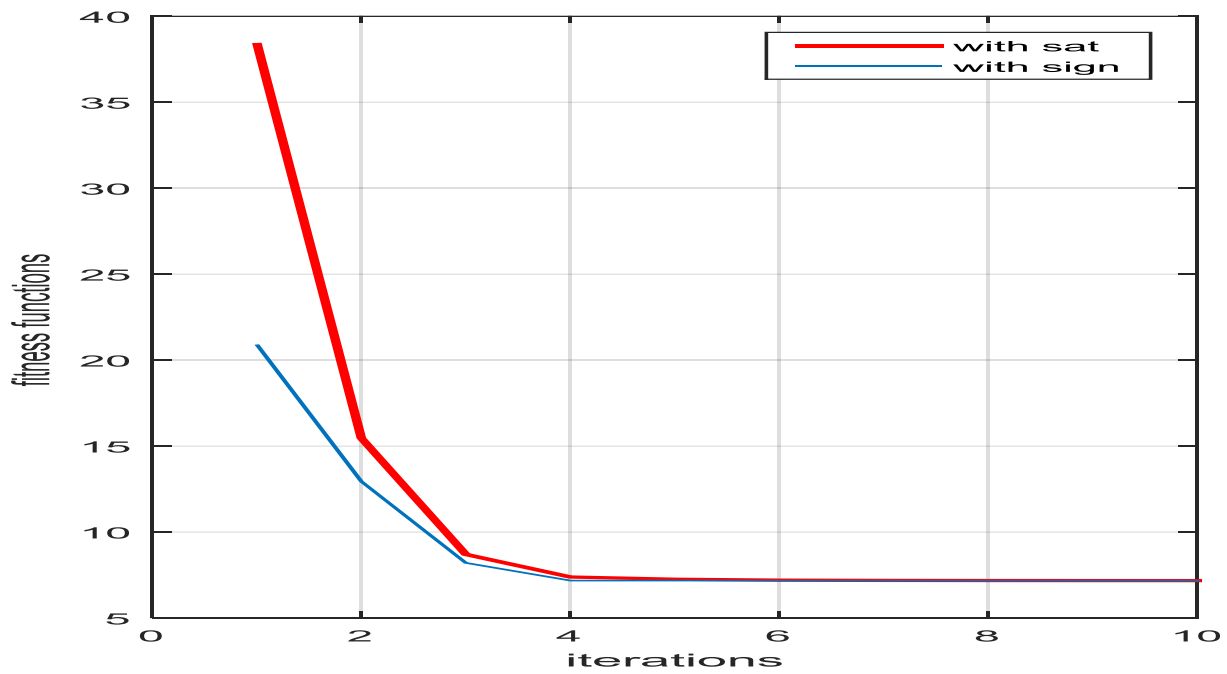


Figure E

FigureIV.3 :Results of optimization by PSO with sat and sign function

IV.3.3.Optimization by GA

- The individual wich optimized is: $[k, \lambda]^T$
- Lower bound= $[0.2 \ 0.01]$, Upper bound= $[5 \ 10]$
- Number of generation= 10

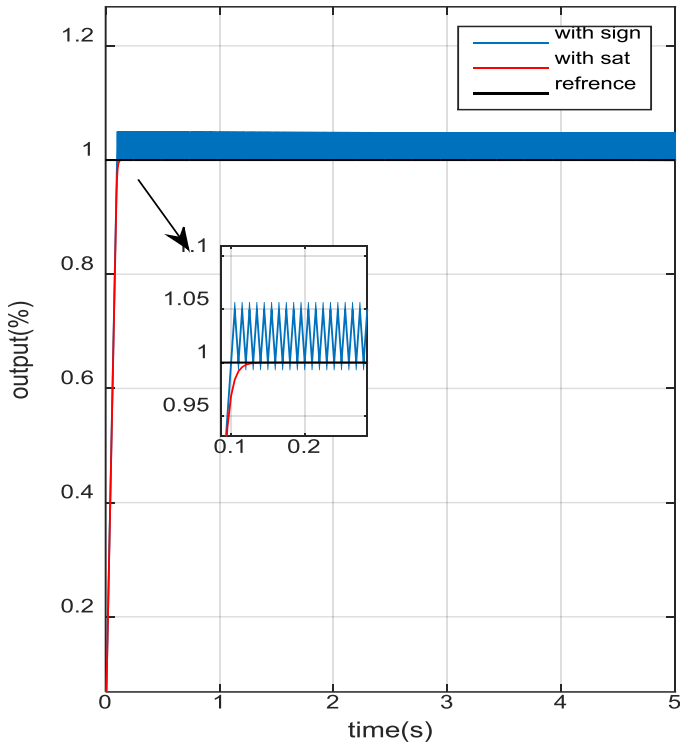


Figure A1

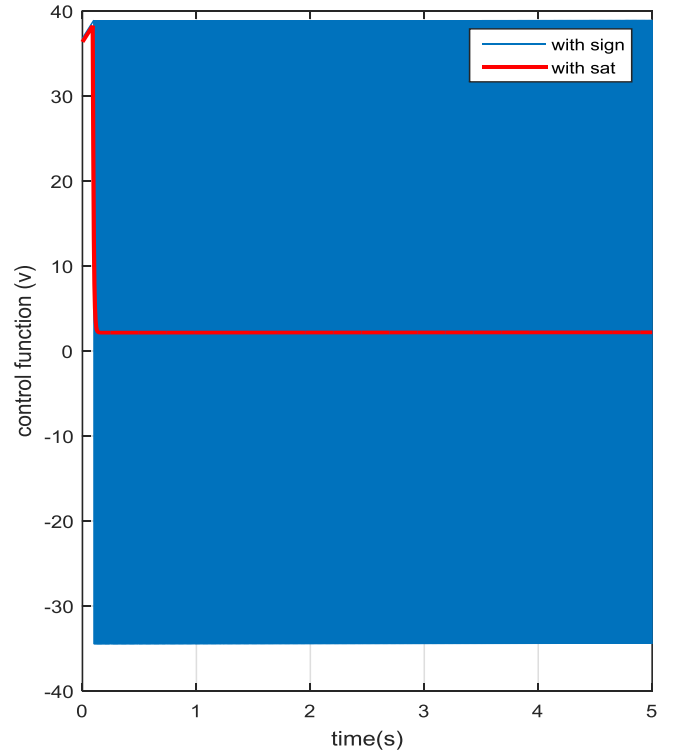


Figure B1

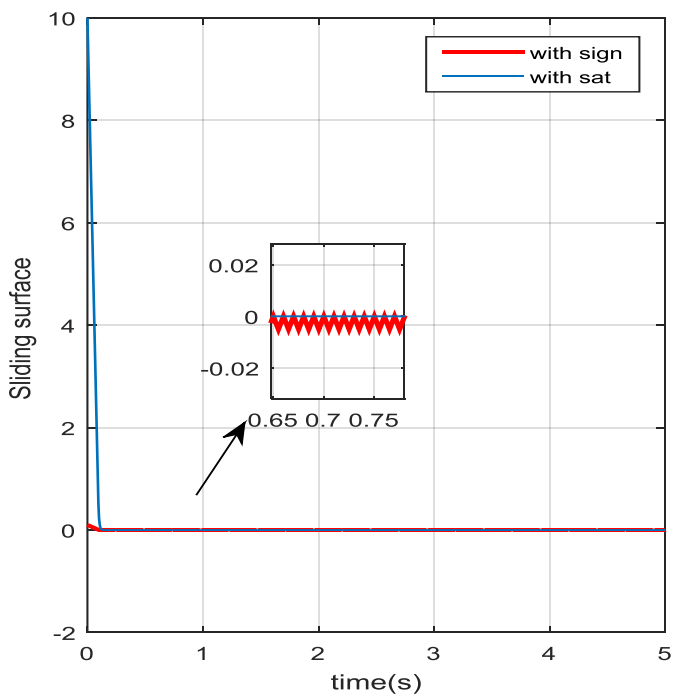


Figure C1

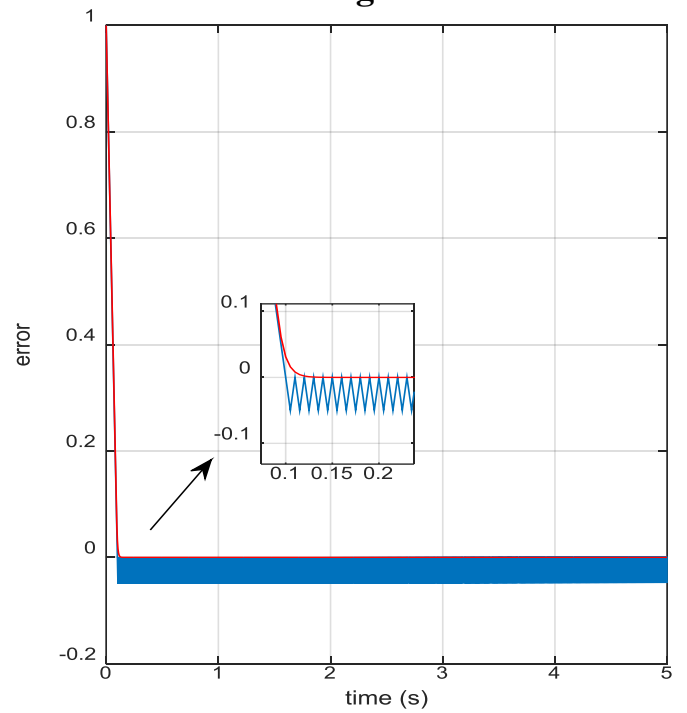


Figure D1

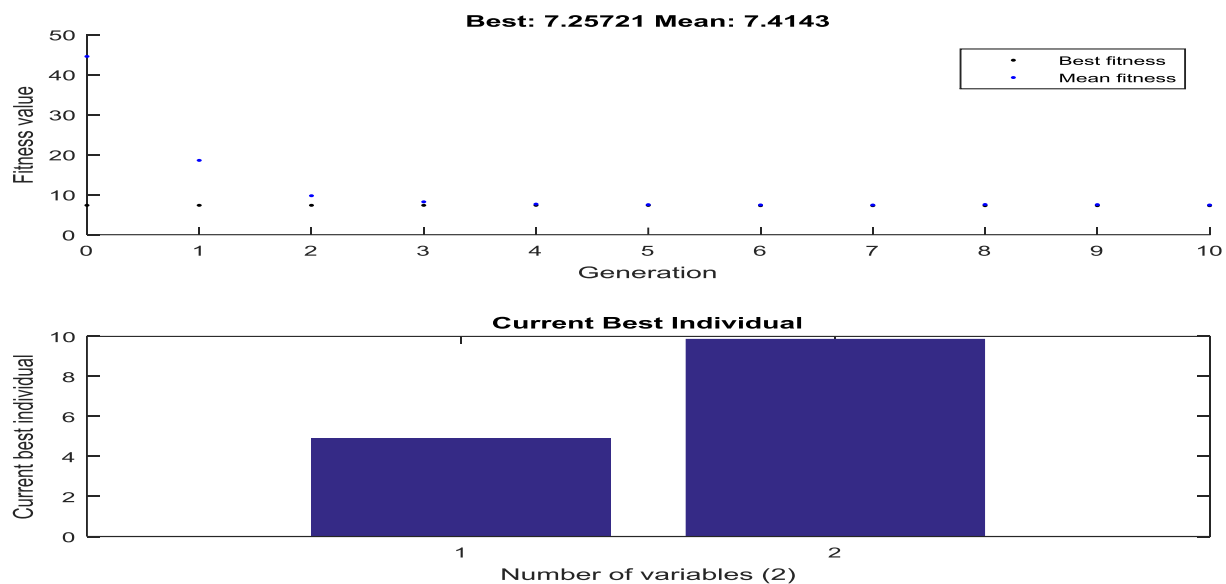


Figure E1.Fitness function with sign function

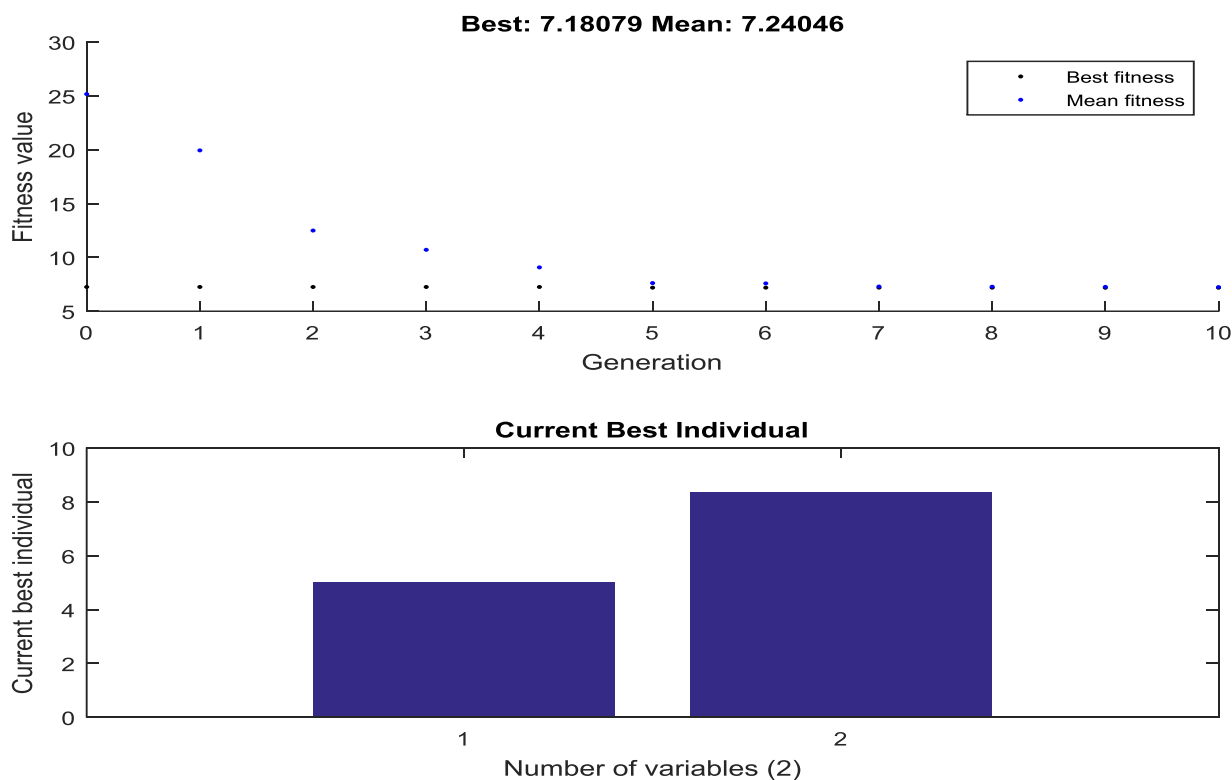


Figure F.Fitness function with sat function

Figure IV.4.:Results of optimization by PSO with sat and sign function

IV.3.4.Discussion

A) Comparaison between sign and sat functions

The figures (IV.3 and IV.4) illustrate the simulation results of optimizing SMC parameters using PSO and GA.

- **Figure A** and **Figure A1** represent the output **Figure B** and **Figure B1** represent the control function. **Figure C** and **Figure C1** represent the sliding surface. **Figure D** and **Figure D1** represent the error **Figure E** , **Figure E1** and **Figure F** represent the fitness function.

In each of the presented graphs, both the sign and sat functions were used. We observe that using the sign function leads to noticeable disturbances in the system. These disturbances arise due to the discontinuous nature of the sign function, which produces binary values (-1 or +1) causing rapid and sudden switching, leading to significant oscillations in the system (chattering). To eliminate these disturbances and achieve a more stable and smoother response, we employed the sat function. The sat function is continuous and saturated, which gradually determines values between -1 and +1, thus reducing sudden switching and oscillations.

B) Comparaison between PSO and GA algorithms

The PSO and GA algorithms show the best results in terms of temporal performance, which includes speed, stability, and accuracy. To determine which algorithm is the most efficient, we rely on two key values:

- **The mean value of the command U_{mean} :**

This value represents the average of the solutions found by the algorithm over time. A lower or more stable average value may indicate a more effective algorithm in finding optimal or near-optimal solutions.

- **The minimum fitness value G_{min} :**

Fitness is a measure of the quality of the solution in the context of optimization. The minimum fitness value G_{min} represents the best solution found by the algorithm. A lower minimum fitness indicates a solution closer to the global optimum. By comparing these two values for each algorithm, we can determine which of the two offers the best overall performance according to the following **table(IV.1)**:

Algorithms Parameters	PSO		GA	
	Sign(t)	Sat(t)	Sign(t)	Sat(t)
K	5	5	4.9998	4.9671
λ	0.1	10	8.3619	9.9236
Gmin	7.1806	7.1786	7.2404	7.4143
Umean	2.9474	2.8611	2.9473	2.8608
abs(e)	0.0017	0.0000	0.0018	0.0000

Table IV.1 :The general performances of system optimized by PSO and GA algorithms

- According to the **table(IV.1)**we observediffernet values of Gmin and Umean in PSO and GA :
- **G min** :a lower value indicates better performance. The PSO shows lower Gmin values than GA in both sign and sat functions ,which means PSO offers better performance than GA.
- **Umean**:a lower value indicates better performance. The values for PSO and GA are very close, but GA slightly outperforms PSO in this metric.

In the last:

If Gmin is considered more critical for the system's performance, then the PSO algorithm would be the preferred choice. However, if Umean is more important, GA might be slightly better. Given PSO's slight edge in Gminand the very close values for Umean, PSO could be considered the overall better algorithm for this specific scenario.

IV.4.Conclusion

In this research, a model for simulating the Sliding Mode Control (SMC) system of the Festo MPS PA unit was presented. We focused on optimizing the system's performance using Particle Swarm Optimization (PSO) and Genetic Algorithm (GA).During the discussion, we compared the sign and satfunctions as well as the performance of GA and PSO in terms of Gmean and Umin. The results showed that both algorithms were capable of enhancing the performance of the SMC system, but they differed in efficiency based on the metrics used.The PSO algorithm proved to be more efficient and faster in reaching the optimal solution due to its ability to adapt to changes in the environment and gradually improve solutions. On the other hand, the GA algorithm demonstrated a greater ability to escape local optima due to its stochastic nature and operations of selection and crossover.This study provides valuable insights into how industrial process control systems can be improved using artificial

intelligence techniques, contributing to enhanced efficiency and reliability in practical applications.

General conclusion

Artificial intelligence (AI) techniques are pivotal in enhancing the performance and efficiency of control systems in industrial automation. They address complex optimization problems that traditional methods struggle with, offering solutions to maximize profits, quality, and efficiency while minimizing time, cost, and risk. Traditional optimization methods are limited by sensitivity to initial estimates and often converge on local optima, especially as problem dimensionality increases. Metaheuristic algorithms like Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) have emerged as effective solutions due to their ability to explore complex search spaces and find global optima.

In this study we focuses on improving the precise performance of control systems using AI methods. Specifically, it applies PSO and GA to enhance control laws in sliding mode control systems. This aims to increase efficiency and accuracy in dynamic industrial environments.

The study compared PSO and GA in optimizing the Sliding Mode Control (SMC) system of the Festo MPS PA unit. The results showed that both algorithms were capable of enhancing the performance of the SMC system, but they differed in efficiency based on the metrics used. The PSO algorithm proved to be more efficient and faster in reaching the optimal solution due to its ability to adapt to changes in the environment and gradually improve solutions. On the other hand, the GA algorithm demonstrated a greater ability to escape local optima due to its stochastic nature and operations of selection and crossover.

In conclusion, the thesis provides valuable insights into leveraging AI-driven optimization techniques to advance industrial control systems. It highlights the efficacy of PSO, GA, thereby contributing to more efficient and reliable industrial automation solutions. This research is crucial for practitioners looking to implement cutting-edge AI technologies to tackle real-world challenges in automation and control engineer.

References

- [1]. Blum, Christian, and Andrea Roli. "Metaheuristics in combinatorial optimization: Overview and conceptual comparison." *ACM Computing Surveys (CSUR)* 35, no. 3 (2003): 268-308.
- [2]. Wang, D. "Enhanced particle swarm optimization through external memory support." In *2005 IEEE Congress on Evolutionary Computation*. Cited 1943 times (2018).
- [3]. Kennedy, J., and R. Eberhart. "Particle swarm optimization." In *Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE, 1942-1948* (1995).
- [4]. Mirjalili, S. "How effective is the grey wolf optimizer in training multi-layer perceptron." *Applied Intelligence* 43, no. 1 (2015): 150-161.
- [5]. Rezaei, H., O. Bozorg-Haddad, and X. Chu. "Grey wolf optimization (GWO) algorithm." In *Advanced Optimization by Nature-Inspired Algorithms*, edited by Omid Bozorg-Haddad, Springer, Singapore, 2018.
- [6]. Genlin, J. "Survey on genetic algorithm." *Computer Applications and Software* 21, no. 3 (2004): 45-52.
- [7]. Marano, G. C., Quaranta, G., & Monti, G. (2011). Modified genetic algorithm for the dynamic identification of structural systems using incomplete measurements. *Computer-Aided Civil and Infrastructure Engineering*, 26(2), 92–110. <https://doi.org/10.1111/j.1467-8667.2010.00659.x>
- [8]. Jamal, R., Men, B., & Khan, N. H. (2020). A novel nature-inspired meta-heuristic optimization approach of GWO optimizer for optimal reactive power dispatch problems. *IEEE Access*, 8, 15711-15721.
- [9]. Edwards, C., & Spurgeon, S. (1998). *Sliding Mode Control: Theory and Applications*. Taylor and Francis.
- [10]. Utkin, V. I. (1993). Sliding mode control design principles and applications to electric drives. *IEEE Transactions on Industrial Electronics*, 40(1), 23-36.
- [11]. Zeinali, M., & Notash, L. (2010). Adaptive sliding mode control with uncertainty estimator for robot manipulators. *Mechanism and Machine Theory*, 45(1), 80–90.
- [12]. Almufti, S. M., Shaban, A. A., Ali, R. I., & Dela Fuente, J. A. (2023). Overview of metaheuristic algorithms. *Polaris Global Journal of Scholarly Research and Trends*, 2(2), 10-32.
- [13]. Luca Maria Gambardella's publication titled "Evolving a Robust Controller for a Simulated Car" presented at the Artificial Life VII conference in 2000.

- [14]. "An introduction to particle swarm optimization algorithm." Analytics Vidhya. Consulted on 24.04.2024. <https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-particle-swarm-optimization-algorithm/>
- [15]. Colorni, A., Dorigo, M., & Maniezzo, V. (1991). Distributed optimization by ant colonies. Proceedings of the First European Conference on Artificial Life, 134-142.
- [16]. Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine, 22(3), 52-67.
- [17]. Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.
- [18]. Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press.
- [19]. Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization: An overview. Swarm Intelligence, 1(1), 33-57.
- [20]. Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In IEEE International Conference on Evolutionary Computation Proceedings (ICEC) (pp. 69-73). IEEE.
- [21]. Cassandras, C. G., & Anastasopoulos, I. M. "Sliding Mode Control: Theory and Applications." This book provides a comprehensive overview of sliding mode control theory, including different modes of operation and applications.
- [22]. Edwards, C., & Spurgeon, S. K. (1998). "Sliding Mode Control: Theory and Applications." CRC Press.
- [23]. Utkin, V. I., Guldner, J., & Shi, J. (2009). "Sliding Mode Control in Electromechanical Systems." CRC Press.
- [24]. Levant, A. (2013). "Sliding Mode Control and Observation." Springer.
- [25]. "Learning Factories - Festo MPS." Festo. Consulted on 03.05.2024. https://www.festo.com/us/en/e/technical-education/educational-concepts/highlights/learning-factories/mps-mechatronics-learning-factories-id_31963/
- [26]. Putawa, R. A., Wardana, A. N. I., & Tenggara, A. P. (2023). "Overview of Metaheuristic Algorithms." J. Phys.: Conf. Ser. 2673 012008. <https://doi.org/10.1088/1742-6596/2673/1/012008>
- [27]. Durand, N. (2004). "Genetic Algorithms and Other Optimization Tools Applied to Air Traffic Management." INPT.
- [28]. Hicham, T. (2016). "Sliding Mode Control: Observation and Parametric Estimation of an Induction Machine with Faults Presented," Mohamed Khider University – Biskra.

- [29]. Ignaciuk, P., & Andrzej, B. (2012). "Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs." Springer Science and Business Media.
- [30]. Liu, J., & Wang, X. (2012). "Advanced Sliding Mode Control for Mechanical Systems." National University of Singapore.
- [31]. Edward, C., & Spurgeon, S. K. (1998). "Sliding Mode Control: Theory and Applications." CRC Press.
- [32]. Vecchio, C. (2008). "Sliding Mode Control: Theoretical Development and Applications to Uncertain Mechanical Systems." Università degli Studi di Pavia.
- [33]. Koszewnik, A., Nartowicz, T., & Pawłuszewicz, E. (2016). "Fractional Order Controller to Control Pump in FESTO MPS® PA Compact Workstation." 2016 IEEE 20th International Conference on Control Systems and Carpathian Conference (ICCS). DOI: 10.1109/CarpathianCC.2016.7501124.
- [34]. Walaa, A. (2017). "System Identification and Model Prediction Control for a Liquid Tank Process." Master thesis, Philadelphia University.
- [35]. Helmich, J. A. (2008). "Process Automation MPS® PA Compact Workstation Manual."
- [36]. Helmich, J. (2008). "Festo MPS PA Compact Workstation Manual."
- [37]. Wolfgang, E. (2008). "MPS PA Compact Workstation, Fluid Lab® - PA - Handling Description."
- [38]. Festo Didactic Technical Documentation. "Data Sheet." (2008).
- [39]. "Datasheets_en_11-05-19.pdf."
- [40]. Romuszewski, P., Mystkowski, A., & Pawłuszewicz, E. (2015). "Modeling and Identification Festo MPS® PA Compact Workstation." *Przegląd Elektrotechniczny*, Vol. R. 91, nr 3, pp. 152-156.
- [41]. Festo Didactic Technical Documentation. "Data Sheet." (2008).