République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 8 Mai 1945 Guelma

Faculté des Sciences et de la Technologie
Département Electronique et Télécommunication
Laboratoire de Contrôle Avancé (LabCAV)

THÈSE

EN VUE DE L'OBTENTION DU DIPLOME DE DOCTORAT EN 3ème CYCLE

Domaine : **Sciences et Technologie** Filière : **Électronique**
Spécialité : **Automatique**

Présentée par

Adel MAKHBOUCHE

Intitulée

Commande Intelligente Basée Sur Les Systèmes D'ordre Fractionnaire

Soutenue le : 17.12.2023 Devant le Jury composé de :

Nom et Prénom	**Grade**		
Mr Djalil BOUDJEHEM	Professeur	Univ. de 8 Mai 1945 Guelma	Président
Mr Badreddine BOUDJEHEM	Professeur	Univ. de 8 Mai 1945 Guelma	Encadreur
Mr Fateh ELLAGGOUNE	Professeur	Univ. de 8 Mai 1945 Guelma	Examinateur
Mr Aissa BELMEGUENAI	Professeur	Univ. de 20 août 1955 Skikda	Examinateur

Année Universitaire : 2023/2024

Abstract

The goal of this thesis is to propose a design control laws based on the combination of advanced control techniques and fractional-order control (FOC), in our case the combination between the computational intelligence (CI) and FOC. This design strategy hardly ever been found in the specialized literature.

This study's purpose can be summarized as follows:

(1) The use of computational intelligence to improve the performance of the fractional-order PID (FOPID) controller. (2) Genetic algorithm (GAs) which are part of evolutionary algorithms (EAs) is used to adjust the controller parameters. (3) The proposed control laws have been applied to control time-delay systems and the liquid level in three tank process system. (4) The validity of the proposed techniques is proven by simulation results in MTLAB in terms of dynamic tracking and disturbance rejection.

Keywords :

Computational intelligence – time-delay systems – evolutionary algorithms – fractional-order control.

Résumé

L'objectif de cette thèse est la proposition de lois de commande basée sur la combinaison de techniques de contrôle avancée et du contrôle d'ordre fractionnaire, dans notre cas la combinaison entre l'intelligence computationnelle et le contrôle d'ordre fractionnaire. Cette stratégie de conception n'a pratiquement jamais été trouvée dans la littérature spécialisée.

Le but de cette recherche se résume comme suit :

(1) L'utilisation de l'intelligence computationnelle pour améliorer les performances du contrôleur PID d'ordre fractionnaire. (2) Les algorithmes génétiques (GAs) qui font partie des algorithmes évolutionnaires (EA) sont utilisés pour régler les paramètres du contrôleur. (3) Les lois de contrôle proposées ont été appliquées pour contrôler les systèmes à retards et le niveau de liquide dans un système à trois réservoirs. (4) La validité des technique proposées sont prouvées par les résultats de simulation dans MTLAB en termes de suivi dynamique et de rejet des perturbations.

Mots clés :

Intelligence computationnelle – systèmes à retards – algorithmes évolutionnistes – contrôle d'ordre fractionnaire.

Acknowledgment

First, thanks to Allah for giving me the will, the patience and favored me over several people to carry the title of a doctor.

I'm so grateful to all the persons who accompanied me to finalize one of the most important projects in my professional career, defending my Doctor of Philosophy in Electronics specialty Automation, i wish i meet their expectations.

I'm so grateful to the MESRS in Algeria for giving me the opportunity to study in best-in-class universities in the northeast of Algeria without any costs.

This thesis has been carried out in the advanced control laboratory (LabCAV, https://labcav.univ-guelma.dz/fr) of the university 08 Mai 1945 - Guelma directed by Prof.Dr **Djalil BOUDGEHEM**, whom i thank a lot for his help.

This thesis is prepared under the supervision of Prof.Dr **Baddredine BOUDJEHEM**, head of the fractional-order systems research team, i want to give him my special thanks for spending his time on all the meetings that we did together, for advising me and giving me the roadmap to follow my research successfully.

I want to thank my professors at the University 20 august 1955 - Skikda for the good training i had in the field of industrial control, specifically: Dr. **Abderazek LACHOURI**, Dr. **Hacene BOUZEKRI**, Dr. **Faouzi BOUCHAREB**, Dr. **Nadir DEROUICHE**, Dr. **Aissa BELMEGUENAI**, Dr. **Rafik DJEMILI**, Dr. **Abdesselam LEBAROUD**, Dr. **Djamel SAYAD**, Dr. **Samir LADACI**, Dr. **Salim OUCHTATI**.

I would like also to thank Dr. **Cristina Ioana MURESAN**, professor, and researcher in the department of automation at the technical university of Cluj-Napoca, Romania for her significant contribution and improvement to our scientific article.

I'm so grateful to **Dmitry BOKOTEY**, CCIE No. 4460, who holds a quadruple Cisco expert-level certifications (CCIE) for his advises and all the lessons that i learnt from him.

I would like to thank **Imed BOUREGA**, former General Manager Algeria / Africa Services Director for Emerson Automation Solutions, for giving me the opportunity to work for Emerson which allowed me to gain a great practical experience in the domain of industrial automation.

Thanks to Dr. **Mohamed Ilyas RAHAL**, Dr. **Abderahmane GANOUCHE**, Dr. **Sami LAIFA** and Dr. **Hamza GASMI** for their advises to accomplish this thesis.

and finally, a special thanks to my parents and my family.

Table of Contents (TOC)

List of figures

List of tables

Nomenclature

MESRS	Ministry of higher education and scientific research of Algeria
PRFU	University-training research projects
LabCAV	Advanced control laboratory
SISO	Single input - single output
CO	Classical optimization
AI	Artificial intelligence
CI	Computational intelligence
EC	Evolutionary computation
EA	Evolutionary algorithm
EP	Evolutionary programming
ES	Evolutionary strategies
GA	Genetic algorithms
GP	Genetic programming
SI	Swarm intelligence
PSO	Particular swarm optimization
ACO	Ant colony optimization
PID	Proportional integral derivative
FPID	Fuzzy proportional integral derivative controller
FFPID	Fuzzy fractional proportional integral derivative
FC	Fractional calculus
FOS	Fractional-order systems
FOC	Fractional-order control
FOMCON	Fractional-order modeling and control
FOPID	Fractional-order proportional integral derivative
FOD	Fractional-order derivative
FOPD	Fractional-order proportional derivative
FOPI	Fractional-order proportional integral
FOBSMC	Fractional-order backstepping sliding mode control
FOSMC	Fractional-order sliding mode control
FOGPC	Fractional-order generalized predictive control
FOPF	Fractional-order potential field
FOESC	Fractional-order extremum seeking controller

FOTSMC	Fractional-order terminal sliding mode control
FOCF	Fractional-order complementary filters
F-MIGO	Fractional Ms constrained integral gain optimization
DFOFTC	Distributed fractional-order finite-time control
DSMO	Distributed sliding-mode observer
AIS	Artificial immune system
IFM	Immune feedback mechanism
ALC	Artificial lymphocytes
CSA	Clonal selection algorithm
FS	Fuzzy systems
FOPDT	First order plus delay time
UAV	Unmanned aerial vehicle
UGV	Unmanned ground vehicle
ANN	Artificial neural networks
FFNN	Feed forward neural network
SRNN	Simple recurrent neural network
TDNN	Time delay neural network
FLNN	Functional link neural network
PUNN	product unit neural networks
CNN	Cascade neural network
LMS	Least mean squares
GD	Gradient descent
TTP	Three tank process

General introduction

Framework and context of the thesis

This thesis is endorsed by the MESRS in the context of a university training research project (PRFU no : A01L08UN240120220001).

This thesis has been carried out in the advanced control laboratory (LabCAV, https://labcav.univ-guelma.dz/fr) of the university 8 Mai 1945 Guelma. The LabCAV is led by Prof.Dr Djalil BOUDGEHEM and it is composed of various research groups : optimization modeling and identification (IMO), systems control, artificial vision and the FOS driven by Prof.Dr Baddredine BOUDJEHEM.

The fractional-order systems research team focuses on studding and analysing models based on fractional calculus (FC), proposing new class of fractional-order controllers and developing new techniques for their tunning, and last but not least improving the fractional-order approximation methods.

The FOS research team is one of the pioneers in this field with hundreds of publications in different prestigious scientific journals worldwide. Including but not limited to [1], [2] and [3].

This thesis is supervised by Prof.Dr Baddredine BOUDJEHEM and comes within the scope of improving the performance of fractional-order controllers when controlling industrial systems. Recently, FOC has gained a great attention by researchers, as it gives more degree of freedom to the controller, improves the performance of the control loop. it has been also proved in different publication that the fractional controllers performance are superior compared with their integer counterparts [4].

Objective of the thesis

This thesis is a continuation and extension of the research teamwork, which focus on enhancing and improving the performance of the fractional-order controllers by using CI part of AI.

The current trend is to develop hybrids of control theories. In doing so, we benefit from the strengths of each method since no one is superior in all situations.

In this thesis, we tried to enhance the performance of the FOPID with the use of AIS and FS, both part of CI. This type of controllers is very promising when controlling industrial systems.

Positioning in the literature

In recent years, the application of fractional calculus (FC) in science and engineering including but not limited to : control theory, signal and image processing, dynamic systems have become a new research area of interest [5] . FC is used in control system theory to build fractional-order controllers such as FOPI, FOPD and FOPID, the particularity of these controllers is the order of the integration and derivative which is real.

FOPID controllers have been used to control different type of processes such as : first order plus delay time (FOPDT) lag-dominant process - higher order process – integrating time-delay process - FOPDT delay-dominant process – poorly damped system – unmanned aerial vehicle (UAV) – unmanned ground vehicle (UGV) ….etc.

The focus of our state of the art will target time-delay systems in details and some different examples of important works covering the liquid level control in three tank process (TTP) system and finally providing an overview of principal works on modeling and control of both UAVs and UGVs which is considered as a trend in the current era of digitalization.

Researchers developed **different control structure** and used **different tuning method** for FOPID when controlling **time-delay systems**.

The first method among which the **indirect tuning methods** which require a process model [6] to develop a **FOC based on the classic feedback control structure**. We can cite here principal publications, the use of **Ziegler-Nichols** in [7], using **Hermite-Biehler** and **Pontryagin theorems** [8], **linear programming formulation** in [9], **F-MIGO** optimization in [10]. And other **direct tuning methods,** such as the tuning rules of FOPID based on the optimization of a performance criteria [11], a **tuning rules of FOPID based on the relay test** [12], new **Ziegler-Nichols approach** for fractional-order controllers in [13].

The second type of control structure is to use the **FOC in an internal model control (IMC) closed loop** [14] [15].

The third method is to use the **FOC in a Smith Predictor (SP) control structure** [16] [17] and [18].

The fourth method **is the combination between FOC and advanced control methods or advanced control strategies**. We can cite the use of **sliding mode control** in [19]. Our first contribution which concerns the control of time-delay systems falls into this category, we used the AIS as advanced control technique to enhance the performance of the FOC.

In another hand, we were interested also to liquid level control as it exists in many industrial applications such as chemical and pharmaceutical processes in the manufacturing environment, The TTP is considered as an interesting example to understand how the level control is tackled, it is a challenging task for automation engineers, as it may influence both pressure and flow of the process, subsequently it is important to maintain level at set point. In this work we attempt to control three-tank liquid-level, which is generally a large lag, time varying and complex system.

The three tanks can be coupled in different ways, in this work we are targeting a special connection, each tank is at different horizontal level, so the inflow rate into a tank is impacted solely by the liquid level of previous tank and the resistance offered by valve, the TTP is represented in **Figure 3.6**. The objective is to fill the tank number three as quickly and smoothly as possible according to a desired value, therefore, the process is a single input-single output process (SISO).

Researchers proposed in the literature different techniques to control the liquid level in three-tank system, especially those based on computational intelligence [20]. The use of hybrid version of evolutionary algorithm and swarm intelligence in [21], in which it has been shown the effectiveness of such bio-inspired techniques to control a three-tank delay system. The use of modern AI also is presented in [22] . Our second contribution uses the fuzzy logic in combination with FOC to solve the liquid level challenge in three tank process.

With the appearance of the **fourth industrial revolution** reflecting the rapid change of the technology due to the increasing of connectivity and the appearance of smart automation, advanced robotics, and the use of artificial intelligence in industrial control, many FC techniques developed to target the modeling and control of UAVs and AGVs, **several control problems** are presented in **UAVs** such as **trajectory tracking**, we can cite an application of **FOPID controller emulated by a neural network** in [23], FOBSMC approach in [24], FOSMC strategies in [25].The **path planning/collision avoidance** problem is also treated in [26]. using a FOPF method. the **attitude control** also has been resolved using different techniques based on FC, such as **the application of FOSMC** to control the attitude of a quadrotor in [27].

For the **state estimation**, a FOCF approach for the attitude estimation [28], **formation control** using **distributed fractional-order finite-time control** (DFOFTC) for a group of UAVs, the DSMO is designed to approximate the tracking error between the leader UAV and its reference, in addition the FOSMC is used to ensure the finite-time convergence of the tracking errors between followers and leader in [29] **fault tolerant control** using fractional-order fault tolerant control under external disturbance and actuator fault in [30]

The same control problems also are encountered in the UGVs for the **trajectory tracking**, using FOGPC in [31], for the **path planning/collision avoidance**, the FOPF has been used in [32].

Overall, the most control techniques applied on UAVs and UGVs are FOD, FOPI, FOPD, FOPI, FOPID, FOBSMC, FOSMC, FOPF, FOGPC, FOESC and FOTSMC.

In conclusion, according to the state of the art presented in this thesis, we found that the actual trend is to develop hybrids paradigms to benefit from the strength of each one, no methods can fulfil all the requirements, hence we have proposed the use of advanced control techniques combined with FOC. The FOC add additional degree of freedom in tuning the parameters of the controllers because the orders of the integro-differentiation are not restricted to integer, it also improves the robustness of the closed loop control under disturbances and improve the performance. The disadvantage is the high-order approximation required for the fractional operator's approximation in continuous or discrete time form.

The main contributions of this thesis are :

- Design of an intelligent controllers in a classical-feedback structure based on the combination of FOC and CI.
- Analysis of the performance of these type of controllers.
- Application of the new approach to control time-delay systems and liquid level in three tank process.

The results obtained have been the subject of the following publications :

Makhbouche, A.; Boudjehem, B.; Birs, I.; Muresan, C.I. Fractional-Order PID Controller Based on Immune Feedback Mechanism for Time-Delay Systems. Fractal Fract. 2023, 7, 53. https://doi.org/10.3390/fractalfract7010053

Thesis Organization

The chapters that make up this thesis are as follows:

Chapter 1 provides an overview of the core definitions of fractional operators including properties and present the fractional-order dynamic models with their analysis.

Chapter 2 provides an overview of most popular technological paradigms under the umbrella of CI part of AI.

Chapter 3 presents the design of a control law using the fuzzy fractional-order PID controller (FFPID), the proposed controller is used to control liquid level in three tank system, the control strategy has been validated by simulation in Matlab.

Chapter 4 provides an overview of the control theory of AIS.

Chapter 5 presents the design of a control law using the combination of FOC and AIS, the proposed controller has been used to control time-delay systems, The validity of the control strategy is proven by simulation in Matlab.

1. Control theory - fractional calculus

1.1 Introduction

The concept of fractional operator was introduced the first time in 17th century. It starts with letters sent by G.W. Leibniz in 1695 which mention the non-integer differentiation. It comes to a simple question that this scientist was asking at that time whether the order of differential operators $\frac{d^2}{dx}$, $\frac{d^3}{dx}$ can be non-integer such $\frac{d^{\sqrt{2}}}{dx^{\sqrt{2}}}$, $\frac{d^{\frac{1}{2}}}{dx^{\frac{1}{2}}}$ …..etc, followed by additional work from well-known mathematicians L. Euler , J.L. Lagrange , P.S. Laplace and J.B.J. Fourier.

N.H. Abel was the first scientist who provide an application of non-integer derivative to solve the isochrone problem, then J. Liouville who makes a significant study on FC by defining a fractional derivative as an infinite series expansion of a function.

Riemann proposed a definition of fractional integration by generalising Taylor series expansion, later, additional contributions made by A.K. Grunwald and A.V. Letnikov who generalize Cauchy integral formulas. We content ourselves in this thesis with the important steps in the history of FC, a detailed chronological bibliography for the FC has been summarized by Professor Bertram Ross in [33].

The power of FC is describing accurately objects and physical phenomena, the most well-known is the diffusive in transport processes which include but not limited to : velocity in fluids, heat in solids, electricity in resistive and capacitive lines…etc.

The application of FC is in many fields such as electrical circuits, chemical processes, signal processing and control engineering…etc. in this study we will focus on the application in control theory and automation which is the subject of this thesis, it has helped a lot to acquire more accurate models for systems dynamics and provide additional degree of freedom for controllers, it means we have to differentiate between fractional models and fractional controllers. Numerous techniques have been created for the identification of fractional dynamics based on the approximation of the fractional operators with rational ratio, a resume of continuous and discreate time approximations can be found in [34].

Different numerical tools for identification and control among which the most important: MATLAB FOMCON ("Fractional-order Modeling and Control") toolbox in [35], MATLAB Ninteger toolbox in [36]. In term of fractional controllers, the FOPID controller has been proposed by [37].

Researchers predict that the field of FC will have a significant impact on control engineering, by providing more accuracy, and precision hence better performance.

1.2 Fractional calculus (FC)

The term differ-integration refers to differentiation or integration. The fundamental fractional operator of derivatives and integrals is ${}_aD_t^\alpha$, where a and t are the limits of the operation. $\alpha \in \mathbb{R}$ denotes the order of the differ-integration; derivatives (for a positive α) and integrals (for a negative α).

The fractional operator definition is as follows :

$$ {}_aD_t^\alpha = \begin{cases} \frac{d^\alpha}{dt^\alpha}, & \alpha > 0 \\ 1, & \alpha = 0 \\ \int_a^t (d\tau)^\alpha, & \alpha < 0 \end{cases} \quad (1.1) $$

It also exists different mathematical definition of fractional-order differ-integration approximated by many techniques. Riemann-Liouville (e.g., in calculus), Grunwald-Letnikov (e.g., Communications and control) & Caputo definitions are the most used.

The GL, RL and Caputo definitions are equivalent under some conditions for a wide class of functions.

1.2.1 Grünwald-Letnikov (G-L) definition

For integer order n, $\frac{d^n f}{[dx]^n}$ which is the nth derivative of the function f (abbreviated by the symbol $f^{(n)}$ for differentiation), we obtain the following :

$$ \frac{d^n}{dt^n} f(t) \equiv f^{(n)}(t) = \lim_{h\to 0} \frac{1}{h^n} \sum_{j=0}^{n} (-1)^j \binom{n}{j} f(t - jh) \quad (1.2) $$

Where $\binom{n}{j} = \frac{n!}{j!(n-j)!}$

The Grünwald-Letnikov definition extends the previous equation to fractional-orders and the corresponding α^{th} represent the differ-integration order of a function $f(t)$ which is described as :

$$ D_t^\alpha f(t) \coloneqq \lim_{h\to 0} \frac{1}{h^\alpha} \sum_{j=0}^{\infty} (-1)^j \binom{\alpha}{j} f(t - jh) \quad (1.3) $$

Oppositely to an integer value n, for a fractional value α , the factorial expression $\binom{n}{j}$ can be replaced by the Euler's Gamma function :

$$ \binom{n}{j} = \frac{\gamma!}{j!(\gamma-j)!} = \frac{\Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)} \quad (1.4) $$

where $n = \frac{t-a}{h}$, and a is a real constant.

The differ-integration is described as :

$$ {}_aD_t^\alpha f(t) = \lim_{h\to 0} h^{-\alpha} \sum_{j=0}^{\left[\frac{t-a}{h}\right]} (-1)^j \binom{\alpha}{j} f(t - jh), \quad (1.5) $$

Where a and t are the limits of operator and h is the computation step size.

1.2.2 Riemann-Liouville (R-L) definition

The R-L fractional integral definition can be expressed as :

$$ {}_0D_t^{-\alpha}f(t) = \frac{1}{\Gamma(a)}\int_0^t \frac{f(\tau)}{(t-\tau)^{1-\alpha}}d\tau \; \forall \; \alpha \in (0,1), t \in (0,\infty) \quad (1.6)$$

and the fractional-order derivative is as follows :

$$ {}_0D_t^{\alpha}f(t) = \frac{1}{\Gamma(\mathrm{n}-\alpha)}\frac{d^n}{dt^n}\int_0^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}}d\tau \quad (1.7)$$

When $\alpha \in (0,1)$ and $f(t)$ is causal (i.e. f (t) = 0 ∀ t < 0).

1.2.3 Caputo definition

According to Caputo's definition the α^{th} order derivative of a function $f(t)$ with respect to time is as follows :

$$ {}_0D_t^{\alpha}f(t) = \frac{1}{\Gamma(\mathrm{n}-\alpha)}\int_\alpha^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha+1-n}}d\tau, \;\; for\; n-1 \le \alpha < n \quad (1.8)$$

1.3 Properties of fractional-order differentiation

1.3.1 Laplace transform

Laplace transform is an important tool in science and engineering, as it helps to solve differential equation, this transformation converts a function of a real variable (t) in the time domain to a function of a complex variable (s) in the frequency domain.

For a function $f(t)$ the Laplace transform $F(s)$ can be written as follows :

$$F(s) = \mathcal{L}[f(t)] = \int_0^\infty f(t)e^{-st}dt \quad (1.9)$$

1.3.2 Linearisation

We assume that a and b are constant then :

$$ {}_0D_t^{\alpha}[af(t) + bg(t)] = a\; {}_0D_t^{\alpha}f(t) + b\; {}_0D_t^{\alpha}g(t) \quad (1.10)$$

1.3.3 Identity operator

$$\text{When } \alpha = 0\text{, then, } {}_0D_t^{0}f(t) = f(t) \quad (1.11)$$

1.3.4 Miscellaneous

If $\alpha = n$ and $n \in \mathbb{Z}_+$(positive integer), then the operator ${}_0D_t^{\alpha}$ will be the classical derivative $\frac{d^n}{dt^n}$

1.4 Fractional-order models

Fractional-order continuous time dynamic systems can be expressed using fractional-order differential equation :

$$\sum_{k=0}^{n} a_k \mathfrak{D}^{kq}y(t) = \sum_{k=0}^{m} b_k \mathfrak{D}^{kq}u(t) \quad (1.12)$$

The system is considered as commensurate order, if in the equation (1.12), all the orders of derivative are integer multiple of a base order q, $a_k, b_k = kq$, where $q \in \mathbb{R}^+$(positive real numbers).

The system is considered as rational order, if in the equation (1.12), the order is $q = 1/r$, $r \in \mathbb{Z}_+$ (positive integers numbers).

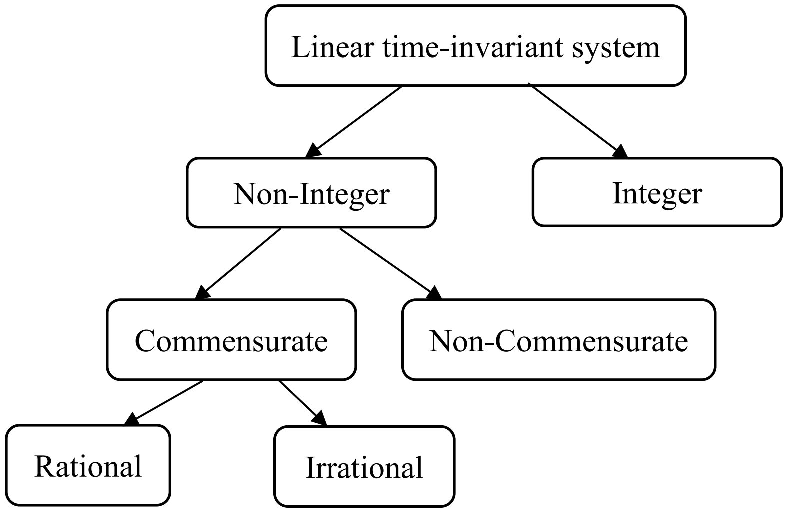

Figure 1.1 Linear time-invariant (LTI) system classification

To obtain the transfer function of the fractional-order system, we apply Laplace transform with Zero initial conditions :

$$G(s) = \frac{Y(S)}{U(s)} = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_0 s^{\alpha_0}} \tag{1.13}$$

1.4.1 Stability analysis

The fractional transfer function $G(s) = Z(s)/P(s)$ is stable if and only if the following condition is satisfied in σ-plane [38] :

$$|\arg(\sigma)| > q\frac{\pi}{2}, \forall \sigma \in C, P(\sigma) = 0 \tag{1.14}$$

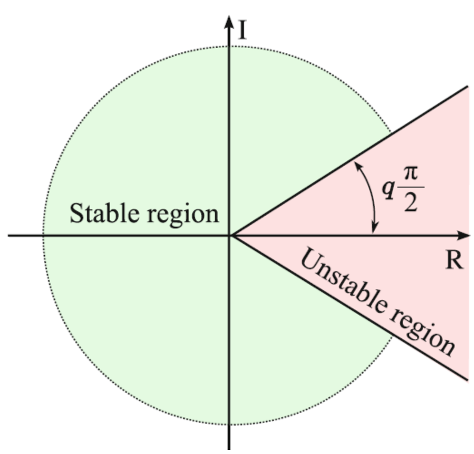

Figure 1.2 Stability regions of a FOS for $0 < q \leq 1, \sigma \coloneqq s^q$

1.4.2 Time domain analysis

The numerical solution for the fractional-order continuous time dynamic systems is obtain using the following equations :

$$\mathcal{y}(t) = \frac{1}{\sum_{i=0}^{n} \frac{a_i}{h^{\alpha i}}} \left[\hat{u}(t) - \sum_{i=0}^{n} \frac{a_i}{h^{\alpha i}} \sum_{j=1}^{\left[\frac{t-a}{h}\right]} (-1)^j \binom{\alpha}{j} \mathcal{y}(t - jh)\right] \quad (1.15)$$

Where

$$\hat{u}(t) = b_m \mathfrak{D}^{\beta_m} u(t) + b_{m-1} \mathfrak{D}^{\beta_{m-1}} u(t) + \cdots + b_0 \mathfrak{D}^{\beta_0} u(t) \quad (1.16)$$

1.4.3 Frequency domain analysis

The frequency domain is obtained when we substitute the $s = jw$, where j is the imaginary unit and $w = (0, \infty)$

The fractional power of the imaginary unit is as follows :

$$j^\alpha = \cos\left(\frac{\alpha\pi}{2}\right) + j \sin\left(\frac{\alpha\pi}{2}\right) \quad \alpha \in \mathbb{R} \quad (1.17)$$

The complex response is as follows :

$$G(jw) = \frac{Y(jw)}{U(jw)} = \frac{b_m (jw)^{\beta m} + b_{m-1} (jw)^{\beta m-1} + \cdots + b_0 (jw)^{\beta_0}}{a_n (jw)^{\alpha n} + a_{n-1} (jw)^{\alpha n-1} + \cdots + a_0 (jw)^{\alpha_0}} \quad (1.18)$$

1.5 Approximation of fractional-order operators (Oustaloup's recursive filter)

The approximation of fractional differ integration of order α in a specified frequency range (ω_b,ω_h) by an integer-order transfer function is done by computing the poles and zeros as follows [39] :

$$s^\alpha \approx K \prod_{k=1}^{N} \frac{s + \omega_k'}{s + \omega_k} \quad (1.19)$$

Where ω_k' , ω_k and K can be evaluated by the following :

$$(Zero\ of\ rank\ k)\ \omega_k' = \omega_b \cdot \omega_u^{\frac{2k-1-\alpha}{N}}$$

$$(Pole\ of\ rank\ k)\ \omega_k = w_b \cdot \omega_u^{\frac{2k-1+\alpha}{N}}$$

$$K = \omega_h^\alpha, \qquad \omega_u = \sqrt{\omega_h / \omega_b}$$

And N is the order of approximation in the specified frequency range (ω_b,ω_h).

Different approximations methods have been developed are resumed in [34]

1.6 FOPID controller

The FOPID controller is a generalization of the classical PID controller. The transfer function of the parallel form of FOPID controller can be described in Laplace domain according to [37] :

$$C_{FOPID} = \frac{U(s)}{E(s)} = K_p + K_i s^{-\lambda} + K_d s^\mu; (\lambda, \mu > 0) \quad (1.20)$$

λ and μ are the fractional-orders of integration and differentiation.

$K_p, K_i, and\ K_d$ are the proportional, integral, and derivative gains.

When $\lambda = \mu = 1$ the controller becomes a classical integer-order PID.

The result in (2.20) can be used to establish the time-domain of the control signal of a FOPID :

$$u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^{\mu} e(t) \quad (1.21)$$

The block diagram of the FOPID is indicated in **Figure 1.3**

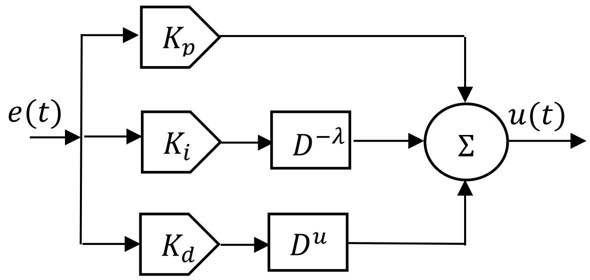

Figure 1.3 Conceptual diagram of a FOPID controller

1.6.1 Optimization based controller

Several aspects need to be considered to optimize the FOPID controller :

- **Type of the plant**: high-order, time delay, non-minimum-phase, unstable, nonlinear…etc. We studied in this work the control of time-delay processes.
- **Optimization criteria**: control system performance in time domain is usually ensured by the minimization of the error signal $e(t)$ which is the difference between the reference signal and the output, hence several criteria are used, ITAE (Integral time-absolute error), ITSE (Integral time-square error), IAE (Integral absolute error), ISE (Integral square error). In our case we used ITAE.
- **Initialization of parameters**: Ziegler Nichols.
- **FOPID controller structure**: classical structure – Smith structure…. etc
- **Parameters to be optimized**: K_p, K_i, K_d, λ *and* μ…. Etc
- **Frequency specifications**: to study the performance in terms of stability and robustness based on the analysis of the open loop frequency response $F(jw) = C(j\omega) \cdot G_p(j\omega)$, where $C(jw)$ is the FOPID controller and $G_p(j\omega)$ is the transfer function of the process.
- The following specifications need to be taken in consideration for the open loop frequency response :

  a. **Gain margin** A_m

$A_m = 1 - |F(j\omega_g)|, \qquad \arg\left(F(j\omega_g)\right) = -\pi$

  b. **Phase margin** φ_m and **gain crossover frequency** ω_c

$\arg\left(F(j\omega_g)\right) = -\pi + \varphi_m, \qquad |F(j\omega_g)| = 1$

  c. **Robustness** to gain variations of the plant :

$$\left.\frac{d\arg\left(F(j\omega_g)\right)}{d\omega}\right|_{\omega=\omega_c} = 0$$

d. **High-frequency noise rejection** with noise attenuation of A dB for the complementary sensitivity function

$$\left|T(j\omega) = \frac{F(j\omega)}{1+F(j\omega)}\right|_{dB} \leq A\ dB \text{ for all frequencies } \omega \gg \omega_t \text{ rad/s.}$$

e. **Disturbance rejection** with a constraint B dB on the sensitivity function

$$\left|S(j\omega) = \frac{1}{1+F(j\omega)}\right|_{dB} \leq B\ dB \text{ for all frequencies } \omega \gg \omega_s \text{ rad/s.}$$

The problem of optimization can be then formulated as follows :

$$\min_{\theta_c} J_c(\cdot)$$

Where $J_c(\cdot)$ is the join cost function on desired system performance which includes specifications for both time and frequency domain and θ_c is the FOPID controller parameters and it is equivalent to :

$$\theta_c = K_p, K_i, K_d, \lambda \text{ } and \text{ } \mu$$

We might have also some constrains on the control signal due to physical equipment limitation :

$$u(t) \in [u_{min}; u_{max}]$$

These constraints are added to the cost function by a penalty function formed by a weighted sum of deviations of design specifications in the frequency domain.

2. Essentials of computational intelligence (CI)

2.1 Introduction

CI is a subdivide of AI, which covers the study of adaptive mechanisms and the ability to learn and/or adapt to new situations which allows the creation of intelligent behaviour. The IEEE computational intelligence society (CIS) cite the following core theories/methods or paradigms used in this field : EC, AIS , FS and ANN. The following **Figure 2.1**
presents an overview of the CI subfields.

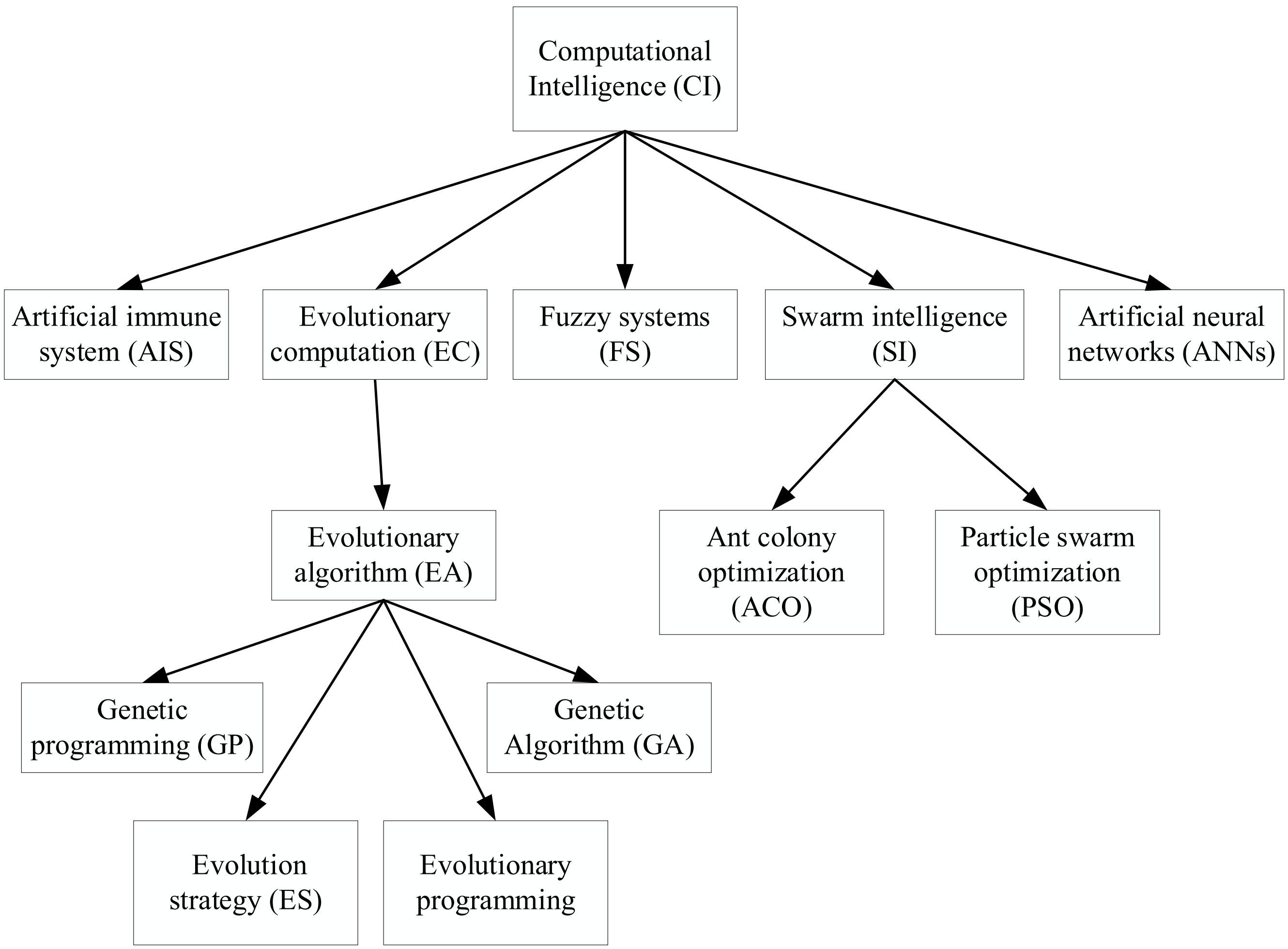

Figure 2.1 Subcategories of computational intelligence

A lot of these methods usually involve metaheuristic optimization algorithms. In the scientific study, metaheuristics are more general and problem-independent, hence can be used to a wide range of optimizations. Metaheuristics are high level algorithmic structure aimed to develop mathematical optimization algorithms used to solve a variety of problems that involves non linearities, non-convexity, large dimension…etc for which classical optimization methods are ineffective or not applicable.

Metaheuristics are iterative and most of them use stochastic operations in their search to find a global optimum. The goal is to use iterations to pass from a solution of poor quality towards an approximate optimal solution in a logical amount of time.

In the real life, when we think which investments can maximize our benefits or what is the best road to follow when travelling is an example of solving optimization problems. We can than conclude that the metaheuristics have been used long time ago before the formal scientific study in the 20th century.

The following **Figure 2.2** shows the important chorological metaheuristics methods.

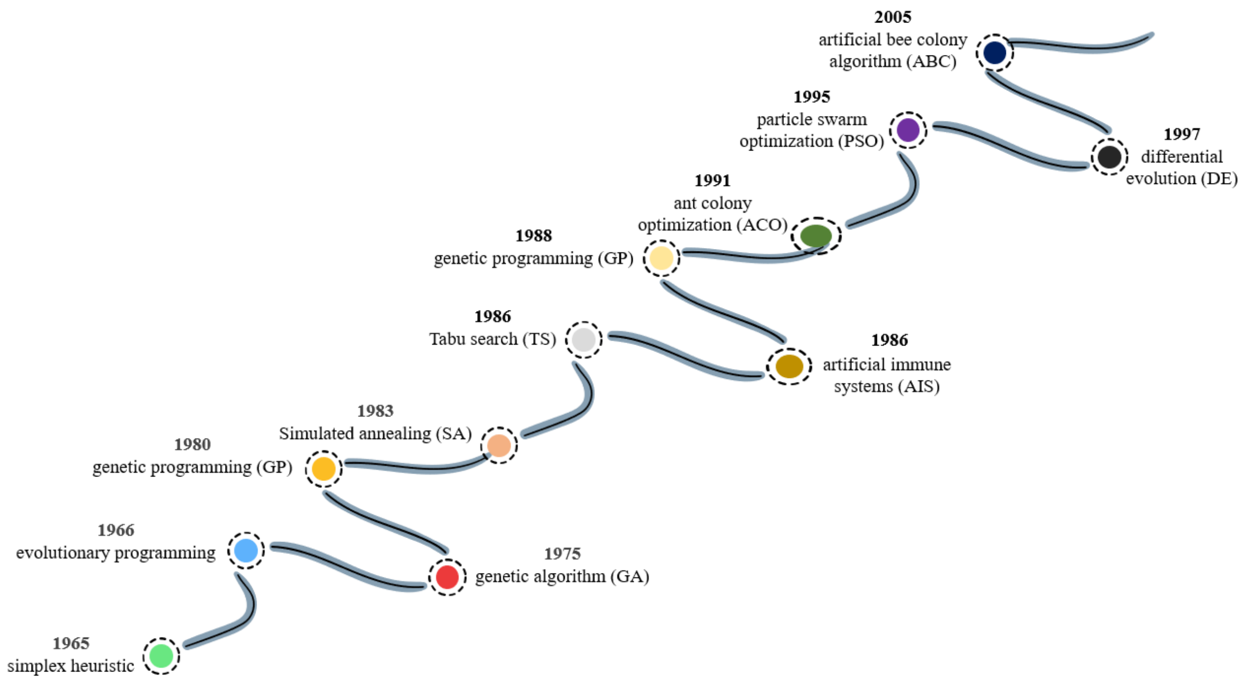

Figure 2.2 Metaheuristics methods

Metaheuristics can be classified in many ways: It can be classified as population based or trajectory based, and to nature inspired or non-nature inspired.

Metaheuristics are also often developed based on metaphors for inspiration, most of them are taken from biology: evolutionary algorithms, swarm intelligence and AIS…etc

2.2 Evolutionary computation (EC)

EC is a subdivide of CI inspired from the phenomenon of natural biological evolution theory developed by [40], it uses the models of the evolution process such as the survival of the fittest and the remove the weak. It is considered as an optimization process which consists of improving the capacity of a system to survive in a dynamical changing environment, EC is used in different applications such as classification, clustering, faut diagnosis, and time series approximations.

The EC differs from the classical optimization (CO) on the search process level, search surface, and the problems where both two methods are shown efficient. The following figures presents the EC vs. CO.

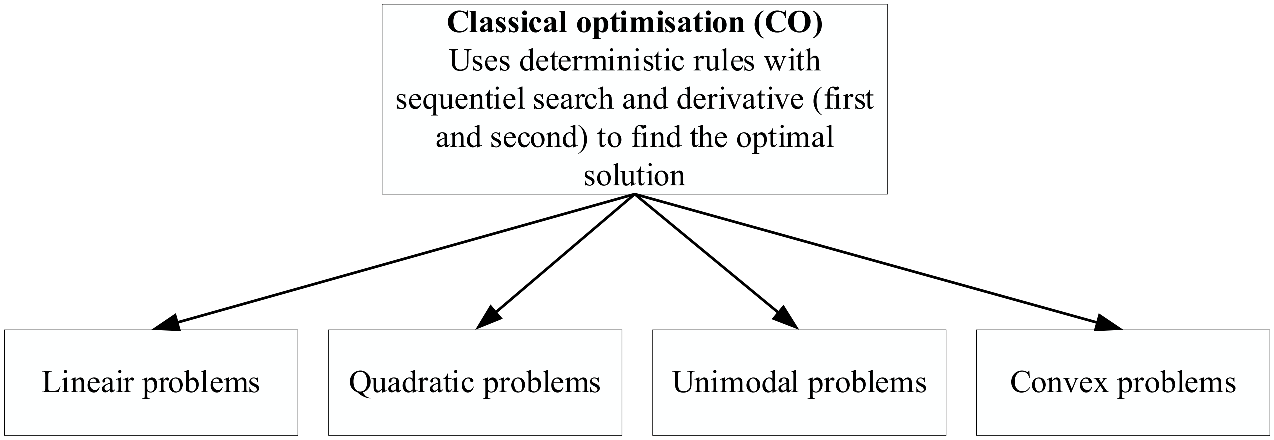

Figure 2.3 Classical optimization

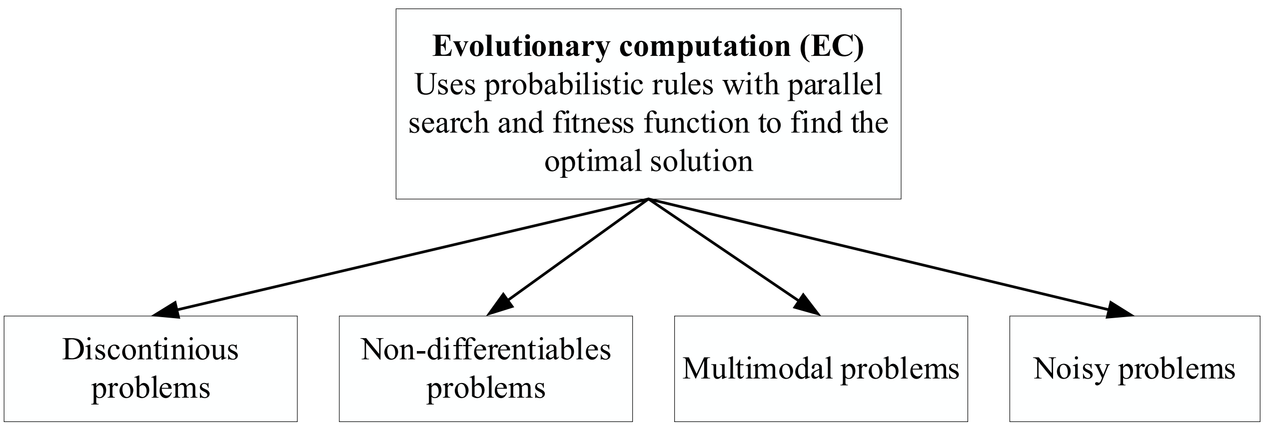

Figure 2.4 Evolutionary computation

The evolutionary algorithms (EA) are stochastic search which use a population of individuals. The characteristics (gene) of individuals can be coded as chromosomes also called genome. The value of a gene is called *Allèle*. In the evolution process, a mathematical fitness function or the objective function is used to determine the individuals that have best characteristics thereby can survive and reproduce. The objective function describes the optimization problem, various type of optimizations can be encounter : unconstrained, constrained, multi-objective and dynamic. **Figure 2.5**

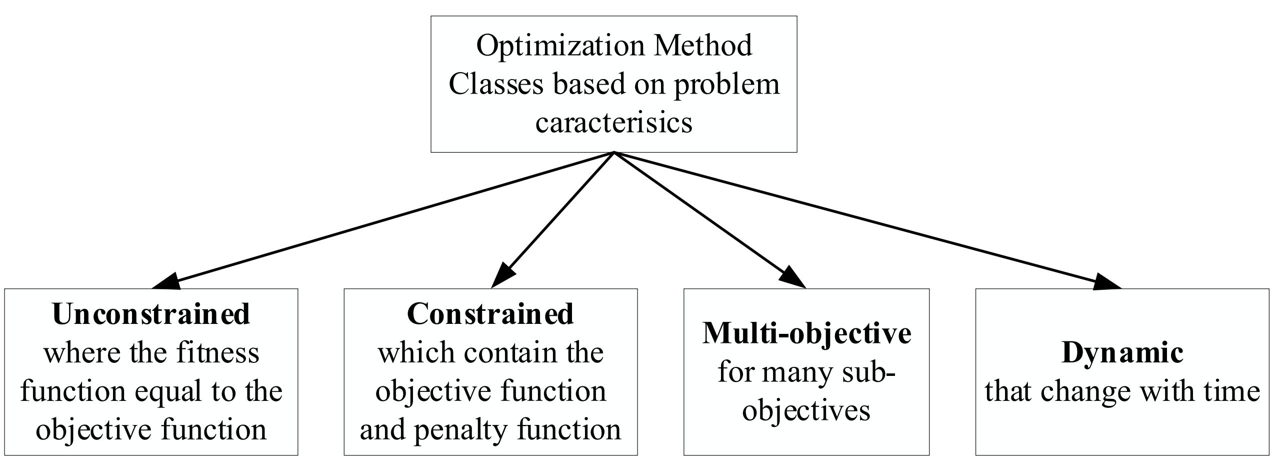

Figure 2.5 Optimization problems

The general structure of EA links stages of selection, crossover (offspring which combines parts of the parent) and mutation which impact the *allèle* of the chromosome of population of individuals.

A generic component of EA is composed of initialisation of population, encoding the solution, using a fitness function to evaluate the strength of individuals, selection and reproduction of operators (crossover and mutation). various implementation of these components leads to different methods of EC, we can cite, EP, GP, GA, ES…etc

We have chosen one example to be discussed in more details, genetic algorithms which have been used on this thesis to tune the parameters of the proposed controller.

2.2.1 Generic operators of EA

a. The **initialization** process consists of generating randomly a new population composed of few/thousands of individuals which represent the solution. The size of the population has an impact on the exploration capabilities and on the time convergence.

b. The **selection** process determine which individuals can achieve the best results, in another term select the best solution in the population based on the fitness related to the objective function of the problem. Different types of selection have been used.

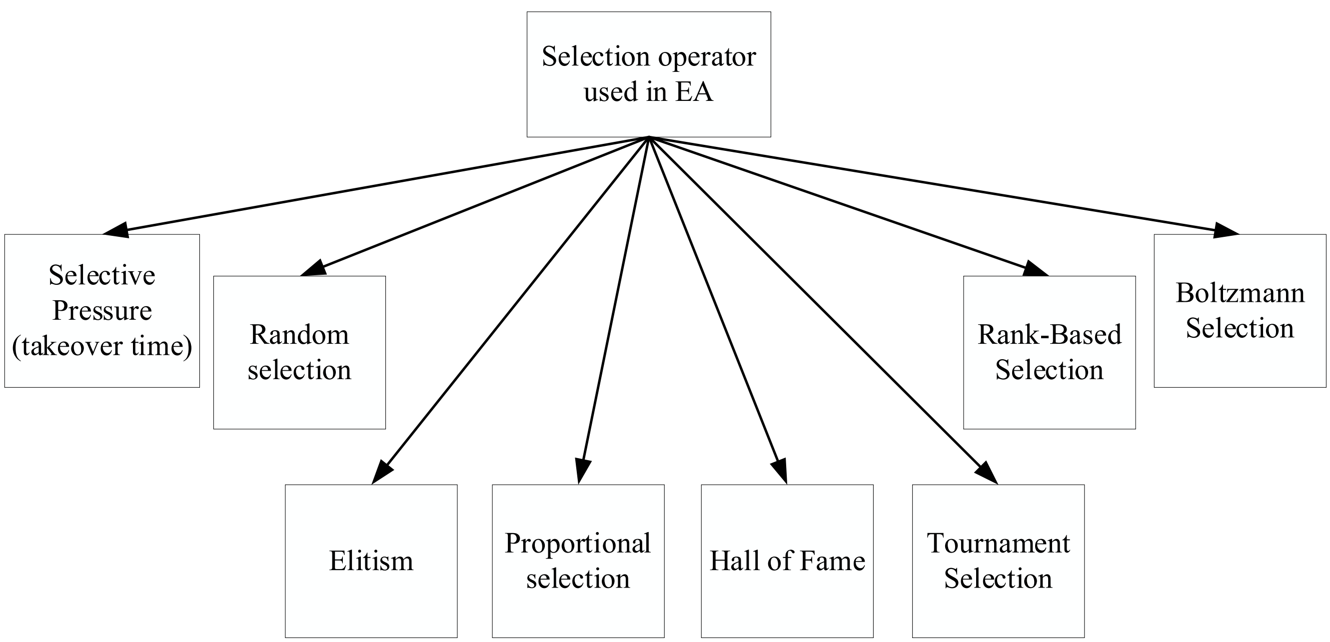

Figure 2.6 Selection types used in EA

<u>Producing of offspring</u>

The producing of offspring is the process of applying mutation and crossover from selected parents.

c. The **cross-over** process is the creation of new population by switching of some parts of chromosome from the parents.

d. The **mutation** process is a random change of the value of genes (random flipping of some digits of the string) to generate new solutions. It is important to choose a low value, to avoid the risk of falling into a random search and preserve the good genetics from the principle of selection. it serves also keep away from local convergence of the algorithm. E.g., when searching for an extremum, the mutation avoids the convergence to a local optimum. We can then resume that the cross-over keep the solution in the same subspace whereas the mutation gives the possibility to increase the diversity.

EA can be stopped using different stopping conditions such as : convergence point when no improvement is made, and it is stable. When the solution is found based on the fitness function or when the execution time is finished (number of generations reaches the limit).

2.2.2 Genetic algorithm (GA)

The start of this field comes back to the pioneer [41]. GA are metaheuristics part of evolutionary algorithms inspired from the biologic theory of evolution. The genetic algorithms mimic the process of genetic evolution, it uses three main operators; selection of the fittest, cross-over to model reproduction and mutation to increase the search space and add diversity. **Figure 2.7**

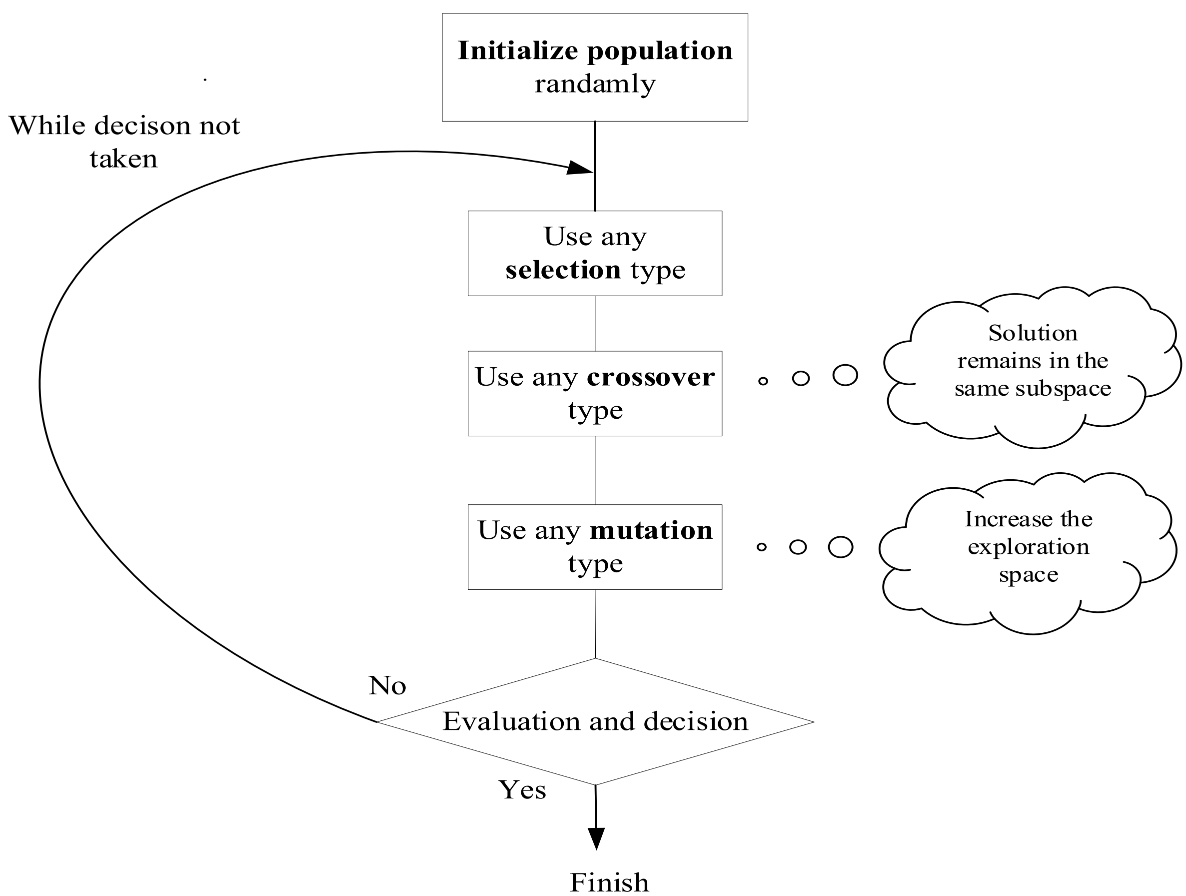

Figure 2.7 Summary diagram of GA

The theoretical computation time of GA is $nln(n)$ where n is the number of variables. In GA, a binary or decimal string is codified as a solution to mimic the chromosome.

Different types of crossovers and mutation used in GA are summarized in the following figures.

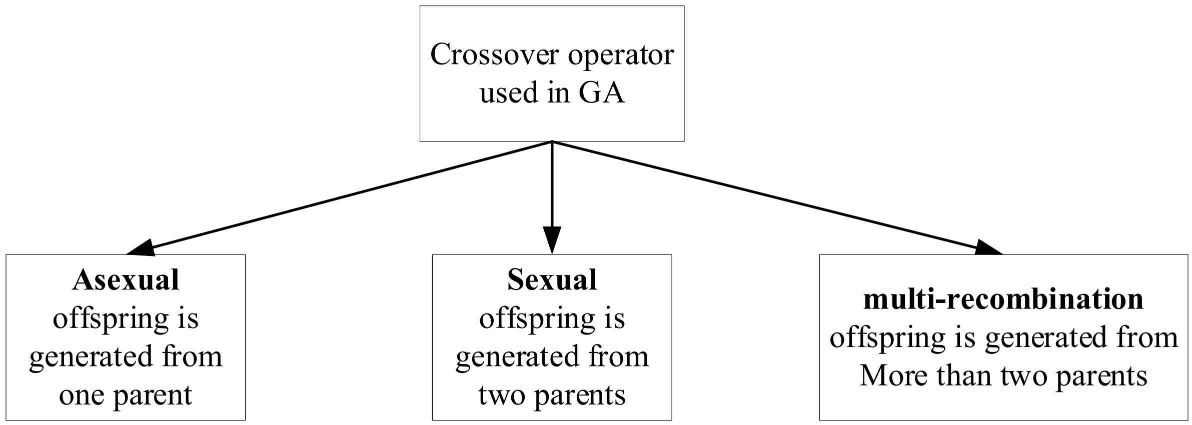

Figure 2.8 Crossover types used in GA

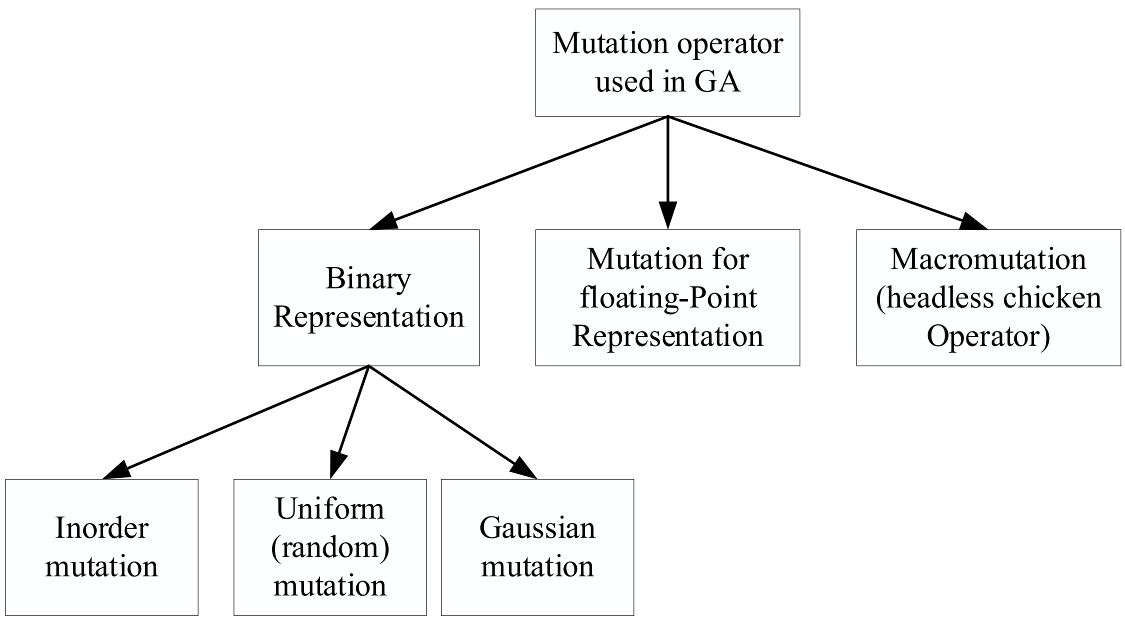

Figure 2.9 Mutation types used in GA

2.3 The swarm intelligence (SI)

The SI referred also as collective intelligence, it is based on social behaviour when living in swarms or colonies. The collaboration of individuals with each other and the interaction between simple individuals can allow the creation of complex, collective and coherent behaviours called emergence can solve problems in an efficient way compared with single and simple individual.

The objective of SI is to model the single behaviour and the local interaction with its environments and with its neighbours to obtain complex behaviours capable of solving problems, case of PSO and ACO.

We have chosen one example to be discussed in more details, particular swarm optimization PSO from swarm intelligence which can be used as a perspective work.

2.3.1 Particle swarm optimization (PSO)

This stochastic optimization metaheuristics developed by [42] is based on the collaboration of individuals with each other, The idea holds that a group of individuals with simple behaviour can possess a complex global organization. The search is based on population procedure.

At the start of the algorithm, each particle (individual) is randomly positioned in a multidimensional search space of the problem. Each iteration pushes the particles to fly (move) their position according to both their own experience and their neighbours to achieve the optimal solution. This gives the following equation of position :

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2.1}$$

- ✓ $x_i(t)$ is the current position of particle i at discreet time t
- ✓ $v_i(t)$ is the current velocity of particle i at discreet time t

The performance of particle is measured by a *fitness function* related to the problem.

There are two important types of algorithms in PSO called global best PSO (gbest PSO) and local best PSO (lbest PSO). gbest PSO use the star social structure, this connected structure is more used for unimodal problems whereas the lbest PSO use the ring social topology which considered as less connected structure used for multi-modal problems. The difference resides also in the size of their neighbours. The neighbours for gbest PSO are the entire swarm while the neighbours for the lbest PSO is a ring social topology. Both algorithms are summarized in [43]

For gbest PSO, $v_{ij}(t)$ is the velocity of particle i in dimension j at discreet time t

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)] \tag{2.2}$$

- ✓ $x_i(t)$ is the position of particle i in dimension j at discreet time t
- ✓ c_1 and c_2 are positive acceleration coefficients.
- ✓ r_1 and r_2 are random vectors that introduce stochastics in the algorithm.
- ✓ $\hat{y}_j$ is the best personal position of particle i since the first-time step.

For lbest PSO, $v_{ij}(t)$ is the velocity of particle i in dimension j at discreet time t

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)] \tag{2.3}$$

- ✓ $x_i(t)$ is the position of particle i in dimension j at discreet time t
- ✓ c_1 and c_2 are positive acceleration coefficients.

- ✓ r_1 and r_2 are random vectors that introduce stochastics in the algorithm.
- ✓ $\hat{y}_j$ is the best position found by the neighbourhood of particle i

An important aspect in optimization algorithms is the exploration and exploitation features which reflects the accuracy and the efficiency of an algorithm. exploration is the capacity of searching in the whole search-space while the exploitation is the concentration around a candidate solution, these features are expressed in the PSO algorithm by the velocity update equation.

2.3.2 PSO parameters

The PSO algorithm is influenced by different parameters : dimension of the problem, number of particles, number of iterations, acceleration coefficients and neighbourhood size.

a. **Number of particles** : a greater number of particles allow to cover large search space however added more computational complexity, the best option is to optimize the swarm size for each problem using cross-validation approaches.

b. **Neighbourhood size** : More neighbourhood increase the interaction in the swarm. The empirical studies show that the best approach is to start with small neighbourhood size for better exploration and then increase proportionally to number of iterations for fast convergence of highly connected structure.

c. **Number of iterations** : it is a problem dependant solution. Few iterations can lead to premature convergence and large number can add computational complexity.

d. **Acceleration constants** : these coefficients used to scale the cognitive and social component in the velocity equation.

2.4 The artificial immune system (AIS)

An AIS is a category of algorithm inspired from the natural immune system (SIN). These algorithms use the characteristics of the immune system in terms of learning and memory for problem solving. it has a strong ability to distinguish between foreign cells invading the body referred to as antigen or non-self and the cells which belongs to the body (Self cells).

The principals theories in natural immune system are:

a. **Classical view** : immune system differentiate between self and non-self, uses lymphocytes which learn to bin the antigen.
b. **Clonal selection theory** : B-cells produces antibodies using a cloning process.
c. **Danger theory** : The immune system can differentiate between dangerous and non-dangerous invaders (antigens)
d. **Network theory** : B-cells are interconnected to form a network which is used to eliminate the antigens.

The artificial immune system are computational models/algorithms inspired from these theories, case of negative selection algorithm [44] inspired from the negative selection of T cells and , CLONALG algorithm [45] inspired from the clonal selection theory of B-cells.

2.5 Fuzzy systems (FS)

FS allow an approximate reasoning, contrary to the binary exact systems used in computers which requires to be either 1 or 0, it has been shown that not all the problems can be resolved using an exact system of two-valued theory, and most of real problems are characterized by uncertainties and incomplete information, fuzzy systems gives the ability to use probability reasoning to solve these problems, the uncertainty in fuzzy logic is considered as non-statistical uncertainty (fuzziness), fuzzy systems have seen extensive use in control engineering. The first remarkable work comes back to Lotfi Zadeh the developer of the fuzzy sets in (Zadeh, 1965)

2.5.1 Fuzzy sets

Fuzzy sets are an extension of two-valued set theory which use the concept of partially truth, and the characteristics of the fuzzy set is described using a membership function, usually it is defined by human expert in the process.

2.6 Artificial neural network (ANN)

ANN were inspired from the human brain modeling. It is composed of around 10-500 billion neurons. An artificial neural network is a model of the biological neurons, the most remarkable work was the perceptron artificial neural network developed by [47] and the Adaline artificial neural network by [48]

A single neurone can be used to learn linear separable function, but for more complex functions a layered neuron network is required.

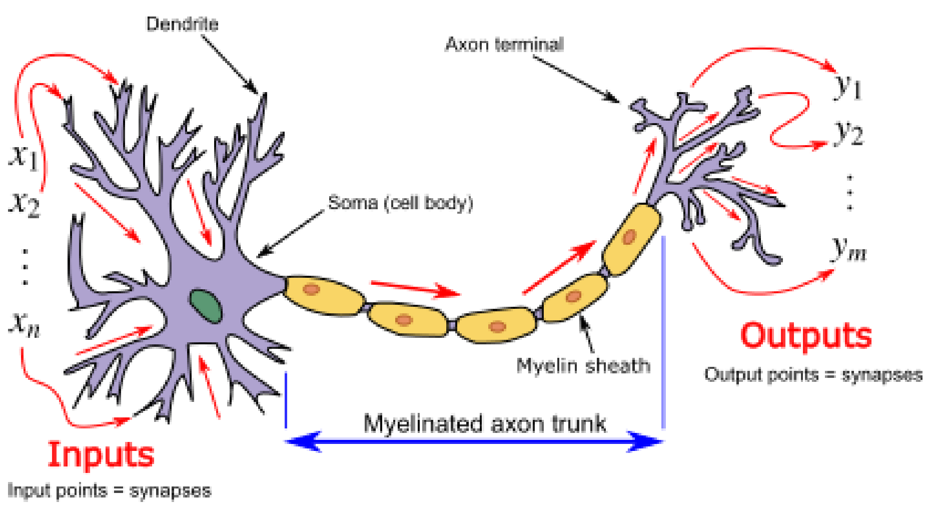

Figure 2.10 Biological neuron model

Source : https://en.wikipedia.org/wiki/Biological_neuron_model

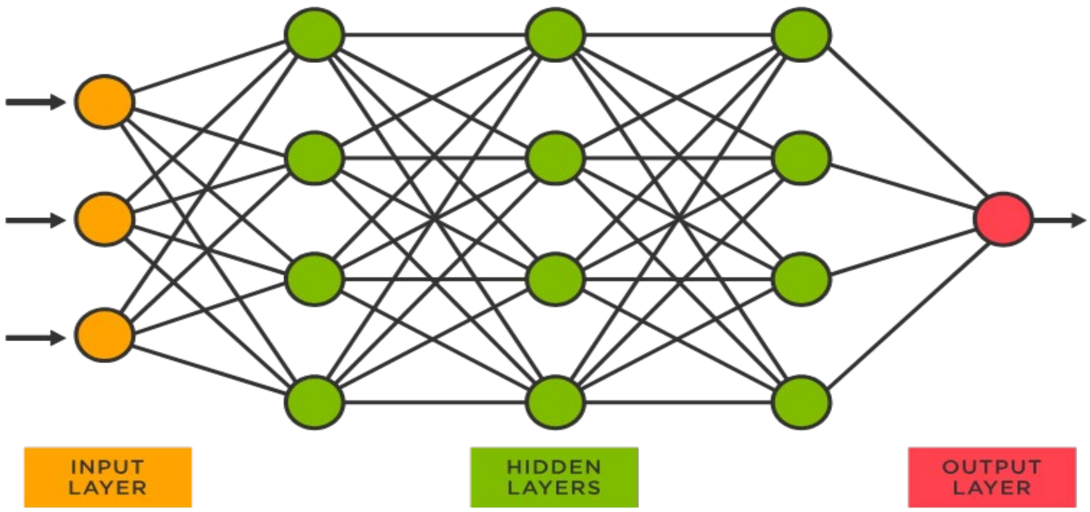

Figure 2.11 Artificial neuron network

ANN have been used in various applications such as classification, pattern matching, optimization, control, times series modeling, function approximation…etc

2.6.1 Training rules and learning types for ANN

An ANN is characterized by its threshold and wights vector, the automated method which consists of determining the values of both wights and threshold is referred to as learning, the principal types of learning are, supervised provided with a training set (input vector and a desired output vector) able to learn under the guidance of a teacher, unsupervised aimed to discover input features without any data set assistance or without any guidance and reinforcement learning able to award the neuron for correct actions and penalize/punish wrong actions. Various training rules have been developed for these learning types.

The **Figure 2.12** provides an overview of the artificial neural learning methods and their subcategories.

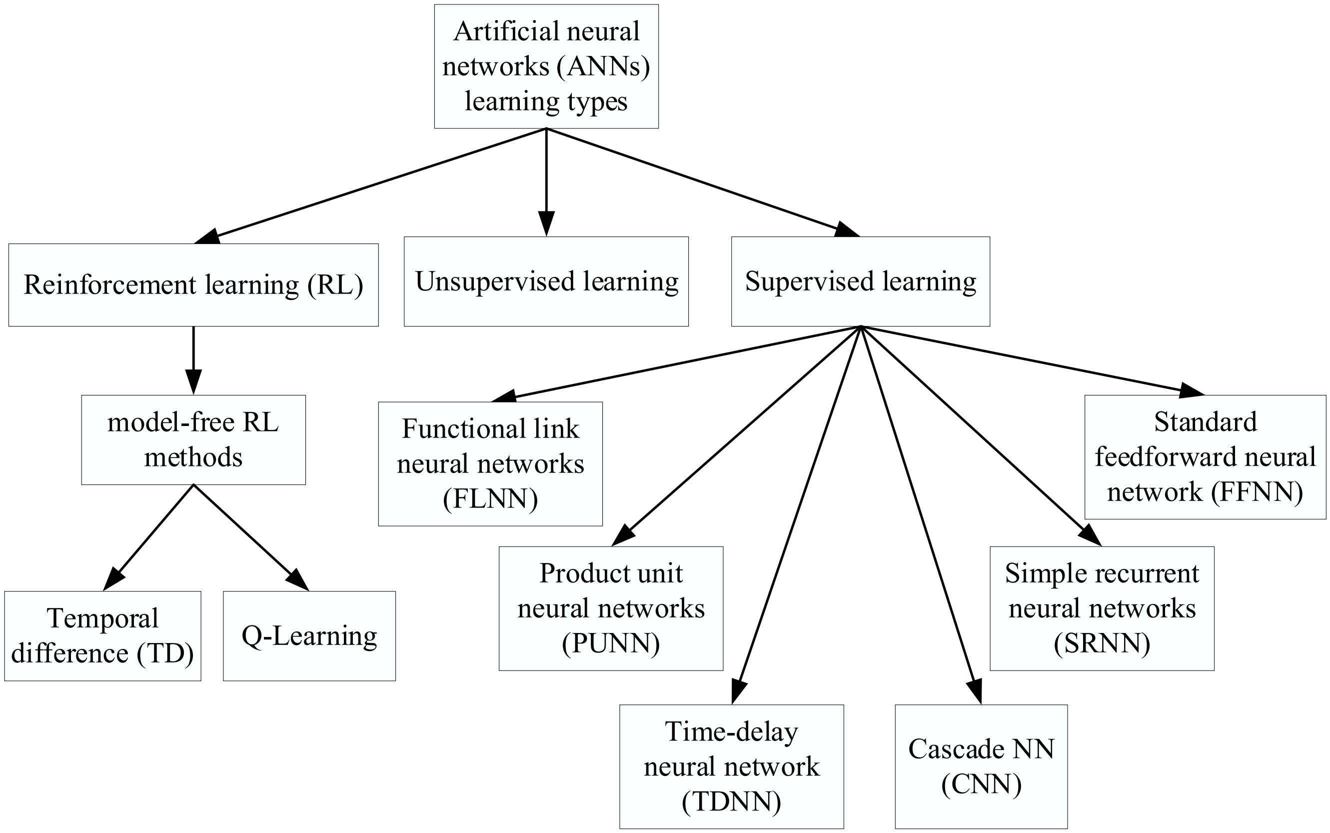

Figure 2.12 ANN learning types

The **Figure 2.13** provides the different types of learning rules used in ANN.

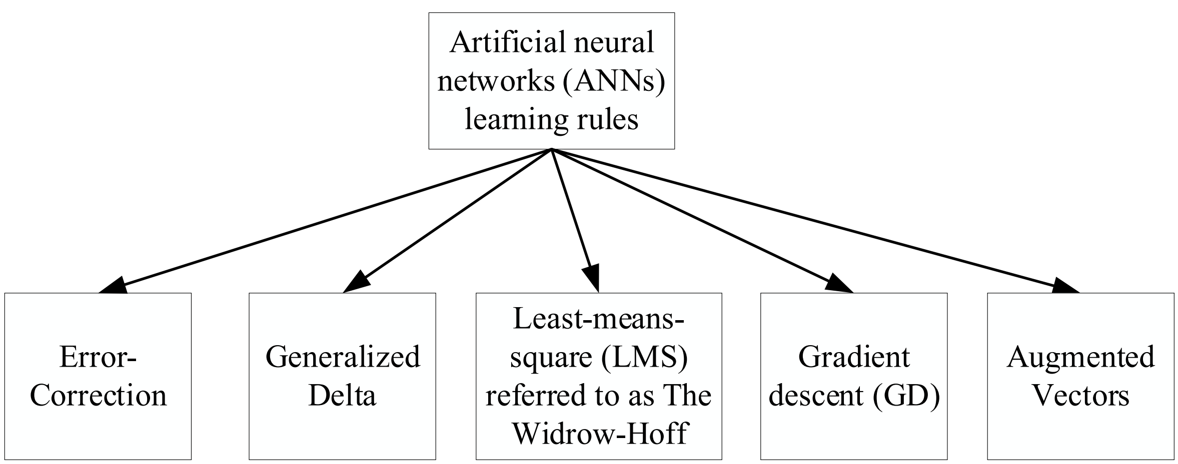

Figure 2.13 ANN learning rules

3. Design of fuzzy fractional PID controller - applied to control three tank system

3.1 Introduction

Fuzzy logic has been used in various fields, due its ability to convert the expert's control knowledge based in human perception to a rule base and fuzzy variable (linguistic variable instead of number). FS is a system that can provide approximate reasoning, it is composed of fuzzification, defuzzification, inferencing and fuzzy knowledge base (both fuzzy sets and rules).

The fuzzification is the process of application of any types of membership functions (example but not limited to : triangular, trapezoidal and exponential functions) to find a fuzzy representation of an input. The inferencing process is the second step which consists of the mapping of the result of fuzzification to the rule base to provide a fuzzy output. A defuzzification is the final process of converting the fuzzy output to a scalar value using different methods such as : center of gravity, max-min method…etc. The **Figure 3.1** provides a high-level overview fuzzy system and integration between its main components.

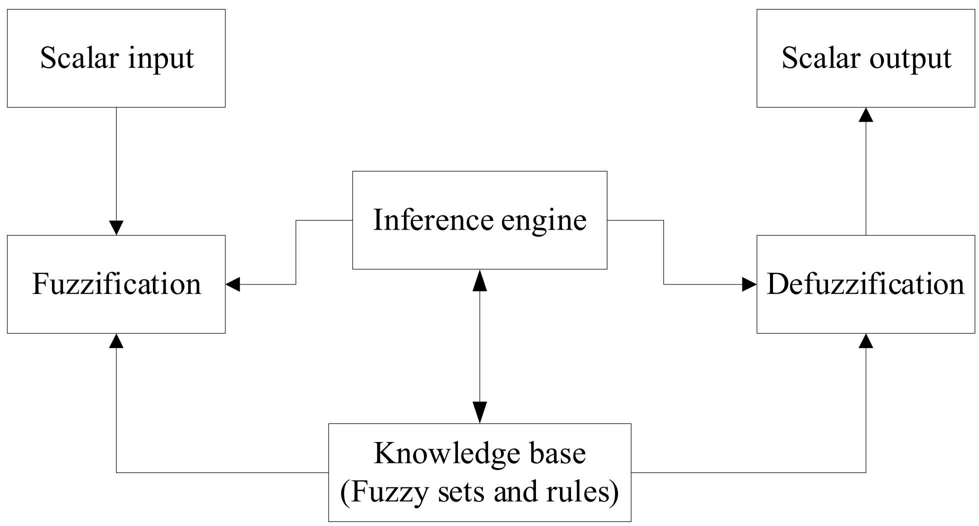

Figure 3.1 Fuzzy reasoning system

3.2 Fuzzy controller

Fuzzy control does not involve mathematics, it uses linguistic sentences instead of numbers, much easier to understand by human, various applications of fuzzy controller developed for the industry, we can cite as an example, steam engine , control of robot, power plants…etc. fuzzy controller is composed of the following :

a) **Knowledge base** which contains the control strategy that describe the dynamic behaviour of the system, the rule base can be static or adaptive rules using EA or ANN learning.
b) **Fuzzifier** converts the rules using membership degrees to fuzzy representation.
c) **Inference engine** performs mapping between fuzzified input and rules base to produce fuzzy output.

d) **De-fuzzifier** converts the fuzzy action coming from the inference engine to a scalar value that can be applied to a system.

The fuzzy controller is considered as a nonlinear function due to the use of nonlinear membership function, inference engine which use min and max operator and the knowledge base which uses a non-linear control strategy. There are two types of fuzzy controller **Mamdani fuzzy controller** [49] and **Takagi-Sugeno Controller** [50], in this thesis , we have used the Mamdani type to describe the system.

We can design a fuzzy logic controller either as a standalone or in conjunction with another type of controller. recent research trend towards the use of hybrid approach which combines various theories. Hence, to enhance robustness and permit a design of intelligent controllers, fuzzy controller can be used with PID controller.

3.3 Fuzzy PID controller (FPID)

The FPID is an easy to understand and implement, as it involves very limited mathematics. The FPID shown in **Figure 3.1** takes two inputs namely the error (e) and change of error $\frac{d}{dt}$ and produces three output control signals according to the **fuzzy inference structure** (FIS). Seven MFs for each of the input leading to a total of 49 rules for each of the three PID parameters. **Figure 3.2** shows the number of inputs/outputs used with Mamdani fuzzification method. In **Figure 3.3** the fuzzy membership function editor is shown, where the number of membership functions, range and type of membership function is chosen.

In our study, we have chosen gaussian membership functions which gives a very smooth output, with a range of output variables MFs in FLCs [- 1, 1]. **Table 3.1** shows the Fuzzy PID rules for developing FIS.

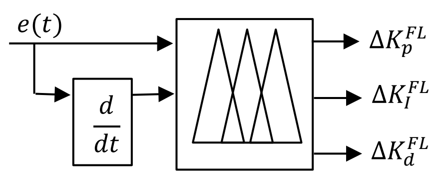

Figure 3.2 FPID controller

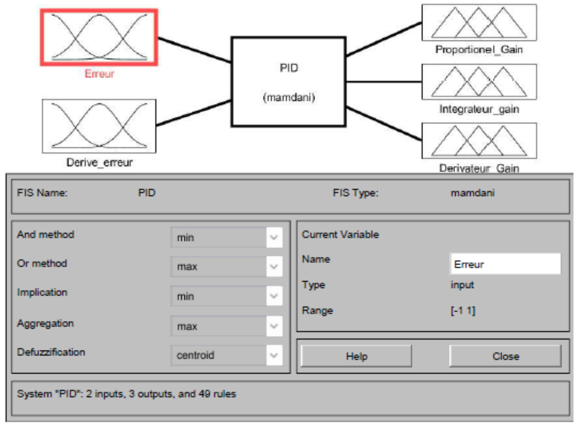

Figure 3.3 FIS for FPID controller

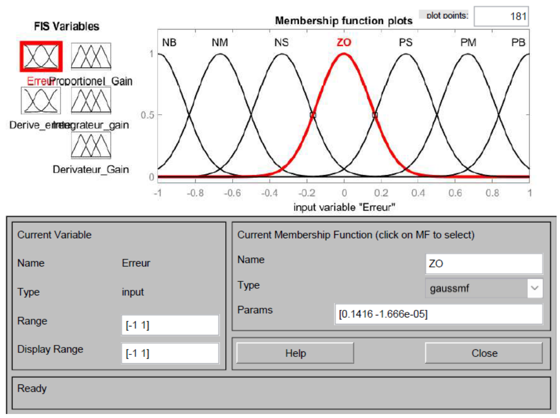

Figure 3.4 Membership function editor for FPID controller

Table 3.1 Fuzzy rules base for developing FIS for a FPID controller

		Change of error						
	K_P K_I K_D	NB	NM	NS	ZO	PS	PM	PB
Error	NB	PB	PB	PM	PM	PS	ZO	ZO
		NB	NB	NM	NM	NS	ZO	ZO
		PS	NS	NB	NB	NB	NM	PS
	NM	PB	PB	PM	PS	PS	ZO	NS
		NB	NB	NM	NS	NS	ZO	ZO
		PS	NS	NB	NB	NM	NS	ZO
	NS	PM	PM	PM	PS	ZO	NS	NS
		NM	NM	NS	NS	ZO	PS	PS
		ZO	NS	NM	NM	NS	NS	ZO

	ZO	PM	PM	PS	ZO	NS	NM	NM
		NM	NM	NS	ZO	PS	PM	PM
		ZO	NS	NS	NS	NS	NS	ZO
	PS	PS	PS	ZO	NS	NS	NM	NM
		NM	NS	ZO	PS	PS	PM	PB
		ZO	ZO	ZO	ZO	ZO	ZO	ZO
	PM	PS	ZO	NS	NM	NM	NM	NB
		ZO	ZO	PS	PS	PM	PB	PB
		PB	NS	PS	PS	PS	PS	PB
	PB	ZO	ZO	NM	NM	NM	NB	NB
		ZO	ZO	PS	PM	PM	PB	PB
		PB	PM	PM	PM	PS	PS	PB

3.4 Fuzzy fractional-order PID controller (FFPID)

The FOPID transfer function already defined in equations (1.20) and (1.21).

Most of the intelligent controllers have a tuned parameters; consequently, FFPID controller is tuned using genetic algorithm (GA). GA is a bio inspired technique based on the evolution of chromosomes and can achieve high efficiency by searching global optimal solution in problem space. It is not impacted by the classical optimization issues related to the stuck in local minima.

For each cycle of GA applied, the result is hybridized with fuzzy logic PID controller for fine tuning. The values obtained from this Hybridize are:

$$\begin{cases} K_p = K_p^{GA} + K_p^{FL} \\ K_i = K_i^{GA} + K_i^{FL} \\ K_d = K_d^{GA} + K_d^{FL} \\ Fractional\ derivatiove = S^{mu,GA} \\ Fractional\ integrator = 1/S\hat{}(lambda, GA) \end{cases} \quad (3.1)$$

A schema of a cooperation between FPID controller and FOPID is presented in **Figure 3.5**

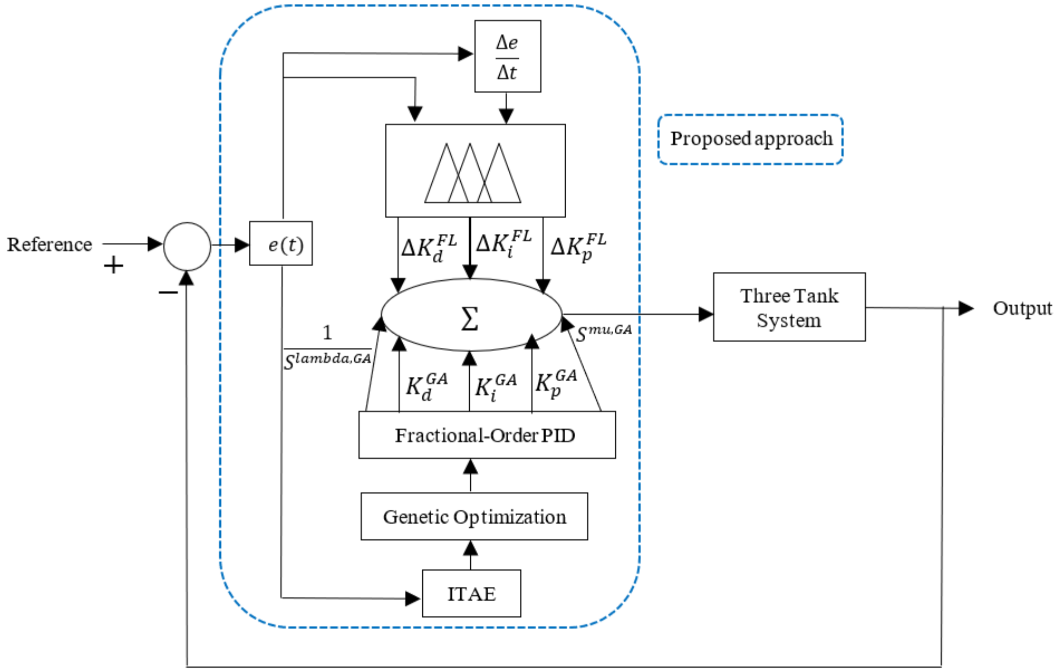

Figure 3.5 Fuzzy fractional-order PID controller

3.5 Application to control three tank systems

The proposed controller is tested on TTP, the simulation shows its effectiveness in terms of time domain performances.

The TTP can be represented as follows :

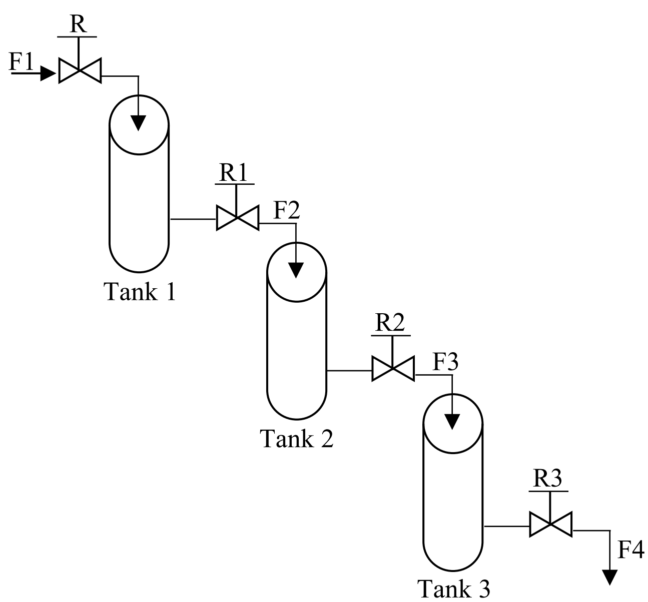

Figure 3.6 Process model: Three tank level system

A three-container tank is usually connected by three first order non periodic inertia links in series. The system's input is the flow in the first tank F1 and the output is the level in the third tank h3.

The mathematical modeling of TTP is obtained using Bernoulli's law. The transfer functions of the systems to be controlled is :

$$Sys = \frac{0.532}{24S^4+44S^3+30S^2+9S+1} \tag{3.2}$$

To find a search space of GA & initialize PID parameters, we applied Ziegler Nichols (ZN) tuning, the tuned gains are listed in the following table:

Table 3.2 PID parameters

K_p	K_i	K_d
4.6604	0.6709	8.0932

GA is then used for optimization, ITAE is used as the fitness function. The fitness function is described as follows :

$$J(x) = \int_{\tau=0}^{\tau=t_{max}} t|e(\tau,x)|d\tau \tag{3.3}$$

The minimization of the fitness function using GA

$$\min_{x\in\Omega} J(x) = \min_{x\in\Omega} \int_{\tau=0}^{\tau=t_{max}} t|e(\tau,x)|d\tau \tag{3.4}$$

$$\text{Subjected to } \Omega = \begin{cases} K_p\ (proportianl\ gain) \\ K_i(integratorgain) \\ K_d\ (derivative\ gain) \\ \lambda\ (fractional\ order\ of\ integration) \\ \mu(fractional\ order\ of\ differentiation) \end{cases} \tag{3.5}$$

Each parameter is limited by minimum and maximum boundary as follows :

$$x_{min} \le x \le x_{max} \tag{3.6}$$

The search space of GA is as follows

$$\Omega = \begin{cases} 0 \le K_p \le 500 \\ 0 \le K_i \le 500 \\ 0 \le K_d \le 500 \\ 0 \le \lambda \le 1 \\ 0 \le \mu \le 1 \end{cases} \tag{3.7}$$

For the implementation of the FOPID, we have use the Oustaloup's recursive approximation with a frequency range of $\omega \in \{10^{-4}, 10^4\}$ rad/s and 10th order.

The final optimal parameters obtained are as follows :

Table 3.3 Fuzzy fractional PID parameters

K_p	K_i	K_d	λ	μ
2.3303	0.39328	9.6891	0,99878	0.99998

For the simulation results of a unit step response, we compare our new approach with PID controller and FPID controller.

The ideal response of the system should have the characteristics of quick settling time, zero steady state error, and stable.

Figure 3.7 and **Figure 3.8** presents the time performance results for a step input with the control signal efforts of the controllers.

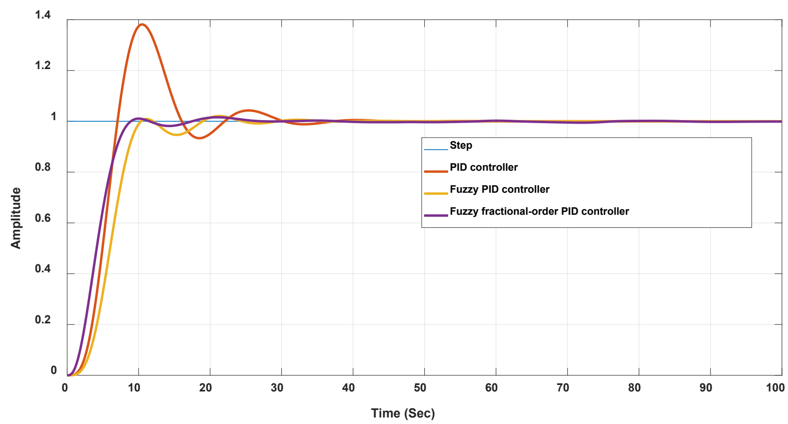

Figure 3.7 Level control using different controllers

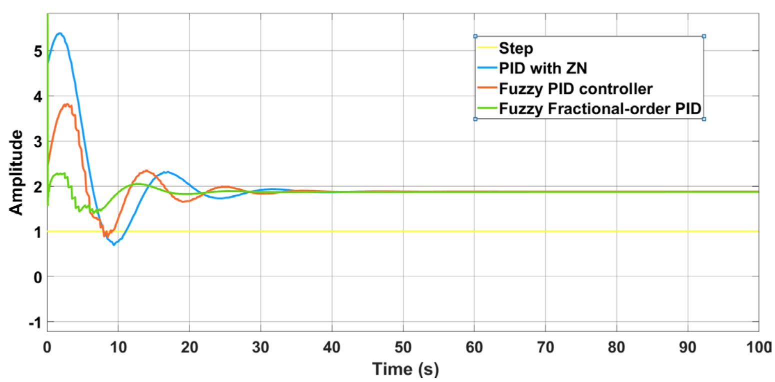

Figure 3.8 Different control signals

The PID controller has oscillations with a peak overshoot. The response of fuzzy logic controller is almost smooth but not as fast as the FFPID controller. The FFPID controller reaches the desired speed with remarkable results and superior performance compared with other controllers. The control signals figure shows that the proposed controller presents a pic effort at the beginning and then less effort after 10s compared with the others.

For load disturbance rejection, PID controller shows good performance compared with both FPID and FFPID controller.

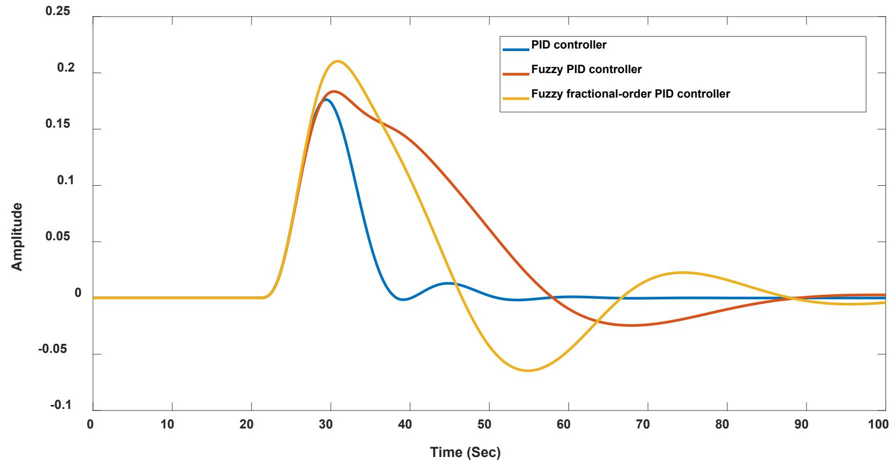

Figure 3.9 Load disturbance rejection for a step injection

3.6 Conclusion

The proposed controller is established for a TTP and seeks a benefit of hybridizing fuzzy control with FOPID. It has been proven that the controller shows high performance and effectiveness for the level control of three tank system. As a perspective work on the tuning of this type of controller, we can use another enhanced biologic algorithm such as artificial immune algorithms, SI.

4. Control theory - artificial immune system

4.1 Introduction

Natural immune system is a robust and adaptative biological mechanism able to protect the body against possible disease due to foreign invaders (referred to as antigen) such as viruses, bacteria, parasites…etc, the antigen is the material that trigger the immune response, to eliminate these foreign invaders. The natural immune system can differentiate between self-cell and non-self-cell . In the other hand antibodies part of self-cells are chemical proteins produced by the immune system to bind the antigen and form an antigen-antibody complex to de-activate it. There are five classes of antibodies: IgG, IgE, IgM, IgA, IgD.

The process of interaction between antigen (non-self-cell) and antibody (self-cell) and its expansion to enhance the antigen-specific immune response leads to the development of several theories:

- ✓ Negative selection theory.
- ✓ Clonal selection theory.
- ✓ Network theory.
- ✓ Danger theory.

Natural immune system is a multilayer system composed of two parts : the innate immune system and the adaptive immune system, both influence each other.

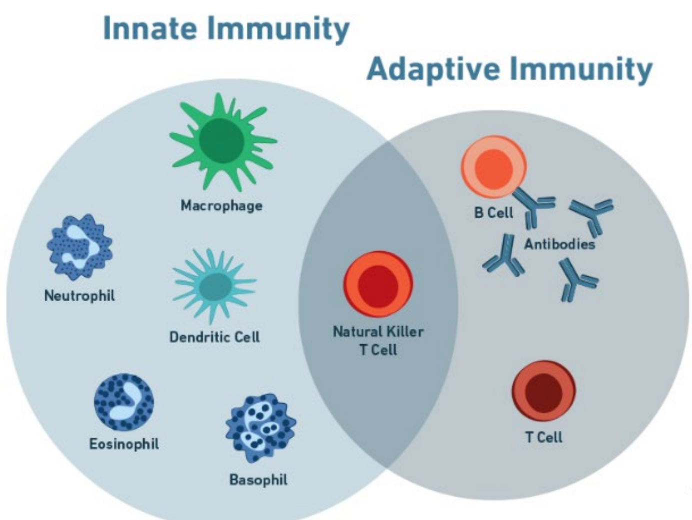

Figure 4.1 Innate immunity vs. Adaptive immunity

Source : Cell Signalling Technology (CST) https://www.cellsignal.com/

4.2 Innate immune System

The innate immune system is the primary response of defence which provides barriers to pathogens through different mechanism using skin, mucous membrane, physiological conditions (PH, temperature, chemical mediators…etc), defence cells and proteins…etc. it is considered as a fast response but non-specific as it responses to germs in the same way and not able to form any cellular memory.

The main functions of the innate immune system are :

- ✓ The physical and chemical barrier against infectious agents.
- ✓ The identification and elimination of foreign bodies present in the body by white blood cells.
- ✓ Activation of adaptive immunity through the presentation of molecular pattern associated with pathogens.

4.3 Adaptive immune System

The adaptive immune system intervenes when the primary innate is not able to destroy the pathogens, it is slower than the innate but more accurate and highly specific.

The immune system is supposed to face repeated antigens (Ag) or non-self-cells, the first time of exposure stimulates the immune response system, this response is handled by the lymphocytes, which will produce different antibodies (Ab), the response to the second exposure of antigens is improved, as one of the important advantages of this immune mechanism is the possibility to remember germs using a cellular memory associated with the first infection. This time the lymphocytes are characterized by speed and accuracy and produces high affinity anti bodies. It may take some days at the beginning to respond and then the second time the immune system reacts immediately. this feature is a principal characteristic of **reinforcement learning** as the interaction with the environment is improved step by step when doing a specific task.

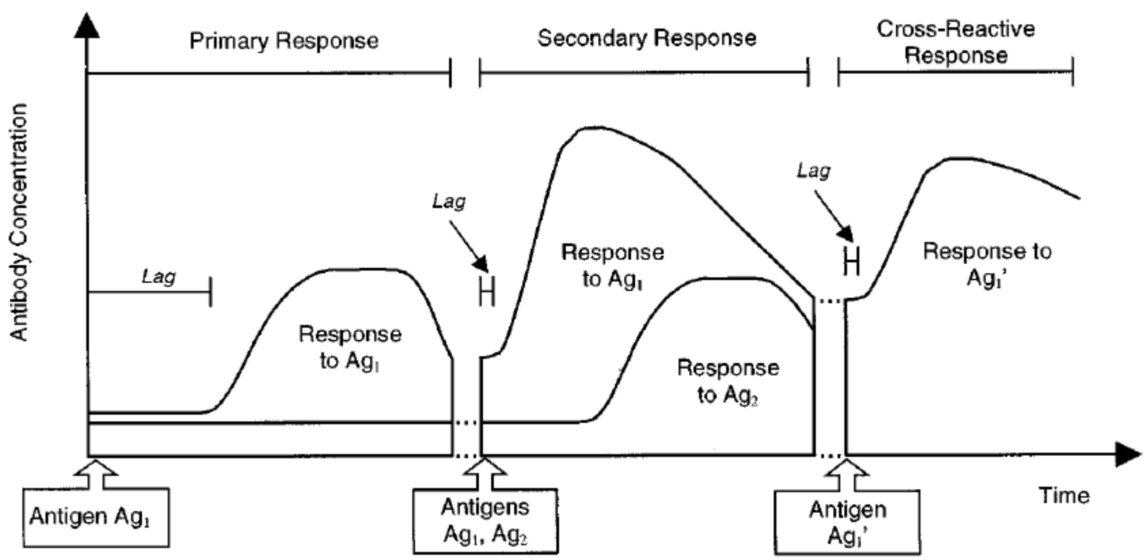

Figure 4.2 Primary, secondary, and cross-reactive response

After an Ag enters to the body, when facing the same infection or having the cross-reactive response (structurally related antigen) will lead to faster response (secondary response).

There are two major characteristics unique to adaptive immunity :

- ✓ The activation of a lymphocyte goes along with a clonal expansion (allowing to amplify the specific immune response to the antigen) and the establishment of a memory.
- ✓ The genes encoding the antigen receptors of lymphocytes are subject to somatic and random recombination of DNA, called somatic recombination. the nature of lymphocyte antigen receptors is acquired and not genetically determined.

The **adaptive immune system** is subdivided based on the lymphocyte types, T-cell lymphocyte for cell mediated immunity and B-cell lymphocyte for humoral immunity, see **Figure 4.3**

4.3.1 Cell mediated immunity

Cell mediated immunity uses T lymphocytes, a type of white blood cells originates in the bone marrow and mature in the thymus. There are two types of T-cells, T-cell-helper which strengthen the B-cells, T-cell-killer for destruction of foreign invaders and T-suppressor used to stabilize the immune system when the germs are killed.

4.3.2 Humoral immunity

Humoral immunity or referred to as **antibody-mediated immunity**. involves B lymphocytes, a type of white blood cells originates in the bone marrow and leave it as mature and then migrate to the lymph nodes, and responsible to produce antibodies to eliminate antigens.

This immune response consists of four major steps :

- ✓ **Recognition of the antigen and clonal selection**: the B lymphocytes that can recognize foreign molecules are selected, and this selection activates the B lymphocytes.
- ✓ **Activation of B lymphocytes**: the selected B lymphocyte will go through serial mitoses, and we will then have clonal proliferation.
- ✓ **Differentiation of B lymphocytes**: part of the B lymphocytes will transform into plasma cells and the other will form memory B lymphocytes. Memory cell will help to sped up the reaction when facing the same antigen.
- ✓ **Antigen neutralization**: the antibodies synthesized by the plasma cells will be released to bind to the antigens.

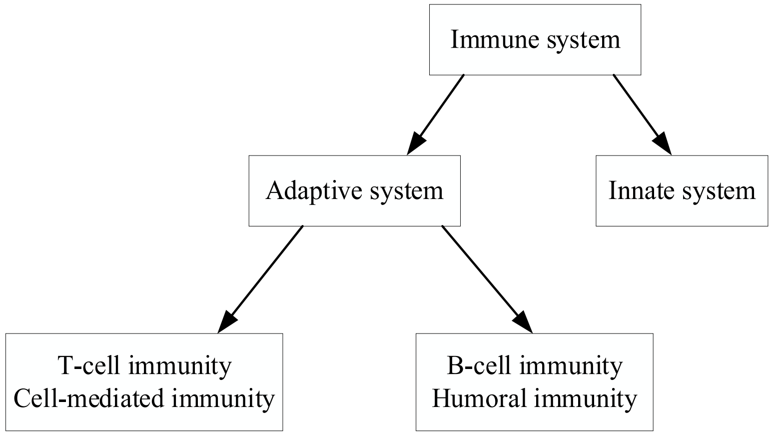

Figure 4.3 T cell immunity vs. B-cell immunity

4.4 Identification of Antigen structure

The identification of the structure of antigen is done through a process referred to as affinity maturation which is composed of cloning process and hyper-mutation process, the cloning is the proliferation of lymphocytes that bind the antigen, and the somatic hyper-mutation process is the creation of mutated clones by the lymphocytes to learn the shape of the antigen to get a hight binding affinity.

4.5 Artificial immune systems

The AIS is a computational technique also known as immuno-logical computation can be applied to solve a wide ranges of engineering applications such as optimization, anomaly detection, pattern recognition and machine learning.

From an automation perspective, it is considered as an intelligent system as it uses the learning mechanism when recognizing patterns and uses memory to take in consideration the past information which make him very powerful, consequently, this information processing technique has attracted the attention of computer science. A lot of algorithms today uses the analogy with biological systems as metaphors to develop some features with aim to resolve problems related to engineering, we can cite as an example but not limited to : neural networks algorithm, genetic algorithm…etc.

4.6 AIS basic algorithms

It has been shown in the natural immune system that the cooperation between B lymphocytes and T lymphocytes secrete antibodies to bind the antigen. Affinity is the number of Lymphokines that are secreted by T-cell to proliferate B-cells into plasma to produce antibodies. A mature T-cell has the capability to distinguish between self-cell and non-self-cell, by analogy artificial lymphocytes (ALC) can detect non-self-patterns with an affinity level.

The basic steps for an AIS algorithm are the initialisation of a population to generate ALCs either randomly or using a training set, determine the stopping condition such as the number of iteration or based on the convergence of ALCs….etc, measurement of antigen affinity, referred to as similarity (matching rule) between ALCs and antigen patterns, measurement of network affinity between two ALCs and finally use a selection method such as negative selection, clonal selection…or other evolutionary techniques with mutation operators to adapt the ALCs.

Similarity can mean **hamming distance** in binary encoding/initialisation, the Euclidean distance in a real variable or it could be correlations (Pearson correlation coefficient between two sets of data) …. etc. the selection method negative or clonal depends on the use case. The mutation is very similar to the ones found in genetic algorithms.

AIS computational models inspired from natural immune theories are as follows :

a. Negative selection model

Also referred to as central tolerance which means tolerance for self-cells, it can differentiate between undesired antigen and self-cells. This process eliminates any T or B lymphocytes reactive to the body itself.

In AIS, the **negative selection algorithm** is a training method mainly used to test ALCs for pattern recognition, the principle is to look for relations and useful pattern in the information. This can be applied on the anomaly detection problems. The idea is to train the ALCs (which represents the mature T-cell in the natural immune system) to be self-tolerant for self-patterns and determine the affinity between ALCs and patterns using a matching rule to recognize any element not part of the self-set, once the non-self is recognized, an appropriate action needs to be taken. The algorithm is described in [44].

b. Clonal selection model

Clonal selection in AIS is having a mechanism of selection of a set of ALCs with hight affinity to a non-self-pattern, and then being cloned and mutated capable of binding specifically the antigen (non-self-pattern). This method was mainly used in science and engineering for pattern recognition than adapted to solve multimodal and optimization. This feature inspired the development of the clonal algorithm (CLONALG). The algorithm is described in [45].

c. Immune network model

The immune network paradigm developed by **Niels Kaj Jerne** in 1974. It is an hypothesis in immunology which announce that B-cells can interconnect to create a network. He was awarded the Nobel Prize in Physiology or Medicine in 1984 for the development of this theory. B-cells are stimulated by antigen and neighbour lymphocytes. In the context of AIS, the ALCs are in a network AIS, an affinity is calculated between ALCs and non-self pattern and/or between an ALC and other ALCs in the network to co-stimulate or co-supress each other, hence adapted to the structure of the antigen.

One of the important applications of this theory is the fault tolerant multi-robot cooperation in [51]

d. Danger theory model

In this AIS model, it differentiates between dangerous and non-dangerous, the main feature in this model is the use of a "stress signal" to determine if a non-self-pattern is dangerous. A good example of the beneficial use of this model to create an adaptable intrusion detection system (IDS) using the stress signal in danger theory, when any abnormal traffic is detected, the IDS will signal an alarm.

4.7 Application areas of AIS

The application of AIS is summarized in detail in [52], they can be grouped in different categories, classification and clustering, anomaly detection, optimisation and learning.

4.7.1 Classification & clustering

The classification is like any machine learning, it requires extraction, recognition and learning. It uses the supervised learning whereas the clustering uses unsupervised learning and it does not use any prediction. The use of CSA approach in [53] to resolve classification problems and immune network theory-based clustering algorithm in [54]

4.7.2 Anomaly detection

The idea of this method is to generate a set of detectors which represents the normal situation, it is possible then to detect an intrusion to the system or suspicious behaviour. One of the important pilar of anomaly detection is to use it in **computer security** to detect viruses or harmful intrusion and respond quickly to this unauthorized change. A practical example is the intrusion detection system used in firewalls to protect computers, one of the approaches is signature-based detection of previously known bad patterns such malware…etc, which has demonstrated its limitation to detect unknown malwares and viruses. In contrast, the natural immune system can respond to any unknown antigens, hence, the second approach which relies on learning mechanism such as AIS are anomaly-based detection capable of detecting a deviation or suspicious behaviour for anu unknown bad patterns. It can be used also in **danger theory**, which means that we do not need to intervene and respond to all intrusion but only when there is a danger.

4.7.3 Optimization

The purpose is to maximize or minimize a given criterion usually called an objective function which includes in automation engineering cost, control signal energy…etc. we can cite some examples, the use of danger model immune algorithm (DMIA) in [55] to solve general optimization problems, clonal selection algorithm (CSA) approach used as optimization algorithm in [56], novel immune clonal algorithm (NICA) in [57] for multiobjective optimization.

4.7.4 Learning

The principle of learning is to acquire knowledge by experience, and apply this knowledge to solve unseen problems, a good example of immune learning is the vaccination. AIS combine the advantages of neural networks and learning classifier systems.

Other minor application areas such as, image processing, control, autonomous navigation of robots…etc

The examples cited above are an example but not limited to, but it still worth to continue searching where the AIS is suitable and best applied.

4.8 AIS Enhancement & Hybridization

4.8.1 AIS Enhancement

The AIS system can also be associated with other techniques/approaches to enhance its functionality or extend its features. We can site the use of fuzzy logic, we can cite the fuzzy recognition algorithms in [58]. the use of probabilistic methods such as bayesian artificial immune system (BAIS) in [59] and gaussian artificial immune system (GAIS) in [60]. Chaos theory and Kernel method. Etc

4.8.2 AIS Hybridization

The AIS system can be combined with other mechanisms to solve difficult problems, nature inspired such as ANNs, GA and SI approaches. Or with machine learning such as support vector machines (SVMs) and k-nearest neighbours (kNN).

5. Design of fractional-order PID controller based on artificial immune system - applied to control time-delay systems

5.1 Introduction

Many studies show the advantages of using bio inspired techniques to add learning capabilities such as the use of artificial neural networks inspired from brain-nervous system or adding a self-learning when using genetic algorithms to solve mathematical optimization problems. Hence the performance of bio-inspired control system is much better than the classical one.

Biological techniques include also endocrine system and immune system. The immune system has the capability to erase viruses and germ thanks to its self-adaptability, learning memory and pattern recognitions.

From control system perspective, many automations controller are inspired from this biological mechanism. Neural networks controllers have been investigated in terms of robustness and performances [61], artificial immune feedback controller in terms of enhancing the performances of such as speed and stability. Etc.

5.2 Design of immune system law

In this chapter we will develop an immune feedback law based on the immune system behaviour. There are two types of immune mechanisms, **innate** considered as general, rapid and without memory. **adaptive** which is specific, slow and with long term memory. And there are two types of immune responses processes, **humoral** response where the antibodies are produced in B-cell to eliminate the foreign germs and a **cellular** response based on T-cell to kill the viruses.

The immune feedback law that we will consider in this study is adaptive. The schema **Figure 4.1** shows the principle of adaptive immune response.

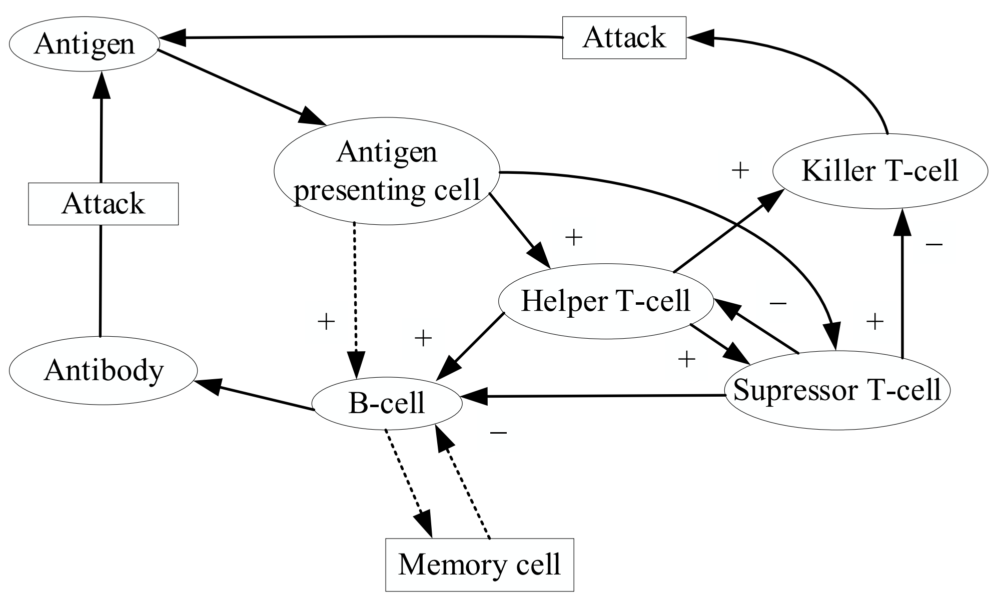

Figure 5.1 Immune system

The biological immune system is too much complex, hence we simplify the schema to the following :

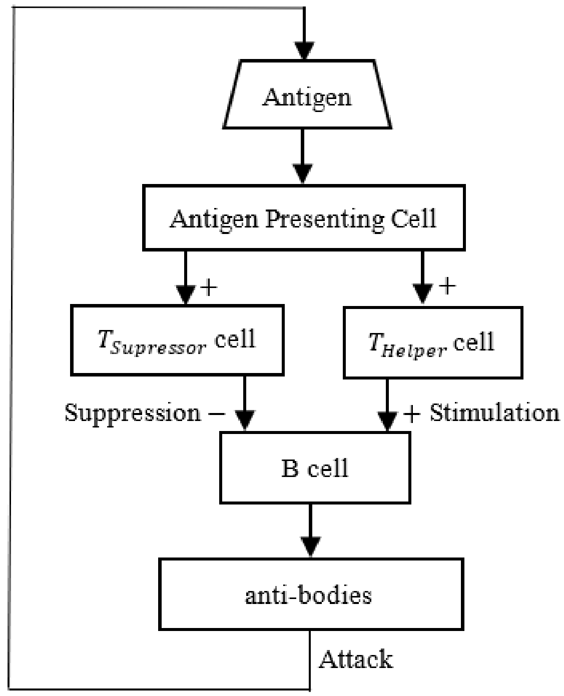

Figure 5.2 Simplified immune system

When the antigen enters the body, an immune system cell presents parts of intruders to T cells, The T helper cells (T_h cells), also known as $CD4^+$ cells or CD4 positive cells stimulate the B-cells to create antibodies and neutralize the antigen, when the antigen start decreasing, The regulatory T cells formerly known as suppressor T cells increase and consequently the B-cell reduces the production of antibodies until the system tends towards the equilibrium.

The immune system can be described by the following equations [62]:

The amount of antigen at the kth generation is

$$\varepsilon(k) = \gamma\varepsilon(k-1) - B(k-d) \tag{5.1}$$

Where γ is the multiplication factor, d is the delay time between the infection and the reaction and B is the amount of b-cells consistency.

The stimulation of B-cells is described by the following :

$$B(k) = T_h(k) - T_s(k) \tag{5.2}$$

The helper T-cell is described by the following :

$$T_h(k) = k_1\varepsilon(k) \tag{5.3}$$

Where k_1 is the stimulation factor.

The suppressor T-cell is described by the following :

$$T_s(k) = k_2 f[B(k), \Delta B(k-d)]\varepsilon(k) \tag{5.4}$$

Where k_2 is the suppressor factor and the function $f(x)$ is a nonlinear function which describe the change of antibodies based on antigen consistency.

We assume that $f(x)$ is defined as follows [63]:

$$f(x) = 1 - \frac{2}{1+\exp(-cx)}, \ C > 0 \tag{5.5}$$

Where c is a parameter that changes the function shape and $-1 < f(x) < 1$

From the previous equations, the consistency of B-cells can be described as follows :

$$B(k) = k_1\varepsilon(k) - k_2 f[B(k), \Delta B(k-d)]\varepsilon(k) \tag{5.6}$$

$$= K\{1 - \eta f[B(k), \Delta B(k-d)]\}\varepsilon(k) \tag{5.7}$$

Equation (4.7) describes the immune feedback law, where $\eta = \frac{k_2}{k_1}$ is called the stabilization factor and the parameter $K = k_1$ control the response speed.

Figure 5.3 shows the feedback law schematically, we assume that the integration of the stimulation $B(k)$ gives the activity of B-cells, and the differentiation of this last gives the antibodies responsible to kill the antigen.

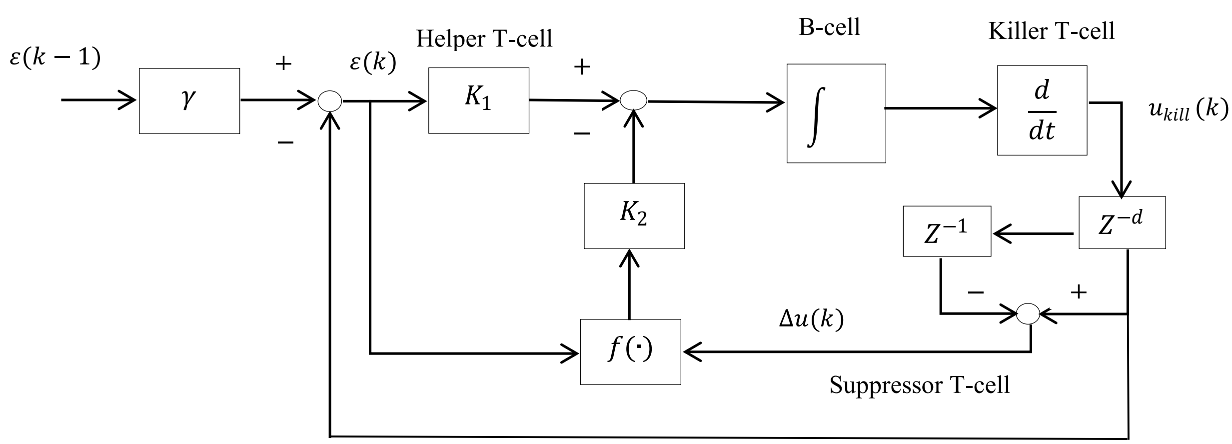

Figure 5.3 Immune feedback law

5.3 Design of immune PID controller

By analogy, the immune system controller can be derived from the immune system law, in control system theory, the kth generation is considered as sampling number and the amount of antigen $\varepsilon(k)$ at the kth generation can be seen as control error $e(k)$ between the desired output and the feedback. The total stimulation of B-cell is considered as the control signal $u(k)$

$$u(k) = K\{1 - \eta f[u(k), \Delta u(k-d)]\}e(k) \tag{5.8}$$

$$= k_{pI}e(k) \tag{5.9}$$

$k_{pI} = K\{1 - \eta f[u(k), \Delta u(k-d)]\}$ is a nonlinear gain of the immune controller, which is like a proportional controller.

To enhance the performance of this last, we have decided to combine it with a PID controller. The transfer function of a PID controller is as follows :

$$u_{PID}(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t) \quad (5.10)$$

K_p, K_i and K_d are the proportional, integrator and derivative gains respectively. When combining the immune controller with the PID we obtain the immune feedback controller based on the PID controller.

The transfer function of the PID controller based on immune feedback is as follows :

$$u(t,x) = K\{1 - \eta f[u(k), \Delta u(k-d)]\} \cdot \left[K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt} e(t)\right] \quad (5.11)$$

The block diagram representing this controller is as follows :

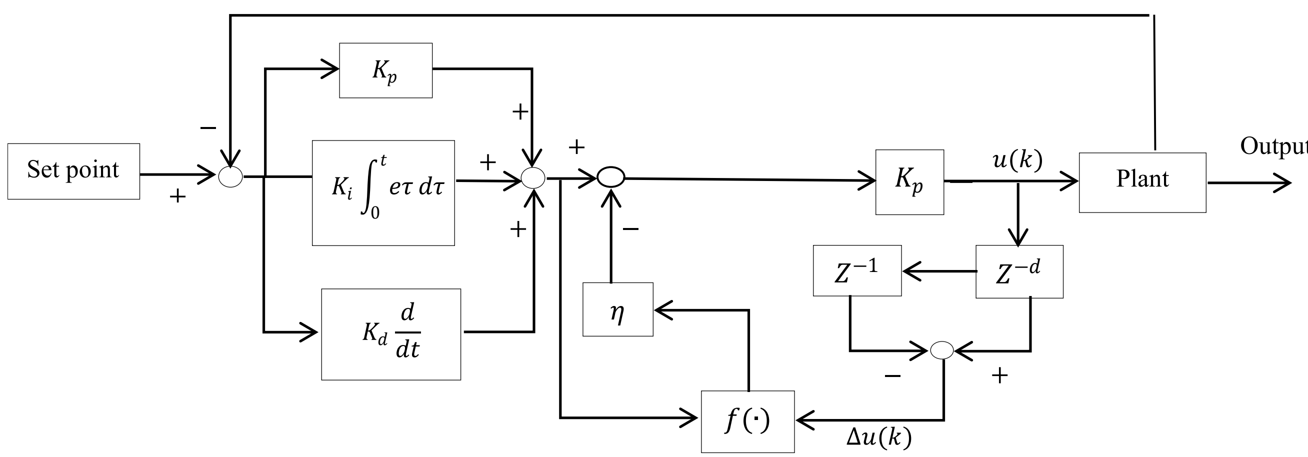

Figure 5.4 Immune PID controller

According to the positive/negative action of $\eta f[u(k), \Delta u(k-d)]$ the immune controller keeps the system stable. The immune controller has 5 parameters, it provides two additional degree of freedom to the PID, when $\eta = 0$ the transfer function becomes an equation of a classical PID controller.

5.4 Design of FOPID based on immune law

The FOPID is considered as a generalization of PID controller, it has two additional degree of freedom, hence it provides more parameters to adjust than the classical PID, all the parameters are optimized using the genetic algorithms by minimizing an objective function based on ITAE criterion.

The idea of this design is to replace the classical PID with a fractional controller.

The FOPID controller can be defined in S-domain as follows [37] :

$$T_{FOPID} = \frac{U(s)}{E(s)} = K_p + K_i S^{-\lambda} + K_d S^{\mu}; (\lambda, \mu > 0) \quad (5.12)$$

The FOPID controller can be defined in time-domain as follows :

$$u_{FOPID}(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^{\mu} e(t) \tag{5.13}$$

K_p, K_i and K_d are the proportional, integrator and derivative gains respectively.

λ and μ are the fractional orders of integration and differentiation.

When combining the FOPID with immune controller we obtain the FOPID controller based on IFM.

The transfer function of the FOPID controller based on immune feedback is as follows [64]:

$$u(t,x) = k_{pI} \cdot PI^{\lambda} D^{\mu}(t) \tag{5.14}$$

where $k_{pI} = K\{1 - \eta f[u(k), \Delta u(k-d)]\}$ and $PI^{\lambda} D^{\mu}(t) = u_{FOPID}(t)$

The simplified equation of is described as follows :

$$u(t,x) = K\{1 - \eta f[u(k), \Delta u(k-d)]\} \cdot \left[K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^{\mu} e(t)\right] \tag{5.15}$$

The FOPID controller based on IFM has 8 parameters, it provides five additional degree of freedom to the PID, when $\eta = 0$ and the λ=μ=1, the transfer function becomes an equation of a classical discrete PID controller.

5.5 Tunning of parameters using genetic algorithms

The FOPID based on IMF controller parameters that needs to be optimized are represented by a vector x = (K_p, K_i, K_d, λ, μ, η, c, K).

The tunning of these parameters is based on the **minimization of ITAE** criterion **using the GA**.

The fitness function is described as follows :

$$J(x) = \int_{\tau=0}^{\tau=t_{max}} t|e(\tau,x)|d\tau \tag{5.16}$$

and the minimization of this fitness function using genetic algorithms can be described as follows :

$$\min_{x\in\Omega} J(x) = \min_{x\in\Omega} \int_{\tau=0}^{\tau=t_{max}} t|e(\tau,x)|d\tau \tag{5.17}$$

$$\text{Subjected to } \Omega = \begin{cases} K_p \ (proportianl\ gain) \\ K_i (integratorgain) \\ K_d \ (derivative\ gain) \\ \lambda \ (fractional\ order\ of\ integration) \\ \mu (fractional\ order\ of\ differentiation) \\ \eta = k_2/k_1 \ (stabilization\ effect) \\ c \ (Constant) \\ K = k_1 \ (Stimulation\ factor) \end{cases}$$

Each parameter is limited by minimum and maximum boundary as follows :

$$x_{min} \le x \le x_{max}$$

5.6 Application to control time-delay systems

The proposed controller is tested on time delay systems, the simulation shows its effectiveness in terms of robustness to gain and time delay variations, time domain performances and disturbance rejection.

Many of process control applications can be good approximated by time-delay systems. To demonstrate our approach, we applied the FOPID based on IFM to control time-delay systems and compared this simulation with immune PID, when changing different parameters : time constant - time delay – and gain parameter to study the robustness and effectiveness of both commands.

The transfer functions of the systems to be controlled are :

First-order plus time-delay system (FOPTD) : $Sys1 = \frac{1}{2.5S+1} e^{-1.8S}$ (5.18)

Delay dominant system : $Sys2 = \frac{1}{2S+1} e^{-4S}$ (5.19)

The fitness function is described as follows :

$$J(x) = \int_{\tau=0}^{\tau=t_{max}} t|e(\tau, x)|d\tau \quad (5.20)$$

The minimization of the fitness function using GA

$$\min_{x\in\Omega} J(x) = \min_{x\in\Omega} \int_{\tau=0}^{\tau=t_{max}} t|e(\tau, x)|d\tau \quad (5.21)$$

The search space for the FOPID parameters is :

$$\Omega = \begin{cases} 0 \le K_p \le 500 \\ 0 \le K_i \le 500 \\ 0 \le K_d \le 500 \\ 0 \le \lambda \le 1 \\ 0 \le \mu \le 1 \\ \le \eta = k_2/k_1 \le \\ 0 \le c \le 50 \\ 0 \le K = k_1 \le 200 \end{cases} \quad (5.22)$$

For the implementation of the FOPID controller, we have use the Oustaloup's recursive approximation with a frequency range of $\omega \in \{10^{-2}, 10^{2}\}$ rad/s and 4th order.

5.7 FOPTD system :

The Ziegler–Nichols tuning method has been used to initialize the GA, the parameters are listed in **Table 5.1**, and the final optimal parameters obtained from GA are listed in **Table 5.2**

Table 5.1 PID parameters computed using ZN tuning rule for FOPDT system

K_p	K_i	K_d
1.7152	0.5848	1.2576

Table 5.2 FOPID based on IFM parameters computed using GA tuning rule for FOPDT system

K_p	K_i	K_d	λ	μ	k	n	c
401.5	65.1	0.016	0.48	0.06	$9.77 \times 10{-5}$	0.0001	20.47

After running GA, the optimization results are shown in **Figure 5.5** comparing the fitness value vs. generation., in addition it gives the percentage of the stopping criteria.

Figure 5.6 presents the comparison between FOPID based on IFM vs. immune PID in the time-domain for a step response and **Figure 5.7** shows the comparison between the control signal for both types of controllers. The time domain specifications are summarized in that **Table 5.3**

From the numerical simulation results, it is shown that the proposed controller has a good performance with less effort, without overshot and with excellent steady error. Whereas the immune PID has a small overshot and requires huge effort to meet the same performance.

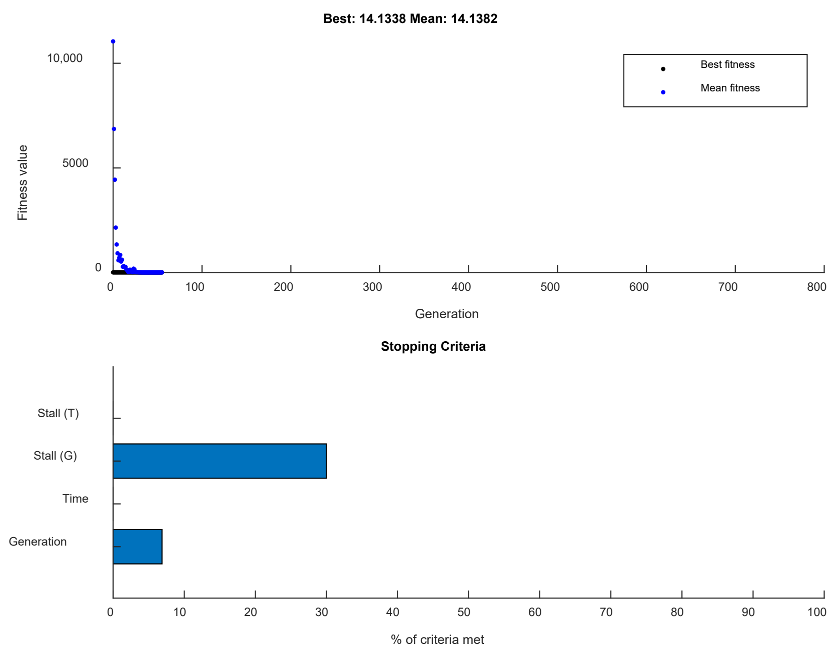

Figure 5.5 GA optimization results for FOPDT system

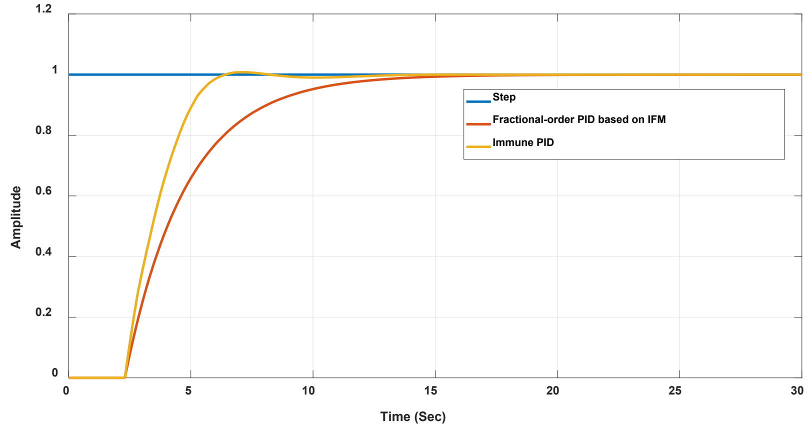

Figure 5.6 Step response of both FOPID using IFM and immune PID for FOPDT system

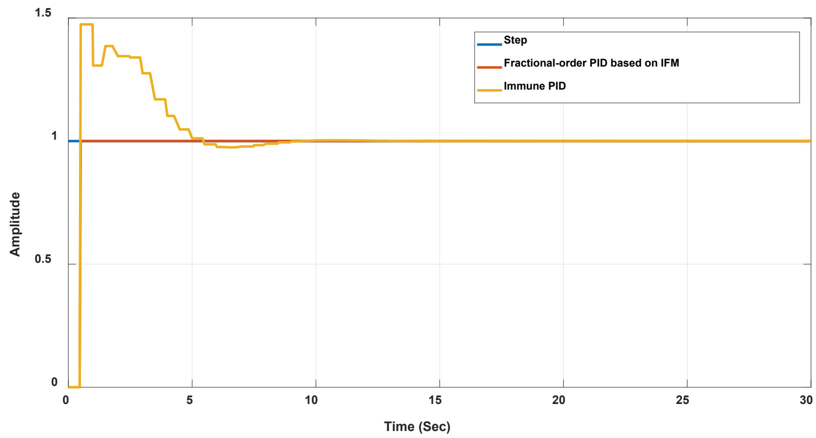

Figure 5.7 Control signal of both FOPID using IFM and immune PID for FOPDT system

Table 5.3 Comparison of time domain performances for FOPDT system

Controller used	Peak (%)	Settling Time (s)	Rise Time (s)	Steady State Error (%)
Immune PID	1.4	5.7	2.76	1
FOPID based on IFM	No overshoot	9.8	5.44	0

To evaluate the robustness, we change the time constant and time-delay for the system to simulate modeling uncertainties.

Figure 5.8 presents a comparison between FOPID based on IFM vs. immune PID in the time-domain for a step response and **Figure 5.9** shows the comparison between the control signal for both types of controllers when increasing and decreasing the time constant.

Figure 5.10 presents a comparison between FOPID based on IFM vs. immune PID in the time-domain for a step response and **Figure 5.11** shows the comparison between the control signal for both types of controllers when increasing and decreasing the time-delay.

From the numerical simulation results, it is shown that the proposed controller shows good performance with less effort and without overshoot, whereas the immune PID controller present a smaller overshoot when decreasing the time constant and a big overshoot when increasing the time delay.

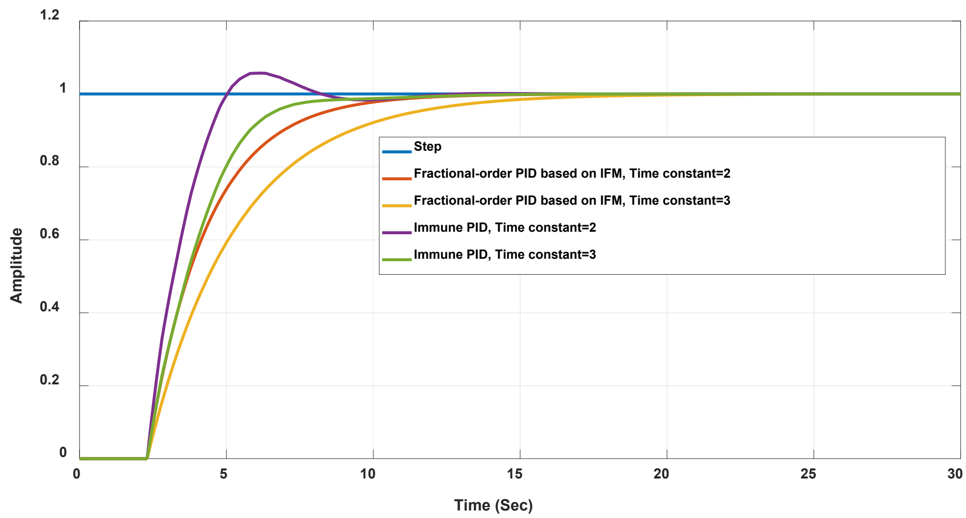

Figure 5.8 Reference tracking results when varying time constant for FOPDT system

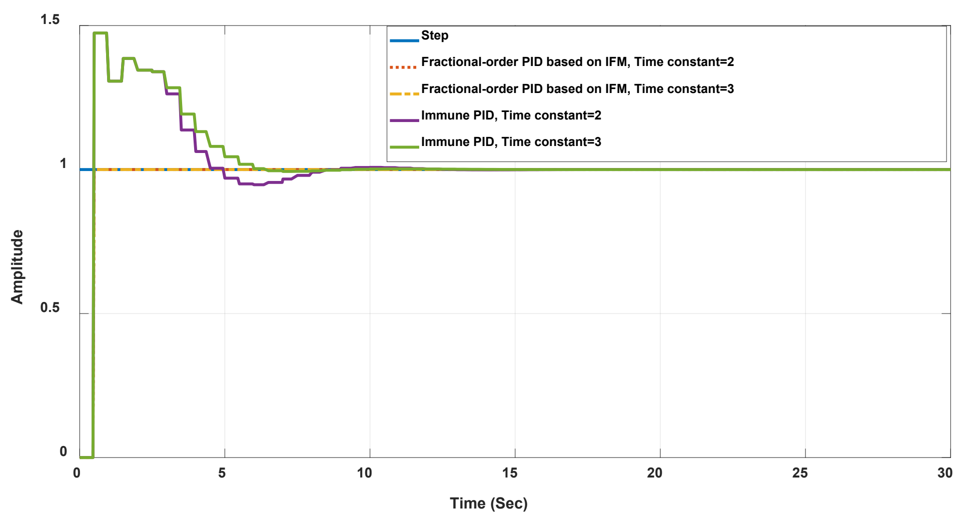

Figure 5.9 Control signal when varying time constant for FOPDT system

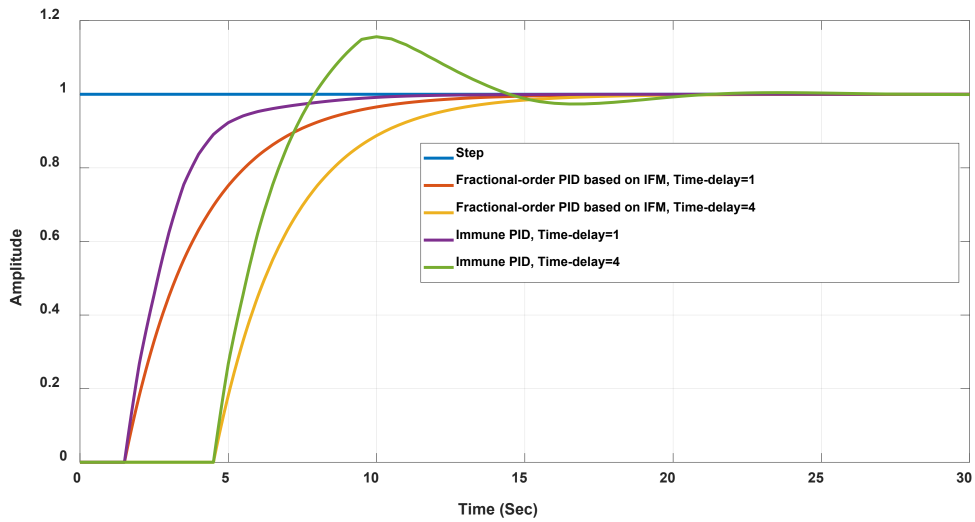

Figure 5.10 Reference tracking results when varying time-delay for FOPDT system

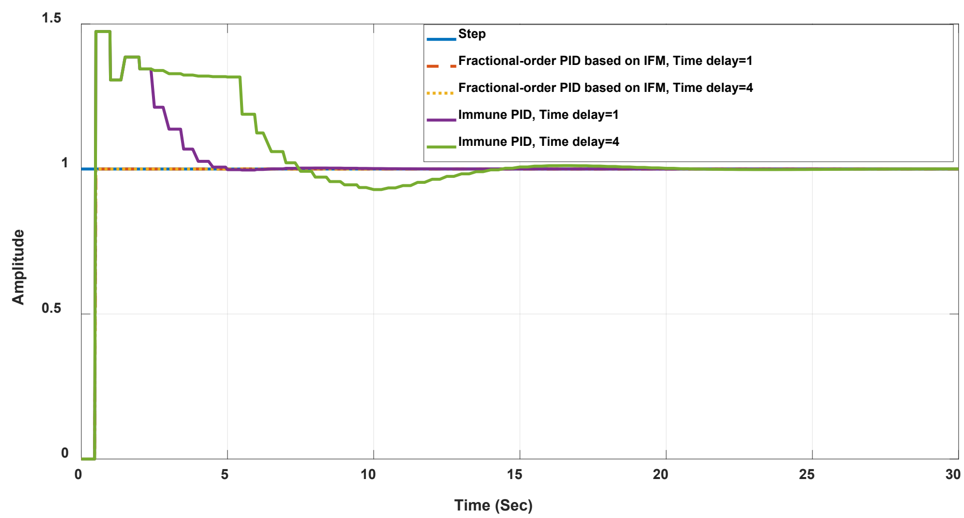

Figure 5.11 Control signal when varying time-delay for FOPDT system

Figure 5.12 shows load disturbance when using a random number as an input, the numerical simulation results show fast setting time and good rejection compared with the immune PID controller.

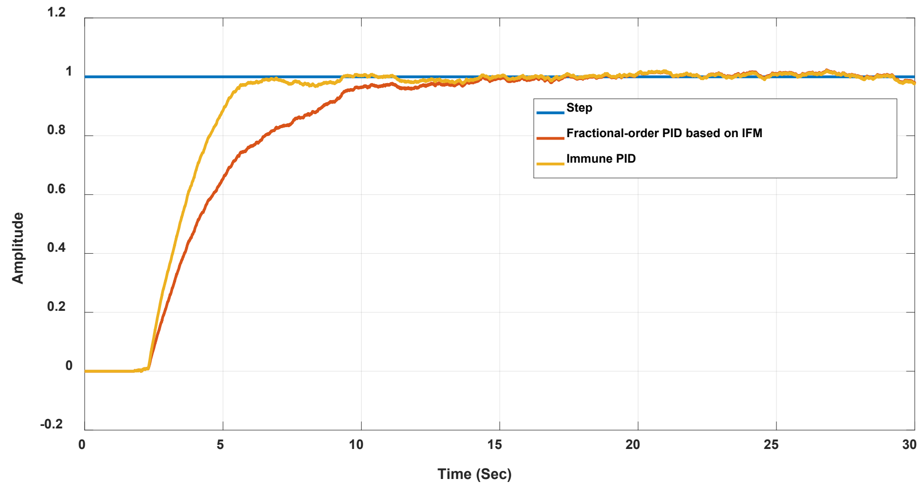

Figure 5.12 Load disturbance rejection for a random number disturbance for FOPDT system

5.8 Delay dominant system

The Ziegler–Nichols tuning method has been used to initialize the GA, the parameters are listed in **Table 5.4**, and the final optimal parameters obtained from GA are listed in **Table 5.5**

Table 5.4 PID parameters computed using ZN tuning rule for delay dominant system

K_p	K_i	K_d
0.9119	0.1661	1.2516

Table 5.5 FOPID based on IFM parameters computed using GA tuning for delay dominant system

K_p	K_i	K_d	λ	μ	k	n	c
1.8257	0.0079346	3.1803	0.066297	0.99368	0.257	4.427	0.093956

After running GA, the optimization results are shown in **Figure 5.13** comparing the fitness value vs. generation., in addition it gives the percentage of the stopping criteria.

Figure 5.14 presents the comparison between FOPID based on IFM vs. immune PID in the time-domain for a step response and **Figure 5.15** shows the comparison between the control signal for both types of controllers. The time domain specifications are summarized in that **Table 5.6**

From the numerical simulation results, it is shown that the proposed controller has the same performance of immune PID controller but with less effort.

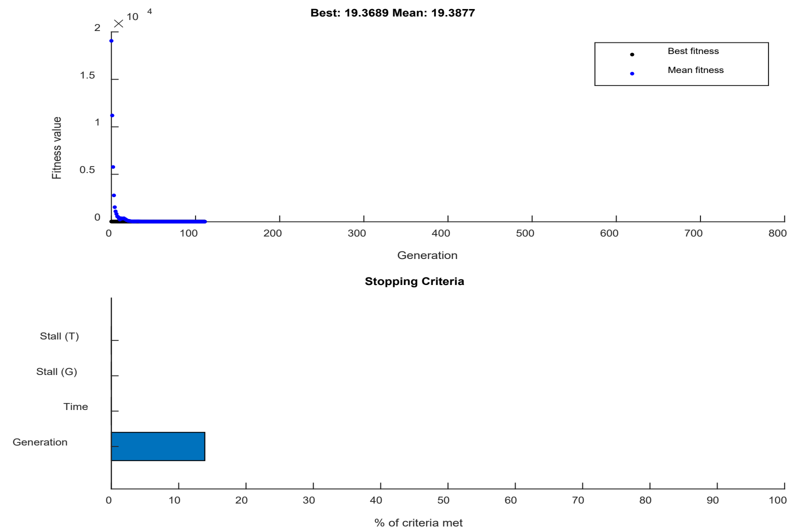

Figure 5.13 GA optimization results for delay dominant system

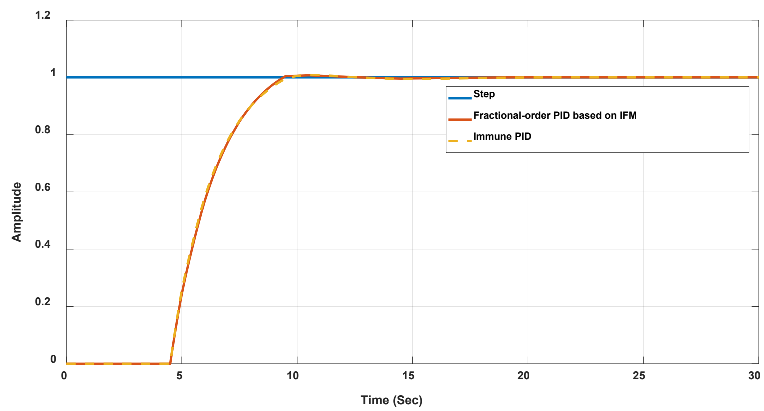

Figure 5.14 Step response of both FOPID using IFM and immune PID for delay-dominant system

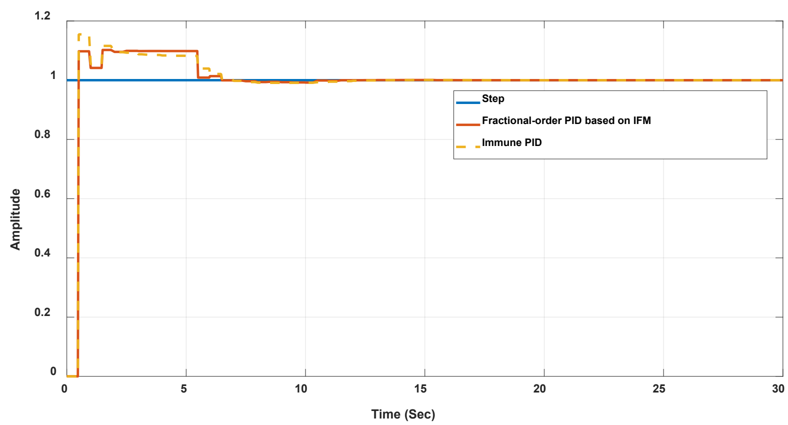

Figure 5.15 Control signal of both FOPID using IFM and immune PID for delay-dominant system

Table 5.6 Comparison of time domain performances for delay-dominant system

Controller used	Peak (%)	Settling Time (s)	Rise Time (s)	Steady State Error (%)
Immune PID	No overshoot	8.6	8.0	0
FOPID based on IFM	No overshoot	8.6	8.0	0

To evaluate the robustness, we change the time-delay for the system to simulate modeling uncertainties.

Figure 5.16 presents a comparison between FOPID based on IFM vs. immune PID in the time-domain for a step response and **Figure 5.17** shows the comparison between the control signal for both types of controllers when increasing and decreasing the time delay.

From the numerical simulation results, it is shown that the proposed controller shows the same performance with slightly less effort.

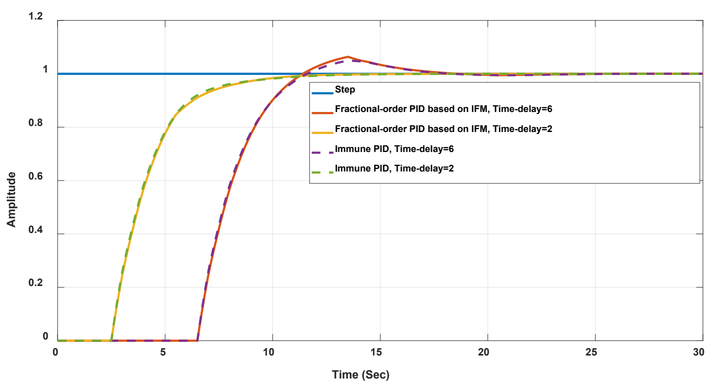

Figure 5.16 Reference tracking results when varying time-delay for delay-dominant system

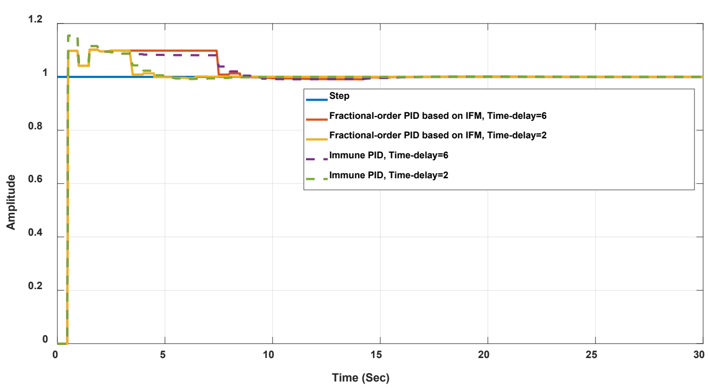

Figure 5.17 Control signal when varying time-delay for delay-dominant system

5.9 Conclusion

A new FOPID controller based on IFM is developed, the controller-parameters are optimized using EA, in this article we have used the GA. The design control law is used to control time delay systems. Comparative simulation results with immune PID controller demonstrate the effectiveness of the proposed control technic in terms of dynamic tracking of the reference and disturbance rejection.

General conclusion and perspective

In recent years, there is a huge focus on techniques used to create machines capable of simulating human intelligence. In this context, the scientific research in the field of CI is experiencing great progress, different CI paradigms reaches a mature and stable phase enough to be used by the industry. In another hand, FOC which uses the possibility of defining non-integer powers of the operators of derivation and integration of automation controllers is being used in different fields specifically in control engineering of various processes as it has been shown that this type of controllers increase the performance and robustness of the control loop.

This thesis has as objective using both advantages of CI and FOC, we have used the CI to increase the degree of freedom of the FOPID controller and enhance its performance when controlling industrial systems. The proposed design controllers have been validated by simulation in MATLAB.

In the introduction, a general review of the state of the art of the principal important work developed to improve the performance of FOC targeting the control of industrial systems, mainly time delay systems, UGVs and UAVs, and then we provide in the first chapter an introduction to FC theory that will help us understand the design of FOC. The second chapter provides a brief overview of CI field part of artificial intelligence with its different subcategories mainly FS and AIS. This will help the reader to have a holistic overview of this scientific field. the theories provide deep explanation that will help us understand the design of the proposed controllers. In the third chapter we proposed the first design control law based on the combination of FS and FOC. The proposed controller has been tested by simulation to control the liquid level in three tank system. The fourth chapter focus on the AIS theory pat of CI and the design of a FOC based on IFM in a classical control loop, The proposed controller has been tested by simulation to control time delay-systems.

As a perspective work, we propose to extend the application of this type of controllers to UGVs and UAVs and other type of processes, and it will be also worth to develop an efficient method to implement these types of controllers. Other powerful techniques can be also used to tune the controller parameters such as the use of swarm intelligence, other evolutionary algorithms…etc online tunning can be also considered.

Bibliography

[1] D. Boudjehem and B. Boudjehem, “Robust Fractional Order Controller for Chaotic Systems,” *IFAC-PapersOnLine*, vol. 49, no. 9, pp. 175–179, Jan. 2016, doi: 10.1016/J.IFACOL.2016.07.525.

[2] B. Boudjehem and D. Boudjehem, “Fractional order controller design for desired response,” *http://dx.doi.org/10.1177/0959651812456647*, vol. 227, no. 2, pp. 243–251, Oct. 2012, doi: 10.1177/0959651812456647.

[3] D. Boudjehem, M. Sedraoui, and B. Boudjehem, “A fractional model for robust fractional order Smith predictor,” *Nonlinear Dyn*, vol. 73, no. 3, pp. 1557–1563, Aug. 2013, doi: 10.1007/S11071-013-0885-9/METRICS.

[4] M. Dulău, A. Gligor, and T. M. Dulău, “Fractional Order Controllers Versus Integer Order Controllers,” *Procedia Eng*, vol. 181, pp. 538–545, Jan. 2017, doi: 10.1016/J.PROENG.2017.02.431.

[5] H. G. Sun, Y. Zhang, D. Baleanu, W. Chen, and Y. Q. Chen, “A new collection of real world applications of fractional calculus in science and engineering,” *Commun Nonlinear Sci Numer Simul*, vol. 64, pp. 213–231, Nov. 2018, doi: 10.1016/J.CNSNS.2018.04.019.

[6] C. I. Muresan, I. Birs, C. Ionescu, E. H. Dulf, and R. De Keyser, “A Review of Recent Developments in Autotuning Methods for Fractional-Order Controllers,” *Fractal and Fractional 2022, Vol. 6, Page 37*, vol. 6, no. 1, p. 37, Jan. 2022, doi: 10.3390/FRACTALFRACT6010037.

[7] A. Yüce, N. Tan, and D. P. Atherton, “Fractional Order PI Controller Design for Time Delay Systems,” *IFAC-PapersOnLine*, vol. 49, no. 10, pp. 94–99, Jan. 2016, doi: 10.1016/J.IFACOL.2016.07.487.

[8] T. Sadalla, D. Horla, W. Giernacki, and P. Kozierski, “Influence of time delay on fractional-order PI-controlled system for a second-order oscillatory plant model with time delay,” *Archives of Electrical Engineering*, vol. 66, no. 4, pp. 693–704, Dec. 2017, doi: 10.1515/AEE-2017-0052.

[9] A. Jajarmi and D. Baleanu, “Suboptimal control of fractional-order dynamic systems with delay argument,” *https://doi.org/10.1177/1077546316687936*, vol. 24, no. 12, pp. 2430–2446, Jan. 2017, doi: 10.1177/1077546316687936.

[10] Y. Q. Chen, T. Bhaskaran, and D. Xue, “Practical tuning rule development for fractional order proportional and integral controllers,” *J Comput Nonlinear Dyn*, vol. 3, no. 2, Apr. 2008, doi: 10.1115/1.2833934/465239.

[11] F. Padula and A. Visioli, “Tuning rules for optimal PID and fractional-order PID controllers,” *J Process Control*, vol. 21, no. 1, pp. 69–81, Jan. 2011, doi: 10.1016/J.JPROCONT.2010.10.006.

[12] C. A. Monje, B. M. Vinagre, V. Feliu, and Y. Q. Chen, “Tuning and auto-tuning of fractional order controllers for industry applications,” *Control Eng Pract*, vol. 16, no. 7, pp. 798–812, Jul. 2008, doi: 10.1016/J.CONENGPRAC.2007.08.006.

[13] C. I. Muresan and R. De Keyser, “Revisiting Ziegler-Nichols. A fractional order approach,” *ISA Trans*, vol. 129, no. Pt A, pp. 287–296, Oct. 2022, doi: 10.1016/J.ISATRA.2022.01.017.

[14] B. A. Francis and W. M. Wonham, “The internal model principle of control theory,” *Automatica*, vol. 12, no. 5, pp. 457–465, Sep. 1976, doi: 10.1016/0005-1098(76)90006-6.

[15] C. I. Muresan, A. Dutta, E. H. Dulf, Z. Pinar, A. Maxim, and C. M. Ionescu, “Tuning algorithms for fractional order internal model controllers for time delay processes,” *http://dx.doi.org/10.1080/00207179.2015.1086027*, vol. 89, no. 3, pp. 579–593, Mar. 2015, doi: 10.1080/00207179.2015.1086027.

[16] Otto J. M. Smith, “Closer Control of Loops with Dead Time,” *Chem Eng Prog*, vol. 53, no. 5, pp. 2017–2019, 1957.

[17] P. Lanusse and A. Oustaloup, “Control of Time-Delay Systems Using Robust Fractional-Order Control and Robust Smith Predictor Based Control,” *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference - DETC2005*, vol. 6 B, pp. 1475–1483, Jun. 2008, doi: 10.1115/DETC2005-84460.
[18] V. Feliu-Batlle, R. Rivas Pérez, F. J. Castillo García, and L. Sanchez Rodriguez, “Smith predictor based robust fractional order control: Application to water distribution in a main irrigation canal pool,” *J Process Control*, vol. 19, no. 3, pp. 506–519, Mar. 2009, doi: 10.1016/J.JPROCONT.2008.05.004.
[19] M. Yousefi and T. Binazadeh, “Delay-independent sliding mode control of time-delay linear fractional order systems,” *https://doi.org/10.1177/0142331216678059*, vol. 40, no. 4, pp. 1212–1222, Dec. 2016, doi: 10.1177/0142331216678059.
[20] S. Tiwari, G. Kaur, and H. Singh, “Analysis of Fuzzy PID and Immune PID Controller for Three Tank Liquid Level Control,” 2011.
[21] R. K. Jatoth, A. K. Jain, and T. Phanindra, “Liquid level control of three tank system using hybrid GA-PSO algorithm,” *2013 Nirma University International Conference on Engineering, NUiCONE 2013*, 2013, doi: 10.1109/NUICONE.2013.6780189.
[22] M. Uvais, P. R. Sarkar, and S. Mukherjee, “Three Tank Interacting System Level Control using Modern AI Techniques,” 2016.
[23] M. Ö. Efe, “Neural network assisted computationally simple PIλD μ control of a quadrotor UAV,” *IEEE Trans Industr Inform*, vol. 7, no. 2, pp. 354–361, May 2011, doi: 10.1109/TII.2011.2123906.
[24] X. Shi, B. Hu, C. Yin, Y. Cheng, and X. Huang, “Design of trajectory tracking controller with fractional-order backstepping sliding mode method for quadrotor UAV,” *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, pp. 5960–5965, Jul. 2018, doi: 10.1109/CCDC.2018.8408175.
[25] A. Govea-Vargas, R. Castro-Linares, M. A. Duarte-Mermoud, N. Aguila-Camacho, and G. E. Ceballos-Benavides, “Fractional Order Sliding Mode Control of a Class of Second Order Perturbed Nonlinear Systems: Application to the Trajectory Tracking of a Quadrotor,” *Algorithms 2018, Vol. 11, Page 168*, vol. 11, no. 11, p. 168, Oct. 2018, doi: 10.3390/A11110168.
[26] A. M. Jensen, D. K. Geller, and Y. Chen, “Monte Carlo simulation analysis of tagged fish radio tracking performance by swarming unmanned aerial vehicles in fractional order potential fields,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 74, no. 1–2, pp. 287–307, Apr. 2014, doi: 10.1007/S10846-013-9949-9/METRICS.
[27] M. Ö. Efe, “Battery power loss compensated fractional order sliding mode control of a quadrotor UAV,” *Asian J Control*, vol. 14, no. 2, pp. 413–425, Mar. 2012, doi: 10.1002/ASJC.340.
[28] C. Coopmans, A. M. Jensen, and Y. Chen, “Fractional-order complementary filters for small unmanned aerial system navigation,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 73, no. 1–4, pp. 429–453, Oct. 2014, doi: 10.1007/S10846-013-9915-6/METRICS.
[29] Z. Yu, Y. Qu, C. Y. Su, and Y. Zhang, “Distributed Fractional-Order Finite-Time Control for Multiple Unmanned Aerial Vehicles,” in *2018 IEEE Conference on Control Technology and Applications, CCTA 2018*, Institute of Electrical and Electronics Engineers Inc., Oct. 2018, pp. 1058–1063. doi: 10.1109/CCTA.2018.8511405.
[30] Z. Yu, Y. Zhang, Y. Qu, and Z. Xing, “Adaptive Fractional-Order Fault-Tolerant Tracking Control for UAV Based on High-Gain Observer,” in *Proceedings of the ASME Design Engineering Technical Conference*, American Society of Mechanical Engineers Digital Collection, Nov. 2017. doi: 10.1115/DETC2017-67479.
[31] I. Tejado, V. Milanes, J. Villagra, and B. M. Vinagre, “Fractional network-based control for vehicle speed adaptation via vehicle-to-infrastructure communications,” *IEEE Transactions on*

Control Systems Technology, vol. 21, no. 3, pp. 780–790, 2013, doi: 10.1109/TCST.2012.2195494.
[32] J. Moreau, P. Melchior, S. Victor, F. Aioun, and F. Guillemard, "Path planning with fractional potential fields for autonomous vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14533–14538, Jul. 2017, doi: 10.1016/J.IFACOL.2017.08.2076.
[33] K. B. Oldham and J. Spanier, *The fractional calculus : theory and applications of differentiation and integration to arbitrary order*. Academic Press, 1974.
[34] B. M. Vinagre, I. Podlubny, A. Hernandez, and A. Hernandez, "Some approximations of fractional order operators used in control theory," *Fract Calc Appl Anal*, vol. 3, no. 3, pp. 231–248, 2000, Accessed: Jun. 12, 2023. [Online]. Available: https://www.researchgate.net/publication/228392562_Some_approximations_of_fractional_order_operators_used_in_control_theory
[35] A. Tepljakov, E. Petlenkov, and J. Belikov, "FOMCON : a MATLAB Toolbox for Fractional-order System Identification and Control," vol. 2, no. 2, 2011.
[36] Duarte Valério, "Toolbox ninteger for MATLAB." Accessed: Oct. 26, 2022. [Online]. Available: http://web.ist.utl.pt/duarte.valerio/ninteger/ninteger.htm
[37] I. Podlubny, "Fractional-order systems and PIλDμ-controllers," *IEEE Trans Automat Contr*, vol. 44, no. 1, pp. 208–214, 1999, doi: 10.1109/9.739144.
[38] D. Matignon, "Stability results for fractional differential equations with applications to control processing," 1996.
[39] D. Xue, Y. Chen, and D. P. Atherton, *Linear feedback control : analysis and design with MATLAB*. Society for Industrial and Applied Mathematics, 2007. Accessed: Jun. 18, 2023. [Online]. Available: https://nl.mathworks.com/academia/books/linear-feedback-control-xue.html
[40] C. Darwin and W. Bynum, *The origin of species by means of natural selection: or, the preservation of favored races in the struggle for life*. 1859. Accessed: Aug. 10, 2023. [Online]. Available: https://www.ias.ac.in/public/Volumes/reso/014/02/0204-0208.pdf
[41] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, 1992. doi: 10.7551/MITPRESS/1090.001.0001.
[42] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995, doi: 10.1109/ICNN.1995.488968.
[43] A. P. Engelbrecht, *Computational Intelligence: An Introduction: Second Edition*. John Wiley and Sons, 2007. doi: 10.1002/9780470512517.
[44] S. Forrest, L. Allen, A. S. Perelson, and R. Cherukuri, "Self-nonself discrimination in a computer," *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 202–212, 1994, doi: 10.1109/RISP.1994.296580.
[45] L. N. De Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 239–251, Jun. 2002, doi: 10.1109/TEVC.2002.1011539.
[46] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965, doi: 10.1016/S0019-9958(65)90241-X.
[47] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol Rev*, vol. 65, no. 6, pp. 386–408, Nov. 1958, doi: 10.1037/H0042519.
[48] B. Widrow and M. Hoff, "Adaptive switching circuits," *IRE WESCON convention record*, pp. 96–104, 1960, Accessed: Jul. 06, 2023. [Online]. Available: https://www.scirp.org/(S(351jmbntvnsjt1aadkposzje))/reference/ReferencesPapers.aspx?ReferenceID=55352

[49] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int J Man Mach Stud*, vol. 7, no. 1, pp. 1–13, Jan. 1975, doi: 10.1016/S0020-7373(75)80002-2.
[50] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Trans Syst Man Cybern*, vol. SMC-15, no. 1, pp. 116–132, 1985, doi: 10.1109/TSMC.1985.6313399.
[51] M. T. Khan and C. W. De Silva, "Autonomous fault tolerant multi-robot cooperation using artificial immune system," *Proceedings of the IEEE International Conference on Automation and Logistics, ICAL 2008*, pp. 623–628, 2008, doi: 10.1109/ICAL.2008.4636225.
[52] Jon Timmis, Peter J. Bentley, and Emma Hart, *Artificial Immune Systems: Second International Conference, ICARIS 2003, Edinburgh, UK, September 1-3, 2003, Proceedings | SpringerLink*. 2003. Accessed: Jun. 24, 2023. [Online]. Available: https://link.springer.com/book/10.1007/b12020
[53] H. Jabeen and A. R. Baig, "CLONAL-GP framework for artificial immune system inspired genetic programming for classification," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6276 LNAI, no. PART 1, pp. 61–68, 2010, doi: 10.1007/978-3-642-15387-7_10/COVER.
[54] B. Chen and C. Zang, "Emergent damage pattern recognition using immune network theory," *Smart Struct Syst*, vol. 8, no. 1, pp. 69–92, 2011, doi: 10.12989/SSS.2011.8.1.069.
[55] Q. Xu, S. Wang, and C. Zhang, "Structural design of the danger model immune algorithm," *Inf Sci (N Y)*, vol. 205, pp. 20–37, Nov. 2012, doi: 10.1016/J.INS.2012.04.011.
[56] R. R. Muddada, F. Zhang, M. K. Tiwari, and W. J. Zhang, "A new index to evaluate solutions in the clonalg algorithm: Structural similarity index," *Advances in Intelligent and Soft Computing*, vol. 66 AISC, pp. 1119–1132, 2010, doi: 10.1007/978-3-642-10430-5_86/COVER.
[57] R. Shang, L. Jiao, F. Liu, and W. Ma, "A novel immune clonal algorithm for MO problems," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 35–50, Feb. 2012, doi: 10.1109/TEVC.2010.2046328.
[58] G. Costa Silva, R. M. Palhares, and W. M. Caminhas, "Immune inspired Fault Detection and Diagnosis: A fuzzy-based approach of the negative selection algorithm and participatory clustering," *Expert Syst Appl*, vol. 39, no. 16, pp. 12474–12486, Nov. 2012, doi: 10.1016/J.ESWA.2012.04.066.
[59] P. A. D. Castro and F. J. Von Zuben, "MOBAIS: A Bayesian artificial immune system for multi-objective optimization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5132 LNCS, pp. 48–59, 2008, doi: 10.1007/978-3-540-85072-4_5/COVER.
[60] P. A. D. Castro and F. J. Von Zuben, "A Gaussian Artificial Immune System for multi-objective optimization in continuous domains," *2010 10th International Conference on Hybrid Intelligent Systems, HIS 2010*, pp. 159–164, 2010, doi: 10.1109/HIS.2010.5600022.
[61] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," *Proc Natl Acad Sci U S A*, vol. 79, no. 8, pp. 2554–2558, 1982, doi: 10.1073/PNAS.79.8.2554.
[62] K. Takahashi and T. Yamada, "Application of an Immune Feedback Mechanism to Control Systems," *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, vol. 41, no. 2, pp. 184–191, Jun. 1998, doi: 10.1299/JSMEC.41.184.
[63] W. Wang, X. Z. Gao, and C. Wang, "A New Immune PID Controller Based on Immune Tuning," *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques*, pp. 1337–1346, Aug. 2007, doi: 10.1007/978-3-540-74282-1_149.

[64] A. Makhbouche, B. Boudjehem, I. Birs, and C. I. Muresan, “Fractional-Order PID Controller Based on Immune Feedback Mechanism for Time-Delay Systems,” *Fractal and Fractional 2023, Vol. 7, Page 53*, vol. 7, no. 1, p. 53, Jan. 2023, doi: 10.3390/FRACTALFRACT7010053.