People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Université de Boumerdes
University of Boumerdes

Institute of Electrical and Electronic Engineering
Department of Power and Control Engineering

Final Year Project Report Presented in Partial Fulfilment of the Requirements for the Degree of

MASTER

In **Control Engineering**

Option: **Control Engineering**

Title:

PID Controller Parameters Tuning Using Metaheuristic Methods For Quadrotor UAV

Presented by:

Abdelaziz Mouloud Aghilas

Supervisor:

Dr. Abdelkarim Ammar

Registration Number: 2021/2022

Acknowledgement

First thanks to god almighty for all his granted blessings on me. i would like to thank my supervisor **Mr Abdelkarim Ammar** for his assistance through this work. This endeavor would not have been possible without **Mr Belmadani Hamza** i am also grateful to my classmates for their contributions by showing me some tips, which helped me to do this work in good conditions.

Dedication

I dedicate this work to my father and my mother, who with love and effort have accompanied me in this process, they always encourage me to push the limits, also my sister may Allah bless her.
to my best friend Redouane who supported me during this work by influencing me by positive energy.
of course, a dedication to all my friends and mates inside and outside university

Abstract

Since quadrotors are gaining more popularity in research area, due to their control complexity and their highly complicated nature. Although there have been many techniques designed to control quadrotor systems, most of them are characterized with high computational burden. Therefore PID controllers are preferred for their simplicity and effective performance, however in order to achieve such level of efficiency their gains need to be tuned properly. In this work, three well-known metaheuristic algorithms are applied for that purpose, namely: PSO, TLBO, and WOA. The three optimizers were employed to extract the optimum PID gains for altitude, Roll, yaw and pitch angles. A comparative study on the basis of the Integral of Absolute Error, Settling time, Rise Time, and overshoot was conducted to examine the introduced approaches. In general the results have demonstrated the superior performance of the PSO algorithm over its counterparts in most the investigated aspects.

Contents

Acknowledgment **I**

Dedication **II**

Abstract **III**

1 General introduction **1**

2 REVIEW **2**
2.1 Introduction: 2
2.2 Review about quadrotor: 2
2.2.1 What is a quadrotor ? 2
2.2.2 Application: 3
2.2.3 Control methods of quadrotor: 3
2.3 PID tuning using meta-heuristic algorithms: 4
2.4 Conclusion 6

3 Quadrotor modelling and control **7**
3.1 Quadrotor dynamics 7
3.1.1 Assumptions 7
3.1.2 Euler newton model 8
3.2 PID controller: 10
3.2.1 Introduction: 10
3.2.2 PID functions for reducing error 10
3.3 Conclusion 11

4 **12**
4.1 Optimization techniques 12
4.1.1 Introduction 12
4.1.2 Meta-heuristic algorithms types 12
4.2 PSO algorithm 12
4.2.1 PSO algorithm description 13
4.2.2 Description 13
4.3 TLBO algorithm 14
4.3.1 Description 14
4.3.2 Mathematical model and optimization algorithm 15
4.4 Whale optimization algorithm WOA 18
4.4.1 Description 18
4.4.2 Mathematical model and optimization algorithm 18
4.5 Conclusion 22

5 Results and discussion **23**
5.1 Introduction 23
5.2 Set up parameters 23
5.3 Objective functions 23
5.4 Results 24
5.4.1 Conclusion: 28

List of Figures

2.1 Quadrotor . . . 2

3.1 The direction of rotation of motors . . . 7
3.2 reference frames . . . 8
3.3 PID controller . . . 10

4.1 PSO algorithm flowchart . . . 14
4.2 TLBO algorithm flowchart . . . 17
4.3 Bubble-net feeding motion[10] . . . 18
4.4 the updated positions path [10] . . . 20
4.5 spiral path for updating position . . . 21
4.6 Whale algorithm flowchart [19] . . . 22

5.1 Altitude step response . . . 24
5.2 Theta angle step response . . . 25
5.3 Phi angle step response . . . 26
5.4 Psi angle step response . . . 28

List of Tables

2.1 Characteristics of Metha-heuristic algorithms 5

5.1 Tables of setted up parameters 23
5.2 The response of the altitude 25
5.3 Gains and final fitness values of altitude Z 25
5.4 The response of the angle θ 26
5.5 Gains and final fitness values for angle θ 26
5.6 The response of the angle ϕ 27
5.7 Gains and final fitness values for angle ϕ 27
5.8 The response of the angel ψ 28
5.9 Gains and final fitness values of angle ψ 28
5.10 The Quadrotor Model Parameters 31

list of abbreviation

UAVs Unmanned Aerial Vehicles

PID Proportional-Integral-Derivative

SMC Sliding Mode Control

GA Genetic Algorithm

PSO Particle Swarm Algorithm

TLBO Teaching-Learning Based Optimization

WOA Whale Optimization Algorithm

Chapter 1

General introduction

Quadrotors, also known as unmanned aerial vehicles (UAVs), are one of the most fascinating issues for researchers. Because of its importance and wide range of uses in both military and civilian duties, many governments are prioritizing military applications of UAVs [1]. The quadrotor is an underactuated mechanical system that is difficult to operate due to its nonlinearity and underactuation. As a result, finding a good quadrotor controller remains a significant task. A growing number of scholars and researchers in the field of control are focusing their attention on quadrotor helicopters. PID control, sliding-mode control, integral backstepping control, and linear quadratic regulator are only a few of the control theories and approaches used to improve the system's performance and behavior under various conditions. It's difficult to fine-tune the parameters of a PID controller to achieve the appropriate transient response.

Nonlinear control approaches can precisely regulate the dynamics of a quadrotor model, but the PID controller can successfully deal with the problem of steady state inaccuracy in the modeling process. A quadrotor helicopter's three location coordinates and three attitude angles can be successfully controlled [2]. Conventional methods used to find PID parameters or manual trials according to engineer's experience have shown their limits in terms of efficiency and adaptability, especially when the application is as complex as a quadrotor control, therefore, a new techniques are used to compute the control parameters and optimize them simultaneously.

Thesis organization:

- **Chapter 1:** This chapter gives a general review on Quadrotors and their control techniques, also it involves optimization techniques and their types.

- **Chapter 2:** This chapter is about the mathematical model of a Quadrotor and its PID controller.

- **Chapter 3:** In this chapter the three used algorithms are explained in details.

- **Chapter 4:** This chapter results were discussed results and a comparative study was performed to conclude the best one among those algorithms.

Chapter 2

REVIEW

2.1 Introduction:

This chapter will focus on the generalities of each of the following: quadrotors, control techniques, and optimization algorithms of a PID controller, which is used to control a quadrotor in this study.

2.2 Review about quadrotor:

2.2.1 What is a quadrotor ?

Quadrotors are becoming more prevalent in the search area, promising efficient work and results by replacing many other old techniques and machines in a variety of fields, simply because they are small and unmanned (no onboard pilot required), allowing them to reach dangerous places that full-size helicopters cannot. Quadrotors can conduct a vertical take-off and landing, as well as forceful maneuvers with steep angles, despite their small size. In contrast to fixed-wing aircraft, which must constantly move forward to generate lift, they may hover [3].

Figure 2.1: Quadrotor

2.2.2 Application:

A quadrotor's flight maneuvers are so unique that they're ideal for a variety of applications, including traffic monitoring, source investigation, power line inspection, a direct attack on vital targets, aerial photography, and videography. A quadrotor can perform the activities stated above more effectively than a ground robot, although a ground robot may have advantages over a quadrotor in terms of load-carrying ability. Ground robots can carry big loads, are resilient in the event of a collision, and are not dependent on sensor tracking to remain stationary in one location, reducing energy consumption [3]. Despite of all the aforementioned advantages of the ground robot, a quadrotor acts as the most favorable robot for tasks where many other robots were not able to accomplish or not capable to do it at all, such as reaching rough terrain and navigation in space. The mechanical simplicity and design of a quadrotor considered as an advantageous point, however, energy aspect is the disadvantageous side.
Fast dynamics due to the low inertia moments also the effect of the aerodynamic and gyroscopic, controlling a quadrotor is such a tricky task especially when it is subjected to the environment and atmospheric disturbances, which will add more complexity to the system.

2.2.3 Control methods of quadrotor:

The quadrotor does not have complex mechanical control linkages due to the fact that it relies on fixed pitch rotors and uses the variation in motor speed for vehicle control [4]. However these advantages come at price as controlling a quadrotor is not easy because of the coupled dynamics and its commonly under actuated design configuration [5].
Many control techniques and algorithms are proposed to act efficiently with quadrotor due to its nature of dynamics. Of course, the one who chooses one of all algorithms should be aware of its advantages and disadvantages, also how to face such drawbacks of that selected technique.
when it is about to categorize techniques and algorithms, it is clear that linear and non-linear control techniques will be the main criterion to categorize. The algorithms proposed to this control system are as follows:

- **Linear quadratic regulator**.
  Assigning any suitable cost function and minimizing it to operate a dynamic system is the familly of LQR optimal control.

- **Sliding mode control (SMC)**:
  Sliding mode control belongs to non-linear control algorithms. It applies a discontinuous control signal to the system to command it to follow a desired path[6]. Using Lyapunov stability theory for a sliding mode controller, had a favorable results to drive a quadrotor with a good stability. the most important thing about sliding mode is, during the linearization, the dynamics are not simplified.

- **Backstepping control**:
  Backstepping is an algorithm that devides the controller into subsystems and stabilizes them progressively[6]. it is not favorable in terms of robustness, but the algorithm can deal with internal disturbances well. By applying Lyapunov stability theory, a good tracking for position and stable angles were achieved. adding an integrator to the algorithm will eliminates the steady-state errors, enhances time response and reduces overshoot.

- **Adaptive control algorithms:**
  The purpose of those algorithms is to adapt to varying and uncertain parameters. A continuous time-varying adaptive controller, which shows good performance, was implemented by Diao et al. In[7], with known uncertainties in mass, moments of inertia and aerodynamic. damping coefficients.

- **Robust control algorithm:**
  Disturbances and uncertainties in the system parameters are well managed by robust control algorithms, so enhancing the performance of a controller for un-modelled system parameters, but the downside of this algorithm is less ability for tracking.

- **Optimal control algorithm:**
  Form a mathematical model of a system, variables are selected to be minimized, then generating a best cost function, but is considered poor in terms of robustness.

- **Feedback linearization:**
  It is a transformation from non-linear system model into a linear system by changing variables, but this will affect the precision of the system during implementation.

- **Intelligent control (fuzzy logic and artificial neural networks):**
  Many artificial intelligence approaches are involved when it is about intelligent control, such as fuzzy logic, machine learning, neural networks and genetic algorithm. Intelligent systems could not be efficient due to uncertainty and mathematical complexity.

- **Hybrid control algorithms:**
  As the name hybridization indicates a combination of different types of algorithms. as discussed previously, each algorithm has its specific downsides, then a hybridization of algorithms will reduce, because advantages and disadvantages of algorithms are overlapped. For example, combining backstepping and sliding mode control.

- **Proportional Integral Derivative (PID):**
  The PID controller is widely used especially in industries processes. PID controller is simple to implement with a good robustness, but this can be achieved with only linear systems. The performance of a PID controller will be reduced while using it to control a quadrotor, due to the non-linearity of the system and the inaccurate mathematical modelling of the dynamics.

2.3 PID tuning using meta-heuristic algorithms:

Meta-heuristic is composed of two Greek words (meta ; heuristic), heuristic refers to ' find' and meta ' beyond, in an upper level' [8]. Algorithms such as meta-heuristics aim to solve complex problems by finding optimal solution. Meta-heuristic algorithms are increasingly used to solve engineering problems and many other disciplines due to their simple concept and implementation. Meta-heuristic algorithms inspire from nature by mimicking biological or physical phenomena, they are attracted by many researchers due to nature as a source of inspiration [8].

Last decades several meta-heuristic algorithms are developed, and many researchers still introduce new ones or enhancing already existing algorithms by making changes in parameters or mathematical functions. Meta-heuristic algorithms are distinguished according to their nature of inspiration and sorted as follows:
Evolution-based: This algorithm inspires from the lows of natural evolution. The process of this algorithm begins by initializing a number of population which are randomly generated, then best individuals are gathered forming a next generation, during each generation a set of all best individuals are selected to be transferred to next generation, until a well defined condition is satisfied. Genetic algorithm (GA) belongs to this family of algorithms, Genetic algorithm is mostly used with complex optimization problems with a considerable time complexity. GA is general algorithm, therefore it can be implemented for all search spaces. GA uses stochastic operators to search for a solution which will avoid local optimum, it deals with large parameter spaces, it can handle discrete and continuous parameters, easy to implement, and can generate many solution in a simple run, all previous mentioned advantages are according to [9].
Swarm-based: They are nature inspired methods, it mimics the social behavior of groups of animals. PSO algorithm is the most popular algorithm in this category, where a social behavior of bird flocking is imitated. Swarm-based algorithms, in general, have some advantages over evolution-based algorithms. Swarm-based algorithms, for example, keep search space information over iterations, whereas evolution-based algorithms destroy all knowledge as soon as a new population forms[10].
Human-based: Human's behavior is the source of inspiration. It acts like humans when they improve their level of searching.some algorithms belonging to this categories are teaching-learning-based optimization (TLBO), Tabu search (TS).
Physic-based: All strategies used in this category are based on the physical laws of the cosmos. This technique's fundamental idea is to imitate a certain physical or chemical law, such as electrical charges, gravity, river systems, chemical reactions, and so on. Simulated Annealing is the most popular Physic-based algorithm (SA). SA is one of the most well-known and often used local search metaheuristics derived from statistical mechanics.

The table bellow shows the various parameters and architectures of the various algorithms previously discussed:

Optimization	**Metaheuristic techniques**
decision variables	position, marks, gene, subjects
solution	population member, flame, learner, child, parent, moth, bee,stream
set of solutions	swarm, water body, class, flames, population
objective function	fitness, energy, nectar amount
iteration	generation, cycles

Table 2.1: Characteristics of Metha-heuristic algorithms

Exploration and exploitation, often known as diversification and intensification respectively, are the two most important aspects of metaheuristic algorithms [11]. The terms diversification and intensification were first discovered in the Tabu Search field. Diversification refers to the abilities of a search algorithm to identify a range of solutions inside different parts of the search space; nevertheless, this process is also known as global search. In contrast to diversification, intensification relates to the concept of improving the search process in order to locate a better solution; nonetheless, this approach is known as local search.

In contrast to diversification, intensification relates to the concept of improving the search process in order to locate a better solution; nonetheless, this approach is known as local search. Many metaheuristics must address both exploration and exploitation in a search space; if both exploration and exploitation are balanced, the implementation's performance will improve [11], [12]. According to [13], A balance between exploration and exploitation is reported to be achieved by contributions from both aspects, one of which contributes by quickly identifying a region in the search space with high-quality solution, while the other contributes by identifying the region in the search space from those that are either explored or do not provide high-quality solution.
Also, because one is known to change the speed of convergence for obtaining global optima, and the other is known to raise the chance of locating an area in search space where the global optima are located, these two elements are closely associated.

2.4 Conclusion

The definition of quadrotors and their applications are highlighted in this chapter, also a brief description about control techniques of a quadrotor, then we defined the meta-heuristic techniques and their natures.

Chapter 3

Quadrotor modelling and control

3.1 Quadrotor dynamics

Quadrotor concept consists of four rotors in cross configuration. Electric DC motors drive propellers through the reduction gears, also they have fixed-pitch blades, which are obviously designed in such a way to create upward lift. Two propellers have a clockwise rotation, while other two have counterclockwise rotation, see figure 3.1, this configuration is to balance the body and eliminates the necessity of tail rotor like in helicopters. The dynamic of a quadrotor lays on the speed of each/pairs of motors, for a vertical take-off or landing, increasing the speed of all propellers simultaneously and by the same amount of speed. Any difference in speed of propellers which are oppositely mounted, then a pitch or roll angles are achieved. A yaw angle is achieved by unbalancing the overall torque, due to the difference of speed between each pair of propellers.

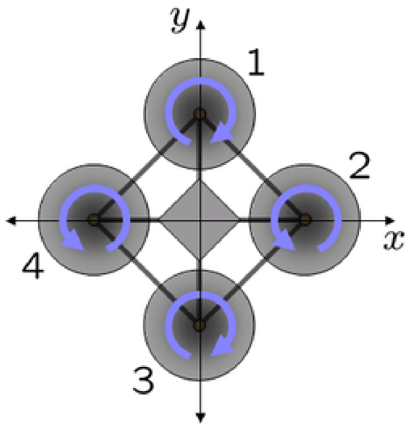

Figure 3.1: The direction of rotation of motors

3.1.1 Assumptions

The model used in this work has a set of assumptions concerning the physical, mechanical and aerodynamics consideration such as:

- The structure is supposed rigid. structure is supposed symmetrical.
- The CoG and the body fixed frame origin are assumed to coincide.

- The propellers are supposed rigid.
- Thrust and drag are proportional to the square of propeller's speed.

For a quadrotor modelling, gyroscopic effects must be considered including: gravity effect, friction, Aerodynamic effects and inertial counter torques.

3.1.2 Euler newton model

The frame proposed here to study the motion of a quadrotor is a six degrees of freedom (DOF), as shown in figure 3.2, we distinguish two types of motions, one is a total of three translations, the second is a three angular moments. All those motions are controlled by four commands. Keeping changing propellers speeds to achieve the desired altitude and torques to change angles. This leads to an under actuated control system.

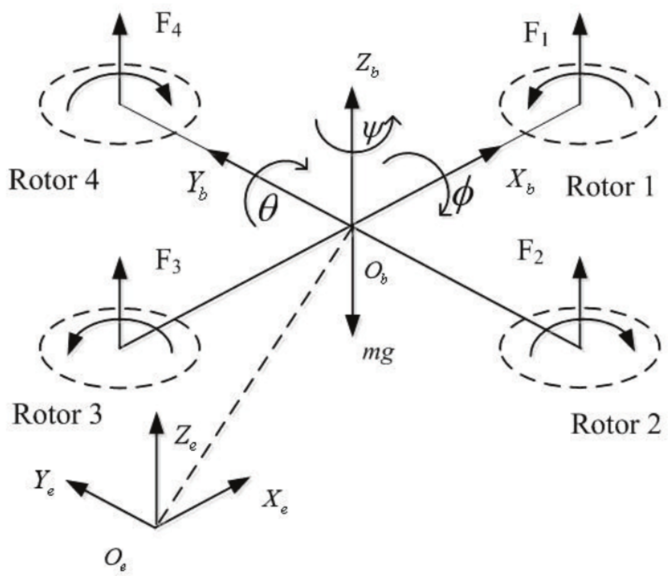

Figure 3.2: reference frames

Assuming an earth-fixed frame E and a body -fixed frame B, as shown in figure 3.2 . The position of the center of mass of the quadrotor can be referred as $\begin{bmatrix} x & y & z \end{bmatrix}^T$ in earth fixed frame (E). The transformation of movements coordinates and the angular velocity from B-frame to E-frame is performed by a rotation matrix, which is characterized as follows:

$$\mathrm{R}(\Theta) = \begin{bmatrix} c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \quad (3.1)$$

where : c( ) = cos( ) and s( ) = sin( )

- ψ: refers to the yaw angle.
- θ: refers to the pitch angle;

- ϕ: refers to the roll angle

;
The vector $\begin{bmatrix} p & q & r \end{bmatrix}^t$ represents the quadrotor angular velocity in B-frame, is also expressed with the first derivative of the roll, pitch, and yaw angles according to equation 3.2.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3.2}$$

We denote the position of the center of mass in earth frame E by **r**, the dynamics of the linear motion is represented by the equation 3.3 which represents the the acceleration of the center of mass.

$$\text{m}= \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \text{R} \begin{bmatrix} 0 \\ 0 \\ \sum F_i \end{bmatrix} \text{ (3.3)}$$

We also define the angular acceleration by equation 3.4 showing the dynamics of angular motion accornding to Euler.

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4) \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} * I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.4}$$

Where L is the distance from the axis of rotation of the rotors to the center of the quadrotor and $I = [I_x, I_y, I_z]^t$ is the inertia matrix.

The equation of motion that describes the movement in space [14]:

$$\begin{cases} \ddot{x} = (cos\psi sin\theta cos\phi + sin\psi sin\phi)\frac{U1}{m} \\ \ddot{y} = (sin\psi sin\theta cos\phi - cos\psi sin\phi)\frac{U1}{m} \\ \ddot{z} = (cos\phi cos\theta)\frac{U1}{m} - g \\ \ddot{\phi} = [lU2 + \ddot{\theta}\ddot{\psi}(Iz - Ix)]/Ix \\ \ddot{\theta} = [lU3 + \dot{\phi}\dot{\psi}(Iz - Ix)]/Iy \\ \ddot{\psi} = [U4 + \dot{\phi}\dot{\theta}(Ix - Iy)]/Iz \end{cases} \tag{3.5}$$

We can finally define control inputs as follows :

$$\begin{bmatrix} U1 \\ U2 \\ U3 \\ U4 \end{bmatrix} = \begin{bmatrix} F1 + F2 + F3 + F4 \\ F4 - F2 \\ F3 - F1 \\ F2 + F4 - F3 - F1 \end{bmatrix} = \begin{bmatrix} k_t \sum_{i=1}^{4} \omega_i^2 \\ k_t(\omega_4^2 - \omega_2^2) \\ k_t(\omega_3^2 - \omega_1^2) \\ k_d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^{2)} \end{bmatrix} \tag{3.6}$$

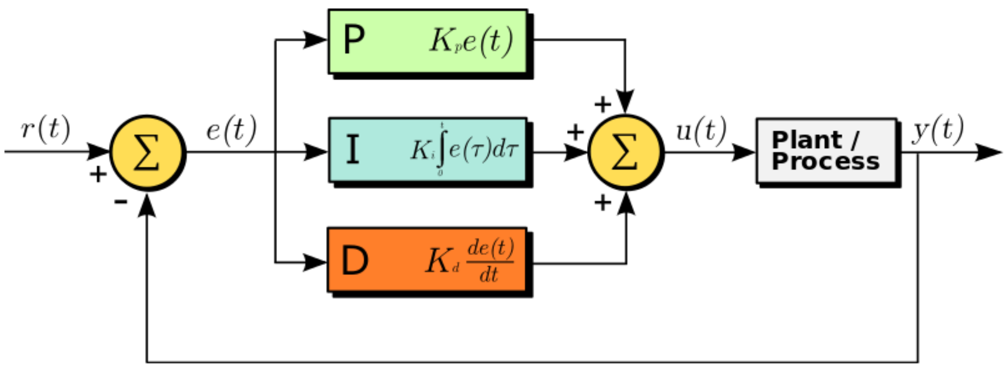

Figure 3.3: PID controller

Where: U1 is the vertical velocity control input, U2 is the roll control input, U3 is the pitch control input, U4 is the yaw control input, ω is the speed of rotors, F1 is the pu11 of rotors.

3.2 PID controller:

3.2.1 Introduction:

In industrial control systems, the proportional–integral–derivative (PID) controller remains the most important component. Its popularity stems from its straightforward structure, which is simple to comprehend and implement. It also delivers consistent results over a wide range of operational situations. Despite their ubiquity, academics and plant operators find it difficult to tune PID settings. Several methods for calculating the PID parameters have been established in the literatures, the first of which was discovered by Ziegler Nichols tuning [15]. A PID controller is a type of feedback controller that's commonly used in industrial control systems, process control, motor driving, and instrumentation.
The above figure 3.3 shows the circuit diagram of a PID controller in a closed loop system.
PID controller consist of three terms proportional, integral and derivative. The PID controller determines how much and how quickly correction is applied by using varying amounts of Proportional, Integral, and Derivative action.

3.2.2 PID functions for reducing error

a general mathematical equation for a PID controller is:

- **Frequency domain:**

  U(S)= $\mathrm{K}_p + \frac{K_i}{S} + K_d S$ (3.7)

- **Time domain:**

  $\mathrm{U}_c(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_d \frac{d}{dt} e(t)$

The proportional block generates an output signal proportional to the Error Signal's magnitude. This block speeds up the response since the closed loop time constant drops with the proportional term but does not change the order of the system as the output is only proportional to the input. The proportional block reduces error but does not eliminate the steady state error.
The integral block creates an output proportional to the duration and magnitude of the Error Signal. The longer the error and the greater the amount, the larger the integral output. The integral block eliminates the offset when the order of the system is increased by 1. This block also increases the system response speed, but with oscillations. The derivative block creates an output signal proportional to the rate of change of the error signal.The faster the error changes, the larger the derivative output. This block mainly lowers the system's oscillatory responsiveness. It has no effect on the offset also it does not change the type or order of the system.
As explained above, the role of each block is such sensitive to the response of a system, therefore for a good design for this controller consists of a well understanding and study of the system to select the right parameters of the three blocks.

3.3 Conclusion

This chapter highlights the dynamics of a quadrotor and its mathematical model, with a review about PID controller.

Chapter 4

4.1 Optimization techniques

4.1.1 Introduction

Engineering is a field which deals with daily technical problems and complexities, as many other areas, engineers always strive for solving problems, they do not convince just by finding a solution, but also how optimal is this solution.
It is clear that, when talking about optimal solutions, we mean a most satisfying results in terms of: efficiency, energy,cost, time complexity and system behaviour or a trade-off between the aforementioned parameters.
Optimizing a PID controller for some application is such a challenging task, this chapter introduces some optimization techniques based on meta-heuristic algorithms, such as particle swarm optimization (PSO), teaching learning-based optimization(TLBO), whale optimization algorithm (WAO).

4.1.2 Meta-heuristic algorithms types

We distinguish two main types of meta-heuristic algorithms based en their principle of working :

- **Individual-based**: This technique lays on only one particle to search for an optimal solution.
- **Population-based**: Where a size of population is considered as possible solutions, each particle share its own experience trying to give an optimal solution. This technique is inspired from many different categories: physics,human, swarm and evolution.

4.2 PSO algorithm

The particle swarm optimization (pso) was firstly introduced by James Kennedy and Russell C.Eberhart in 1995 [16], to simulate the behaviour of birds flocking. Investigating through the defined search space and explore most acceptable solutions, in general to minimize a certain objective function.

4.2.1 PSO algorithm description

4.2.2 Description

This algorithm first initiates the number of particles, which represent a swarm. each particle is considered as a candidate to generate a solution. For each iteration , every particle tries to get closer to some value such that it minimizes the fitness function. Particles generate new solution based on their previous best value achieved (it always keeps its value during its journey) and the global best (the particle that produces the best values to minimize the cost function), then an update occurs for both local and global values if any particle produces a value closer to the optimal one then of those found from all elapsed iterations.
Two parameters characterize each particle during its trajectory toward a solution are: position and velocity. A mathematical equation describes the relationship between those two parameters as follows:

$$v_{ij}^{k} = \omega v_{ij}^{k} + c1 \times rand_1() \times (best_{ij} - x_{ij}^{k}) + c2 \times rand_2() \times (gbest_j - x_{ij}^{k}) \tag{4.1}$$

$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1} \tag{4.2}$$

Where: $1 \leqslant i \leqslant I$; $1 \leqslant j \leqslant J$; $1 \leqslant k \leqslant K$

- K: The number of iterations
- I: Is the population size
- J: Is the number of values holding each particle
- v_{ij}^{k}:Is the velocity of particle i in the dimension J at actual iteration k.
- x_{ij}^{k}: Is the position of the particle i at actual iteration.
- ω: Is the inertia weight.
- c1,c2: Are the individual and social cognetive respectively.
- rand(): Is the uniformly distributed random number (r1,r2).

From equation 4.1, we deduce the existence of an internal communication between particles to adjust their next position ad velocity, which means each particle keeps tracking its personal best (Pbest) and global best (gbest).

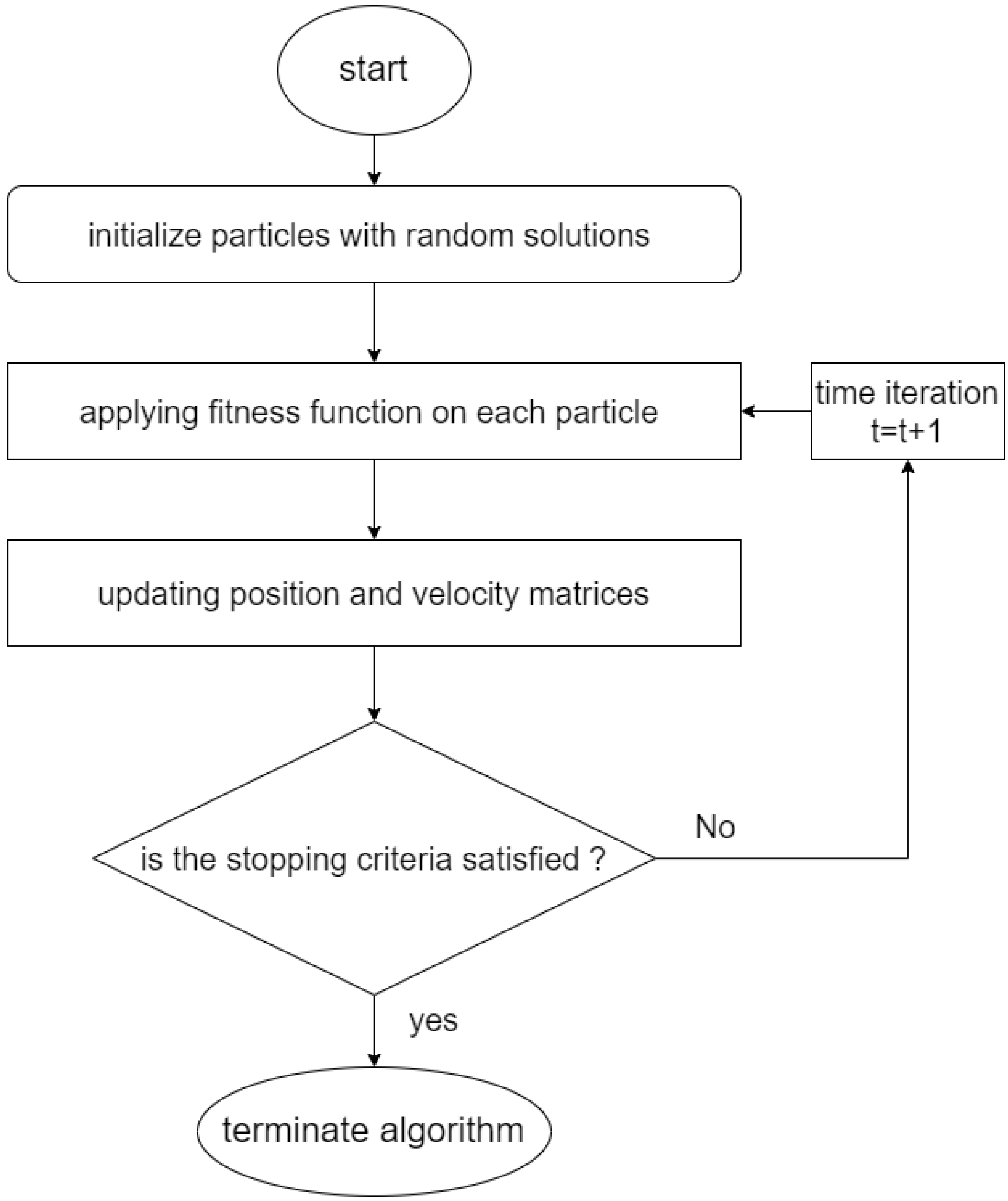

Figure 4.1: PSO algorithm flowchart

4.3 TLBO algorithm

4.3.1 Description

TLBO stands for teaching-learning-based-optimization, it belongs to a family of population based meta-heuristic methods for optimization. TLBO is an inspired algorithm which is proposed by R.V.Rao et al, at 2011 based on the effect of learners in class [17]. In real world, teaching and learning are two activities where an exchange of knowledge and information is established, via communication protocols. The interaction between teacher and learners will systematically leads to an evolution and improvements of learner's level. This algorithm acts exactly the same way with previous described process. It is used to solve multi-dimensional, linear and non-linear problem efficiently

4.3.2 Mathematical model and optimization algorithm

TLBO algorithm may suggest two new solutions two phases, one solution during one phase. Those two phases are known as follows:

1. **Teacher phase:** When initializing the algorithm, an element is selected to be the teacher based on its fitness function value. Obviously, the task of a teacher is to increase the mean result of learners and tries to bring the mean value as close as possible to its fitness value, but this is impossible even in real world, a new mean solution is generated, then the difference between the fitness value of the teacher and mean result of the learners in each subject described by equation 4.3, subjects will simulate variables in this work.

   $\textbf{DIFFERENCE\_}MEAN_{j,i} = r_i \left(x_{j,Kbest,i} - T_F M_{j,i}\right)$ (4.3)

   The teacher is represented by $x_{j,Kbest,i}$, where j is the subject in which we perform the difference mean. other two variablesr_i an T_F.

   - r_i: A random value varying from 0 to 1,

   - T_F: A teaching factor which influences the rate of change of the new mean and can take only two values [1,2]. The program selects one of them randomly during the execution as according to equation 4.4.

     $\mathrm{T}_F = round[1 + rand(0,1)2 - 1]$ (4.4)

   - **K**: Is iteration number.

   next, an update occurs to the actual solution in this phase according to equation4.5.

   $\mathrm{x}'_{j,K,i} = x_{j,K,i} + DIFFERENCE\_MEAN$ (4.5)

   An essential condition for the new value $x'_{j,K,i}$ to be accepted to replace the old one is to produce a better fitness value comparing to old one, otherwise the old solution is kept. All solutions either updated or not will undergo a next step which is learner phase.

2. **Learner phase:** The fact that students communicate and share their competences, an improvement could also be happened. A learner will interact with other learners and grasp more knowledge. This algorithm will simulate this situation in two cases by two equations 4.6 and 4.7, each equation for one case:

- **When:** $x^{'}_{total-p,i} > x^{'}_{toal-Q,i}$

  $\mathrm{x}^{''}_{j,p,i} = x^{'}_{j,p,i} + r_i(x^{'}_{j,p,i} - x^{'}_{j,Q,i})$ (4.6)

- **When:** $x^{'}_{total-p,i} < x^{'}_{toal-Q,i}$

  $\mathrm{x}^{''}_{j,p,i} = x^{'}_{j,p,i} - r_i(x^{'}_{j,p,i} - x^{'}_{j,Q,i})$ (4.7)

This process is performed K times to obtain a desired fitness value that corresponds to variables needed in our optimization.

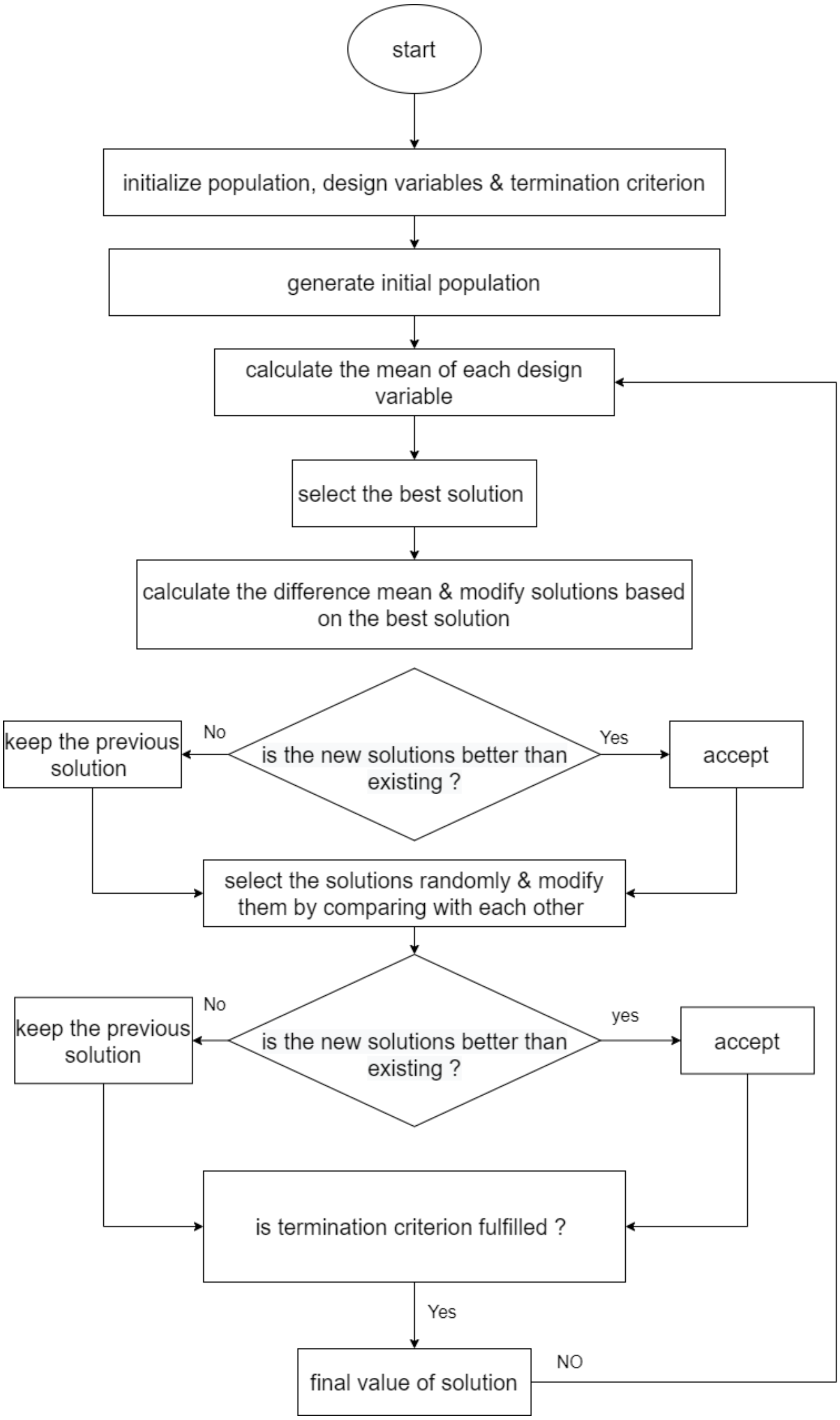

Figure 4.2: TLBO algorithm flowchart

4.4 Whale optimization algorithm WOA

4.4.1 Description

Whales are classified as mammals living in oceans. According to a scientific study, whales share some common cells with humans which make them intelligent and emotional [18]. Whales are well known by their predator character, especially humpback whales, they are the biggest baleen whales which have a size of school bus.
Whales parade their intelligence during hunting operations, scientists observed a very amazing technique used by whales to trap the prey. They dive 12 m under the surface of the ocean and create bubble in spiral shape around the prey starting from the bottom to the surface as shown in figure 4.3

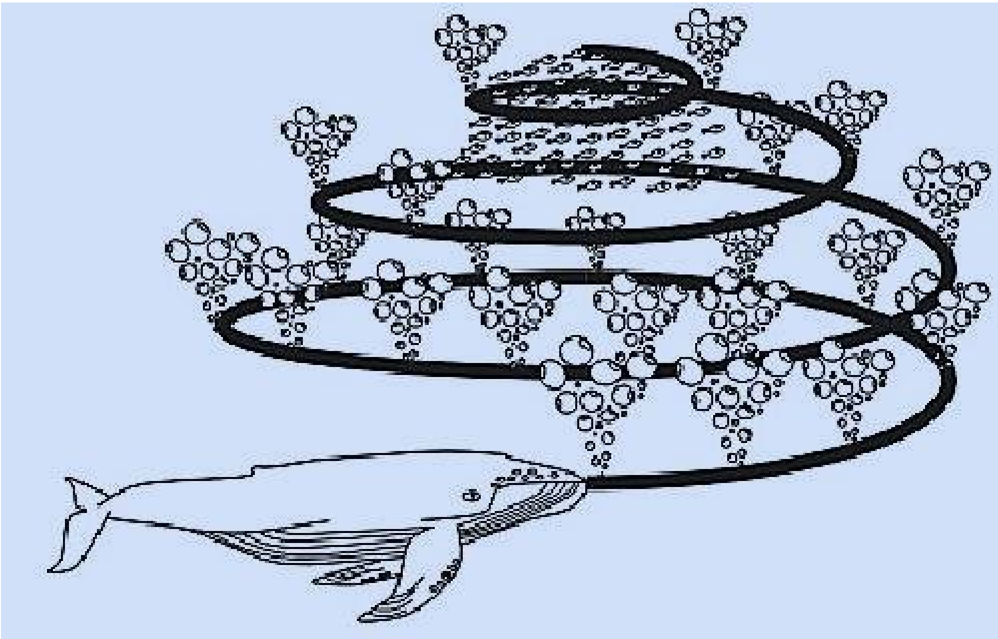

Figure 4.3: Bubble-net feeding motion[10]

4.4.2 Mathematical model and optimization algorithm

It is worth knowing that this algorithm consists of prey encircling, spiral bubble-net feeding maneuver, and search for prey.

(a) **Encircling prey:**
In nature, whales recognize the exact location of prey due to their instinct. This behaviour can not be applied to a mathematical model which means, in the beginning of this process the position of best optimal solution is not known yet. Since this algorithm is population based, the actual candidate providing the the best solution is considered as the target prey, so all other candidates will update their positions towards that best one, the equation bellow describes this motion.

$$\overrightarrow{D} = \left|\overrightarrow{C}.\overrightarrow{X}^*(t) - \overrightarrow{X}(t)\right| \tag{4.8}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X}^*(t) - \overrightarrow{A}.\overrightarrow{D} \tag{4.9}$$

Remark: In case of any new best solution is generated with every iteration$\overrightarrow{X}^*$ is updated.
the convergence factor $\overrightarrow{A}$ and the vector $\overrightarrow{c}$ are obtained as follows:

$$\overrightarrow{A} = 2\overrightarrow{a}.\overrightarrow{r} - \overrightarrow{a} \tag{4.10}$$

$$\overrightarrow{C} = 2.\overrightarrow{r} \tag{4.11}$$

The vector $\overrightarrow{a}$ decreases linearly from 2 to 0 during iterations and $\overrightarrow{r}$ is random vector in [0,1].
Every agent updates its position X towards the actual bestX^*, according to the combination of two vectors $\overrightarrow{A}$ and $\overrightarrow{C}$ which depends on the random value of $\overrightarrow{r}$. This allows agents to locate many different positions surrounding the actual best solution, this what make sense of encircling the prey.

(b) **bubble-net attacking method**
From a view of algorithm architecture, this step is considered as an exploitation phase, where two approaches explained as follows:

i. **Shrinking encircling method:** Analytically, equation (4.10) varies continuously with the variation of the value of a , then a new position will be set anywhere in between the previous position and the best agent, by letting the value of A varying in the interval $0 \leqslant A \leqslant 1$. As shown in figure4.4.

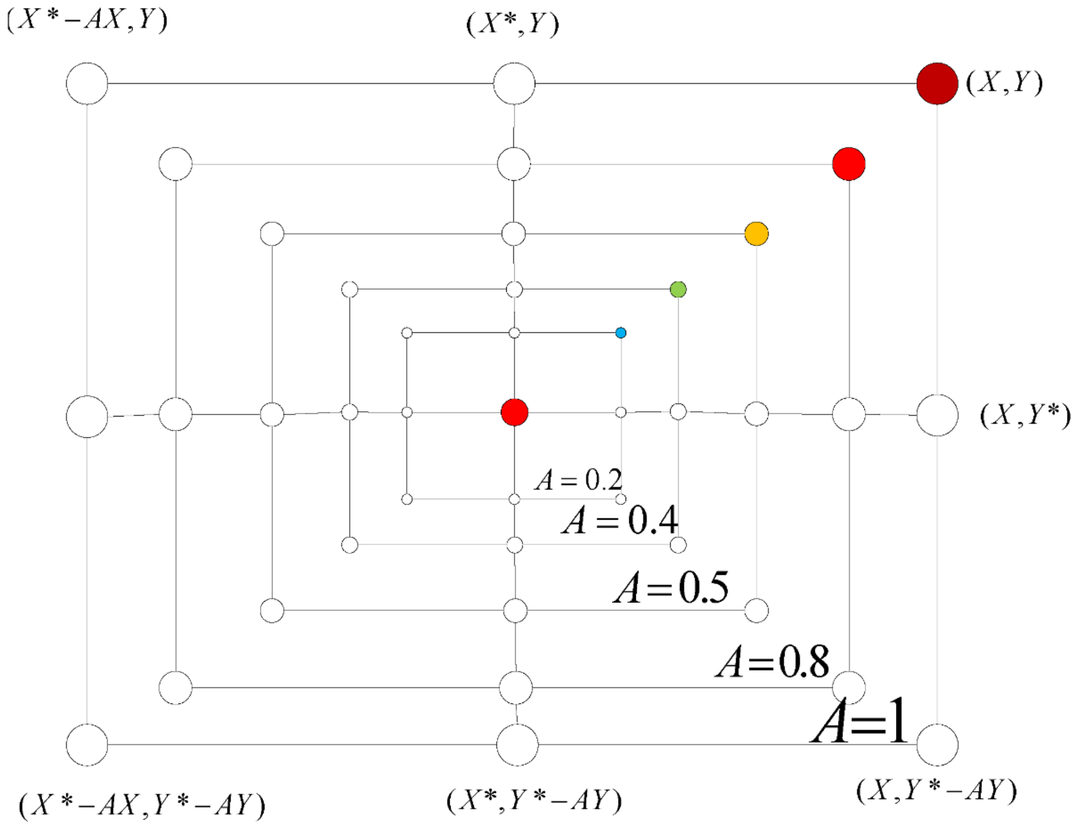

Figure 4.4: the updated positions path [10]

ii. **Spiral updating position :**
This approach computes the distance between the current position X(t) and the best agent $X^*(t)$, then updates its position according to the following equation:

$$\overrightarrow{X}(t+1) = \overrightarrow{D}' e^{bl}.cos(2\pi l) + \overrightarrow{X}^*(t) \tag{4.12}$$

where $\overrightarrow{D}'$, b is a constant for defining the shape of the logarithmic spiral, l is a random number varying in the interval [-1,1]. This approach is depicted in figure 4.5.

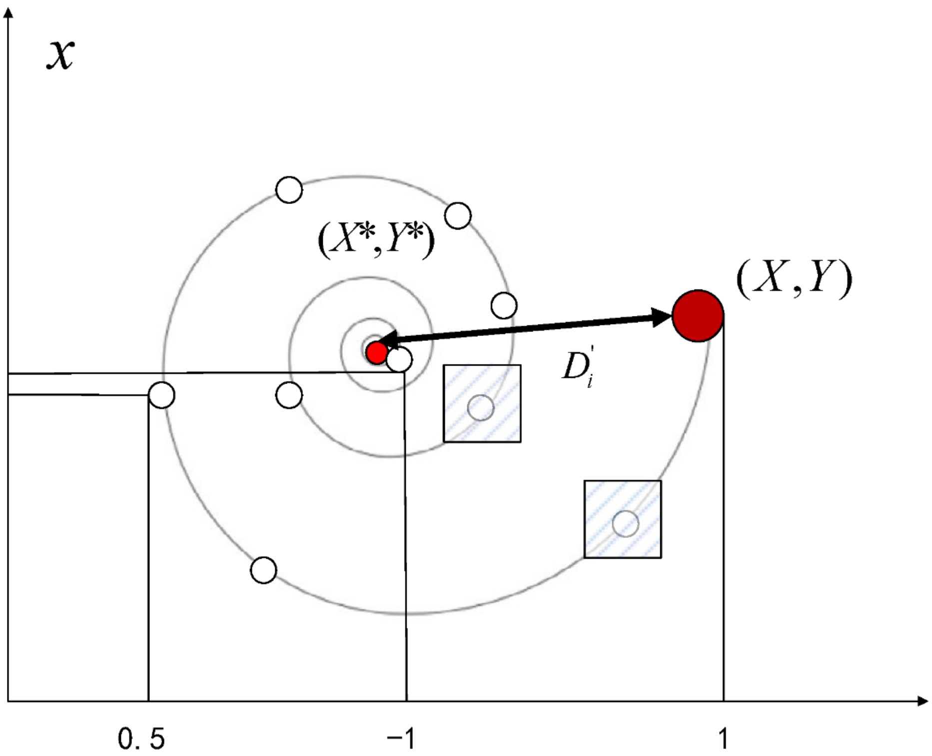

Figure 4.5: spiral path for updating position

In reality, whales use those two approaches at the same time,so imitating this behaviour, we use both equations shown by (4.13).

$$\overrightarrow{X}(t+1) = \begin{cases} \overrightarrow{X}^*(t) - \overrightarrow{A}.\overrightarrow{D} & if\ p < 0.5 \\ \overrightarrow{X}(t+1) = \overrightarrow{D}'e^{bl}.cos(2\pi l) + \overrightarrow{X}^*(t) & if\ p > 0.5 \end{cases} \tag{4.13}$$

(c) **Search for prey:**
This also lays on $\overrightarrow{A}$ variations. But in this case, $\overrightarrow{A}$ can take values greater than 1 or less than -1, which makes a very large search space, where the agent updates his position according to a randomly selected agent, whereas in previous phase (exploitation) the best agent was the reference towards which other's agent positions are updated. This phase is called exploitation, where a global search is done due to the unbounded vector length $\left|\overrightarrow{A}\right| > 1$

$$\overrightarrow{D} = \left|\overrightarrow{C}.\overrightarrow{X_{rand}} - \overrightarrow{X}\right| \tag{4.14}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_{rand}} - \overrightarrow{A}.\overrightarrow{D} \tag{4.15}$$

X_{rand} is a random position agent selected during exploitation

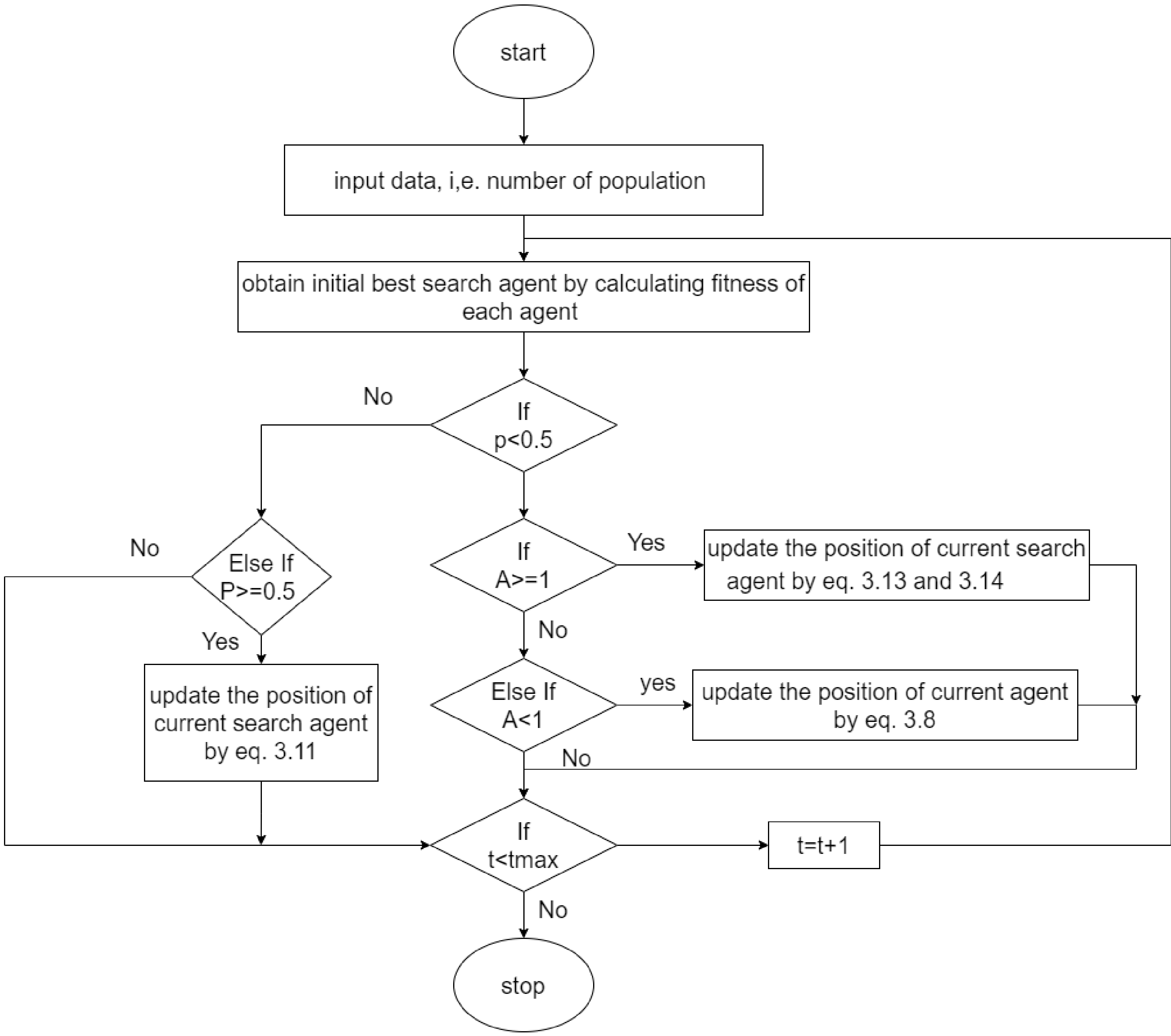

Figure 4.6: Whale algorithm flowchart [19]

4.5 Conclusion

This chapter discusses the three meta-heuristic algorithms with their properties, to show how they can optimize problems to provide optimal solutions.

Chapter 5

Results and discussion

5.1 Introduction

In this chapter, the results of optimization algorithms of a PID parameters, applied to control a quadrotor are shown, then the results of each algorithm are discussed and compared with each other. Based on the response and the objective function. We will get four responses as output from a quadrotor, each response will be analyzed and compared with the same one but different algorithm.

5.2 Set up parameters

From the previous chapter also from review, meta-heuristic algorithms have as parameters: number of particles (search agents), number of iteration and some constants of some algorithms. Those mentioned parameters will be organized in the table bellow:

	number of particles	number of iterations	specific parameters
PSO	30	20	C1= 2 ; C=1.5
TLBO	15	20	-
WOA	30	20	-

Table 5.1: Tables of setted up parameters

Remark: TLBO algorithm will have a half number of search agents then other two algorithms, because TLBO generates a new solution during each phase, and it has two phases in only one iteration, so to give a fair shake for all algorithms, it is balanced either by setting number of population or number of iteration.

5.3 Objective functions

Performance indices are calculated to be used as quantitative measures to assess a system's performance in control system design and analysis or for optimal control purposes. A control system is considered to be an optimum system when the system parameters are adjusted so that the index used in the design reaches its minimum value while the controlled system's constraints are observed [20]. A particular category of objective function defined as the fitness function minimizes or optimizes certain preference goals.

- Integral of Absolute-error: $IAE = \int_0^T |e(t)|\, dt$
- Integral of Squared-error: $ISE = \int_0^T e^2(t)dt$
- Integral Time Squared-error: $ITSE = \int_0^T te^2(t)dt$
- Integral Time Absolute-error: $ITAE = \int_0^T t\, |e(t)|\, dt$

The error e(t) is the difference between the set-point r(t) and the output y(t) of the controller. In this optimization application, we used ITAE objective function.

5.4 Results

The responses of algorithms are compared to reference 1.

1. **The altitude response (Z):**
   The response obtained from Matlab Simulink of altitude (Z) by three algorithms are depicted in figure 5.1

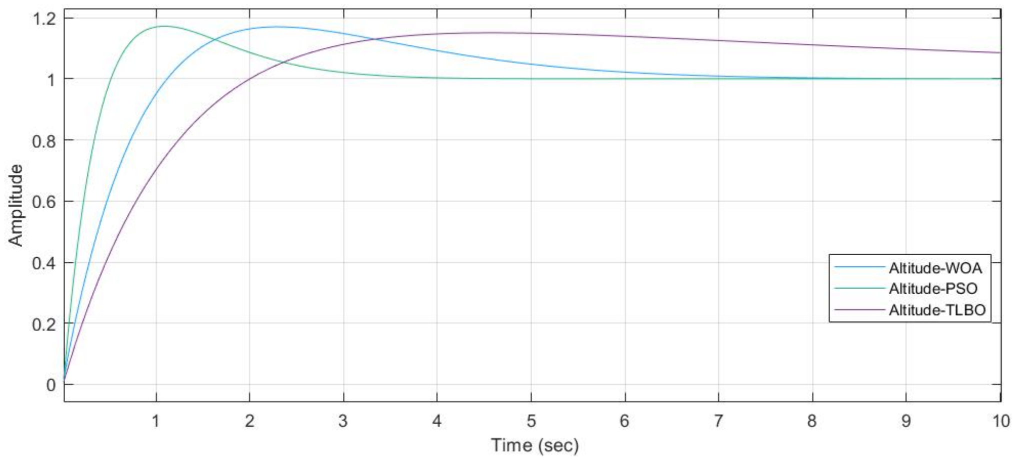

Figure 5.1: Altitude step response

Analysis
For the sake of making a clear and easy comparison, we put the obtained data in a form of a table, as follows:

	Rise Time (s) Tr	Peak Time (s) Tp	Settling Time (s) Ts	Overshoot Mp(%)	steady state-error
PSO	0.5	1.1	4	16	0
TLBO	2	4	-	15	0.0.09
WOA	0	0	0	16	0

Table 5.2: The response of the altitude

	Kp	Ki	Kd	Fitness function
PSO	214.8	204.716	62.857	0.4293
TLBO	3.52	1.282	0.581	1.174
WOA	97.029	11.41	100	5.999

Table 5.3: Gains and final fitness values of altitude Z

Comparison:
The response of PSO algorithm has a lowest rise time but slightly higher overshoot over the two others, also has the quickest settling time, whereas TLBO performs a steady state error despite of its fitness function value which is less than the WOA. PSO generates a best response for the altitude. The values of fitness functions shows clearly the performance of PSO algorithm.

2. **θ angle response:**

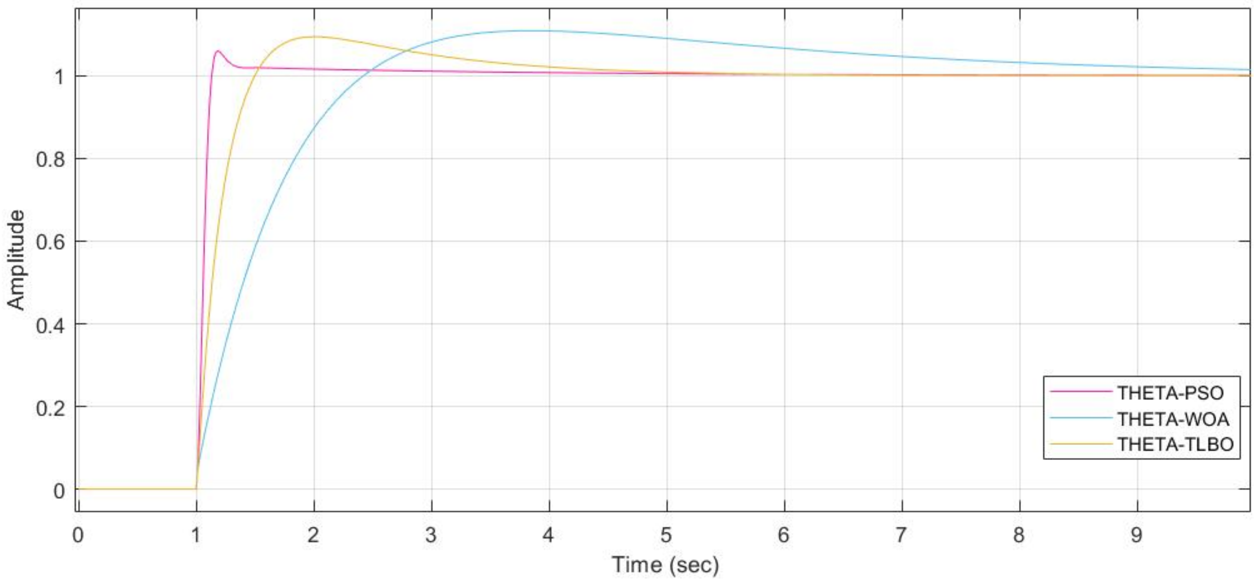

Figure 5.2: Theta angle step response

Analysis:

	Rise Time (s) Tr	Peak Time (s) Tp	Settling Time (s) Ts	Overshoot Mp(%)	steady state-error
PSO	1.15	1.2	5	4	0
TLBO	1.4	2	5	15	0
WOA	2.45	3.6	-	16	0.01

Table 5.4: The response of the angle θ

	Kp	**Ki**	**Kd**	**Fitness function**
PSO	14.335	5.382	0.927	1.1877
TLBO	4.663	4.189	3.347	2.955
WOA	5.005	1.461	4.139	6.035

Table 5.5: Gains and final fitness values for angle θ

Comparison:
At first glimpse, a best response refers to PSO algorithm with lowest overshoot and quickest rise and settling time, also TLBO shows good results with higher overshoot compared with PSO response. WOA has latest rise time and settling time with highest overshoot. This results were directly related to the obtained fitness values. Then fitness values were so objective to make a decision.

3. ϕ **response:**

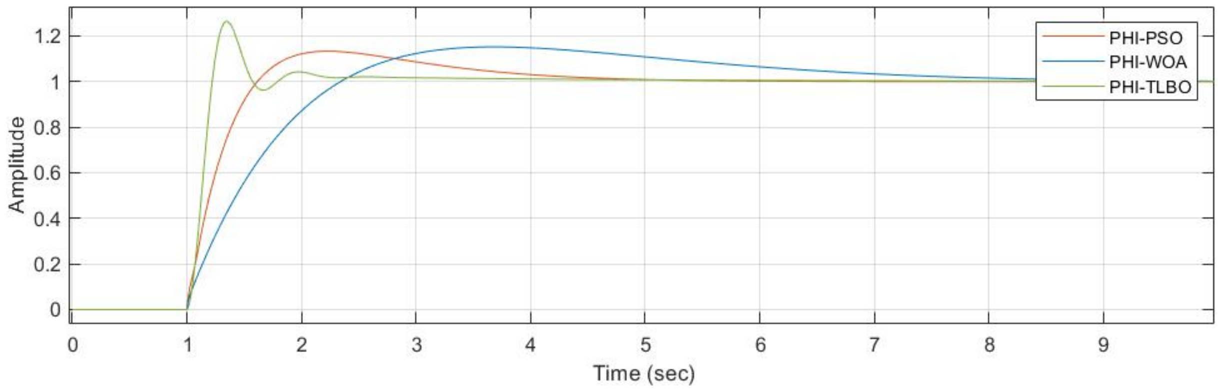

Figure 5.3: Phi angle step response

Analysis:

	Rise Time (s) Tr	Peak Time (s) Tp	Settling Time (s) Ts	Overshoot Mp(%)	steady state-error
PSO	1.1	1.15	3	24	0
TLBO	0.25	1.45	5	2	0
WOA	2.6	4.2	8.4	22	0

Table 5.6: The response of the angle ϕ

	Kp	**Ki**	**Kd**	**Fitness function**
PSO	18.529	14.647	5.597	1.187
TLBO	2.843	0.692	0.315	2.955
WOA	5.644	2.442	4.096	6.035

Table 5.7: Gains and final fitness values for angle ϕ

Comparison: TLBO algorithm has small oscillations and overshoot, but with low rise time and settling time. PSO also shows good response with quick rise and settling times, compared to TLBO which slower response. We can consider PSO algorithm the best in terms of fitness function value.

4. ψ **angle response:**

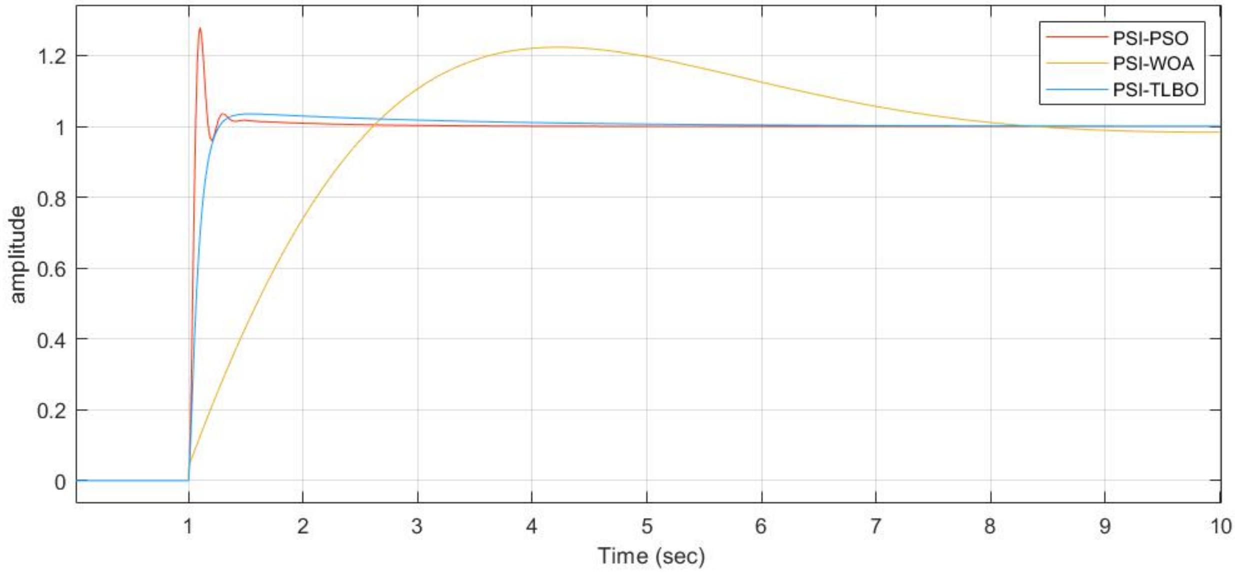

Figure 5.4: Psi angle step response

Analysis:

	Rise Time (s) Tr	Peak Time (s) Tp	Settling Time (s) Ts	Overshoot Mp(%)	steady state-error
PSO	1.1	1.15	3	24	0
TLBO	0.25	1.45	5	2	0
WOA	2.6	4.2	8.4	22	0

Table 5.8: The response of the angel ψ

	Kp	**Ki**	**Kd**	**Fitness function**
PSO	13.877	15.942	0.337	0.0959
TLBO	8.135	3.868	0.725	0.345
WOA	8.802	4.876	9.711	4.5161

Table 5.9: Gains and final fitness values of angle ψ

Comparison: At first look, a best response of angle ψ is generated by TLBO algorithm, with smaller overshoot and small rise time. PSO algorithm produced an overshoot with small oscillations but quickest rise time and settling time, this is shown by the resulted fitness function value by PSO which is well optimized, in other hand, WOA has a greater values of fitness function, this really confirms its response in figure 5.4.

5.4.1 Conclusion:

To make a decision about which algorithm has a best response, we will consider the overall response of the system to different algorithms. Previously, we compared algorithms in every single parameter, but when it comes to decide the best one on the whole system, PSO algorithm was nominated to be the best optimization algorithm among the tested ones on this system and with that specific objective function. This is based on the responses obtained by this algorithm with quick rise and settling times also the values of fitness function, which are the main objective in an optimization problem.

General conclusion:

This study showed a multiple responses from three meta-heuristic algorithms, used to optimize a PID controller by minimizing the error signal. Results of each algorithm were discussed, and we selected a best algorithm, this is does not mean that the two other algorithms are limited to optimize this controller, but with setting conditions and initial parameters fairly, algorithms are given a fair shake to generate solutions within restricted regions. Because every meta-heuristic algorithm is characterized by its exploration and exploitation phases, then a specific chosen parameters for every algorithm, will give him a well balance between the two phases. Also if other objective functions were used, surely different results will be obtained.

Future work:

This work was so motivating for us to do more improvements on this system, so we are looking forward to enhance this controller by using more improved FOPID (Fractional PID), and many other optimization algorithms with different objective functions, the aim of this is to do real implementation for this optimized system.

Appendix

Parameter	**Value**
$Ix = 7.5 \times 10^{-3}\ Kg.m^2$	Quadrotor moment of inertia around X axis
$Iy = 7.5 \times 10^{-3}\ Kg.m^2$	Quadrotor moment of inertia around Y axis
$Iz = 1.3 \times 10^{-2}\ Kg.m^2$	Quadrotor moment of inertia around Z axis
$Jr = 6.5 \times 10^{-5}\ Kg.m^2$	Total rotational moment of inertia around the propeller axis
$b = 3.13 \times 10^{-5} N.s^2$	Thrust factor
$7.5 \times 10^{-7}\ N.m.s^2$	Drag factor
$l = 0.23$ m	Distance to the center of the Quadrotor
$m = 0.65$ Kg	Mass of the Quadrotor in Kg
$g = 9.81\ m/s^2$	Gravitational acceleration

Table 5.10: The Quadrotor Model Parameters

Bibliography

[1] M. N. Duc, T. N. Trong, and Y. S. Xuan, "The quadrotor mav system using pid control", in *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, IEEE, 2015, pp. 506–510.

[2] Q. Jiao, J. Liu, Y. Zhang, and W. Lian, "Analysis and design the controller for quadrotors based on pid control method", in *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, IEEE, 2018, pp. 88–92.

[3] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors", *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[4] X. Huo, M. Huo, and H. R. Karimi, "Attitude stabilization control of a quadrotor uav by using backstepping approach", *Mathematical Problems in Engineering*, vol. 2014, 2014.

[5] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor", *IEEE Robotics and Automation magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[6] A. Zulu and S. John, "A review of control algorithms for autonomous quadrotors", *arXiv preprint arXiv:1602.02622*, 2016.

[7] C. Diao, B. Xian, Q. Yin, W. Zeng, H. Li, and Y. Yang, "A nonlinear adaptive control approach for quadrotor uavs", in *2011 8th Asian Control Conference (ASCC)*, IEEE, 2011, pp. 223–228.

[8] A. A. Muazu, A. S. Hashim, and A. Sarlan, "Review of nature inspired metaheuristic algorithm selection for combinatorial t-way testing", *IEEE Access*, 2022.

[9] T. Li, G. Shao, W. Zuo, and S. Huang, "Genetic algorithm for building optimization: State-of-the-art survey", in *Proceedings of the 9th international conference on machine learning and computing*, 2017, pp. 205–210.

[10] S. Mirjalili and A. Lewis, "The whale optimization algorithm", *Advances in engineering software*, vol. 95, pp. 51–67, 2016.

[11] B. Morales-Castañeda, D. Zaldivar, E. Cuevas, F. Fausto, and A. Rodrıguez, "A better balance in metaheuristic algorithms: Does it exist?", *Swarm and Evolutionary Computation*, vol. 54, p. 100 671, 2020.

[12] A. Ala'a, A. A. Alsewari, H. S. Alamri, and K. Z. Zamli, "Comprehensive review of the development of the harmony search algorithm and its applications", *IEEE Access*, vol. 7, pp. 14 233–14 245, 2019.

[13] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison", *ACM computing surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[14] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying", Epfl, Tech. Rep., 2007.

[15] G. Shabib, A. Moslem, and A. Rashwan, "Optimal tuning of pid controller for avr system using modified particle swarm optimization", in *Proceedings of the 11th WSEAS international conference on nural networks and 11th WSEAS international conference on evolutionary computing and 11th WSEAS international conference on Fuzzy systems*, 2010, pp. 104–110.

[16] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm", in *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, IEEE, vol. 5, 1997, pp. 4104–4108.

[17] R. V. Rao and V. D. Kalyankar, "Parameter optimization of modern machining processes using teaching–learning-based optimization algorithm", *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 524–531, 2013.

[18] P. R. Hof and E. Van der Gucht, "Structure of the cerebral cortex of the humpback whale, megaptera novaeangliae (cetacea, mysticeti, balaenopteridae)", *The Anatomical Record: Advances in Integrative Anatomy and Evolutionary Biology: Advances in Integrative Anatomy and Evolutionary Biology*, vol. 290, no. 1, pp. 1–31, 2007.

[19] A. Sengupta and D. K. Das, "Design of 3-dof pid stabilizer using whale optimization algorithm to eliminate low frequency oscillation in power system", in *2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, IEEE, 2019, pp. 1–6.

[20] S. B. Chiranjeevi *et al.*, "Implementation of fractional order pid controller for an avr system using ga and aco optimization techniques", *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 456–461, 2016.