People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Mohamed Khider University of Biskra

Ref:.............

A thesis submitted to the department of electrical engineering in candidacy for:

Degree of Doctorate in Electrical Engineering

***Speciality:** Automatic Control*

***Option:** Modeling and Control of Dynamic Systems*

Optimization of non-linear control aerodynamic systems using metaheuristic algorithm

Optimisation des commandes non linéaires des systèmes aérodynamiques par les méthodes méta-heuristiques

Author:

Hossam Eddine GLIDA

Discussed publicly ... In front the jury consists of:

President:	Abderrazak DEBILOU	Prof	Univ.Biskra
Supervisor:	Latifa ABDOU	Prof	Univ.Batna 2
Co-supervisor:	Chouki SENTOUH	Prof	LAMIH-UMR France
Examiner:	Mohamed Djamel MOUSS	Prof	Univ.Batna 2
Examiner:	Aicha GUERGAZI	Prof	Univ.Biskra
Examiner:	Hassina MEGHRERBI	Prof	Univ.Biskra

Academic year: **2021/2022**

Abstract

This thesis is part of the project "modelisation and control dynamic systems" carried by the laboratory of LMSE. This project aims to develop and optimize new control approaches for the UAV quadrotor tracking control. This thesis consisted of the modelling of the quadrotor, and then analysing, designing and implementing new optimal control strategies based on the model-free concept. In this context, the aim of the thesis is to propose new control strategies based on the model-free concept. The proposed strategies help to compensate the disturbances and model uncertainties. Regarding our work, we have proposed different control techniques for quadrotor control. First, an optimal model-free backstepping control law applied to a quadrotor UAV has been proposed. In addition to this work, the dynamic system has been estimated through a new proposed fuzzy strategy and merged with the BC under the model-free concept. Finally, an optimal fuzzy model-free control has been designed based on decentralized fuzzy control. The objective of these control strategies is to achieve the best tracking with unknown nonlinear dynamics and external disturbances. These proposed approaches are validated through analytical and experimental procedures and the effectiveness checked and compared with regard to the related controllers in the presence of disturbances and model uncertainties.

Keywords: Quadrotor, nonlinear control, model-free control, optimization, metaheuristic algorithms.

الملخص

هذه الأطروحة جزء من مشروع "النمذجة و التحكم في الانظمة الديناميكية" الذي يدعمه مختبر LMSE. يهدف هذا المشروع إلى تطوير وتحسين أساليب التحكم الجديدة للتحكم في تتبع الطائرات بدون طيار رباعية المحرك. تتكون هذه الأطروحة من نمذجة رباعي المحركات ، ثم تحليل وتصميم وتنفيذ استراتيجيات تحكم أمثل جديدة بناءً على مفهوم خالٍ من النمذجة. في هذا السياق ، تهدف هذه الأطروحة إلى اقتراح استراتيجيات تحكم جديدة تعتمد على مفهوم الخالي من النمذجة. تساعد الاستراتيجيات المقترحة على تعويض الاضطرابات ونموذج عدم اليقين. فيما يتعلق بعملنا ، فقد اقترحنا تقنيات تحكم مختلفة للتحكم في تتبع الرباعي. في البداية ، تم اقتراح قانون تحكم Backstepping (BC) مثالي خالٍ من نمذجة النظام يتم تطبيقه على طائرة بدون طيار رباعية المحركات. بالإضافة إلى هذا العمل ، تم تقدير النظام الديناميكي من خلال إستراتيجية غامضة مقترحة جديدة ودمجها مع BC ضمن مفهوم خالٍ من النموذج. أخيرًا ، تم تصميم تحكم ضبابي مثالي خالٍ من نمذجة النظام على أساس التحكم اللامركزي الضبابي. الهدف من استراتيجيات التحكم هذه هو تحقيق أفضل تتبع مع ديناميكيات غير خطية غير معروفة واضطرابات خارجية. يتم التحقق من صحة هذه الأساليب المقترحة من خلال الإجراءات التحليلية والتجريبية ويتم التحقق من فعاليتها ومقارنتها فيما يتعلق بوحدات التحكم ذات الصلة في وجود اضطرابات وشكوك في النموذج.

الكلمات الرئيسية: رباعية المحرك ، النمذجة ، التحكم الخالي من النماذج ، التحسين ، الخوارزميات metaheuristic.

Dedicated

to both my parents,
the most important people in my life.

Acknowledgements

I want to use this small section to thank the people that really helped me, to give the right acknowledgements, and to describe my PhD experience. First of all, I want to thank ALLAH for providing me the strength and the blessing to complete this thesis.

I wish to express my deep thanks to my supervisors **Pr. Latifa ABDOU** and **Dr. Chouki SENTOUH** for their support and guidance throughout this work.

My thanks also go to the jury members for having agreed to review and evaluate this work:

Pr. Abderrazak DEBILOU, professor at university of Biskra and the jury president.

Pr. Mohamed Djamel MOUSS, professor at university of Batna 2.

Pr. Aicha GUERGAZI, professor at niversity of Biskra.

Pr. Hassina MEGHRERB, professor at university of Biskra.

I also extend my sincere thanks and appreciation to all the teachers especially **Dr. Abdelghani CHELIHI**, he was my main mentor for all the three years, being a very respectful person and teacher and, as his student, my main focus was to try to reach his level.

Also, I would like to thank my colleague **Dr. Gabriele PEROZZI**, who has supported and guide me in my work with a critical and wise eye and created remarkable memories during my study.

Finally, I thank my family and my friends for their patience, encouragement, and support which were very useful to me during my work.

Contents

List of Figures

List of Tables

List of Abbreviations

UAVs	**U**nmanned **A**erial **V**ehicles
FAA	**F**ederal **A**viation **A**dministration
QoE	**Q**uality **of** **E**xperience
PID	**P**roportional-**I**ntegral-**D**erivative
LQR	**L**inear **Q**uadratic **R**egulator
BC	**B**ackstepping **C**ontroller
SMC	**S**liding **M**ode **C**ontrolle
MFC	**M**odel-**F**ree **C**ontroller
6-DOF	**S**ix **D**egree **Of** **F**reedom
ESC	**E**lectronic **S**peed **C**ontroller
PWM	**P**ulse **W**idth **M**odulation
CS	**C**uckoo **S**earch
BA	**B**at **A**lgorithm
FPA	**F**lower **P**ollination **A**lgorithm
GA	**G**enetic **A**lgorithm
GP	**G**enetic **P**rogramming
DE	**D**ifferential **E**volution
PSO	**P**article **S**warm **O**ptimization
MFPA	**M**odified Flower **P**ollination **A**lgorithm
BFPA	**B**inary **F**lower **P**ollination **A**lgorithm
CFPA	**C**hotic **F**lower **P**ollination **A**lgorithm
SISO	**S**ingle-**I**nput **S**ingle-**O**utput
MFBC	**M**odel-**F**ree **B**ackstepping **C**ontroller
OMFBC	**O**ptimal **M**odel-**F**ree **B**ackstepping **C**ontroller
RMSE	**R**oot **M**ean **S**quared **E**rror
MaxAE	**M**aximum **A**bsolute **E**rror

FS	**F**uzzy **S**ystem
FMFC	**F**uzzy **M**odel-**F**ree **C**ontroller
OFMFC	**O**ptimal **F**uzzy **M**odel-**F**ree **C**ontroller
FLC	**F**uzyy **L**ogic **C**ontrol

List of Symbols

$\wp$	m	Position vector
x, y, z	m	The position system
Θ	rad	Angular vector
ϕ, θ, ψ	rad	Angles of the quadrotor
$R_{x\phi}, R_{y\theta}, R_{z\psi}$	-	Rotational matrix for each axes
R	-	Global rotation matrix
u, v, w	m/s	Linear velocities of the quadrotor in body frame
p, q, r	rad/s	Angular velocities of the quadrotor
F_i	N	Lift force of each rotor
$F_{x,y,z}$	N	Disturbances Forces
u_z	N	Total lift force
$\tau_{\phi,\theta,\psi}$	N.m	The aerodynamic moments
$\tau^d_{\phi,\theta,\psi}$	N.m	The disturbance moments
g	m/s^2	Gravity acceleration
I	$kg.m^2$	Inertia matrix of the quadrotor
I_x, I_y, I_z	$kg.m^2$	The quadrotor inertia
$\Omega_1, \Omega_2, \Omega_3, \Omega_4$	rad/s	Rotor angular speed
Ω_r	rad/s	Propeller angular rate
d	m	Arm length of the quadrotor
b	-	The rotor′s reaction torque constant
u_ϕ, u_θ, u_ψ	N.m	Control inputs
u_{rot}	V	Rotor input
i_{rot}	A	Current in the electrical circuit of the rotor
L_{rot}	H	Rotor inductance
τ_m	$N.m$	Rotor torque

τ_d	$N.m$	Load torque

Remark The other symbols will be defined in the corresponding chapters.

Chapter 1

Introduction

1.1 State of art

1.1.1 Background

The last decade has seen an explosive increase in using Unmanned Aerial Vehicles (UAVs) for several areas of life. Federal Aviation Administration (FAA) declared that UAVs number will reach 3.2 million flying units in 2022 [42]. This future prediction will lead to the development of a novel application of such aircraft and also provide new business opportunities. A wide range of tasks can be achieved by these robots creating many useful applications. AAI corporation used aerosonde UAS to observe melt-ponds over Arctic Sea ice [44]. In Midtre Lovenbreen (Svalbard), the UAV type 323 Heliball is used to detect photosynthetic pigments in glacial ecosystems based on LASER-induced fluorescence emission mounted on the aircraft [88]. In Svalbard glaciers, they use UAV to map the glacier terminus using a digital camera mounted on Cryowing UAS [84]. The quality of experience of video streaming applications (QoE) is being enhanced using drone cellular networks project [27]. In [45], researchers suggest a distributed swarm control system for managing multiple agricultural UAVs controlled remotely by a system operator. Furthermore, other papers requiring UAVs have been performed in order to improve the connectivity of ground networks [16], coordinate with ground vehicles [40], collect data from ground sensors [24], recover connectivity in harsh environments [81] and secure communications with the ground [103]. However, there still exists many technical points and trials during the development of UAVs, which still remain completely unstudied and unsolved. Also, many requirements related to these aircraft are neglected, such as their high controlled robustness , their restricted-energy capacity, and their optimal tracking. Consequently, a strong knowledge of the current UAV-assisted solutions is needed to address the actual issues and to improve the functionality of applications.

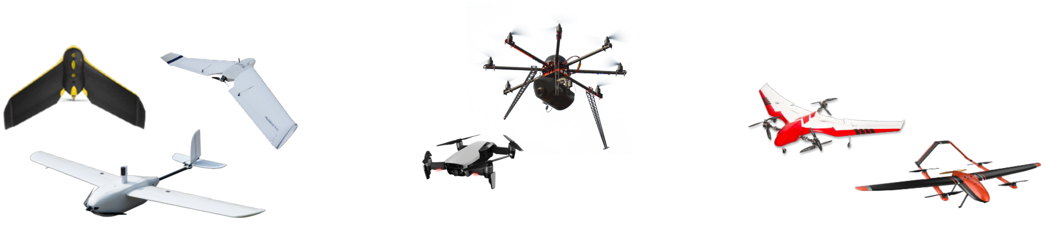
1.1.2 UAVs discrimination

This subsection presents the different types of UAVs based on their technical aspects. First, we begin with UAV categories based on size and weight. Next, we present the possible UAV categories based on the aerodynamic features as well as we investigate their level of autonomy.

Size and weight classification: Finally, UAVs can also be categorised based on their size and weight. For example, the Dutch Human Environment and Transport Inspectorate classify UAVs according to their weight [89]. In particular, we find the heavy UAVs that exceed the weight of 150 kg. On the other side, they are light weight UAVs. In [20], researchers give more specific classification, taking into account the UAV type based on its aerodynamic properties. They consider that the large fixed-wing UAVs are those which have a weight between 20 kg and 150 kg. Otherwise, the small UAVs don't exceed 20 kg.

Furthermore, they suppose that the small UAVs which can be classified further, as mini drones are those whose weight ranges between grams to several kilograms.

Aerodynamic features classification: According to aerodynamic features, it can be considered three types of UAVs; fixed-wing, rotary-wing and hybrid (see figure 1.1). The fixed-wing possesses a predefined airfoil of static and fixed wings that enable lift based on the UAV forward airspeed [38]. The UAV control is achieved by controlling the elevators, ailerons and rudder wings. These construction characteristics allow UAV to turn in 3D rotation space around the roll, pitch and yaw angles, respectively. The rotary-wing type is made of different rotors that produce the proper power necessary for the takeoff [38], this type does not require a forward airspeed for lifting in contrast to the first one. Regarding the control which is accomplished through the torque and thrust of the rotors and the airflow needed for lifting depending on the number of rotors also. Finally, the hybrid-wing combines between rotary-wing and fixed-wing. In particular, this type possesses rotors for taking off and landing, but also includes fixed-wings using for covering a large area [89].



(A) Fixed-wing UAVs (B) Rotary-wing UAVs (C) hybrid-wing UAVs

FIGURE 1.1: UAVs types

Level of autonomy classification: Each UAV is characterised by a rate of autonomy, where the difference between an automatic and autonomous system has to be defined. The functionality of an automatic plan is related to the operator, who has preprogrammed the system to perform a specific operation without his intervention. On the other side, an autonomous system is characterised by the existence of specific laws that can supply a type of improvement in several cases. This option does not exist in automatic systems. We can specify four types of autonomy in modern UAV systems which are the human-operated system, human delegated system, human supervised system and fully autonomous systems [89]. The first type leads to manage and control all tasks of the unmanned system through the system operator. The second type is a higher level than the first one, by keeping the ability to take autonomously some restricted decisions. The third level can take different decisions through the supervision of the operator system. In this case, both the system operator and the unmanned system are able to execute various decisions based on the input data. The last level (fully autonomous systems) which is responsible for all its operations. In this case, the unmanned system gets data from the system operator and convert it into specific actions. In case of an emergency, the system operator could intervene in the function of the unmanned system [89].

1.2 Motivation and literature review

Control systems have covered many aspects of life. Control technologies are invading our area to encourage humans to be more productive and active, where there could be found in several applications like computers, vehicles, factories systems, chemical and physical processes, medical and agricultural applications. It is obvious that this topic has a key role in the research nowadays. As the control technology is so practical and useful, as it could cause serious problems in case that it runs wrong way, where any fatal error in its process could lead to a tragic failure. Therefore, diverse research works have focused on improving control strategies for robots like UAVs quadrotor, used in several applications and which will be discussed in this chapter.

The main reasons for the UAV quadrotor study is their high mobility and their ability to achieve tasks in full autonomy. Furthermore, due to many limitations of the quadrotor like their underactuation, parametric and nonparametric uncertainties, low computational power, etc. They make perfect simulator for

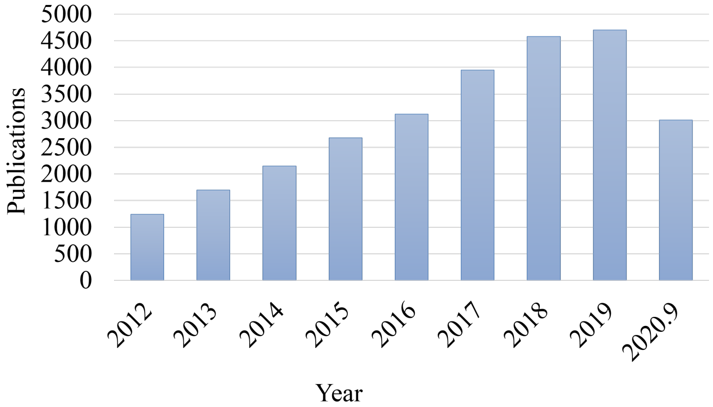


FIGURE 1.2: Publication search results in Google scholar.

novel theoretical techniques in control domain. Looking for keywords, UAV quadrotor control in the Google scholar, it is observed that research related to UAVs control has increased rapidly since 2012 as shown in Figure. 1.2.
Many different control approaches have been proposed within the last 9 years, such as classical linear approaches, classical nonlinear approaches, adaptive approaches, robust techniques, artificial intelligence techniques and robust adaptive approaches...etc. In this section, we will investigate some related control approaches applied to the UAV.

1.2.1 Classical linear control approaches

It can be find two different techniques based on the linear model, which are the PID and LQ techniques. PID is a linear control loop feedback mechanism which is vastly used in industry and in various applications requiring continuously modulated control. In [76], authors have treated the effect of dynamic disturbances of load introduced by the instantaneous increase Payload mass and how these disturbances affect helicopters and quad-rotors under PID flight control. Building a gravitational feedback PID controller for the control of a small helicopter with an experimental validation is proposed [36]. LQ is an optimal control theory based on a specific optimality criterion that must be achieved for a given system, the LQR approach was applied for the stabilization of the Octorotor type drone in [4].

1.2.2 Conventional nonlinear control approaches

The conventional non-linear control techniques for trajectory tracking control of drones and the most used are backstepping control (BC) and sliding mode control (SMC) as conventional techniques, where there is a lot of work in this direction, some proposed works use the backstepping controller for tracking control of a quadrotor in [11, 21]. The sliding mode control was applied to the quadrotor in several works [28, 25, 73].

1.2.3 Adaptive control approaches

The quadrotor has many parameters which could be parametric uncertainties like variation of mass, inertia and aerodynamic coefficients, in this case it requires the use of adaptive estimation techniques. The backstepping with feedback linearization technique was presented in [17] while a passivity-based adaptive control for mixed quadrotor is proposed in [39]. An adaptive integral backstepping technique has been proposed to deal with external disturbances in [100]. One of the most typical techniques for adaptive control involves using the model reference adaptive control technique [65].

1.2.4 Robust approaches

The UAVs are subject to external effects, like uncertainties and external disturbances as the wind. To deal with these effects, the system requires the control robust features. This can be performed using different robust methods. For example, a robust control for attitude quadrotor was presented in [61]. A new terminal sliding mode control has been proposed in [93] to achieve finite-time convergence and ensure the asymptotic convergence of sliding mode, this proposed method suffered from discontinuity in the switching signal which led to the chattering phenomenon. This effect has been addressed in [92]. Another improvement in [77], authors proposed robust backstepping-based controller that induces integral sliding mode.

1.2.5 Artificial intelligence control approaches

Many papers investigated the including of the artificial intelligence strategies with the control operation to enhance the performance using the data obtained during the flight process. These techniques consider the non-parametric and the non-modelled effects, which makes them outclass the adaptive methods. Many approaches related to these techniques can be defined, the most known in literature are the fuzzy approach, learning control and neural network. The fuzzy control system is widely used in UAVs control area, where a fuzzy logic control has been proposed in [51, 64], an optimized fuzzy logic controller type 2 has been tested experimentally for the quadrotor in [74]. Neural networks have been applied successfully in the robotic control domain especially on the drone systems. A feedback control based on neural networks is applied in [23], the reinforcement learning algorithm is proposed for quadrotor tracking control in [43, 55].

1.2.6 Robust adaptive control approaches

Robust adaptive control are hybrid both adaptation and robustness approaches will yield adaptive robust control law that can deal with almost all nonlinearities. The adaptive law can drive the system to instability in case of small disturbances, which lead to robust feature as robust term which is required to deal with the unmodeled dynamics. For instance, an adaptive robust controller for position and attitude tracking control is proposed in [58, 78], a robust hierarchical controller using adaptive radical basis function neural network and sliding mode control is presented in [60] and a robust adaptive backstepping control based on terminal sliding mode term in [53].

1.2.7 Model-free control approaches

For the UAV, the accurate kinematical and dynamical models are hard to reach because of the strong coupling effect and nonlinearities. Unlike the most approaches based on the precise model knowledge to obtain stable control, the model-free control techniques (MFC) have been proposed often for uncertainties systems control. Indeed, we can specify a few modest studies which have investigated the MFC in different versions and show satisfactory results. For instance, robust model-free control applied to a quadrotor UAV in [5], where the authors proposed a comparative study between two different controllers which are the LQ-based MFC and the BC-based MFC and the results have been obtained from a real time test. Model-free-based terminal SMC have been designed in [90], the given work have combined the intelligent PID and the sliding-mode technique that make the proposed control without need to any prior knowledge of the model. Another investigation for the MFC has been introduced in [9], the mentioned work is a conventional strategy that uses the time-delay estimation approach.

As far as we know, control approaches and strategies aren't limited to the control approaches mentioned above, it can be find more different techniques in literature. A review paper [66] introduces 33 different control approaches which address the UAV control. Among these approaches, we find the data-driven control, differential flatness based, Fault tolerant control, visual servo control, fractional-order control, neurobiologically-inspired control, null space control ... etc.

1.3 Trends and challenges

The previously mentioned works have shown satisfactory results in tracking control. However, some of them are subject to a specific limitation that have to address. The linear techniques like PID or LQ are easy to implement on closed-loop control, but they are limited to linear systems, which require some assumptions that have be to considered. In this case, a nonlinear control design is considered as the first challenge in our work. The requirement of accurate knowledge of the system is the most problem that can be find in nonlinear classical techniques. So, the design of a model-free control law without any prior knowledge of system dynamics is another challenge that have to be addressed.

In addition, to deal with the unmodeled dynamics such as variant parameters and external disturbances effects, the designed control should be adaptive law with including robust term to deal with the small unmodeled variation. The drawback of parameters control selection is one of the research subject nowadays, where most works are based on the trial-and-error method which is leading in effect to great limitation of these approaches. Some works proposed the use of the fuzzy system to compute these parameters which increase the complixity of the control structure that leads to make it unimplementable in real time. Hence, an addition of optimization strategies as an offline process gives an optimal control feature and decrease this drawback. Finally, the obtained results of the designed controller have to be acceptable in a way that evaluated for real time experimentals. The work presented in this thesis highlights the following elements:

- The quadrotor modelization is given in details for a simulation purpose;
- To reduce the unmodeled dynamics in the quadrotor system, a model-free based-control is developed;
- The decentralized architecture of the designed control law is considered in order to decrease the strong coupling between the subsystems;
- The disadvantage of the parameters control selection is addressed via including a metaheuristic optimization algorithms.
- The proposed controller has a strong robustness against time-varying uncertainties, unknown nonlinearities, and external disturbances and has characteristics such as simplicity, continuous control signals, and the ability to be implemented in real-time.

1.4 List of productions during the PhD

1.4.1 Journal articles

- GLIDA, Hossam Eddine, ABDOU, Latifa, CHELIHI, Abdelghani, et al. Optimal model-free backstepping control for a quadrotor helicopter. Nonlinear Dynamics , 2020, vol. 100, No. 4, p. 3449-3468.

- Glida HE, Abdou L, Chelihi A, Sentouh C, Perozzi G. Optimal model-free fuzzy logic control for autonomous unmanned aerial vehicle. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering. June 2021.

1.4.2 Conference Proceedings articles

- Glida, H. E., Chelihi, A., Abdou, L., and Sentouh, C. (2020). Disturbance observer-based optimal backstepping controller for an attitude quadrotor helicopter, In020 1st international conference on communications, control systems and signal processing (ccssp).

- Glida, H. E., Abdou, L., Chelihi, A., Sentouh, C., and Perozzi, G., (2020). Optimal Robust Model Free Control for Altitude of a Mini-Drone Using PSO Algorithm, Conference: 2nd International Conference on Electronic Engineering and Renewable EnergyAt: Saidia, Morocco (ICEERE).

- Glida, H. E., Abdou, L., Chelihi, A., and Sentouh, C. (2019). Optimal direct adaptive fuzzy controller basedon bat algorithm for uav quadrotor, In 2019 8th international conference on systems and control (icsc).

- Glida, H., Abdou, L., and Chelihi, A. (2019). Optimal fuzzy adaptive backstepping controller for attitude control of a quadrotor helicopter, In 2019 international conference on control, automation and diagnosis(iccad).

1.4.3 Collaboration articles

- Hasseni, S.-E.-I., Abdou, L., Glida, H.-E. (2019). Parameters tuning of a quadrotor pid controllers by using nature-inspired algorithms. Evolutionary Intelligence, 1–13.

- Guettal, L., Chelihi, A., Touba, M. M., Glida, H., and Ajgou, R. (2020a). Adaptive fuzzy-chebyshevnetwork-based continuous sliding mode controller for quadrotor unmanned aerial vehicle, In 2020 1stinternational conference on communications, control systems and signal processing (ccssp).

- Guettal, L., Chelihi, A., Touba, M. M., Glida, H., and Ajgou, R. (2020b). Neural network-based adaptive backstepping controller for uav quadrotor system, In 2020 1st international conference on communications,control systems and signal processing (ccssp).

- Guettal, L., Glida, H., and Chelihi, A. (2020). Adaptive fuzzy-neural network based decentralizedbackstepping controller for attitude control of quadrotor helicopter, In 2020 1st international conferenceon communications, control systems and signal processing (ccssp).

1.5 Thesis structure

Chapter 2 is entirely devoted to the formulation of a quadrotor model. In order to give more details for the quadrotor model, the whole dynamics model of the quadrotor is extracted through Euler-Newton formalism. The physical model is given in the state-space form to show the influence of the control signals and the underactuated effect on the system state of the quadrotor. Using this state-space representation, control laws and estimation algorithms are designed in the next chapters.

Chapter 3 introduces a stat of art on metaheuristic algorithms used during our work. Also, we give the mathematical representation of each algorithm to clarify their optimization process and we find in the same chapter their different applications in many fields.

In chapter 4, we propose two different optimal model-free approaches based on backstepping control strategy. First, a new estimation method is introduced based on the backstepping approach. Next, we focus on the backstepping fuzzy hybrid control. The proposed controllers are compared with the conventional backstepping to show their robustness.

The Last chapter is devoted to the intelligent model-free control strategy. We present the synthesis of optimal fuzzy model-free control as a new strategy for the design of direct control laws.

Chapter 2

Mathematical modeling of a quadrotor UAV

2.1 Introduction

The goal of this chapter is to give modelling detail of the mathematical dynamic of the UAV quadrotor. Identical to all aircrafts, the modelization of an UAV quadrotor is relatively difficult to establish and needs a fairly thorough mechanical study. This study considers the deformation of certain bodies of the drone as well as the strong couplings between several phenomena, namely the dynamics of the fuselage, the dynamics of the engines, the aerodynamic effects and the gyroscopic effects.

Many researches have been inspired by the dynamics of a rigid body which is associated with a fuselage to approach modelling and to which are added the aerodynamic forces generated by the rotors, such as the model established by Hamel and chriette [18], based on Newton's formalism and which integrates the dynamics of actuators. Lozano and Castello give a detailed modelization for an X4 [62] based on the Euler-Lagrange method. In addition, a modelling based on evolutionary methods is proposed in [37].

In this chapter, we present a detailed description of mathematical modelization of an UAV quadrotor based on Newton's formalism with six degrees of freedom as well as its basic movements.

2.2 Characteristics and configurations

The quadrotor is considered as one of the UAVs that have the most complex dynamic, considering its six-degree-of-freedom (6-DOF) nonlinearities (i.e., nonlinear system, strong coupling system and underactuated system), whose mathematical dynamic model is derived via Newton– Euler formalism. To understand its movement, we have to place two coordinates: The first is a static Galilei coordinate of the earth frame $E_{(x_e,y_e,z_e)}$ and the other one is $B_{(x_b,y_b,z_b)}$ body-fixed frame placed on the structure coincident to the vehicle inertia axes as shown in Figure. 2.1. The solid structure of the quadrotor is assumed to be symmetrical; also, the thrust and drag are proportional to the square of the propellers speed. The quadrotor is able to move in 3D space according to (see Figure 2.2):

- Vertical movement (lift) is obtained from the rotation of all four propellers at the same time with the same rotation speed;
- The displacement along the x axis occurs following a small rotation around y axis, this rotation θ called pitch is created from the difference in the lift of rotors 1 and 3;
- The displacement along the y axis occurs following a small rotation around x axis, this rotation ϕ called roll is created from the difference in the lift of rotors 2 and 4;

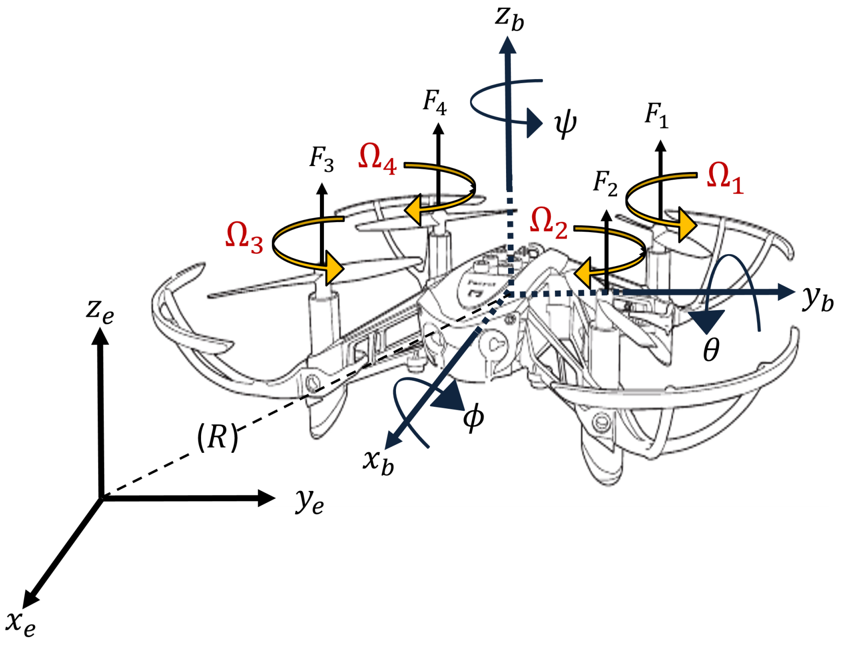


FIGURE 2.1: Quadrotor configuration.

- Yaw ψ rotation around z axis requires two rotors on the same axis to rotate in one direction while the other two in the other direction.

2.3 Kinematics

2.3.1 Rotation matrix

The rotation of a rigid body in space can be described by the orientation between the frames $E_{(x_e,y_e,z_e)}$ and $B_{(x_b,y_b,z_b)}$ and will be given by an orthogonal rotation matrix. Choosing the Euler representation, 3 elementary rotations of (ϕ, θ, ψ) angles around the (x, y, z) axes respectively can be defined as [35]

$$R_{x\phi} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{pmatrix}, R_{y\theta} = \begin{pmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{pmatrix}, R_{z\psi} = \begin{pmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.1}$$

where s and c denote sin(.) and cos(.), respectively. $R_{x\phi}$ is the matrix of rotation angle ϕ around B_{x_b} called roll rotation matrix. $R_{y\theta}$ is the matrix of rotation angle θ around B_{y_b} called pitch rotation matrix. $R_{z\psi}$ is the matrix of rotation angle

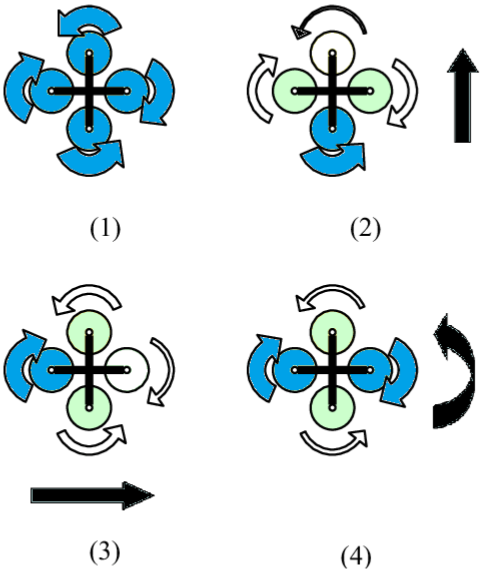


FIGURE 2.2: The concept of the direction control of the quadrotor. The arrow's size is proportional to the corresponding rotor's speed. (1) Going up, (2) Move forward, (3) Move to the right, (4) Rotate counter-clockwise.

ψ around B_{z_b} called yaw rotation matrix. Where the global rotation matrix can be defined as:

$$R = R_{z\psi}.R_{y\theta}.R_{x\phi} = \begin{pmatrix} c\theta c\psi & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ c\theta s\psi & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix} \tag{2.2}$$

2.3.2 Angular rates

The passage between the angular rates $\dot{\Theta} = (\dot{\phi}, \dot{\theta}, \dot{\psi})^T$ in $E_{(x_e,y_e,z_e)}$ and the body angular rates $\omega = (p, q, r)^T$ in $B_{(x_b,y_b,z_b)}$ which are physically measured by the gyroscope is made through the transformation matrix as [72]

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \tag{2.3}$$

2.3.3 Force dynamics

Initially, according to Newton Euler theory it is known that any rigid body in motion is subject to a sum of forces. In this purpose, we can consider a sum of forces that will be applied to the quadrotor; When the propellers which are fixed on the four rotors start to turn, it will push the air downward, which create a vertical aerodynamic lift force. The mass of the vehicle has a negative gravity force effect according to the z-axis. The quadrotor is subject to ground effect forces obtained from the lift force at the moment when the vehicle is near the ground (see figure 2.3), but in general, this effect is neglected because the altitude is varied within a small range. Finally, the quadrotor can be subjected to an external disturbance force like wind effect.

Then, the position dynamics of the drone in the body frame yield

$$m\begin{pmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{pmatrix} + m\begin{pmatrix}p\\ q\\ r\end{pmatrix} \times \begin{pmatrix}u\\ v\\ w\end{pmatrix} = \begin{pmatrix}\sum_{i=1}^{4} F_i\\ \sum_{i=1}^{4} F_i\\ \sum_{i=1}^{4} F_i\end{pmatrix} + mR^{-1}\begin{pmatrix}0\\ 0\\ -g\end{pmatrix} + R^{-1}\begin{pmatrix}F_x^d\\ F_y^d\\ F_z^d\end{pmatrix} \tag{2.4}$$

where (u, v, w) are the linear velocities expressed in body frame, m is the mass of the drone, g is the gravity acceleration in Earth frame, F_i is the magnitude

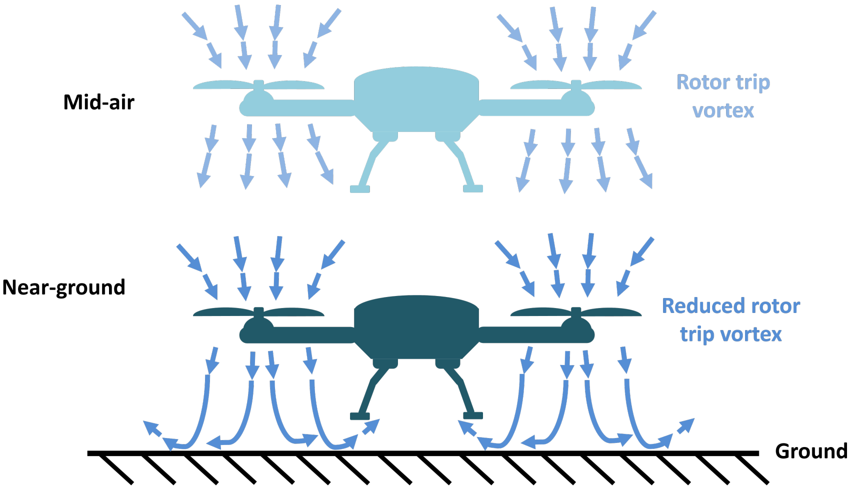


FIGURE 2.3: Vortex change in near-ground/mid-air manoeuvre under ground effect.

of the lift force for each rotor $i = 1, 2, 3, 4$, $F^d_{x,y,z}$ is the magnitude of the disturbances force.(2.4) can be represented in the earth frame

$$m \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = R \begin{pmatrix} \sum_{i=1}^4 F_i \\ \sum_{i=1}^4 F_i \\ \sum_{i=1}^4 F_i \end{pmatrix} + m \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} + \begin{pmatrix} F^d_x \\ F^d_y \\ F^d_z \end{pmatrix} \tag{2.5}$$

The final expression of the position dynamics (2.5) can be rewritten as

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = \frac{1}{m} \begin{pmatrix} (c\phi s\theta c\psi + s\phi s\phi)u_z + F^d_x \\ (s\phi s\theta c\psi - c\phi s\phi)u_z + F^d_y \\ (c\phi c\theta)u_z - mg + F^d_z \end{pmatrix} \tag{2.6}$$

where $u_z = \sum_{i=1}^4 F_i$ is the total lift force. By using the kinematics of a rigid body (2.6), we can obtain the velocity and position integrating the linear acceleration as

$$\wp = (x, y, z)^T = \int (\dot{x}, \dot{y}, \dot{z})^T dt = \iint (\ddot{x}, \ddot{y}, \ddot{z})^T dt \tag{2.7}$$

2.3.4 Moment Dynamics

For the rotary motion of the drone, it is necessary to establish sum of moment torques with respect to the Earth frame as

$$I \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = - \begin{pmatrix} p \\ q \\ r \end{pmatrix} \times I \begin{pmatrix} p \\ q \\ r \end{pmatrix} + \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} + \begin{pmatrix} -Jp\Omega_r \\ Jq\Omega_r \\ -J\dot{\Omega}_r \end{pmatrix} + \begin{pmatrix} \tau^d_\phi \\ \tau^d_\theta \\ \tau^d_\psi \end{pmatrix} \tag{2.8}$$

where $\tau_{\phi,\theta,\psi}$ are the external aerodynamic moments in the body frame, $\tau^d_{\phi,\theta,\psi}$ are the disturbances torques, $\Omega_r = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$ is the propeller angular generated by the angular velocity Ω_i for each rotor $(i = 1, 2, 3, 4)$, J is the propeller inertia and I represents the quadrotor inertia matrix as

$$I = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \tag{2.9}$$

whith (I_x, I_y, I_z) are the inertia of the drone around (x, y, z) axes respectively. The vector resulting from the moments applied to the structure of the quadrotor $\tau = (\tau_\phi, \tau_\theta, \tau_\psi)^T$ can be computed as

$$\begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \begin{pmatrix} d(F_1 - F_3) \\ d(F_2 - F_4) \\ b(F_1 - F_2 + F_3 - F_4) \end{pmatrix} \tag{2.10}$$

d is the distance between the center of mass of the quadrotor and the axis of rotation of the rotor, and b is the rotor's reaction torque constant. The final representation of the rotation system can be obtained from (2.8-2.10), where we get $(p, q, r)^T \approx (\phi, \theta, \psi)^T$

$$\begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \frac{1}{I_x}[(I_y - I_z)\dot{\theta}\dot{\psi} - J\Omega_r\dot{\theta} + du_\phi + \tau_\phi^d] \\ \frac{1}{I_y}[(I_z - I_x)\dot{\phi}\dot{\psi} + J\Omega_r\dot{\theta} + du_\theta + \tau_\theta^d] \\ \frac{1}{I_z}[(I_x - I_y)\dot{\phi}\dot{\theta} + u_\psi + \tau_\psi^d] \end{pmatrix} \tag{2.11}$$

where $(u_\phi, u_\theta, u_\psi)^T = (\tau_\phi, \tau_\theta, \tau_\psi)^T$ are the total torques generated by the forces F_i. We can obtain the angular velocity and Euler angles by integrating the acceleration angular as

$$\Theta = (\phi, \theta, \psi)^T = \int (\dot{\phi}, \dot{\theta}, \dot{\psi})^T dt = \iint (\ddot{\phi}, \ddot{\theta}, \ddot{\psi})^T dt \tag{2.12}$$

From (2.6) and (2.11), the overall dynamic of the quadrotor can be represented in the earth frame as

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} \frac{1}{m}[(c\phi s\theta c\psi + s\phi s\phi)u_z + F_x^d] \\ \frac{1}{m}[(s\phi s\theta c\psi - c\phi s\phi)u_z + F_y^d] \\ \frac{1}{m}[(c\phi c\theta)u_z + F_z^d] - g \\ \frac{1}{I_x}[(I_y - I_z)\dot{\theta}\dot{\psi} - J\Omega_r\dot{\theta} + du_\phi + \tau_\phi^d] \\ \frac{1}{I_y}[(I_z - I_x)\dot{\phi}\dot{\psi} + J\Omega_r\dot{\theta} + du_\theta + \tau_\theta^d] \\ \frac{1}{I_z}[(I_x - I_y)\dot{\phi}\dot{\theta} + u_\psi + \tau_\psi^d] \end{pmatrix} \tag{2.13}$$

We can observe that the quadrotor has a nonlinear dynamic with strong coupling and an interconnection between the state variables. In some additional works [56, 3], the dynamic model of the quadrotor (2.13) can be found as linear model by using the linearization theory

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{pmatrix} (\theta u_z + F_x^d)/m \\ (-\phi u_z + F_y^d)/m \\ (u_z + F_z^d - mg)/m \\ u_\phi/I_x \\ u_\theta/I_y \\ u_\psi/I_z \end{pmatrix} \tag{2.14}$$

2.4 Actuator dynamics

The rotor used for the UAV is composed of an Electronic Speed Controller (ESC), brushless motor and a Propeller. The rotors are controlled by DC-motors using Pulse Width Modulation (PWM) signals, which connect the electrical and the mechanical quantities together. The DC-motor can be described by the following dynamic equation [72]

$$\begin{cases} L_{rot}\frac{di_{rot}}{dt} = u_{rot} - R_{rot}i_{rot} - K_e\Omega \\ J\dot{\Omega} = \tau_m - \tau_d \end{cases} \tag{2.15}$$

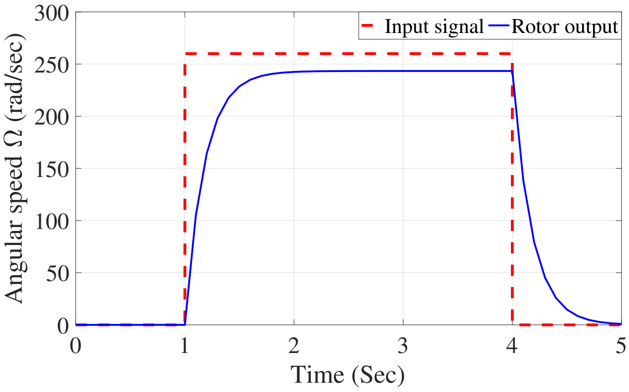


FIGURE 2.4: Rotor model response.

where u_{rot} is the rotor input, i_{rot} is the current in the rotor, L_{rot} is the rotor inductance, R_{rot} is the rotor internal resistance, K_e is the Electromagnetic Force constant, τ_m and τ_d are the rotor torque and the rotor load respectively. Experimentally, the dynamic of a motor with very low inductance can be approximated by a linearization around an operating point Ω_0 to get a first-order transfer function as

$$G(s) = \frac{0.936}{1 + 0.178s} \tag{2.16}$$

It can be seen from figure. 2.4 that the rotor model response has a small time-delay.

2.5 State-space representation

Considering equation (2.13) with neglecting the external disturbances, six second-order differential equations are considered to constitute the quadrotor nonlinear model under the designed conditions with four control inputs namely, $U = [u_z, u_\phi, u_\theta, u_\psi]^T$. The position and the attitude of quadrotor denoted as $\wp$ and Θ respectively are set as the outputs, namely $y = [x, y, z, \phi, \theta, \psi]^T$. The dynamic equation (2.13) is considered as an affine nonlinear system which can be written in the following form

$$\dot{X} = F(X) + G(X)U \tag{2.17}$$

where $X \in \Re^{12}$ represents the state variables, $F(X) \in \Re^{12\times 1}$ and $G(X) \in \Re^{12\times 4}$ are nonlinear functions matrix, with

$$\begin{aligned} X &= [x_{x,1}, x_{x,2}, x_{y,1}, x_{y,2}, x_{z,1}, x_{z,2}, x_{\phi,1}, x_{\phi,2}, x_{\theta,1}, x_{\theta,2}, x_{\psi,1}, x_{\psi,2}]^T \\ &= [x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T \end{aligned} \tag{2.18}$$

and,

$$F(X) = \begin{pmatrix} x_{x,2} \\ 0 \\ x_{y,2} \\ 0 \\ x_{z,2} \\ -g \\ x_{\phi,2} \\ \frac{I_y-I_z}{I_x} x_{\theta,2} x_{\psi,2} - \frac{J_r\Omega_r}{I_x} x_{\theta,2} \\ x_{\theta,2} \\ \frac{I_z-I_x}{I_y} x_{\phi,2} x_{\psi,2} + \frac{J_r\Omega_r}{I_y} x_{\phi,2} \\ x_{\psi,2} \\ \frac{I_x-I_y}{I_z} x_{\theta,2} x_{\phi,2} \end{pmatrix}, G(X) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{cx_{\phi,1}sx_{\theta,1}cx_{\psi,1}+sx_{\psi,1}sx_{\phi,1}}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{cx_{\phi,1}sx_{\theta,1}sx_{\psi,1}-sx_{\psi,1}cx_{\phi,1}}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{sx_{\phi,1}cx_{\theta,1}}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{l}{I_x} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{l}{I_y} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \end{pmatrix} \tag{2.19}$$

It should be noted that the previous quadrotor model is obtained after some simplifications in its physical conception, such as the quadrotor has a rigid and symmetric body. The four rotors and propellers of the quadrotor have the same characteristics. The roll, the pitch and the yaw angles, ϕ, θ and ψ are bounded as $\frac{-\pi}{2} < \phi < \frac{\pi}{2}$, $\frac{-\pi}{2} < \theta < \frac{\pi}{2}$ and $-\pi < \psi < \pi$. These limitations are put to ensure the controllability of the overall system.

Since the quadrotor is an underactuated system, in which the six outputs $y_i \in \{x, y, z, \phi, \theta, \psi\}$ are controlled only by four inputs $u_i \in \{u_z, u_\phi, u_\theta, u_\psi\}$, it is difficult to control all the six subsystems independently. To overcome this problem, two virtual control inputs u_x and u_y are created to drive the Cartesian

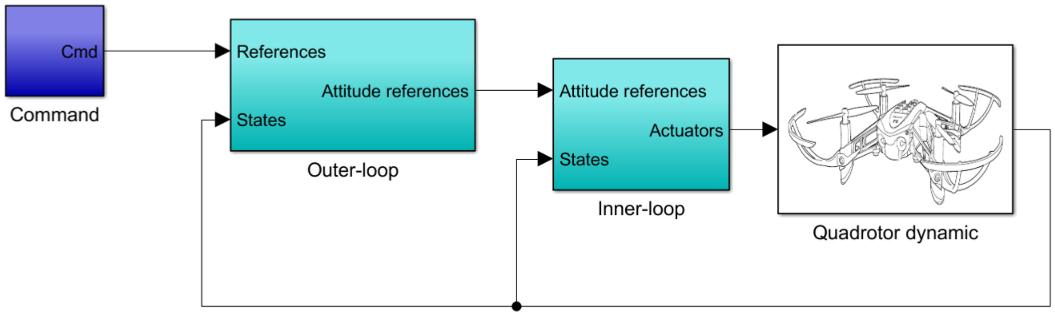


FIGURE 2.5: Quadrotor system control with two cascade feedback strategy (inner/outer-loop).

position subsystems x and y, respectively. From equation (2.13), their expressions are chosen as follows

$$\begin{cases} u_x = (cx_{\phi,1}sx_{\theta,1}cx_{\psi,1} + sx_{\psi,1}sx_{\phi,1})u_z \\ u_y = (cx_{\phi,1}sx_{\theta,1}sx_{\psi,1} - sx_{\psi,1}cx_{\phi,1})u_z \end{cases} \tag{2.20}$$

Then, the desired roll ϕ_d angle and the desired pitch angle θ_d can be computed as

$$\begin{cases} \phi_d = (u_x s\psi_d - u_y c\psi_d)/u_z \\ \theta_d = (u_x c\psi_d + u_y s\psi_d)/u_z \end{cases} \tag{2.21}$$

Remark 2.1. Obviously, from equation (2.21), ϕ_d and θ_d are well defined since u_z is nonsingular, i.e. u_z is strictly positive since $u_z = \sum_{i=1}^{4} F_i$ that expresses the safely flight condition.

The quadrotor control is based on cascade feedback which concerns the following steps. First, the quadrotor system is divided into two cascade loops including the inner-loop (attitude dynamics loop) and the outer-loop (position dynamics loop). Then, the control strategy is designed for each subsystem in each loop individually. Finally, the outer-loop is connected with the inner-loop in a way that provides a reference for the inner-loop. Figure 2.5 illustrates a simplified scheme of the cascade feedback strategy.

2.6 Summary

This chapter presented the dynamic modelling using Euler-Network formalism to establish a model which describes almost all the physical phenomena acting on the quadrotor. The obtained model is a nonlinear model with a strong coupling between the state variables. In addition, this model is an underactuated system which has six outputs controlled with four control inputs and thus their direct application for synthesis of control is complicated. To overcome this problem, some acceptable simplifications are used which lead to create two virtual inputs. Last, the overall quadrotor system is rewritten in the state-space representation to get better visibility. Using this state-space form, control laws are computed in the next chapters.

Chapter 3

Metaheuristic Algorithms

3.1 Introduction

In the last decade, optimization has become a key issue in many applications such as engineering process, business activities, and industrial designs etc. The optimization target may be to find some parameters that minimize energy consumption and costs, or maximize profit, production, performance, and efficiency. It is no exaggeration to say that optimization is necessary for everything, because resources, time and money are always limited in real world applications, therefore, we must find solutions to optimize the use of these precious resources under various constraints.

Since most real-world applications are often highly non-linear, they require sophisticated optimization tools. Nowadays, computer simulations are becoming an indispensable tool to solve these optimization problems with different efficient search algorithms. One can define in literature various optimization algorithms, where the metaheuristic algorithms show their performance in several computation.

In this part, we will put the point on some metaheuristic algorithms that are used in our work and which are the Cuckoo Search algorithm (CS), the Bat Algorithm (BA) and the Flower Pollination Algorithm (FPA).

3.2 Optimization algorithms classification

In literature, it can be specified three classes of optimization algorithms: conventional or classical methods, evolutionary algorithms, and swarm intelligence algorithms.

Classical optimization methods: This class of optimization strategy is continuous optimization functions that are differential; The idea of their use involves the computation of the derivative degree of the objective function to find the maximum or minimum. The obtained target solution could be locally or globally but the target of optimization is to achieve the best global solution. Generally, these optimization strategies are based on the gradient methods and supply deterministic solutions indoors continuous search space. We can name for example, the simplex method, Kuhn–Tucker method, Newton's method and Lagrangian method.

Evolutionary optimization algorithms: The algorithms from this class, developed in 1950, are research optimization techniques inspired by the biological evolution of species. They draw inspiration from the evolution of living things (Darwin's theory of natural selection of species, mating, reproduction, crossover, and mutation operators) to solve optimization problems. The principle of these algorithms is that individuals who have inherited characters well adapted to their environment tend to live long enough to reproduce, while the weakers tend to disappear. Genetic algorithm (GA), genetic programming (GP), and differential evolution (DE) are three of the most popular evolutionary algorithms.

Swarm intelligence algorithms: These algorithms have computational intelligence built into them and inspired from nature; They are developed based on the study of natural processes that could be physical, chemical, or biological and the behavior of animals, birds and insects. They are mostly population-based and use their intelligence swarm of group to achieve the optimal solution. The Cuckoo Search algorithm (CS), Bat Algorithm (BA) and Flower Pollination Algorithm (FPA) are the most algorithms known in literature that have been discussed in this chapter. They are characterized by their good perception capabilities, adaptive and flexible in the environment utilizing their accessible resources.

3.3 How to choose an optimization algorithm?

In many cases, for a given type of problem, we might not know which algorithm is relevant until we try it. There may be a set of efficient algorithms to solve such problems. In some cases, such algorithms can still be developed. Even for existing algorithms, the choice depends largely on the expertise of the decision-maker, the resources available and the type of problem. Ideally, the best available algorithms and tools should be used to solve a given problem. However, the correct use of these tools may still depend on the user's experience. In addition, resources such as computing costs, software availability and solution production time will also be important factors in deciding which algorithms and methods to use.

For classical algorithms like gradient algorithm and simplex methods, we know what types of problems they can typically solve. However, for new algorithms, as with most algorithms inspired from nature, it needs to conduct extensive studies to validate and test their performance. This is the case for our work.

It is interesting to point out that any specific knowledge of a particular problem is always useful for the appropriate choice of the best and most efficient methods for the optimization procedure.

3.4 Optimization using cuckoo search algorithm (CS)

3.4.1 Inspiration

Cuckoo Search algorithm (CS)is an optimization algorithm, which has been developed by [99], and is one of the meta-heuristic algorithms inspired from the cuckoo behavior. To describe the cuckoo search algorithm steps, three idealized rules one has to point:

- Each cuckoo can lay one egg in a randomly chosen nest through time;
- The best high quality egg nests (solutions) will be carried over to future generations;
- The number of available reception nests is fixed, and a host can discover a strangled egg with a probability P_a. In this case, the host bird can either throw the egg away or abandon the nest in order to build a new nest in a new location.

The quality or the fitness function of a solution can simply be proportional to the objective function.

3.4.2 CS algorithm

The CS algorithm searches for the best solution used local x_i and global v_i random search nests, which contain random values of the parameters required to be optimized and controlled by the probability parameter P_a at each generation i through the following equations

$$x_i^{t+1} = x_i^t + \delta s \otimes H\left(P_a - \varepsilon\right) \otimes \left(x_p^t - x_k^t\right) \tag{3.1}$$

$$v_i^{t+1} = x_i^t + \delta L(s, \lambda) \tag{3.2}$$

where $\otimes$ denotes entry-wise product of two vectors, δ is positive step size scaling factor, s is the step size, H is a Heavy-side function, $\varepsilon \in [0, 1]$ is a random number, x_p^t and x_k^t two different solutions are selected randomly, $L(s, \lambda)$ represents lévely flights distribution used to define the step size of random walk. The local and global random walk is controlled by the probability factor as

$$q_i^{t+1} = \begin{cases} x_i^{t+1} \; if \text{ rand}\left(0, 1\right) > P_a \\ v_i^{t+1} \; otherwise \end{cases} \tag{3.3}$$

The target solution will be selected by minimizing the cost of an objective function J as

$$x_i^{best} = \begin{cases} q_i^{t+1} \; if \text{ J}(q_i^{t+1}) < \text{J}(x_i^t) \\ x_i^t \; otherwise \end{cases} \tag{3.4}$$

The main steps of the CS can be outlined as the algorithm shown in Algorithm 1.

3.4.3 Applications

The CS has proved its efficiency in many areas of optimization applications such as in [85], where an improved cuckoo search algorithm based on compact and parallel techniques for three-dimensional path planning of unmanned robots. CS algorithm has been employed for automatic different image processing applications contrast enhancement showing promising performance in [46]. Regarding the optimization in the control domain, it can determine diverse works that use the CS algorithm to tune the parameters of the designed controller [2, 15, 32, 101, 35]. In literature, we can find many cases for this algorithm

Algorithm 1 Pseudo-code of the cuckoo search algorithm

```
Cost function min $J$
Initialize $n$ population of host nests $x_i$
while t < Max Generation do
    Evaluate the local random search $x_i^t$ by Lévly flights.
    Generate new solution using the global search. Eq(3.2)
    A fraction of worse nests are abandoned via probability factor $p_a$ to build new ones $q_{i+1}$. Eq(3.3)
    Evaluate the cost function $J(q_{i+1})$
    if $J(q_{i+1}) < J(x_i)$ then
        Found the new solution $x_i^{best}$. Eq(3.4)
    end if
    Keep the best solutions (or nests with quality solutions).
end while
Rank the solutions and find the current best.
```

like modified CS [67, 95, 70] and hybrid with other algorithms [80, 52]. The mentioned studies have shown that CS can perform significantly and prove powerful convergence in many applications.

3.5 Optimization using bat algorithm (BA)

3.5.1 Basics

By studying the behaviour of the bat which uses a type of sonar called echolocation, a bat algorithm (BA) was developed by Xin-She Yang [96] based on three laws:

- The bats fly at a certain altitude to search for a target and to identify the kind of this target which can be prey or food. Also, to differentiate between them and to know the distance from the target or to detect obstacles, bats use echolocation;

- Bats fly randomly at location x_i with velocity v_i and use wavelength and loudness in the search for prey, and can adjust the frequency Q_i of pulses depending on the proximity of the target;

- The loudness can vary in different ways but it can determines its variation in a range of positive value A_0 to the value of small A_{min}.

Algorithm 2 Pseudo-code of bat algorithm (BA)

```
Cost function Min $J$
Initialize $Np$ Bat population: locations $x_i$
and velocities $v_i$ with random solution
Initialize frequencies $f_i$, pulse rates $r_i$ and the loudness $\vartheta_i$
Find the best solution $x^{best}$ in the initial population
while $J >$ desired tolerance do
    Generate new solutions by update frequency, velocities and location and
    update velocities Eqs. (3.5-3.7),
    if $rand > r_i$ then
        Select a solution $x_i^t$ among the best solutions
        Generate a local solution around the selected best solution
    end if
    if $(rand < \vartheta_i$ and $J(x_i^{new}) < J(x_i))$ then
        Accept the new solutions
        Increase $r_i$ and Reduce $\vartheta_i$
    end if
    Rank the bats and find the current best $x^{best}$
end while
Output the best solution found $x^{best}$
```

Based on these approximations, the main rules of BA can be summarized in the pseudo code shown in algorithm 2.

3.5.2 BA process

The virtual bats fly at a certain position x_i and velocities v_i which have to define in a d-dimensional search space that is updated for each generation i. The new solutions x_i^t and velocities v_j^t at time step t are generated through the frequency f_i as:

$$f_i = f_{min} + (f_{min} - f_{max}) \times rand(0,1) \tag{3.5}$$

$$v_i^{new} = v_i^t + (x_i^t - x_i^{best}) \times f_i \tag{3.6}$$

$$x_i^{new} = x_i^t + x_i^{new} \tag{3.7}$$

For each generation a new solution is generated locally using random walk as [98]

$$x_i^{new} = x_i^{best} + rand(-1,1) \times \hat{\vartheta}_i^{(t)} \tag{3.8}$$

while $\hat{\vartheta}^{(t)}$ is the average loudness of bats, where the new bat solution x_i^{new} is generated as:

$$x_i^{new} = \begin{cases} x_i^{new} \text{ If } rand(0,1) > \hat{r}_i^{(t)} \\ x_i \text{ otherwise} \end{cases} \tag{3.9}$$

The evolution rules pulse rate $\hat{r}^{(t)}$ and loudness $\vartheta^{(t)}$ are as follows:

$$\hat{r}_i^{(t+1)} = r_0^t(1 - \exp(-\gamma t)) \tag{3.10}$$

$$\hat{\vartheta}_i^{(t+1)} = \sigma\vartheta^{(t)} \tag{3.11}$$

The best solution will be selected based on objective function values

$$x_i^{best} = \begin{cases} x^{new} \text{ If } \mathrm{J}(x^{new}) < \mathrm{J}(x_i^t) \\ x_i^t \; otherwise \end{cases} \tag{3.12}$$

Remark 3.1 It can be seen that it can capture the BA essentially from the standard PSO. If we replace the variations of the frequency f_i by a random parameter and setting $\vartheta = 0$ and $r_i = 1$. So it can be considered that the PSO is a special cases of BA.

3.5.3 Applications

The bat algorithm has been used in almost every area of optimization to solve several engineering problems. We briefly highlight some of the applications, such as sizing of battery energy storage for micro grid operation management [10], power system stabilizers [7, 79], image processing [102, 47], among the BA applications, tune parameter control has been addressed in many works [54, 87]. There exist other developed algorithms that are inspired from the standard BA like directional bat algorithm, evolutionary discrete bat algorithm, binary bat algorithm and hybrid bat algorithm with other metaheuristic algorithms.

3.6 Optimization using flower pollination (FPA)

3.6.1 Characteristics of flower pollination

The Flower Pollination Algorithm (FPA) is a metaheuristic algorithm introduced by Xin-She Yang [97], which is inspired by the biological flower pollination in a flowering plant. This algorithm is based on the following rules:

- Processes of global pollination can be achieved using pollen of the same flower, and using pollen from a flower of some other plant of the same species (the pollen-carrying pollinators move in a way that obeys the Levy flight).
- Abiotic and self pollination occur for local pollination without the help of any insect.
- Flower constancy can be developed with the help of pollinators such as insects. Reproduction probability constancy is proportional to the similarity of the two flowers involved.
- A switch probability $\wp$ can control the interaction of local pollination and global pollination, slightly biased toward local pollination because of physical proximity and wind.

This idea can be extended to carry multiple flowers and multiple pollen gametes in order to solve multi-objective optimization problems.

3.6.2 FPA process

To express the updating mathematical formulas for the FPA, it has to convert the mentioned rules into roper updating equations. For example, let the solution of the t^{th} iteration of flower i be S_i^t and the global best flower of population vector be S^*. The iteration flower constancy can be represented as

$$S_i^{t+1} = S_i^t + \gamma L(\lambda)(S^* - S_i^t) \tag{3.13}$$

Algorithm 3 Pseudo-code of the flower pollination algorithm

```
Objective function Min $J$
Initialize $N_P$ population of flowers $S_i$ with random solution.
Find the best solution $S_i^*$ in the initial population.
Define a switch probability $\wp \in [0,1]$
while t < Max Generation do
    Global search process Eq (3.13) using Lévly flights.
    if $rand(0,1) < \wp$ then
        Do local pollination Eq (3.15).
    end if
    Evaluate the cost function $J(S^{t+1})$
    if $J(S^t) > J(S^{t+1})$ then
        Replace by the new solution $S^{t+1}$.
    end if
    Keep the best solution.
end while
Rank the solutions and find the best one.
```

where γ is a constant, and $L(\lambda)$ is the strength of the pollination (the step size of the Levy flight). The Levy distribution is used to represent the flight of the insects as in [97]

$$L \sim \frac{\lambda \nu(\lambda) sin(\pi\lambda/2)}{\varpi} \frac{1}{l^{1+\lambda}} \tag{3.14}$$

where ϖ is a constant which is the same equal to Archimedes constant, λ is a constant, $\nu(\lambda)$ represents the standard gamma function, $l = \frac{U}{|V|^{1/\lambda}}$, $U \sim N(0,\sigma^2)$, $V \sim N(0,1)$ with $\sigma^2 = (\frac{\nu(1+\lambda)}{\lambda\nu((1+\lambda)/2)} \cdot \frac{sin(\varpi\lambda/2)}{2^{(\lambda-1)/2}})^{1/\lambda}$.

In some flowers, self pollination takes place with the following local search dynamic

$$v_i^{t+1} = S_i^t + \varepsilon(S_n^t - S_k^t) \tag{3.15}$$

where $\varepsilon \in [0,1]$ is a random number, S_n^t, S_k^t are the pollen from different flower positions in the population. For each flower, the interaction or switching common global pollination to intensive local pollination can be controlled using the switching probability $\wp$ as

$$S_i^{t+1} = \begin{cases} v_i^{t+1} \text{ If } rand(0,1) > \wp \\ S_i^{t+1} \; Otherwise \end{cases} \tag{3.16}$$

The selection operator of the best current global flower depends from the flower having the least value of the objective function J during the pollination as

$$S_i^{best} = \begin{cases} S_i^{t+1} \text{ If } \mathrm{J}(S_i^{t+1}) < \mathrm{J}(S_i^t) \\ S_i^t \; otherwise \end{cases} \tag{3.17}$$

To give more clarification, the main steps of flower pollination algorithm have been summarized in Algorithm 3.

3.6.3 Applications

Flower pollination algorithm can be applied in diverse range of real-world scenarios for solving optimization problems like power system [1, 22, 63], estimation of photovoltaic parameters [6, 94, 12], image processing [31, 69, 101]. We can find successful works for the algorithm in wireless and network system domain [41, 82], regarding control domain, it is important to mention reached works which are using the FPA to search for the optimal parameters of the designed controller [22, 14, 75]. In addition, other modified FPA techniques have been developed through the standard one such as modified flower pollination algorithm (MFPA) , Binary FPA (BFPA), Chotic-Based FPA (CFPA) and hybridization of standard FPA with other algorithms.

3.7 Other algorithms

There are many other metaheuristic algorithms in literature and more algorithms are appearing. However, we could not discuss all of them. Rather, we provide a subset of these algorithms so that readers can see the search mechanisms of various algorithms. In this chapter, we can highlight some metaheuristic algorithms in diverse applications such as:

- Dolphin echolocation algorithm [49, 48];
- Eagle strategy algorithm [29, 50];
- Bee colony algorithm [26, 68, 86];

- Gravitational search algorithm [57, 71, 33];
- Black-hole algorithm [30, 91];
- Differential search algorithm [8, 19].

It could find many optimization algorithms that we didn't mention in this list and other hybrid algorithms which are developed by combining two or more algorithms inspired from the original enhance overall search efficiency.

3.8 Summary

An overview of nature-inspired optimization algorithms and their various applications have been given in this chapter and some of other algorithms have been listed.

Metaheuristic algorithms are nature-inspired and are developed to show their superior innate capacity to solve challenging problems efficiently. Many nonlinear applications in the real world are based requiring state-of-the-art optimization strategies to deal with them. These algorithms are flexible, adaptive, self-organized and population-based on pure interactions amid search process. The metaheuristic algorithms process based on increasing the variety of solutions decreases the possibility of getting trapped in the local optimal solution, hence these algorithms can deal with complex problems quite efficiently with shortest time complexity. These algorithms will be used to address the selection problem of the designed parameters of the control law in the next chapter.

Chapter 4

Optimal model-free-based backstepping control

4.1 Introduction

The UAV dynamics has many characteristics that complicate the process of designing the tracking control or the stabilization like nonlinearities, under-actuated, strong coupling system and external disturbances, etc. Since these characteristics are considered as unknown disturbances, we cannot resolve this problem by using classical controllers. For this purpose, model-free-based control is required.

In this chapter, we present a design of several optimal model-free controllers for the flight control of an UAV quadrotor. The proposed methods consist of using a model-free-based backstepping controller optimized by different metaheuristic algorithms. We begin this work by an introduction to the nonlinear model presentation that will be used in the current work. Then, we present the conventional backstepping controller (BC) that its development is based on an accurate knowledge of the dynamic system. Since our goal is to develop a model-free approach, a new estimator law is designed later and included to the BC in order to estimate the unknown dynamic model with disturbances and the control process is optimized via cuckoo search algorithm (CS). The designed strategy is well proved via Lyapunov theory and validated in different simulation cases. We close the chapter with the design of an optimal fuzzy model-free control approach, where this approach is proved through Lyapunov theory. The proposed strategy verified its performances within different simulation cases for the quadrotor attitude.

4.2 Nonlinear system representation

Recall that a nonlinear system is a system which has at least one nonlinear function in its dynamic. Recall that a nonlinear system is a system that has at least one nonlinear function in its dynamic. Typically, the behavior of a nonlinear system is described in mathematics by a nonlinear system of equations that can be presented in specific classes.

4.2.1 Single-Input Single-Output (SISO) nonlinear system representation

The position and attitude dynamics of the quadrotor, by recalling the model (2.17), can be introduced in i^{th} interconnected SISO system as

$$\begin{cases} \dot{x}_{i,1} = x_{i,2} \\ \dot{x}_{i,2} = f_i(X) + g_i(X)u_i \\ y_i = x_{i,1} \end{cases} \tag{4.1}$$

where $X \in \Re^{12}$ represents the state variables, $u_i \in \{u_x, u_y, u_z, u_\phi, u_\theta, u_\psi\} \in \Re$ is the control input, $f_i(X) \in \Re$ and $g_i(X) \in \Re$ are nonlinear smooth functions of the i^{th} subsystem. With,

$$\begin{aligned}
&f_x(X) = 0, f_y(X) = 0, f_z(X) = -g,\\
&f_\phi(X) = (I_y - I_z)/I_x x_{\theta,2} x_{\psi,2} - J_r \Omega_r / I_x x_{\theta,2},\\
&f_\theta(X) = (I_z - I_x)/I_y x_{\phi,2} x_{\psi,2} + J_r \Omega_r / I_y x_{\phi,2},\\
&f_\psi(X) = (I_x - I_y)/I_z x_{\theta,2} x_{\phi,2},\\
&g_x(X) = 1/m, g_y(X) = 1/m, g_z(X) = cx_{\phi,1} cx_{\theta,1}/m,\\
&g_\phi(X) = l/I_x, g_\theta(X) = l/I_y, g_\psi(X) = 1/I_z.
\end{aligned} \tag{4.2}$$

In order to design a model that considers the effect of the disturbance, the above dynamic model of the quadrotor can be reformulated as follows

$$\dot{x}_{i,2} = f_i(X) + g_i(X) u_i + h_i(t) \tag{4.3}$$

It should be mentioned that the disturbances $h_i(t)$ are added to the quadrotor model for considering the nonmodeled effects such as aerodynamic drag force, friction and wind effects. It's necessary to ensure the boundedness of all closed-loop system signals, for this purpose, the following assumptions are made.

Assumption 4.1 The reference trajectory y_{id} for $i \in \{x, y, z, \phi, \theta, \psi\}$ and their time derivatives $\dot{y}_{id}, \ddot{y}_{id}$ are assumed to be known, smooth and bounded.

Assumption 4.2 The state vector X is available for measurement, smooth and bounded, which means no observer is needed.

4.2.2 Modified representation

In order to reduce the number of nonlinear functions in the presented model (4.3), some manipulations are suggested in a way that can obtain a simple presentation with one nonlinear function by some manipulations as follow [35]:

By multiplying the equation (4.3) on the left by $g_i(X)^{-1}$, one has

$$0 = -g_i(x)^{-1}\dot{x}_{i,2} + g_i(X)^{-1}f_i(X) + u_i + g_i(X)^{-1}h_i(t) \tag{4.4}$$

By adding $\dot{x}_{i,2}$ to both sides of the equation (4.4), yields

$$\dot{x}_{i,2} = (1 - g_i(x)^{-1})\dot{x}_{i,2} + g_i(X)^{-1}f_i(X) + u_i + g_i(X)^{-1}h_i(t) \tag{4.5}$$

In order to get one nonlinear function which represents the global nonlinearities, we put

$$\Psi_i(X) = (1 - g_i(X)^{-1})\dot{x}_{i,2} + g_i(X)^{-1}(f_i(X) + h_i(t)) \tag{4.6}$$

Using (4.6), it can get a final model with the following compact form

$$\begin{cases} \dot{x}_{i,1} = x_{i,2} \\ \dot{x}_{i,2} = \Psi_i(X) + u_i \\ y_i = x_{i,1} \end{cases} \tag{4.7}$$

To ensure the controllability of the system (4.7), one assumes the following assumption.

Assumption 4.3 The nonlinear function $g_i(X)$ is assumed different from zero without changing its signal, it is assumed that $g_i > 0, \forall t > 0$.

Remark 4.1 Considering the generalities which have been used to get the nonlinear model (2.17), it is easy to check from **assumption 4.3**.

4.3 Classical backstepping controller

The backstepping controller (BC) design methodology provides a useful tool of designing in UAVs control area which has been successfully applied in several works [11, 21]. This is due to its ability to deal with the nonlinear problems and strict mathematical proof processes.

4.3.1 Control design

Our main objective in this section is to design BC for the closed-loop system (4.7) considering the nonlinear dynamical function $f_i(X)$ and the nonlinear control function $g_i(X)$ are known and the external disturbance function is $h_i(t) = 0$.

To design the BC, the following steps are introduced to realize the above-mentioned objective [35]:

Step 1. The tracking error and its derivative equations can be defined as

$$\begin{aligned} e_{i,1} &= y_{id} - x_{i,1} \\ \dot{e}_{i,1} &= \dot{y}_{id} - x_{i,2}, i \in \{x, y, z, \phi, \theta, \psi\} \end{aligned} \tag{4.8}$$

To ensure the stability and the convergence of the first tracking errors $e_{i,1}$, let's define the positive Lyapunov function

$$V_1 = \sum_{i \in \{x,y,z,\phi,\theta,\psi\}} V_{i,1} = \frac{1}{2} \sum_{i \in \{x,y,z,\phi,\theta,\psi\}} e_{i,1}^2 \tag{4.9}$$

where $V_{i,1} = \frac{1}{2} e_{i,1}^2$ is a local Lyapunov function of i^{th} subsystem. The time derivative of V_1 by substituting (4.8) can be found as follows

$$\dot{V}_1 = \sum_{i \in \{x,y,z,\phi,\theta,\psi\}} e_{i,1}(\dot{y}_{id} - x_{i,2}) \tag{4.10}$$

To satisfy the condition $\dot{V}_1$, $x_{i,2}$ are designed as virtual control input as

$$x_{i,2} = \dot{y}_{id} + \alpha_{i,1} e_{i,1} \tag{4.11}$$

where $\alpha_{i,1} > 0$ is designed parameter chosen by the designer. Substituting (4.11) into (4.10), one has

$$\dot{V}_1 = - \sum_{i \in \{x,y,z,\phi,\theta,\phi\}} \alpha_{i,1} e_{i,1}^2 \leq 0 \tag{4.12}$$

Step 2. Let's define the second tracking error

$$e_{i,2} = \dot{x}_{i,1} - x_{i,2} = \dot{x}_{i,1} - \dot{y}_{id} - \alpha_{i,1} e_{i,1} \tag{4.13}$$

Considering (4.7), the time derivative of (4.13) is described as follows

$$\dot{e}_{i,2} = \Psi_i(X) + u_i - \ddot{y}_{id} - \alpha_{i,1} \dot{e}_{i,1} \tag{4.14}$$

Similar to the previous step, consider the augmented Lyapunov function given by

$$V_2 = V_1 + \frac{1}{2} \sum_{i \in \{x,y,z,\phi,\theta,\phi\}} e_{i,2}^2 = \frac{1}{2} \sum_{i \in \{x,y,z,\phi,\theta,\phi\}} e_{i,1}^2 + e_{i,2}^2 \tag{4.15}$$

Its time derivative is

$$\dot{V}_2 = \sum_{i \in \{x,y,z,\phi,\theta,\phi\}} -\alpha_{i,1} e_{i,1}^2 + e_{i,2} (\Psi_i(X) + u_i - \ddot{y}_{id} - e_{i,1} - \alpha_{i,1} e_{i,1}) \tag{4.16}$$

Step 3. To ensure the stability of the overall system and the convergence of $e_{i,2} \to 0$ for $t \to \infty$, the feedback control inputs u_i are designed as

$$u_i = \ddot{y}_{id} - \Psi_i(X) + e_{i,1} + \alpha_{i,1} \dot{e}_{i,1} - \alpha_{i,2} e_{i,2} \tag{4.17}$$

where $\alpha_{i,2} > 0$ is designed parameter. Using (4.8) and (4.13), one can rewrite (4.17) as

$$u_i = \ddot{y}_{id} - \Psi_i(X) + (1 - \alpha_{i,1}^2) e_{i,1} + (\alpha_{i,1} + \alpha_{i,2}) e_{i,2} \tag{4.18}$$

Substituting (4.18) into (4.16), one gets

$$\dot{V}_2 = \sum_{i \in \{x,y,z,\phi,\theta,\psi\}} -\alpha_{i,1} e_{i,1}^2 - \alpha_{i,2} e_{i,2}^2 \leq 0 \tag{4.19}$$

Since $\dot{V}_2$ is semi-negative definite, the control law (4.18) asymptotically stabilizes the system (4.7). ■

TABLE 4.1: Parameters of the quadrotor

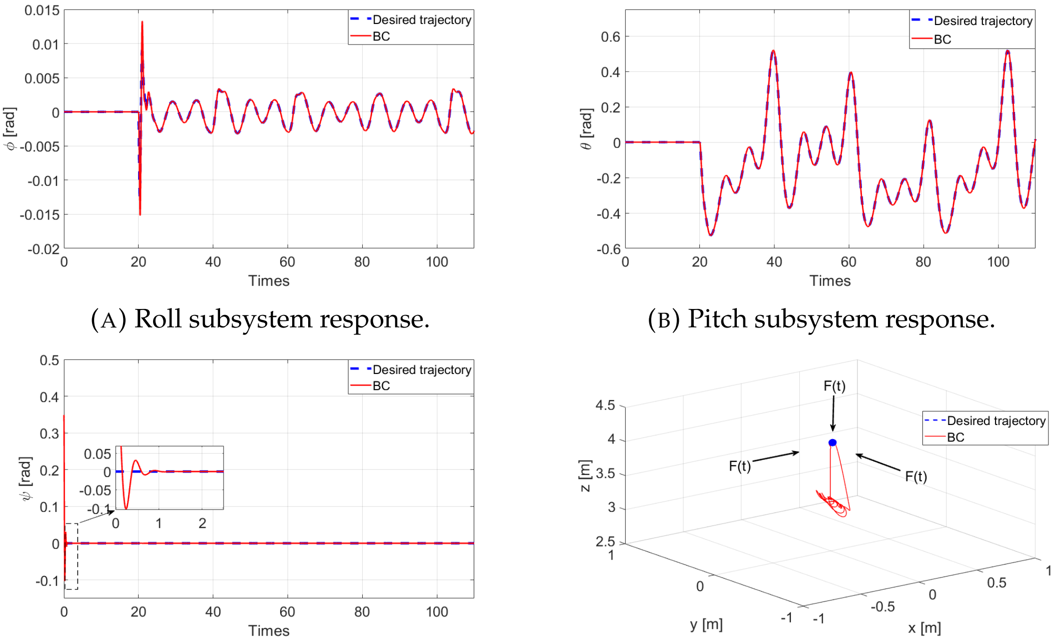
Parameters	Value
I_x	$7.5 \times 10^{-3}\ kg.m^2$
I_y	$7.5 \times 10^{-3}\ kg.m^2$
I_z	$1.3 \times 10^{-2}\ kg.m^2$
J_r	$6 \times 10^{-5}\ kg.m^2$
g	$9.81\ m/s^2$
d	$0.23\ m$
m	$0.65\ kg$

4.3.2 Simulations

In this section, simulation tests are given using MATLAB/Simulink environment. The parameters of the quadrotor used in the following simulations are shown in Table 4.1. The controller parameters have been arbitrarily chosen with taking into account the stability condition Eq.(4.19). Two different simulation cases are validated. The first case tested the stability performances under external disturbance. The second case discusses the trajectory tracking performances under external disturbance.

Case 1. The results of the proposed controller in stabilizing case are obtained for stable hoverating state. The desired trajectory is defined as
$y_{xd} = 0m$, $y_{yd} = 0m$, $y_{zd} = 4m$, $y_{\psi d} = 0rad$. The initial position of the quadrotor is: $\wp_0 = [0, 0, 3.5]^T m$ which respects to initially hover to certain altitude without moving, the initial rotation is: $\Theta_0 = [0, 0, 0.34]^T rad$.
It can be seen from figure 4.1 that the backstepping control law introduced in this section cannot achieve the stabilization of the quadrotor under the external disturbances.

Case 2. In this case, the quadrotor is assumed to track the designed path that is made up of a set of line stretches (square trajectory) with altitude $y_{zd} = 4m$ and $y_{\psi d} = 1rad$ as



(A) Roll subsystem response.

(B) Pitch subsystem response.

(C) Yaw subsystem response.

(D) Position system response.

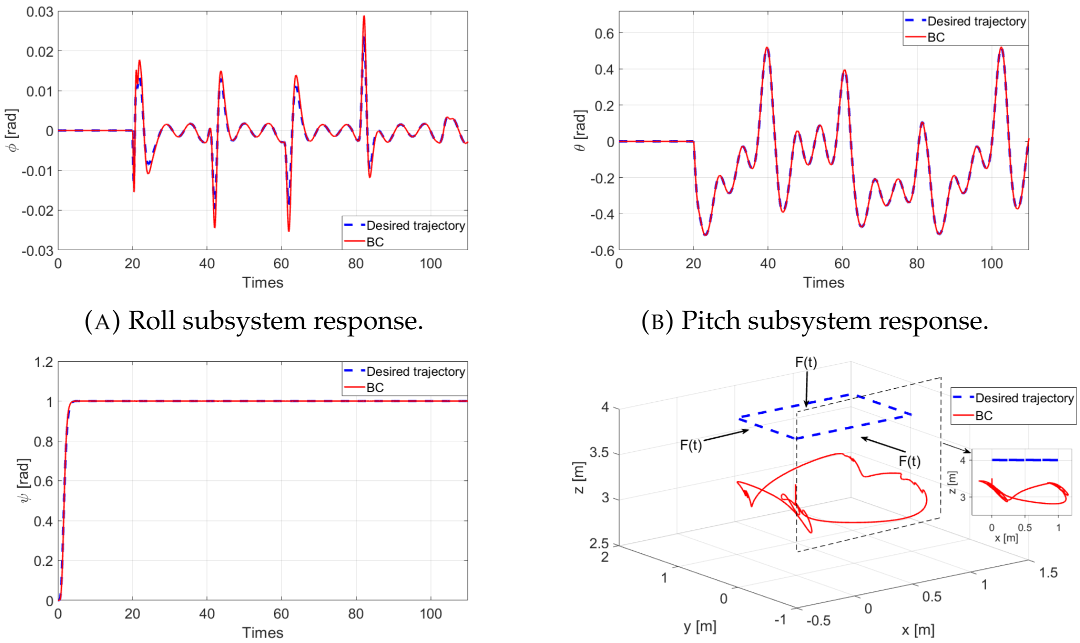
FIGURE 4.1: Quadrotor responses (case 1).

$$y_{xd} = \begin{cases} 0m, \ t < 20s \\ 1m, \ 20s \leq t < 40s \\ 1m, \ 40s \leq t < 60s \\ 0m, \ 60s \leq t < 80s \\ 0m, \ 80s \leq t \end{cases}, \quad y_{yd} = \begin{cases} 0m, \ t < 20s \\ 0m, \ 20s \leq t < 40s \\ 1m, \ 40s \leq t < 60s \\ 1m, \ 60s \leq t < 80s \\ 0m, \ 80s \leq t \end{cases}$$

Figure 4.2 shows the responses of the quadrotor system controlled by the BC, when it can observe that the control strategy is very weak under the perturbations. Since BC is a controller that is developed from the accurate knowledge of the dynamic system, its performances remain feeble against the unknown nonlinearities. Therefore, the integration of an estimation law to deal with unmodelled dynamics is required.

4.4 Optimal model-free backstepping controller

In this section, we will investigate model-free control (MFC) for UAV quadrotor system. The MFC technique has several advantages which makes it favourable for many real application systems, such as in [5, 59]. The main feature of the



(A) Roll subsystem response.

(B) Pitch subsystem response.

(C) Yaw subsystem response.

(D) Position system response.

FIGURE 4.2: Quadrotor responses (case 2).

MFC law is that it depends only on the real-time measurement data of the controlled plant without the requirement of the accurate knowledge dynamics. However, most of them consider that the controlled system can be represented in finite time by a very-local model, where the stability of the overall closed-loop system is not absolutely proved.

4.4.1 Control design

In this work, the control target is to develop a robust nonlinear model-free backstepping controller (MFBC) for each i^{th} subsystem (4.7) considering that the total dynamic functions $\Psi_i(X)$ is unknown function along the control path. This objective can be achieved by using the classical backstepping control law (4.18) combined with adaptive nonlinear estimator of $\Psi_i(X)$. The drawback of controller parameters selection is addressed via including the cuckoo search algorithm (CS) which has been presented in the previous chapter. This final addition gives an optimal behaviour feature on the designed controller. The block diagram of the overall control system is shown in Figure 4.3.

Thus, the model-free counterpart of control law (4.18) that turns out to be that

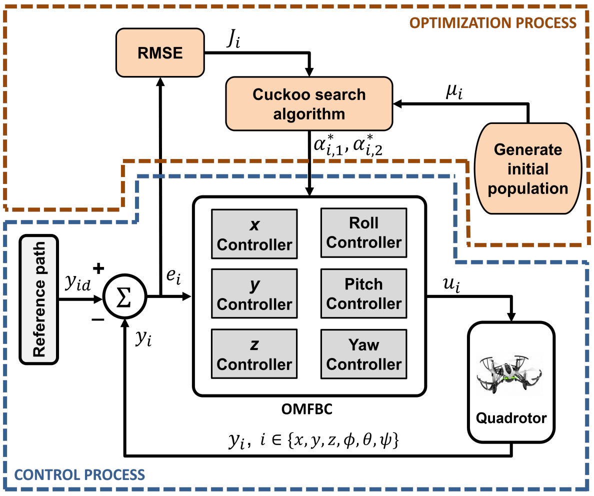


FIGURE 4.3: OMFBC structure.

of final MFBC is defined as follows [35]

$$u_i = \ddot{y}_{id} + (1 - \alpha^2_{i,1})e_{i,1} + (\alpha_{i,1} + \alpha_{i,2})e_{i,2} - \hat{\Psi}_i(X) \tag{4.20}$$

where $\hat{\Psi}_i(X)$ is the adaptive estimator of the total dynamic functions $\Psi_i(X)$. Unlike the classic MFC theory which uses an ultra-local model to define the control law [5], the estimation functions $\hat{\Psi}_i(X)$ in the proposed MFBC are constructed by two parts, the first part is given by the following dynamics

$$\dot{\hat{\Psi}}_i(X) = \gamma_{i,1}(x_{i,2} - \hat{\pi}_i) \tag{4.21}$$

where $\gamma_{i,1} \gg 0$ is the estimation parameter, its value is chosen larger than zero and $\hat{\pi}_i$ is the adaptive factor. Established on the above discussion, the following theorem is given to illustrate the MFBC performance of the overall closed-loop system.

Theorem 4.1 Considering the system given in (4.7) with satisfied **assumptions 1-3**, the proposed MFBC law (4.20) with the estimator law (4.21) guarantee the overall stability closed-loop system and the asymptotic convergence of the tracking error towards zero, i.e., $e \to 0$ for $t \to \infty$.

Proof: Let us consider the candidate Lyapunov function as

$$V_3 = V_2 + \sum_{i \in \{x,y,z,\phi,\theta,\psi\}} \frac{1}{2\gamma_{i,2}} \tilde{\Psi}_i^2 + \frac{1}{2} \tilde{\pi}_i^2 \tag{4.22}$$

where $\tilde{\Psi}_i = \Psi_i - \hat{\Psi}_i$ represents the estimation error and $\tilde{\pi}_i = x_{i,2} - \hat{\pi}_i$ is the adaptive factors respectively.

The time derivative of V_3 is

$$\begin{aligned} \dot{V}_3 = \dot{V}_2 + \textstyle\sum_{i \in \{x,y,z,\phi,\theta,\psi\}} \frac{1}{\gamma_{i,1}} \tilde{\Psi}_i \dot{\tilde{\Psi}}_i + \tilde{\pi}_i \dot{\tilde{\pi}}_i \\ = \textstyle\sum_{i \in \{x,y,z,\phi,\theta,\psi\}} e_{i,1}\dot{e}_{i,1} + e_{i,2}\dot{e}_{i,2} + \frac{1}{\gamma_{i,1}} \tilde{\Psi}_i (\dot{\Psi}_i - \dot{\hat{\Psi}}_i) + \tilde{\pi}_i (\dot{x}_{i,2} - \dot{\hat{\pi}}_i) \end{aligned} \tag{4.23}$$

Using (4.15), yields

$$\begin{aligned} \dot{V}_3 = \textstyle\sum_{i \in \{x,y,z,\phi,\theta,\psi\}} -\alpha_{i,1} e_{i,1}^2 + e_{i,2}(\Psi_i + u_i - \ddot{y}_{id} - e_{i,1} - \alpha_{i,1}\dot{e}_{i,1}) \\ -\frac{1}{\gamma_{i,1}} \tilde{\Psi}_i \dot{\hat{\Psi}}_i + \tilde{\pi}_i (\dot{x}_{i,2} - \dot{\hat{\pi}}_1) \end{aligned} \tag{4.24}$$

Considering (4.8) and (4.13), Eq. (4.24) becomes

$$\begin{aligned} \dot{V}_3 = \textstyle\sum_{i \in \{x,y,z,\phi,\theta,\psi\}} -\alpha_{i,1} e_{i,1}^2 + e_{i,2}(\Psi_i + u_i - \ddot{y}_{id} - (\alpha_{i,1} + \alpha_{i,2}) e_{i,2} \\ +(\alpha_{i,1}^2 - 1) e_{i,1}) - \frac{1}{\gamma_{i,1}} \tilde{\Psi}_i \dot{\hat{\Psi}}_i + \tilde{\pi}_i (\dot{x}_{i,2} - \dot{\hat{\pi}}_1) \end{aligned} \tag{4.25}$$

Substituting (4.20) in (4.25), yields

$$\dot{V}_3 = \sum_{i \in \{x,y,z,\phi,\theta,\psi\}} -\alpha_{i,1} {e_{i,1}}^2 - \alpha_{i,2} {e_{i,2}}^2 - \frac{1}{\gamma_{i,1}} \tilde{\Psi}_i \dot{\hat{\Psi}}_i + \tilde{\pi}_i (\dot{x}_{i,2} - \dot{\hat{\pi}}_1) \tag{4.26}$$

To ensure the performance of the estimation process, the adaptive factor law is defined as

$$\dot{\hat{\pi}}_i = \hat{\Psi}_i + u_i - \gamma_{i,2}(\hat{\pi}_i - x_{i,2}) \tag{4.27}$$

where $\gamma_{i,2}$ is positive constant. Substituting (4.21) and (4.27) into (4.26), one has

$$\dot{V}_3 = \sum_{i \in \{x,y,z,\phi,\theta,\psi\}} -\alpha_{i,1} {e_{i,1}}^2 - \alpha_{i,2} {e_{i,2}}^2 - \tilde{\Psi}_i \tilde{\pi}_i + \tilde{\pi}_i (\dot{\chi}_{i,2} - (\hat{\Psi}_i + u_i - \gamma_{i,2}(\hat{\pi}_i - x_{i,2}))) \tag{4.28}$$

Invoking (4.7), Eq. (4.28) becomes

$$\begin{aligned}\dot{V}_3 = \textstyle\sum_{i\in\{x,y,z,\phi,\theta,\psi\}} -\alpha_{i,1}{e_{i,1}}^2 - \alpha_{i,2}{e_{i,2}}^2 - \tilde{\Psi}_i\tilde{\pi}_i \\ +\tilde{\pi}_i(\Psi_i + u_i - (\hat{\Psi}_i + u_i - \gamma_{i,2}(\hat{\pi}_i - x_{i,2})))\end{aligned} \tag{4.29}$$

After some mathematical simplifications, it can find

$$\begin{aligned}\dot{V}_3 &= \textstyle\sum_{i\in\{x,y,z,\phi,\theta,\psi\}} -\alpha_{i,1}e_{i,1}^2 - \alpha_{i,2}e_{i,2}^2 - \tilde{\Psi}_i\tilde{\pi}_i + \tilde{\pi}_i(\tilde{\Psi}_i - \gamma_{i,2}\tilde{\pi}_i) \\ &= \textstyle\sum_{i\in\{x,y,z,\phi,\theta,\psi\}} -\alpha_{i,1}e_{i,1}^2 - \alpha_{i,2}e_{i,2}^2 - \gamma_{i,2}\tilde{\pi}_i^2 \leq 0\end{aligned} \tag{4.30}$$

Since $\alpha_{i,1}, \alpha_{i,2}$ and $\gamma_{i,2}$ are positive constants, it could get $\dot{V}_3$ is semi-negative definite function and $V_3 \in L_\infty$. This implies that the overall stability in the closed-loop system can be guaranteed. Thus, the theorem is proved. ■

Remark 4.2. The designed controller (4.20) is only related to the tracking errors $e_{i,1}$ and $e_{i,2}$ where no dynamic model information is needed. Therefore, the proposed control law is model-free without local identification, which makes it easy to be implemented in real-time.

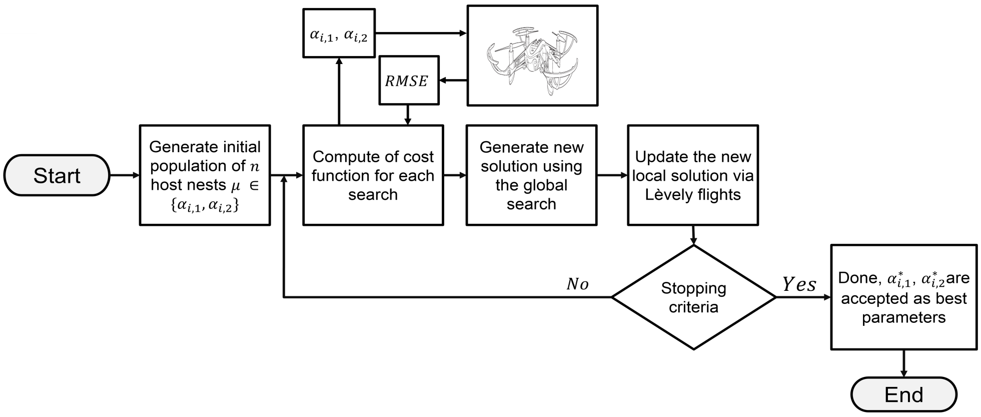

$\alpha_{i,1}$, $\alpha_{i,2}$
RMSE
Start
Generate initial population of n host nests $\mu \in \{\alpha_{i,1}, \alpha_{i,2}\}$
Compute of cost function for each search
Generate new solution using the global search
Update the new local solution via Lèvely flights
Stopping criteria
No
Yes
Done, $\alpha^*_{i,1}$, $\alpha^*_{i,2}$are accepted as best parameters
End

FIGURE 4.4: Flowchart of CS process.

4.4.2 Optimization process using CS

Usually, the numerical values of the designed control parameters are picked randomly by trial-and-error during the simulation process, when a compromise should be made between the dynamic response speed and steady-state accuracy. To overcome this limitation, the CS that presented in section 3.5 is included to get the optimal parameters for the proposed optimal model-free backstepping controller (OMFBC) against minimizing an objective function J_i for $i \in \{x, y, z, \phi, \theta, \psi\}$. The step by step procedure is given bellow. The same figure 4.4 illustrates the following steps of the optimization process.

Step 1. Set: Number of nests n, maximum number of iterations, the fraction of nests gain P_a, the step size s and the constant δ.

Step 2. The initial feasible solutions are generated according to Low-bound μ_L and Up-bound μ_U have the same size of the dimensional problem, these bounds are limited to satisfy the stability condition (4.30). The identification of the initial global best solution coming through minimizing the objective function value J_i such as the root mean squared error (RMSE) which is computed as

$$J_i = \sqrt{\frac{\sum_j^N (y_{id} - x_{i,1})^2}{N}} \tag{4.31}$$

where N is the simulation sampling time size.

Step 3. Use the switch probability P_a, the random ε, the step size s and the constant δ to generate the new global solution via Eqs (3.1,3.2).

Step 4. Evaluate the process to get a new local solution and select the best target through evaluate Eqs (3.3,3.4).

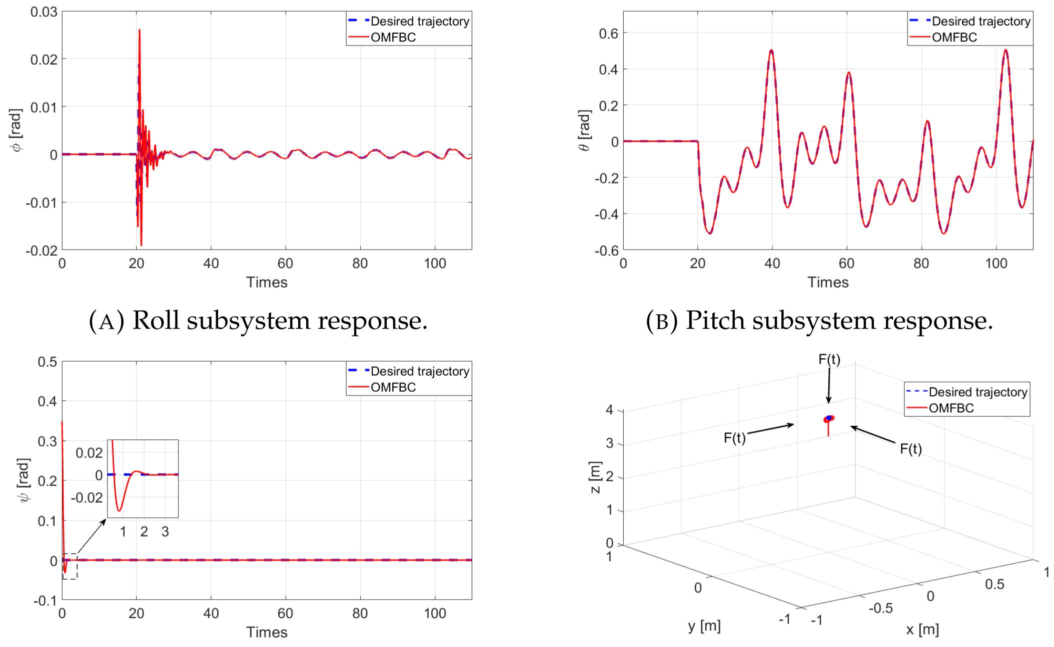
Step 5. keep the evaluation of the optimization process till the stopping criterion is satisfied.

4.4.3 Simulations

In this section, we use the same quadrotor parameters that have been used in the previous simulation tests. Firstly, the OMFBC is optimized using CS in a bounded search space $\mu_i \in [\mu_L, \mu_U] = [1, 10]$ and the parameters of CS

TABLE 4.2: CS Parameters

Parameters	**Symbol**	**Value**
Number of nests	n	**25**
Fraction of nests	P_a	**0.25**
Step size scaling factor	δ	**1.5**
Step size	s	**0.01**



(A) Roll subsystem response.

(B) Pitch subsystem response.

(C) Yaw subsystem response.

(D) Position system response.

FIGURE 4.5: Quadrotor responses (case 1).

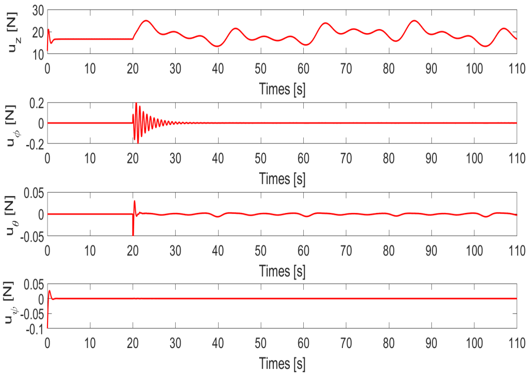
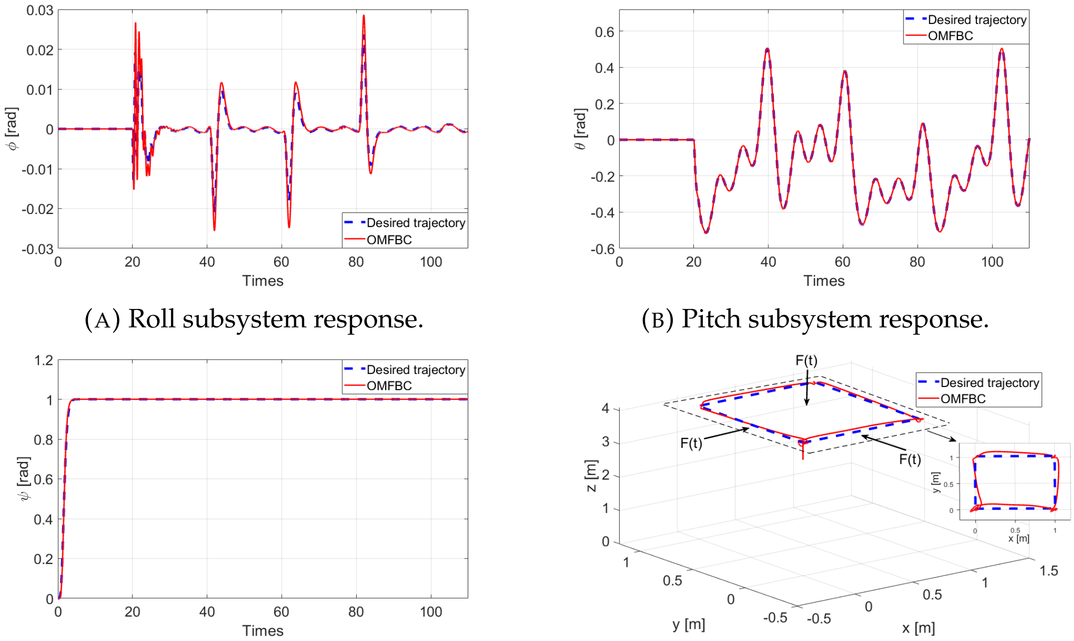


FIGURE 4.6: Control inputs of OMFBC (case 1).

are summarized in Table 4.2, where the optimal control parameters obtained from the CS optimization process are given in Table 4.3, where the RMSE for the position systems is chosen as RMSE-P=$\sum_{i\in\{x,y,z\}} J_i$, and the same control parameters obtained for the rotation's subsystem where the RMSE for the rotation systems is chosen as RMSE-R=$\sum_{i\in\{\phi,\theta,\phi\}} J_i$. Also, the same simulation cases have been validated to test the performances of the proposed controller.

Case 1. This case discusses the performances of OMFBC in stabilization mode under external perturbation. Compared with the results obtained using the BC, Figure 4.5 shows the effectiveness of the proposed controller against



(A) Roll subsystem response.

(B) Pitch subsystem response.

(C) Yaw subsystem response.

(D) Position system response.

FIGURE 4.7: Quadrotor responses (case 2).

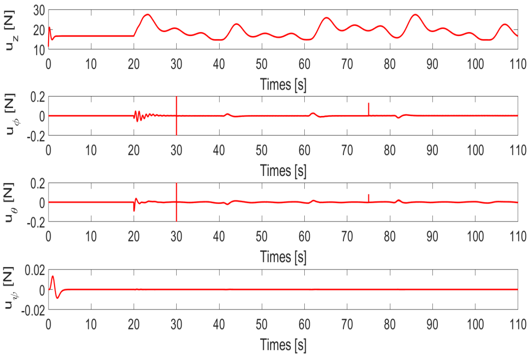


FIGURE 4.8: Control inputs of OMFBC (case 2).

the external disturbances, where it can see that the stabilization system can be achieved by applying the OMFBC. The input signals can be seen in Figure 4.6, where the figure demonstrates that the signals are well smooth and some variation at 20*sec* caused by the changing of the position system at the same time.

Case 2. This section proves the superiority of the proposed controller under the external disturbances in fly tracking mode. Figure 4.7 displays the quadrotor responses where it can observe that the OMFBC is well achieving the control target in presence of perturbation. Figure 4.8 presents the control

TABLE 4.3: Control gains.

Subsystem:	x, y	z	ϕ	ψ
Parameter:	$\alpha_1^* = 3.9551$	$\alpha_1^* = 3.9551$	$\alpha_1^* = 6.0706$	$\alpha_1^* = 6.0706$
	$\alpha_2^* = 5.9116$	$\alpha_2^* = 5.91165$	$\alpha_2^* = 6.9681$	$\alpha_2^* = 6.9681$
	$\gamma_1 = 900$	$\gamma_1 = 500$	$\gamma_1 = 300$	$\gamma_1 = 300$
	$\gamma_2 = 100$	$\gamma_2 = 100$	$\gamma_2 = 90$	$\gamma_2 = 90$

signal inputs provided by the controller, where it is clear that the signal is smooth as well which means that the control is more realizable in real-time. For more visibility, a comparison study error based performances index RMSE and maximum absolute error (MaxAE) are given in Table 4.4, as it's shown on this table that the OMBC gets the minimum index values, that can validate the superiority of the proposed controller than the BC.

It can conclude that the proposed OMFBC can deal with the external disturbances due to its development which does not require any prior knowledge of the dynamics and the included adaptive laws with the optimization strategies increase the robustness of the control law behaviour.

TABLE 4.4: RMSE and MaxAE of the proposed OMFBC and BC.

		RMSE		**MaxAE**	
Controller	**Subsystem**	**Case 1**	**Case 2**	**Case 1**	**Case 2**
BC	**Rotation [rad]**	9.1×10^{-3}	1.5×10^{-2}	0.349	0.349
	Position [m]	7.584×10^{-1}	1.027	0.59	0.592
OMFBC	**Rotation [rad]**	3.8×10^{-3}	4×10^{-3}	0.35	0.341
	Position [m]	6.4×10^{-2}	6.38×10^{-2}	0.45	0.46

4.5 Optimal fuzzy model-free backstepping controller

In this section, an optimal fuzzy model-free controller (OFMFC) for the attitude of a quadrotor is proposed in the presence of unknown nonlinear dynamics, parameter uncertainties and external disturbances. The integration of the intelligent systems as the fuzzy system (FS) is the subject of nowadays. The main FS feature is its ability to estimate any unknown nonlinear function. The selection of the designed control law parameters is addressed via including the flower pollination algorithm (FPA) described in the previous section. The block diagram of the control system is shown in Figure 4.9.

Before the design of the control law, it should recall the representation system given in (4.1). Since the input function $g_i(X)$ of the attitude quadrotor system has known constant values they can be set by the user, one can note that $g_i(X) = g_i$ for $i \in \{\phi, \theta, \psi\}$. Addition to the previous section when the dynamic system is considered with an unknown disturbance function $h_i(t)$, it also considers the parametric uncertainties which can occur on the dynamic model (4.3) to give the following form [34]

$$\dot{x}_{i,2} = f_i(X) + \Delta f_i + (g_i + \Delta g_i)u_i + h_i(t), i \in \{\phi, \theta, \psi\} \tag{4.32}$$

where Δf_i and Δg_i are the parametric variations. Let's put the total uncertainty and the external disturbances function $D_i = \Delta f_i + \Delta g_i u_i + h_i$ which assumes to satisfy the inequality $|D_i| \leq d_i$, where $d_i > 0$ is known constant. One can gets the presentation form presented in (4.32) in the following form

$$\dot{x}_{i,2} = f_i(X) + g_i u_i + D_i(t), i \in \{\phi, \theta, \psi\} \tag{4.33}$$

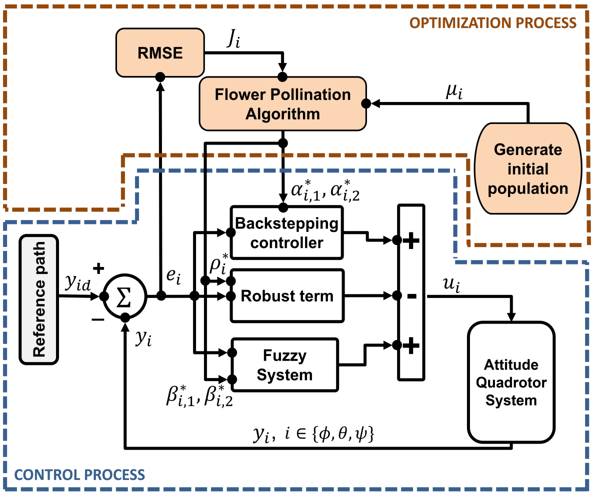

OPTIMIZATION PROCESS
RMSE
J_i
Flower Pollination Algorithm
μ_i
Generate initial population
$\alpha^*_{i,1}, \alpha^*_{i,2}$
Backstepping controller
ρ^*_i
Robust term
Fuzzy System
$\beta^*_{i,1}, \beta^*_{i,2}$
Reference path
y_{id}
+
−
Σ
e_i
y_i
+
-
+
u_i
Attitude Quadrotor System
$y_i,\ i \in \{\phi, \theta, \psi\}$
CONTROL PROCESS

FIGURE 4.9: OFMFC structure.

4.5.1 Control design

In this work, our objective is to design fuzzy control law based on FS to approximate the unknown nonlinear function $f_i(X)$ of each attitude subsystem along with parameters uncertainties and external disturbances.
First, let's assume that the unknown nonlinear function $f_i(X)$ in (4.32) can be approximated, over a compact set Ξ_X by fuzzy systems as follows [13]

$$f_i(X) = \xi_i^T(x_{i,1}, x_{i,2})\pi_i^* + \Delta_i \tag{4.34}$$

where $\xi_i^T = [\xi_1, ..., \xi_L]^T$ is the basic fuzzy function vector of L fuzzy rules "If-Then" whose its elements are computed from the selected membership functions $\mu_{F_{i,1}^l}$ and $\mu_{F_{i,2}^l}$ as

$$\xi_l = \frac{\prod_{i\in\{\phi,\theta,\psi\}(\mu_{F_{i,1}^l}\mu_{F_{i,2}^l})}}{\sum_{k=1}^L \prod_{i\in\{\phi,\theta,\psi\}(\mu_{F_{i,1}^l}\mu_{F_{i,2}^l})}}, l = 1, ..., L \tag{4.35}$$

where $F_{i,1}^k$ and $F_{i,2}^k$ are the fuzzy set of $x_{i,1}$ and $x_{i,2}$ respectively, π_i^* is the optimal parameter that is minimizing the approximation error Δ_i and satisfies $\pi_i^* = \text{argmin}[\sup_{X\in\Xi_X} |\hat{f}_i(X, \pi_i) - f_i(X, \pi_i)|]$. From the above analysis, we can write

$$f_i(X) - \hat{f}_i(X, \pi_i) = \xi_i^T(x_{i,1}, x_{i,2})\tilde{\pi}_i + \Delta_i \tag{4.36}$$

where $\hat{f}_i(X, \pi_i)$is the estimate of the nonlinear function $f_i(X)$ and $\tilde{\pi}_i = \pi_i^* - \hat{\pi}_i$ is the parameter estimation error.
Assumption 4.4. The approximation error Δ_i is considered small and bounded $|\Delta_i| < \overline{\Delta}_i, \forall X \in \Xi_X$, according to the theory of the universal approximation and $\overline{\Delta}_i$ is unknown positive constant.
Based on (4.36), we propose the following control law

$$u_i = u_{if} + u_{ir}, i \in \{\phi, \theta, \psi\} \tag{4.37}$$

The control law is composed of two terms. The first term is a fuzzy control term u_{if}, introduced to deal with the unknown nonlinear system, its expression inspired form (4.18) as follows [34]

$$u_{if} = g_i^{-1}(\ddot{y}_{id} - \hat{f}_i + (1 - \alpha_{i,1}^2)e_{i,1} + (\alpha_{i,1} + \alpha_{i,2})e_{i,2}) \tag{4.38}$$

The second term is a robust control term introduced to deal with disturbances and approximation errors, which has the following form

$$u_{ir} = -g_i^{-1}(\rho_i e_{i,2} + \hat{\Delta}_i \text{sgn}(e_{i,2})) \tag{4.39}$$

where $\rho_i > 0$ is a designed parameter and $\hat{\Delta}_i$ represents the adaptive gain to estimate the value of $\overline{\Delta}_i$. In order to reach the target control, the following adaptive laws are given

$$\begin{aligned} \dot{\hat{\pi}}_i &= \beta_{i,1} e_{i,2} \xi_i - \beta_{i,1}\beta_{i,2}\hat{\pi}_i \\ \hat{\Delta}_i &= \delta_i \int_0^t |e_{i,2}| dt \end{aligned} \tag{4.40}$$

where $\beta_{i,1}$, $\beta_{i,2}$ and δ_i are positive constant. Established on the above discussion, the following theorem concludes the stability of the closed-loop system.

Theorem 4.2. Considering the attitude system (4.32). Suppose that **assumption 4.4** is satisfied. The control law defined in (4.37-4.39), with adaptive laws given in (4.40) guarantees that the closed-loop system is stable and all signals are bounded.

Proof: Let's consider the candidate Lyapunov function

$$V_4 = V_2 + \sum_{i \in \{\phi,\theta,\psi\}} \frac{1}{2\beta_{i,1}} \tilde{\pi}_i^T \tilde{\pi}_i + \frac{1}{2\lambda_i} \tilde{\Delta}_i^2 \tag{4.41}$$

where $\tilde{\Delta}_i = \overline{\Delta}_i - \hat{\Delta}_i$. The time derivative of (4.41) is

$$\dot{V}_4 = \dot{V}_2 \sum_{i \in \{\phi,\theta,\psi\}} -\frac{1}{\beta_{i,1}} \tilde{\pi}_i^T \dot{\hat{\pi}}_i - \frac{1}{\lambda_i} \tilde{\Delta}_i \dot{\hat{\Delta}}_i \tag{4.42}$$

Using the derivative Lyapunov function obtained in (4.16) and invoking (4.16), Eq(4.42) becomes

$$\begin{aligned}\dot{V}_4 = \textstyle\sum_{i\in\{\phi,\theta,\psi\}} -e_{i,1}e_{i,2} - \alpha_{i,1}e_{i,1}^2 + e_{i,2}(f_i(X) + g_i u_i + D_i - \ddot{y}_{id} \\ +\alpha_{i,1}(e_{i,2} + \alpha_{i,1}e_{i,1})) - \tfrac{1}{\beta_{i,1}}\tilde{\pi}_i^T\dot{\hat{\pi}}_i - \tfrac{1}{\lambda_i}\tilde{\Delta}_i\dot{\hat{\Delta}}_i\end{aligned} \tag{4.43}$$

Substituting (4.38-4.40) into (4.43), it follows that

$$\begin{aligned}\dot{V}_4 = \textstyle\sum_{i\in\{\phi,\theta,\psi\}} -\alpha_{i,1}e_{i,1}^2 - \alpha_{i,2}e_{i,2}^2 + e_{i,2}(\xi_i^T\tilde{\pi}_i + \Delta_i + D_i - \rho_i e_{i,2} - \hat{\Delta}_i \mathrm{sgn}(e_{i,2})) \\ -\tfrac{1}{\beta_{i,1}}\tilde{\pi}_i^T\dot{\hat{\pi}}_i - \tfrac{1}{\lambda_i}\tilde{\Delta}_i\dot{\hat{\Delta}}_i\end{aligned} \tag{4.44}$$

By substituting (4.40) into (4.44), one has

$$\begin{aligned}\dot{V}_4 = \textstyle\sum_{i\in\{\phi,\theta,\psi\}} -\alpha_{i,1}e_{i,1}^2 - \alpha_{i,2}e_{i,2}^2 - \rho_i e_{i,2}^2 \\ -\beta_{i,2}\tilde{\pi}_i^T\hat{\pi}_i - (\tfrac{1}{\lambda_i}\tilde{\Delta}_i\dot{\hat{\Delta}}_i + \hat{\Delta}_i \mathrm{sgn}(e_{i,2})) + e_{i,2}(\Delta_i + D_i)\end{aligned} \tag{4.45}$$

with some simplification, we get

$$\begin{aligned}\dot{V}_4 = \textstyle\sum_{i\in\{\phi,\theta,\psi\}} -\alpha_{i,1}e_{i,1}^2 - \alpha_{i,2}e_{i,2}^2 - \rho_i e_{i,2}^2 \\ -\beta_{i,2}\tilde{\pi}_i^T\hat{\pi}_i - \overline{\Delta}_i|e_{i,2}| + e_{i,2}(\Delta_i + D_i)\end{aligned} \tag{4.46}$$

Using the Young's inequality, yields:

$$\begin{aligned}\beta_{i,2}\tilde{\pi}_i^T\hat{\pi}_i = \beta_{i,2}\tilde{\pi}_i^T(\pi_i^* - \tilde{\pi}_i) \leq \tfrac{\beta_{i,2}}{2}\|\pi_i^*\|^2 - \tfrac{\beta_{i,2}}{2}\|\tilde{\pi}_i\|^2 \\ \overline{\Delta}_i|e_{i,2}| = (\tilde{\Delta}_i + \hat{\Delta}_i)|e_{i,2}| \leq \tfrac{e_{i,2}^2}{2} + \tfrac{\tilde{\Delta}_i^2}{2} \\ e_{i,2}(\Delta_i + D_i) \leq \tfrac{e_{i,2}^2}{2} + \tfrac{\overline{\Delta}_i^2}{2} + \tfrac{\overline{d}_i^2}{2}\end{aligned} \tag{4.47}$$

By using **assumption 4.4** and the above inequality, (4.46) can be written as

$$\dot{V}_4 \leq \sum_{i\in\{\phi,\theta,\psi\}} -\alpha_{i,1}e_{i,1}^2 - (\alpha_{i,2} + \rho_i)e_{i,2}^2 - \frac{\beta_{i,2}}{2}\tilde{\pi}_i^T\tilde{\pi}_i - \frac{\tilde{\Delta}_i^2}{2} + \frac{\overline{d}_i^2}{2} \tag{4.48}$$

let's be $\varrho = \min\{2\alpha_{i,1}, 2(\alpha_{i,2} + \rho_i), \beta_{i,2}/\beta_{i,1}, \delta_i\}$ and $\vartheta = \overline{d}_i^2/2$, the result of (4.48) becomes

$$\dot{V}_4 \leq \sum_{i\in\{\phi,\theta,\psi\}} -\varrho V_4 + \vartheta \tag{4.49}$$

Integrating (4.48), we can get

$$V_4(t) \leq \sum_{i\in\{\phi,\theta,\psi\}} V_4(0)e^{-\sigma t} + \frac{\vartheta}{\lambda}, \forall t \tag{4.50}$$

Because ϑ/λ is bounded, and $V_4(0)e^{-\sigma t}$ is non-increasing and bounded, the following result is obtained:

$$\lim_{t\to\infty} V_4(0)e^{-\sigma t} + \frac{\vartheta}{\lambda} < \infty \tag{4.51}$$

In addition, since $\dot{V}_4$ is bounded, using Barbalat's Lemma [83], it can be shown that $\lim_{t\to\infty} V_4(t) = 0$. As a result, the stability of the proposed control system can be guaranteed and makes $\lim_{t\to\infty} e_{i,1}, e_{i,2} = 0$. Thus, the theorem is proved. ■

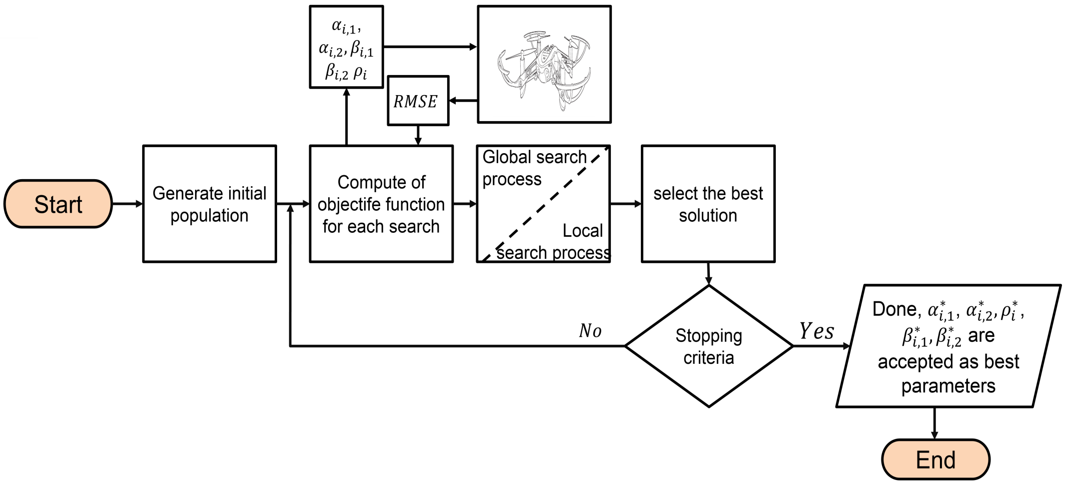
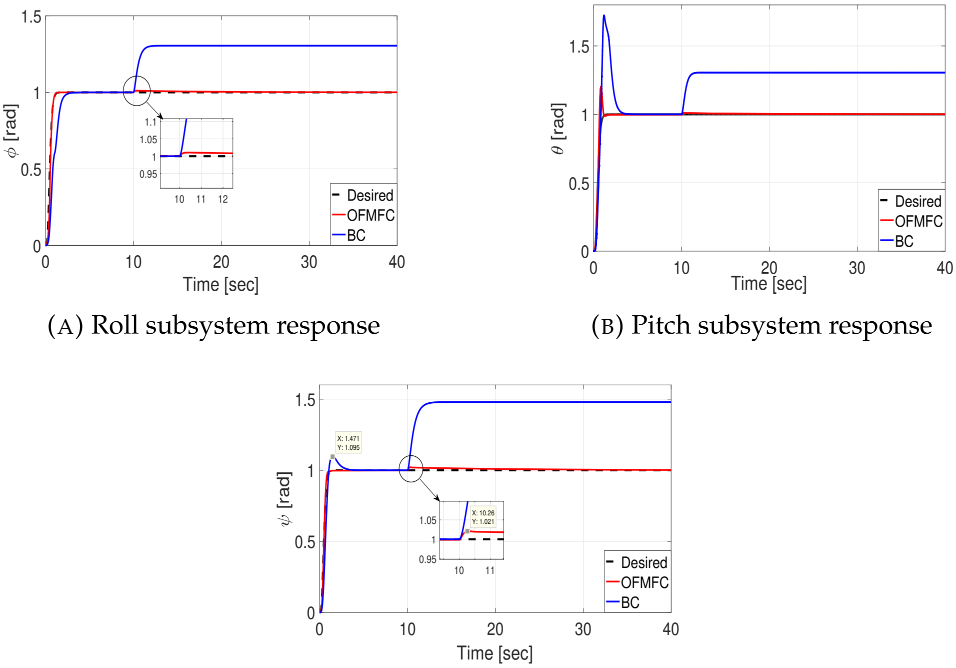

$\alpha_{i,1}$, $\alpha_{i,2}, \beta_{i,1}$ $\beta_{i,2}$ ρ_i
$RMSE$
Start
Generate initial population
Compute of objectife function for each search
Global search process
Local search process
select the best solution
No
Stopping criteria
Yes
Done, $\alpha^*_{i,1}$, $\alpha^*_{i,2}, \rho^*_i$, $\beta^*_{i,1}, \beta^*_{i,2}$ are accepted as best parameters
End

FIGURE 4.10: Flowchart of FPA process.

4.5.2 Optimization process using FPA

In the previous section, the OFMFC designed which doesn't need any prior knowledge about the dynamic system and depends only on local measurements. In another hand, a satisfactory degree of the controller performance depends on the best selection of the parameters. Usually, the parameters of



(A) Roll subsystem response

(B) Pitch subsystem response

(C) Yaw subsystem response

FIGURE 4.11: Quadrotor attitude responses (case 1).

the control law are selected by a trial-and-error. Even if the parameters are properly chosen, it is not guaranteed that optimal parameters are selected. For this purpose, the FPA is integrated with the designed control law to tune the controller parameters against minimizing an objective function. This addition makes the control law reaches the optimal feature. The step by step optimization process is described in the following items and illustrated in the flowchart of figure 4.10.

Step 1. Initialize the flowers population size Np, the maximum number of iterations, the switching probability $\wp$, the step size λ, and the constant γ.

Step 2. Generate the initial possible solutions according to the lower-bound S_L and the upper-bound S_U, having the same size as the dimensional problem. Compute the RMSE (4.31) of each initial solution, and identify the initial global best solution.

Remark 4.3. The selection of the upper\lower bound is an arbitrary chosen from the user that has to respect and satisfy the stability condition (4.49).

Step 3. Use the switching probability $\wp$; if $rand < \wp$, keep the global pollination solution extracted via (3.13); else, perform a local pollination (3.15).

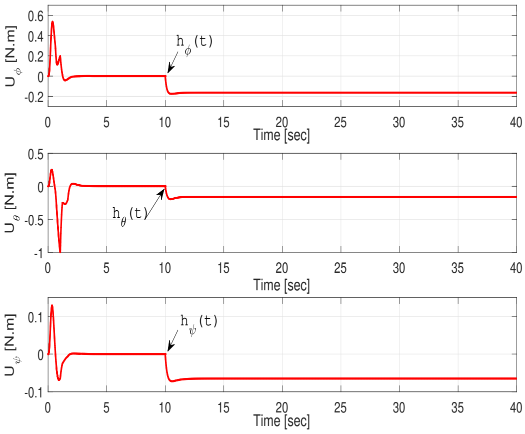


FIGURE 4.12: Control inputs of OFMFC (case 1).

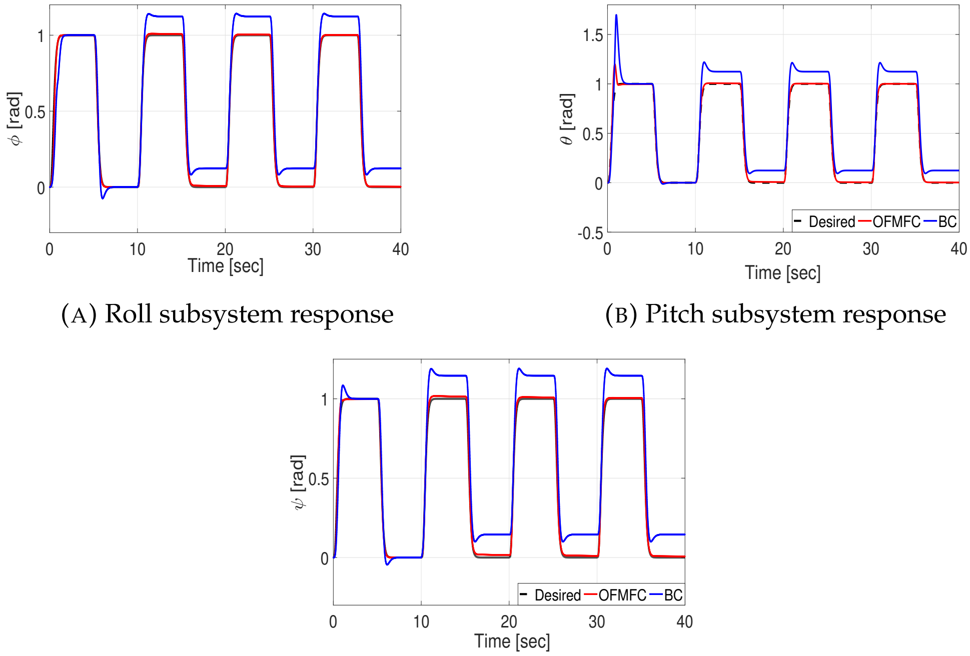
Step 4. Finding new solutions for all flowers. Replace old solutions with the new best solutions then repeat the steps and bend the current best solution until the stopping criterion is satisfied.

4.5.3 Simulations

To test the performance of the designed attitude controller, the OFMFC is validated using Matlab/Simulink. Firstly, the designed controller is optimized using FPA where the parameters of this optimization algorithm are: switch probability $\wp = 0.08$, the step size $\lambda = 0.1$ and $\gamma = 1.5$. The population size $Np = 10$, and the number of iterations are set to 100. Two simulation cases are validated and compared the optimal proposed controller which contains the optimal parameters given in Table 4.5 and the classical BC. The first case tests the proposed controller under the external disturbances, while the second case considers the uncertainties model. The initial condition for rotation system is taken as $\Theta_0 = [0, 0, 0]^T rad$ for both cases.

TABLE 4.5: Control gains.

Subsystem:	ϕ	θ	ψ
Parameter:	$\alpha_1^* = 52.439$	$\alpha_1^* = 46.281$	$\alpha_1^* = 46.281$
	$\alpha_2^* = 9.884$	$\alpha_2^* = 11.653$	$\alpha_2^* = 8.946$
	$\beta_1 = 10.118$	$\beta_1 = 10.627$	$\beta_1 = 7.150$
	$\beta_2 = 10.648$	$\beta_2 = 11.253$	$\beta_2 = 8.791$
	$\rho = 0.682$	$\rho = 0.390$	



(A) Roll subsystem response

(B) Pitch subsystem response

(C) Yaw subsystem response

FIGURE 4.13: Quadrotor attitude responses (case 2).

Case 1. In this case the attitude system is regarded as stable in certain angles as $y_{\phi d} = 1\ rad$, $y_{\theta d} = 1\ rad$, $y_{\psi d} = 1\ rad$. The simulation results of this case are depicted in figures 4.11,4.12. It can be seen from figure 4.11 that the responses of the OFMFC track the desired path with negligible error, while the responses using BC are leaving the desired path under the external disturbance (at $t > 10\ sec$). The attitude control inputs are depicted in Figure 4.12, where it is obviously that the control signals are smooth and they reach the desired control at the moment when the disturbance is included.

Case 2. In order to validate the controller in tracking motion, a square wave reference signals are considered for attitude angles and a total uncertainty and the external disturbances function D_i are included at $t > 10\ sec$ where the variation inertia matrix is considered as $\Delta I = 50\% I$ in this moment.The simulation results of this case are shown in figures 4.13,4.14. Figure 4.13 presents the attitude system responses, where it can observe the superiority of the proposed controller than the BC in presence of the disturbance. Figure 4.14 depicts the control signals behaviours under the perturbation, the figure shows that the signals are totally smooth and bounded that make these signal realizable in

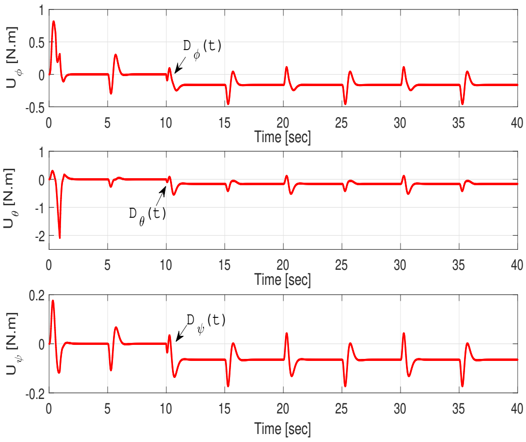
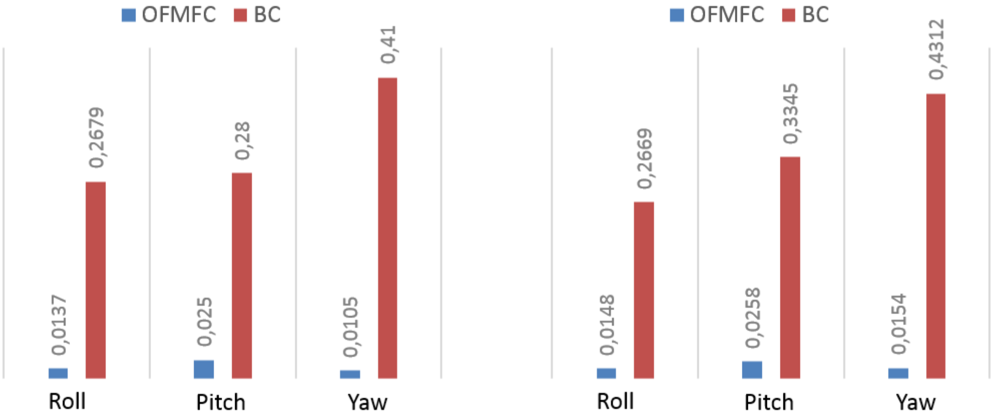


FIGURE 4.14: Control inputs of OFMFC (case 2).

real-time, and also can see that they reach the desired control in the case of the disturbances. To investigate the performance of the designed controllers more quantitatively, the RMSE values of the tracking errors were also calculated and depicted in Figure 4.15. Thus, these tests confirm clearly the performance and robustness property of the proposed OFMFC with respect to the BC.

It can conclude that the designed controller in this section is successfully used



(A)

(B)

FIGURE 4.15: RMSE values of the quadrotor attitude, (A) RMSE values for case 1, (B) RMSE values for case 2

on the attitude control and absolutely can deal with the unmodeled dynamics due to its independent structure from the model and the included robust term.

4.6 Summary

In this chapter, different model-free techniques based on the backstepping controller are introduced to compensate the unmodeled quadrotor dynamics and uncertainties. The stability analysis is analytically proved via Lyapunov theory and then verified by various simulation tests. The obtained results confirm the robustness of the designed approaches in the presence of disturbances, where the classical backstepping controller failed to compensate and adapt to the injected disturbance. Indeed, also the optimization algorithms that are included have a big addition in a way that we can obtain optimal control behaviour. Finally, it can say that the proposed control strategies show superior performance among the BC tested in terms of trajectory tracking, stability and robustness under the disturbances. The simplicity of the control law structure is one of the important items in the real-time area, where the control law have to be designed in a way that can be easily implemented in real-time. In another hand, the robust model-free controller is the control that is absolutely independent from the model and directly implemented. The next chapter will highlight to a type of model-free control that have a direct and simple structure.

Chapter 5

Intelligent optimal model-free control

5.1 Introduction

Intelligent techniques like fuzzy systems have been widely used in many control applications due to their flexibility to adapt with large complex systems and the design of a fuzzy model-free strategy is one of the main targets which our work cares out.

In this chapter, an optimal fuzzy model free for a quadrotor system using the bat algorithm is presented. The concept of this method is to decouple the whole system into six single-input single-output (SISO) subsystems. Then, a fuzzy adaptive controller and a compensated control term are employed using tracking error for each subsystem. After, the bat algorithm is included to define the local optimal gains of the proposed controller. The control law scheme completely overcomes the singularity problem that occurs in indirect adaptive control. The stability of the quadrotor control system, which is based on the Lyapunov theory, is assured. The obtained results prove the robustness of the proposed techniques even with external disturbances.

5.2 Nonlinear system representation

As we have mentioned in the previous chapter (Section 4.3) that the nonlinear model can be represented in several mathematical representations that define specific classes. Consider the overall dynamic system (2.17), it can represent theses dynamics by i^{th} subsystems described by a generalized state equation as

$$\begin{cases} \dot{\chi}_i(t) = v_i(t) \\ \dot{v}_i(t) = A_i(t) + B_i(t)u_i + d_i(t) \\ y_i(t) = \chi_i(t) \end{cases} \tag{5.1}$$

where $\chi_i(t) \in \{x(t), y(t), z(t), \phi(t), \theta(t), \psi(t)\}$ is the state vector and $v_i(t)$ its derivative which are available for measurement, $A_i(t) \in \Re$ is the nonlinear dynamic function with $i \in \{x, y, z, \phi, \theta, \psi\}$, $B_i(t)$ is the input control function of the i^{th} subsystem and $d_i(t) \neq 0$ is the external disturbance that the quadrotor can be subjected to, $u_i(t)$ is the control inputs. With,

$$\begin{aligned} &A_x(t) = 0, A_y(t) = 0, A_z(t) = -g, \\ &A_\phi(t) = \left(I_y - I_z\right) v_\theta v_\psi / I_x - J_r \Omega_r v_\theta / I_x, \\ &A_\theta(t) = (I_z - I_x) v_\phi v_\psi / I_y + J_r \Omega_r v_\phi / I_y, \\ &A_\psi(t) = (I_x - I_y) v_\phi v_\theta / I_z, \\ &B_z(t) = (c\chi_\phi(t) s\chi_\theta(t) c\chi_\psi(t) + s\chi_\phi(t) s\chi_\psi(t))/m, \\ &B_y(t) = (c\chi_\phi(t) s\chi_\theta(t) c\chi_\psi(t) - c\chi_\phi(t) c\chi_\psi(t))/m, \\ &B_z(t) = (c\chi_\phi(t) c\chi_\theta(t))/m, \\ &B_\phi(t) = l/I_x,\ B_\theta(t) = l/I_y,\ B_\psi(t) = 1/I_z. \end{aligned} \tag{5.2}$$

Assumption 5.1. The external disturbance function $d_i(t)$ is assumed unknown and bounded for a given constant $\overline{d}_i$ such that $|d_i(t)| \leq \overline{d}_i$ for $i \in \{x, y, z, \phi, \theta, \psi\}$.

Assumption 5.2. The input control function $B_i(t)$ assuming difference to zero for all $t > 0$ and supposed to be as $B_i(t) = B_{0i} + \Delta B_i(t)$, where $B_{0i} > 0$ is the nominal value of $B_i(t)$ that can be known or unknown, and $\Delta B_i(t)$ is an unknown part which presents the modeling errors or parametric variations.

Remark 5.1. Since $|\phi| < \pi/2$ and $|\theta| < \pi/2$, it can prove that the input control function $B_z(t) > 0$ for all $t > 0$. Also that, I is a symmetric positive definite inertia matrix with respect to the body frame and $|I_{x,y,z} \pm \Delta I_{x,y,z}| > 0$, it can prove that the input control function $B_\phi(t) > 0$, $B_\theta(t) > 0$ and $B_\psi(t) > 0 \; \forall t > 0$.

Let's define the local errors as

$$\begin{aligned} e_{\chi i} &= \chi_{i,d} - \chi_i \\ e_{vi} &= v_{i,d} - v_i \end{aligned} \tag{5.3}$$

where $\chi_{i,d}$ and $v_{i,d}$ are the desired and the velocity references respectively wich assumed bounded and smooth. Using representation (5.1) and Eqs(5.3), it can define the tracking error equation as

$$\dot{e}_i = \Lambda_i e_i + \Gamma_i(\ddot{\chi}_{i,d} - A_i(t) - B_i(t)u_i - d_i(t)) \tag{5.4}$$

with,

$$e_i = [e_{\chi i}, e_{vi}]^T, \Lambda_i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \Gamma_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.5}$$

In order to get an ideal control law, we have to assume that all dynamic functions are known and the external disturbance $d_i(t) = 0$, and using feedback linearization technique one gets

$$u_i^* = B_i(t)^{-1}(-A_i(t) + \ddot{\chi}_{i,d} + K_i^T e_i) \tag{5.6}$$

where $K_i = [k_{\chi i}, k_{vi}]$ is a gain vector defined by the designer to have a stable closed-loop system. Substituting (5.6) into (5.5), yields

$$\dot{e}_i = (\Lambda_i - \Gamma_i K_i^T) e_i \tag{5.7}$$

According to the above analysis, the ideal control law (5.6) is easily obtained if the nonlinear functions $A_i(t)$ and $B_i(t)$ are known and the quadrotor system is not subject to any parametric uncertainty or external disturbance $d_i(t) = 0$. However, the quadrotor system is flies in an uncertain environment where it can subject to any unknown disturbances like winds or unmodeled dynamics. In this case, the above designed method cannot be implemented. Hence, the design of intelligent adaptive control law is required to estimate the control law $u_i^*(t)$ direcly and can deal with the unknown nonlinearities.

5.3 Fuzzy model-free control design

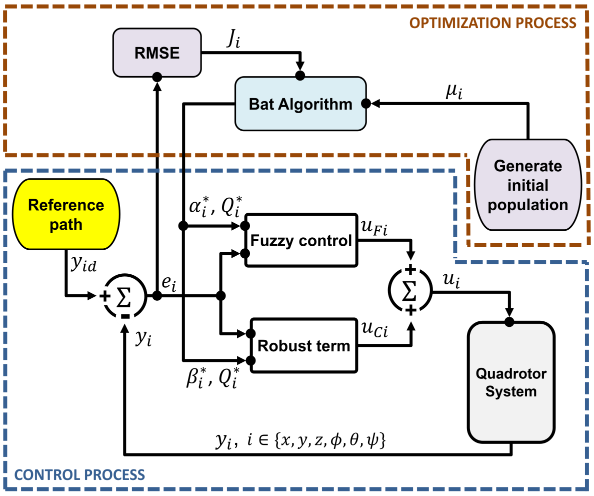


FIGURE 5.1: OFMFC structure.

As we mentioned in the previous chapter that the fuzzy system can estimate any continuous function by N fuzzy rules "IF-THEN". Using this feature, it can design a direct fuzzy model-free controller to estimate directly the ideal control law (5.6) along with parameters uncertainties and external disturbance. Hence, the ideal control law (5.6) can be estimated by fuzzy system as follows [34]

$$u_i^* = \xi_i^T(e_i)\pi_i^*(t) + \Delta_i(t) \tag{5.8}$$

where $\xi_i^T(e_i) = [\xi_1(e_i), ..., \xi_N(e_i)]^T$ is the basic fuzzy function vector of N fuzzy rules, its elements are calculated from the selected membership functions $\mu_{\breve{o}^n_{\chi i}}(e_{\chi i})$ and $\mu_{\breve{o}^n_{vi}}(e_{vi})$ as

$$\xi_{i,n}(e_i) = \frac{\mu_{\breve{o}^n_{\chi i}}(e_{\chi i})\mu_{\breve{o}^n_{vi}}(e_{vi})}{\sum_{k=1}^N \mu_{\breve{o}^n_{\chi i}}(e_{\chi i})\mu_{\breve{o}^n_{vi}}(e_{vi})} \tag{5.9}$$

where $\breve{o}^n_{\chi i} \in \breve{o}^1_{\chi i}, ..., \breve{o}^{m_{\chi i}}_{\chi i}$ and $\breve{o}^n_{vi} \in \breve{o}^1_{vi}, ..., \breve{o}^{m_{vi}}_{vi}$ are the fuzzy sets of $e_{\chi i}$ and e_{vi} respectively, $\pi_i^* = [\pi_{1,i}^*, ..., \pi_{N,i}^*]^T$ is the optimal parameter minimizing the estimation error $\Delta_i(t)$.

Assumption 5.3. The approximation error $\Delta_i(t)$ for $i \in \{x, y, z, \phi, \theta, \psi\}$ is considered small and bounded $\forall\ t > 0$ as $|\Delta_i(t)| < \overline{\Delta}_i$, according to the theory of the universal approximation, where $\overline{\Delta}_i$ is unknown positive constant.

Considering the subsystems (5.1), the desired unknown ideal control law (5.6) is considered to stabilize the i^{th} subsystems theoretically. Then, an adaptive control law based on FLC is used instead to approximate this unknown control signal as

$$u_i = u_{Fi} + u_{Ci}u_{Fi} \tag{5.10}$$

The control law is composed of two terms. The first term is a fuzzy control term that is introduced to estimate the control law as follow

$$u_{Fi} = \xi_i^T(e_i)\widehat{\pi}_i(t) \tag{5.11}$$

where $\widehat{\pi}_i$ represents the estimation of π_i^*. The second term is a robust control term u_{Ci} that must be designed to deal with the total uncertainties and estimation error and it has the following form

$$u_{Ci} = \hat{\omega}_i \text{sgn}(e_i^T P_i \Gamma_i) \tag{5.12}$$

where $\operatorname{sgn}(.)$ is the sign function, $\hat{\omega}_i$ is the estimate of $\overline{\omega}_i$. Considering **assumption 5.2**, substituting (5.6) into (5.4), the tracking error Eq. (5.7) becomes

$$\begin{aligned} \dot{e}_i &= \Lambda_i e_i + \Gamma_i [B_i(t)(u_i^* - u_i) - K_i^T E_i - d_i(t)] \\ &= (\Lambda_i - \Gamma_i K_i^T) e_i + \Gamma_i [B_{0i}(u_i^* - u_i) + \Delta B_i(t)(u_i^* - u_i) - d_i(t)] \end{aligned} \tag{5.13}$$

Using equations (5.10), (5.11) and (5.13), one gets

$$\dot{e}_i = (\Lambda_i - \Gamma_i K_i^T) e_i + \Gamma_i [B_{0i} \xi_i^T(e_i) \tilde{\pi}_i - B_{0i} u_{Ci} + \omega_i] \tag{5.14}$$

where $\tilde{\pi}_i = \pi_i^* - \widehat{\pi}_i$ is the parameter estimation error and $\omega_i = B_{0i}\Delta_i(t) + \Delta B_i(t)(u_i^* - u_i) - d_i(t)$ is considered as an overall approximation error that satisfies $|\omega_i| \le \overline{\omega}_i$ with $\overline{\omega}_i$ is an unknown positive constant. In order to reach the target control, the following adaptive laws are given

$$\begin{aligned} \dot{\hat{\pi}}_i &= \alpha_i e_i^T P_i \Gamma_i \xi_i(e_i) \\ \dot{\hat{\omega}}_i &= \beta_i |e_i^T P_i \Gamma_i| \end{aligned} \tag{5.15}$$

where α_i and β_i are positive designing constants, and $P_i = P_i^T$ being a positive definite symmetric matrix solution of the following Lyapunov matrix equation

$$P_i(\Lambda_i - \Gamma_i K_i^T) + (\Lambda_i - \Gamma_i K_i^T)^T P_i = -Q_i \tag{5.16}$$

where $Q_i = Q_i^T$ is a diagonal positive definite matrix which will be determined by the optimization process in the next section. Established on the above discussion, the following theorem concludes the stability of the closed-loop system.

Theorem. Consider dynamic system (5.1). Suppose that assumptions 1-3 are satisfied. The control law defined in (5.10) with the proposed termes (5.11,5.12) guarantees that all signals in the closed-loop are bounded and the tracking error asymptotically converges to zero (i.e.,$e(t) \to 0$ for $t \to \infty$).

Proof: Let's consider the following candidate Lyapunov function as

$$V(t) = \sum_{i \in \{x,y,z,\phi,\theta,\psi\}} V_i(t) \tag{5.17}$$

with,

$$V_i(t) = \frac{1}{2}e_i^T P_i e_i + \frac{B_{0i}}{2\alpha_i}\tilde{\pi}_i^T\tilde{\pi}_i + \frac{B_{0i}}{2\beta_i}\tilde{\omega}_i^2 \tag{5.18}$$

where $\tilde{\omega}_i = \omega_i - \hat{\omega}_i$. The time derivative of (5.18) is given as

$$\dot{V}_i(t) = \frac{1}{2}\dot{e}_i^T P_i e_i + \frac{1}{2}e_i^T P_i \dot{e}_i - \frac{B_{0i}}{\alpha_i}\tilde{\pi}_i^T\dot{\hat{\pi}}_i - \frac{B_{0i}}{\beta_i}\tilde{\omega}_i\dot{\hat{\omega}}_i \tag{5.19}$$

Substitution (5.14) into (5.19) one has

$$\begin{gathered}\dot{V}_i(t) = \tfrac{1}{2}e_i^T(P_i(\Lambda_i - \Gamma_i K_i^T) + (\Lambda_i - \Gamma_i K_i^T)^T P_i)e_i \\ +e_i^T P_i\Gamma_i[B_{0i}\xi_i^T(e_i)\tilde{\pi}_i - B_{0i}u_{Ci} + \omega_i] - \tfrac{B_{0i}}{\alpha_i}\tilde{\pi}_i^T\dot{\hat{\pi}}_i - \tfrac{B_{0i}}{\beta_i}\tilde{\omega}_i\dot{\hat{\omega}}_i\end{gathered} \tag{5.20}$$

Using definition (5.16), yields

$$\begin{gathered}\dot{V}_i(t) \leq -\tfrac{1}{2}e_i^T Q_i e_i + \tilde{\pi}_i^T B_{0i}(e_i^T P_i\Gamma_i\xi_i^T(e_i) - \tfrac{1}{\alpha_i}\dot{\hat{\pi}}_i) \\ -e_i^T P_i\Gamma_i B_{0i}u_{Ci} + e_i^T P_i\Gamma_i\omega_i - \tfrac{B_{0i}}{\beta_i}\tilde{\omega}_i\dot{\hat{\omega}}_i\end{gathered} \tag{5.21}$$

Substitute (5.15) in the previous inequality, it will become

$$\begin{gathered}\dot{V}_i(t) \leq -\tfrac{1}{2}e_i^T Q_i e_i - e_i^T P_i\Gamma_i B_{0i}u_{Ci} + e_i^T P_i\Gamma_i\omega_i \\ -\tfrac{B_{0i}}{\beta_i}\tilde{\omega}_i\dot{\hat{\omega}}_i\end{gathered} \tag{5.22}$$

Then, by replacing the control law defined in (5.12) and using Schwartz inequality, the latter results become

$$\begin{gathered}\dot{V}_i(t) \leq -\tfrac{1}{2}e_i^T Q_i e_i - \mid e_i^T P_i\Gamma_i \mid B_{0i}\hat{\omega}_i + \mid e_i^T P_i\Gamma_i\omega_i \mid \overline{\omega}_i - \tfrac{B_{0i}}{\beta_i}\tilde{\omega}_i\dot{\hat{\omega}}_i \\ \leq -\tfrac{1}{2}e_i^T Q_i e_i - \mid e_i^T P_i\Gamma_i \mid B_{0i}\hat{\omega}_i - \tfrac{B_{0i}}{\beta_i}\tilde{\omega}_i\dot{\hat{\omega}}_i\end{gathered} \tag{5.23}$$

Finally, by applying the adaptive law $\hat{\omega}_i$ defined in (5.15), we find

$$\dot{V}_i(t) \leq -\frac{1}{2}e_i^T Q_i e_i \leq 0 \tag{5.24}$$

Since $\dot{V}_i(t) \leq -\frac{1}{2}e_i^T Q_i e_i \leq 0$ is a negative-semi definite function, so it follows that $V_i(t) \leq V_i(0)$ which means that $V(t) \in L_\infty$ and the signals $e_i, \dot{e}_i, \tilde{\omega}_i, \tilde{\pi}_i$ and u_i are bounded. Define the following term

$$\dot{T}_i(t) = \frac{1}{2}e_i^T Q_i e_i \tag{5.25}$$

By integrating (5.25), one has

$$\int_0^t T_i(\tau)d\tau = \dot{V}_i(0) - \dot{V}_i(t) \tag{5.26}$$

Because $\dot{V}_i(0)$ is bounded, and $\dot{V}_i(t)$ is non-increasing and bounded, one can get

$$\lim_{t\to\infty} \int_0^t T_i(\tau)d\tau < \infty \tag{5.27}$$

In addition, since $\dot{T}_i$ is bounded, using Barbalat's Lemma [83], it can be shown that $\lim_{t\to\infty} T_i(t) = 0$. As a result, the stability of the proposed control system can be guaranteed and makes $\lim_{t\to\infty} e_i = 0$. Thus, The stability of the proposed control system is guaranteed by Theorem. ■

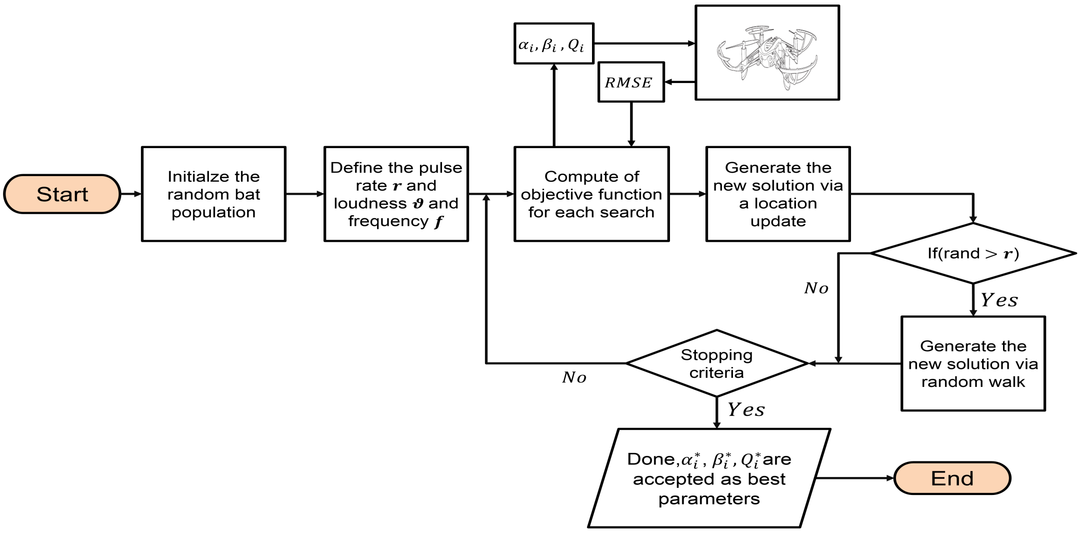

α_i, β_i, Q_i
$RMSE$
Start
Initialze the random bat population
Define the pulse rate r and loudness ϑ and frequency f
Compute of objective function for each search
Generate the new solution via a location update
If(rand > r)
No
Yes
Generate the new solution via random walk
Stopping criteria
No
Yes
Done,$\alpha_i^*, \beta_i^*, Q_i^*$are accepted as best parameters
End

FIGURE 5.2: Flowchart of BA process.

5.4 Optimization process using BA

In this section, we consider the optimization of the control parameters using the bat algorithm (BA) to ensure maximum performances even when changing the reference trajectory. From equation (5.15), it should be emphasized that the proposed control law becomes optimal if the existing parameters α_i, β_i and P_i are derived from the Lyapunov equation (5.16) via Q_i have optimal values. Hence, the BA is used to reach the target solutions α_i^*, β_i^* and Q_i^* against minimizing the objective function J_i for $i \in \{x, y, z, \phi, \theta, \psi\}$ such as the RMSE given in (4.30). The following steps summarize the optimization process in a way that can find the optimal solution which defines the optimal parameters also the flowchart of the BA is given in figure 5.2.

Step 1. Initializing the population and parameters: Let the bat's population size be Np and the initial population x_l is randomly generated. The initial population is calculated for

$$x_l = \{\alpha_i, \beta_i\, Q_i\}, 1 \leq l \leq Np \tag{5.28}$$

The bat parameters for position optimization are initialized for the optimization process. Then, calculate the objective function J_i among all the population and find the best position x_l^{best}.

Step 2. Updating the position of the search process: The bat search for the target optimum values through the updating of the position and find the best location. Updating the new location is calculated via Eqs. (3.5-3.7). For each iteration, the objective function is calculated for each best generation to evaluate the performances of the BA algorithm, where the best target has the minimum objective function value for each search.

Step 3. Random walk: A new random walk added to the generated location allows the bats to get more degree of freedom for search using Eqs. (3.8-3.9).

Step 4. Termination: Repeat all the steps until the current best solution is obtained or the condition criterion is reached. The optimal parameters that are defined will be integrated into the proposed controller as an optimal solution for the simulation purpose.

5.5 Simulations

TABLE 5.1: Parameters of the quadrotor

Parameters	Value
I_x	$0.068 \times 10^{-3}\ kg.m^2$
I_y	$0.92 \times 10^{-3}\ kg.m^2$
I_z	$0.1366 \times 10^{-2}\ kg.m^2$
J_r	$6 \times 10^{-5}\ kg.m^2$
g	$9.81\ m/s^2$
l	$0.065\ m$
m	$0.068\ kg$

In this section, the performances of the proposed controller have been tested using Simulink for Parrot Minidrone platform. The parameters of the minidrone used in these simulations are shown in Table 5.1. The bat algorithm has been included in offline loop to compute the OFMFC optimal parameters locally in a search space:
$x_{min} = \{\alpha_{min}, \beta_{min}, q_{1i,min}, q_{2i,min}\} = \{0.1, 0.1, 0.1, 0.1\}, x_{max} = \{\alpha_{max}, \beta_{max}, q_{1i,max}$ $, q_{2i,max}\} = \{10, 10, 10, 10\}$, the set of the space search is an arbitrary choice defined by the user. The parameters of the bat algorithm are given in Table 5.2. The input variables of each local fuzzy controller are $e_{\chi i}$ and e_{vi} . For each variable, we have three Gaussian membership functions and two sigmoid curve functions. The Gaussian functions are defined as

TABLE 5.2: BA Parameters

Parameters	Symbol	Value
Population size	Np	10
Constant	γ	0.5
Minimum frequency	f_{min}	0
Maximum frequency	f_{max}	2

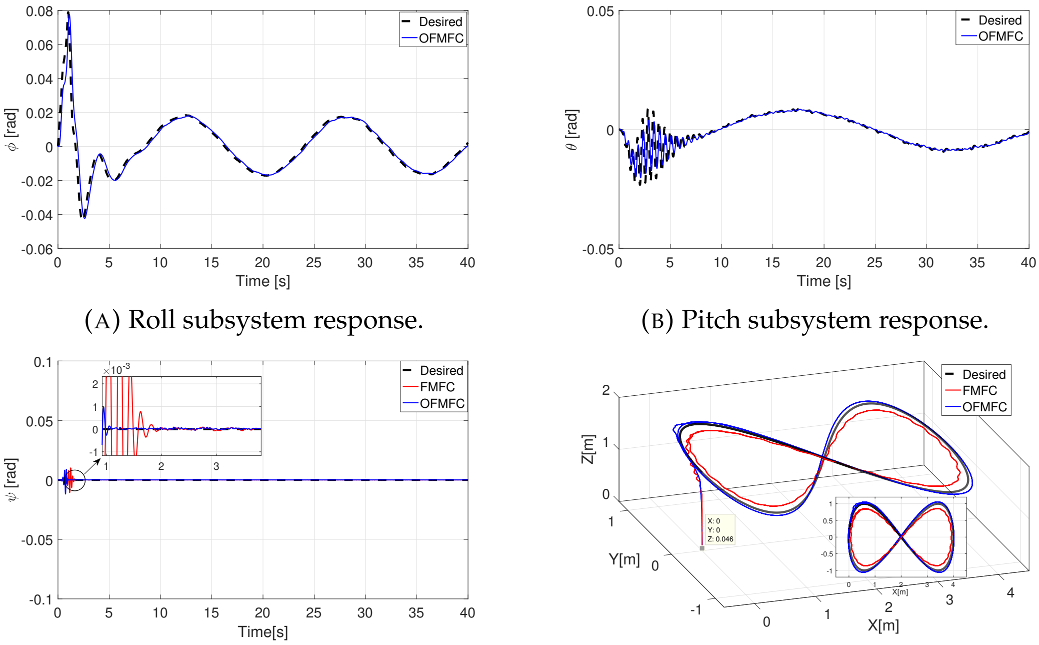
$$\mu_{\check{o}_i^l}(e_{ji}) = \exp\left\{-\frac{1}{2}\left(\frac{e_{ji} - C_i}{\sigma_i}\right)\right\}, \; l = 2:4, \; j = \;\chi, v$$

where $C_{x,y,z} = [-2,0,2]$, $\sigma_{x,y,z} = 0.5$, $C_{\phi,\theta,\psi} = [1.04,0,1.04]$ and $\sigma_{\phi,\theta,\psi} = 0.37$. The sigmoid functions are defined as follows:

$$\mu_{\breve{o}_i^l}(e_{ji}) = \frac{1}{1+\exp^{-\tau_i(e_{ji}-\varrho_i)}}, \quad l = 1,...,5, \; j = \chi, v$$

where $\tau_{x,y,z} = [-2,2]$, $\varrho_{x,y,z} = 5$, $\tau_{\phi,\theta,\psi} = [-2.09,2.09]$ and $\varrho_{\phi,\theta,\psi} = 2.3$.

To evaluate the OFMFC performances, two cases are carried out on the implement of the corresponding control strategy. The first case is a comparative study between the proposed controller with optimal parameters and the FMFC



(A) Roll subsystem response.

(B) Pitch subsystem response.

(C) Yaw subsystem response.

(D) Position system response.

FIGURE 5.3: Quadrotor responses (case 1).

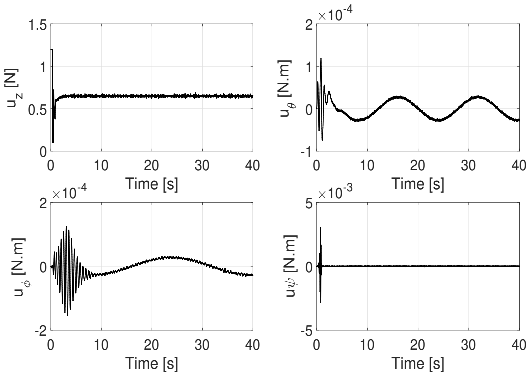


FIGURE 5.4: Control inputs of OMFBC (case 1).

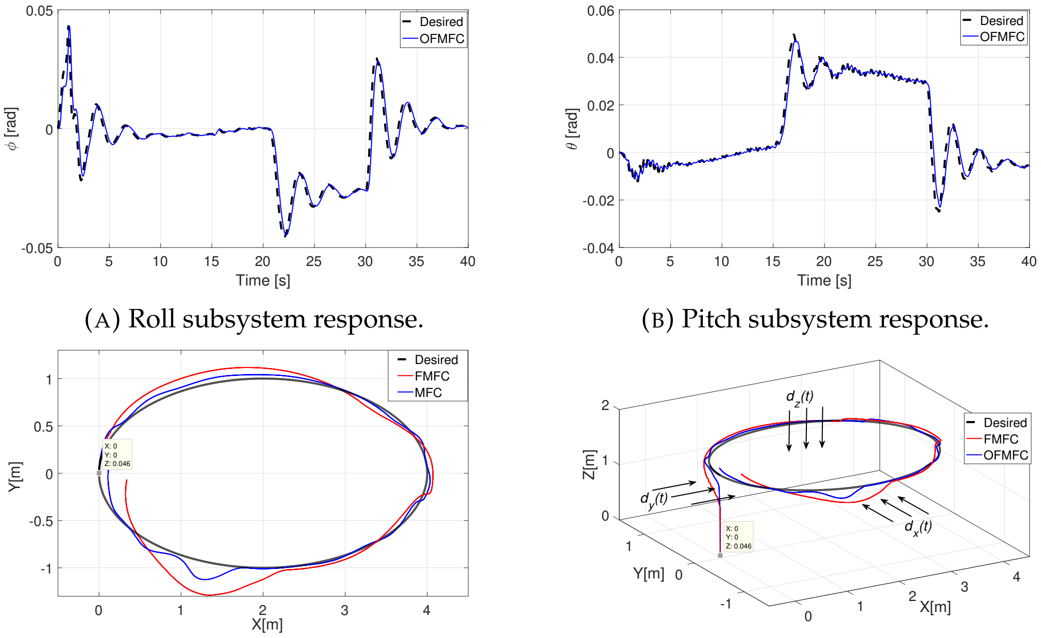
whose parameters were arbitrarily chosen. The second case discusses the performance of the proposed strategy in the presence of the external disturbances.
Case 1. In this case, the proposed OFMFC is compared with FMFC whose parameters have been chosen arbitrarily. The desired trajectory is defined as:

$$\begin{cases} x_d = 2(1 - cos(0.2t))m \\ y_d = sin(0.4t)m \\ z_d = 1.5m \\ \psi_d = 0rad \end{cases}$$

The initial conditions of the Mini-Drone are $y_0 = [0, 0, 0.04, 0, 0, 0]^T$. The simulation results of this case are depicted in Figures 5.3-5.4. It can observe from the obtained results in the same figures the superiority of the proposed controller than the other controller due to the optimization process which ensures locally optimal parameters to minimize the tracking error between the desired and the actual path. It can be seen the input signals in Figure 5.4, where it is obvious that the signals are smooth and bounded which absolutely means that the proposed control can be real-time implemented.
Case 2. In this case, the proposed strategy has been tested under the external disturbances $d_{x,y,z}(t)$, where the simulation results are obtained with a circular trajectory in 3D space. The desired path is given as: $x_d = 2(1 - \cos(0.2t))m$, $y_d = \sin(0.2t)m$, $z_d = 1.5m$ and $\psi_d = 0rad$. The disturbances which are imposed on the altitude and the position are given as:

$$\begin{cases} d_x(t) = 0.2N, 10 < t < 35\ sec; \\ d_y(t) = 0.2N, 12.5 < t < 35\ sec; \\ d_z(t) = 0.5N, 15 < t < 35\ sec. \end{cases} .$$

The simulation results of this case are depicted in Figures 5.5-5.6. It can be seen from the same figures the best tracking performed by the OFMFC, where the proposed controller can overcome the imposed disturbances. In addition, the FMFC also can achieve satisfactory tracking but the optimal can reach to the



(A) Roll subsystem response.

(B) Pitch subsystem response.

(C) Yaw subsystem response.

(D) Position system response.

FIGURE 5.5: Quadrotor responses (case 2).

optimal behaviour with minimum error tracking control only with OFMFC. Figure 5.6 shows the control signal provided by the proposed control, where it's clear that the controller can reach the desired signal to compensate the imposed disturbances.

5.6 Summary

In this chapter, we have proposed an optimal fuzzy model-free control for the trajectory tracking problem of a quadrotor. The control law is designed in a

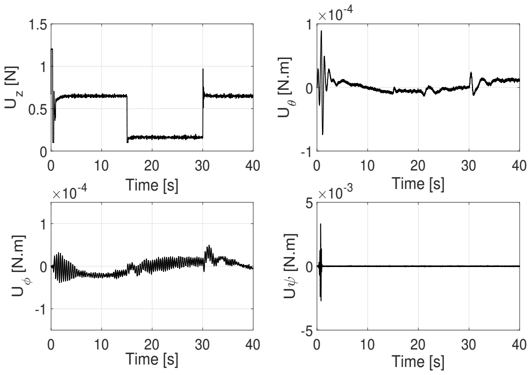


FIGURE 5.6: Control inputs of OMFBC (case 2).

way that allows it to deal with disturbances effects. To address the control parameters selection drawback the bat algorithm is added to tune the proposed controller parameters. To test the effectiveness of the proposed strategy different cases have been validated in the simulation part; the proposed optimal controller was compared with another controller in two different environments. The obtained results have proved the robustness and null steady-state error tracking ability of the proposed control strategy. Moreover, the simplicity and model-free features, as well as the continuous control signals, make it easier to implement the proposed control strategy in a real quadrotor.

Chapter 6

Conclusion & Open problems

The growth of interest in UAVs, like quadrotor drone, mini-drone and above all lightweight drones has opened a big challenge in many research fields in recent years. Hence, the main motivation for this research work was the synthesis of stabilizing control laws for this type of device.

The research given in this thesis concerns the development of flight tracking controller for the quadrotor system. Therefore, new strategies and techniques are proposed in this study to get optimal tracking control for the quadrotor. The results of the thesis are presented next.

6.1 Thesis Summary

The proposed control strategies in this thesis are applied on the quadrotor UAV. Hence, the description of the quadrotor dynamics and its motion are required before proposing any algorithm related to it. Chapter 2 is intended to give details at the nature of this UAV by extracting its dynamics via Newton–Euler formalism that represent its motion. The extracted nonlinear model has been successfully used to different quadrotor tracking control strategies in the following chapters.

The proposed control strategies are based on an optimization process that makes the drone fight with optimal behaviour. Therefore, the necessity of proposing a closer look at metaheuristic optimization algorithms to be applied to the control law was the main mission of chapter 3. A detailed description of three different optimization algorithm (Cs, BA, FPA) was given that allows the understanding of the optimization process for each algorithm.

In order to make the drone flight and track the given path is necessary to develop a tracking control for this flight machine. Chapter 4 is focusing on designing optimal model-free controllers based on the backstepping approach for the quadrotor. A new nonlinear function estimation technique was combined with the BC to compensate the unmodeled dynamics and uncertainties. The fuzzy system combined with BC is another approach that has been proposed and investigated. The stability analysis for these control strategies is analytically proved and then verified by numerical flight tests.

Based on the difficulties that have been found in the previously proposed techniques, chapter 6 has shown the design of an intelligent optimal model-free controller for the quadrotor. The proposed strategy could offer an optimal tracking control with null study error based on the direct fuzzy technique. The results show that the proposed control has the best performance with respect to compared design controllers.

6.2 Open problems

Further research studies can be made to improve the results of this thesis:

- Regarding the quadrotor model, the given model can be further improved by adding specific neglected terms to be tested.

- The optimization area has large fields and filled by many optimization algorithms, it can test and validate other algorithms which are not tested and not validated in this research.

- Regarding the control part, MFC based on BC can be further improved by developing auxiliary systems to address the explosion of complexty drawback that can be found in the BC.

- Investigation in other intelligent control design like neural network and machine learning can be a big improvement for the MFC.

- The safety is always required and any fault could lead to a catastrophic failure. Therefore, fault-tolerant control could be a big deal for this type of machine to reduce the fault effects and increase the robustness of the proposed control law.

Bibliography

[1] Almoataz Y Abdelaziz, Ehab S Ali, and SM Abd Elazim. “Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems”. In: *energy* 101 (2016), pp. 506–518.

[2] AY Abdelaziz and ES Ali. “Cuckoo search algorithm based load frequency controller design for nonlinear interconnected power system”. In: *International Journal of Electrical Power & Energy Systems* 73 (2015), pp. 632–643.

[3] Latifa Abdou, Hossam-Eddine Glida, et al. “Parameters tuning of a quadrotor PID controllers by using nature-inspired algorithms”. In: *Evolutionary Intelligence* (2019), pp. 1–13.

[4] Victor G Adir and Adrian M Stoica. “Integral LQR control of a star-shaped octorotor”. In: 4.2 (2012), p. 3.

[5] Younes Al Younes et al. “Robust model-free control applied to a quadrotor UAV”. In: *Journal of Intelligent & Robotic Systems* 84.1-4 (2016), pp. 37–52.

[6] DF Alam, DA Yousri, and MB Eteiba. “Flower pollination algorithm based solar PV parameter estimation”. In: *Energy Conversion and Management* 101 (2015), pp. 410–422.

[7] ES Ali. “Optimization of power system stabilizers using BAT search algorithm”. In: *International Journal of Electrical Power & Energy Systems* 61 (2014), pp. 683–690.

[8] Hilal Alkan and c C a u g layan Balkaya. “Parameter estimation by Differential Search Algorithm from horizontal loop electromagnetic (HLEM) data”. In: *Journal of Applied Geophysics* 149 (2018), pp. 77–94.

[9] Jaemin Baek and Jinmyung Jung. “A Model-Free Control Scheme for Attitude Stabilization of Quadrotor Systems”. In: *Electronics* 9.10 (2020), p. 1586.

[10] Bahman Bahmani-Firouzi and Rasoul Azizipanah-Abarghooee. “Optimal sizing of battery energy storage for micro-grid operation management using a new improved bat algorithm”. In: *International Journal of Electrical Power & Energy Systems* 56 (2014), pp. 42–54.

[11] Mohd Ariffanan Mohd Basri, Abdul Rashid Husain, and Kumeresan A Danapalasingam. “Enhanced backstepping controller design with application to autonomous quadrotor unmanned aerial vehicle”. In: *Journal of Intelligent & Robotic Systems* 79.2 (2015), pp. 295–321.

[12] Rabah Benkercha, Samir Moulahoum, and Bilal Taghezouit. “Extraction of the PV modules parameters with MPP estimation using the modified flower algorithm”. In: *Renewable Energy* 143 (2019), pp. 1698–1709.

[13] Abdelhamid Bounemeur, Mohamed Chemachema, and Najib Essounbouli. “Indirect adaptive fuzzy fault-tolerant tracking control for MIMO nonlinear systems with actuator and sensor failures”. In: *ISA transactions* 79 (2018), pp. 45–61.

[14] Oscar R Carvajal, Oscar Castillo, and José Soria. “Optimization of membership function parameters for fuzzy controllers of an autonomous mobile robot using the flower pollination algorithm”. In: *Journal of Automation Mobile Robotics and Intelligent Systems* 12 (2018).

[15] S Chaine, M Tripathy, and Divesh Jain. “Non dominated Cuckoo search algorithm optimized controllers to improve the frequency regulation characteristics of wind thermal power system”. In: *Engineering science and technology, an international journal* 20.3 (2017), pp. 1092–1105.

[16] Ursula Challita et al. “Machine learning for wireless connectivity and security of cellular-connected UAVs”. In: *IEEE Wireless Communications* 26.1 (2019), pp. 28–35.

[17] Young-Cheol Choi and Hyo-Sung Ahn. "Nonlinear control of quadrotor for point tracking: Actual implementation and experimental tests". In: *IEEE/ASME transactions on mechatronics* 20.3 (2014), pp. 1179–1192.

[18] Abdelhamid Chriette. "Contribution a la commande et a la modelisation des helicopteres: Asservissement visuel et commande adaptative". PhD thesis. Evry-Val d'Essonne, 2001.

[19] Pinar Civicioglu and Erkan Besdok. "Bernstain-search differential evolution algorithm for numerical function optimization". In: *Expert Systems with Applications* 138 (2019), p. 112831.

[20] Bart Custers, JanJaap Oerlemans, and Bas Vergouw. "Het Gebruik Van Drones. Een Verkennend Onderzoek Naar Onbemande Luchtvaartuigen (The Use of Drones: An Exploratory Study on Unmanned Aerial Vehicles (UAVs))". In: *Custers BHM, Oerlemans JJ & Vergouw SJ (2015), Het gebruik van drones. Een verkennend onderzoek naar onbemande luchtvaartuigen (Wetenschappelijk Onderzoek-en Documentatiecentrum) Onderzoek en beleid WODC* 313 (2015).

[21] Abhijit Das, Frank Lewis, and Kamesh Subbarao. "Backstepping approach for controlling a quadrotor using lagrange form dynamics". In: *Journal of Intelligent and Robotic Systems* 56.1-2 (2009), pp. 127–151.

[22] Puja Dash, Lalit Chandra Saikia, and Nidul Sinha. "Flower pollination algorithm optimized PI-PD cascade controller in automatic generation control of a multi-area power system". In: *International Journal of Electrical Power & Energy Systems* 82 (2016), pp. 19–28.

[23] Travis Dierks and Sarangapani Jagannathan. "Output feedback control of a quadrotor UAV using neural networks". In: *IEEE transactions on neural networks* 21.1 (2009), pp. 50–66.

[24] Dariush Ebrahimi et al. "UAV-aided projection-based compressive data gathering in wireless sensor networks". In: *IEEE Internet of Things Journal* 6.2 (2018), pp. 1893–1905.

[25] Mehmet Önder Efe. “Sliding mode control for unmanned aerial vehicles research”. In: *Recent Advances in Sliding Modes: From Control to Intelligent Mechatronics*. Springer, 2015, pp. 239–255.

[26] Farzaneh Ezazi et al. “A new method for multicomponent mixture separation cascade optimization using artificial bee colony algorithm”. In: *Progress in Nuclear Energy* 124 (2020), p. 103371.

[27] Ludovico Ferranti et al. “DUPLICATE: Drone cellular networks: Enhancing the quality of experience of video streaming applications”. In: *Ad Hoc Networks* 80 (2018), pp. 130–141.

[28] Nuradeen Fethalla et al. “Robust observer-based dynamic sliding mode controller for a quadrotor UAV”. In: *IEEE access* 6 (2018), pp. 45846–45859.

[29] Amir Hossein Gandomi et al. “Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization”. In: *Computers & Mathematics with Applications* 63.1 (2012), pp. 191–200.

[30] Wei Gao. “Investigating the critical slip surface of soil slope based on an improved black hole algorithm”. In: *Soils and foundations* 57.6 (2017), pp. 988–1001.

[31] Reham Gharbia et al. “Multi-spectral and panchromatic image fusion approach using stationary wavelet transform and swarm flower pollination optimization for remote sensing applications”. In: *Future Generation Computer Systems* 88 (2018), pp. 501–511.

[32] Meysam Gheisarnejad. “An effective hybrid harmony search and cuckoo optimization algorithm based fuzzy PID controller for load frequency control”. In: *Applied Soft Computing* 65 (2018), pp. 121–138.

[33] Chen Giladi and Avishai Sintov. “Manifold learning for efficient gravitational search algorithm”. In: *Information Sciences* 517 (2020), pp. 18–36.

[34] Hossam-Eddine Glida, Latifa Abdou, and Abdelghani Chelihi. "Optimal fuzzy adaptive backstepping controller for attitude control of a quadrotor helicopter". In: *2019 International Conference on Control, Automation and Diagnosis (ICCAD)*. IEEE. 2019, pp. 1–6.

[35] Hossam Eddine Glida et al. "Optimal model-free backstepping control for a quadrotor helicopter". In: *Nonlinear Dynamics* (2020), pp. 1–20.

[36] Bryan Godbolt, Nikolaos I Vitzilaios, and Alan F Lynch. "Experimental validation of a helicopter autopilot design using model-based PID control". In: *Journal of Intelligent & Robotic Systems* 70.1-4 (2013), pp. 385–399.

[37] Huajun Gong, Hao Xu, and Fahmida N Chowdhury. "A neuro-augmented observer for a class of nonlinear systems". In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE. 2006, pp. 2497–2500.

[38] Higinio González-Jorge, Joaquin Martínez-Sánchez, Martín Bueno, et al. "Unmanned aerial systems for civil applications: A review". In: 1.1 (2017), p. 2.

[39] ChangSu Ha et al. "Passivity-based adaptive backstepping control of quadrotor-type UAVs". In: *Robotics and Autonomous Systems* 62.9 (2014), pp. 1305–1315.

[40] Blake Hament and Paul Oh. "Unmanned aerial and ground vehicle (UAV-UGV) system prototype for civil infrastructure missions". In: *2018 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE. 2018, pp. 1–4.

[41] Nguyen Thi Hanh et al. "An efficient genetic algorithm for maximizing area coverage in wireless sensor networks". In: *Information Sciences* 488 (2019), pp. 58–75.

[42] *Hank Price Federal Aviation Administration (FAA), 2019. Forecast fiscal.* https://www.faa.gov/news/fact$_sheets/news_story.cfm?newsId$22594. 2017, pp. 04–26.

[43] Jemin Hwangbo et al. "Control of a quadrotor with reinforcement learning". In: *IEEE Robotics and Automation Letters* 2.4 (2017), pp. 2096–2103.

[44] Jun Inoue, Judith A Curry, and James A Maslanik. "Application of Aerosondes to melt-pond observations over Arctic sea ice". In: *Journal of Atmospheric and Oceanic technology* 25.2 (2008), pp. 327–334.

[45] Chengquan Ju, Shuhan Yao, and Peng Wang. "Resilient post-disaster system reconfiguration for multiple energy service restoration". In: *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*. IEEE. 2017, pp. 1–6.

[46] Ammar Mansoor Kamoona and Jagdish Chandra Patra. "A novel enhanced cuckoo search algorithm for contrast enhancement of gray scale images". In: *Applied Soft Computing* 85 (2019), p. 105749.

[47] Chiranjeevi Karri and Umaranjan Jena. "Fast vector quantization using a Bat algorithm for image compression". In: *Engineering Science and Technology, an International Journal* 19.2 (2016), pp. 769–781.

[48] A Kaveh and N Farhoudi. "A new optimization method: Dolphin echolocation". In: *Advances in Engineering Software* 59 (2013), pp. 53–70.

[49] A Kaveh and N Farhoudi. "Dolphin monitoring for enhancing metaheuristic algorithms: Layout optimization of braced frames". In: *Computers & Structures* 165 (2016), pp. 1–9.

[50] D Kesavaraja and A Shenbagavalli. "QoE enhancement in cloud virtual machine allocation using Eagle strategy of hybrid krill herd optimization". In: *Journal of Parallel and Distributed Computing* 118 (2018), pp. 267–279.

[51] Do Wan Kim. "Fuzzy model-based control of a quadrotor". In: *Fuzzy Sets and Systems* 371 (2019), pp. 136–147.

[52] J Senthil Kumar et al. "Optimizing renewable based generations in AC/DC microgrid system using hybrid Nelder Mead Cuckoo Search algorithm". In: *Energy* 158 (2018), pp. 204–215.

[53] Moussa Labbadi and Mohamed Cherkaoui. "Robust adaptive backstepping fast terminal sliding mode controller for uncertain quadrotor UAV". In: *Aerospace Science and Technology* 93 (2019), p. 105306.

[54] SK Lakshmanaprabu, U Sabura Banu, and PR Hemavathy. "Fractional order IMC based PID controller design using Novel Bat optimization algorithm for TITO Process". In: *Energy Procedia* 117 (2017), pp. 1125–1133.

[55] Nathan O Lambert et al. "Low-level control of a quadrotor with deep model-based reinforcement learning". In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 4224–4230.

[56] Daewon Lee, H Jin Kim, and Shankar Sastry. "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter". In: *International Journal of control, Automation and systems* 7.3 (2009), pp. 419–428.

[57] Zhenyu Lei et al. "An aggregative learning gravitational search algorithm with self-adaptive gravitational constants". In: *Expert Systems with Applications* (2020), p. 113396.

[58] Shushuai Li, Yaonan Wang, and Jianhao Tan. "Adaptive and robust control of quadrotor aircrafts with input saturation". In: *Nonlinear Dynamics* 89.1 (2017), pp. 255–265.

[59] Zhi Li, Xin Ma, and Yibin Li. "Model-free control of a quadrotor using adaptive proportional derivative-sliding mode control and robust integral of the signum of the error". In: *International Journal of Advanced Robotic Systems* 15.5 (2018).

[60] Zhi Li, Xin Ma, and Yibin Li. "Robust tracking control strategy for a quadrotor using RPD-SMC and RISE". In: *Neurocomputing* 331 (2019), pp. 312–322.

[61] Hao Liu et al. "Robust attitude control of uncertain quadrotors". In: *IET Control Theory and Applications* 7.11 (2013), pp. 1583–1589.

[62] Rogelio Lozano, Pedro Castillo, and Alejandro Dzul. "Global stabilization of the PVTOL: real-time application to a mini-aircraft". In: *IFAC Proceedings Volumes* 37.21 (2004), pp. 235–240.

[63] Satya Dinesh Madasu, MLS Sai Kumar, and Arun Kumar Singh. "A flower pollination algorithm based automatic generation control of interconnected power system". In: *Ain Shams Engineering Journal* 9.4 (2018), pp. 1215–1224.

[64] MJ Mahmoodabadi and N Rezaee Babak. "Robust fuzzy linear quadratic regulator control optimized by multi-objective high exploration particle swarm optimization for a 4 degree-of-freedom quadrotor". In: *Aerospace Science and Technology* 97 (2020), p. 105598.

[65] Mostafa Mohammadi and Alireza Mohammad Shahri. "Adaptive nonlinear stabilization control for a quadrotor UAV: Theory, simulation and experimentation". In: *Journal of Intelligent & Robotic Systems* 72.1 (2013), pp. 105–122.

[66] Tiago P Nascimento and Martin Saska. "Position and attitude control of multi-rotor aerial vehicles: A survey". In: *Annual Reviews in Control* 48 (2019), pp. 129–146.

[67] Thang Trung Nguyen and Dieu Ngoc Vo. "Modified cuckoo search algorithm for multiobjective short-term hydrothermal scheduling". In: *Swarm and evolutionary computation* 37 (2017), pp. 73–89.

[68] Milo v s Nikoli ´c et al. "Bee Colony Optimization Metaheuristic for Fuzzy Membership Functions Tuning". In: *Expert Systems with Applications* (2020), p. 113601.

[69] Kok Meng Ong et al. "Effective moving object tracking using modified flower pollination algorithm for visible image sequences under complicated background". In: *Applied Soft Computing* 83 (2019), p. 105625.

[70] Pauline Ong and Zarita Zainuddin. "Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction". In: *Applied Soft Computing* 80 (2019), pp. 374–386.

[71] Danilo Pelusi et al. “Improving exploration and exploitation via a Hyperbolic Gravitational Search Algorithm”. In: *Knowledge-Based Systems* 193 (2020), p. 105404.

[72] Gabriele Perozzi. “Safe exploration of an aerodynamic field by a mini drone”. PhD thesis. Ecole Centrale Lille, 2018.

[73] Gabriele Perozzi et al. “Trajectory tracking for a quadrotor under wind perturbations: sliding mode control with state-dependent gains”. In: *Journal of the Franklin Institute* 355.12 (2018), pp. 4809–4838.

[74] Pedro Ponce et al. “Experimental fuzzy logic controller type 2 for a quadrotor optimized by ANFIS”. In: *IFAC-PapersOnLine* 48.3 (2015), pp. 2435–2441.

[75] Devendra Potnuru, K Alice Mary, and Ch Sai Babu. “Experimental implementation of Flower Pollination Algorithm for speed controller of a BLDC motor”. In: *Ain Shams Engineering Journal* 10.2 (2019), pp. 287–295.

[76] Paul EI Pounds, Daniel R Bersak, and Aaron M Dollar. “Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control”. In: *Autonomous Robots* 33.1-2 (2012), pp. 129–142.

[77] Heriberto Ramirez-Rodriguez et al. “Robust backstepping control based on integral sliding modes for tracking of quadrotors”. In: *Journal of Intelligent & Robotic Systems* 73.1-4 (2014), pp. 51–66.

[78] Hadi Razmi and Sima Afshinfar. “Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV”. In: *Aerospace Science and Technology* 91 (2019), pp. 12–27.

[79] DK Sambariya and R Prasad. “Robust tuning of power system stabilizer for small signal stability enhancement using metaheuristic bat algorithm”. In: *International Journal of Electrical Power & Energy Systems* 61 (2014), pp. 229–238.

[80] J Sasikala, D Sujitha Juliet, et al. "Optimized vessel detection in marine environment using hybrid adaptive cuckoo search algorithm". In: *Computers & Electrical Engineering* 78 (2019), pp. 482–492.

[81] Sanaa Sharafeddine and Rania Islambouli. "On-demand deployment of multiple aerial base stations for traffic offloading and network recovery". In: *Computer Networks* 156 (2019), pp. 52–61.

[82] Jaspreet Singh, Ranjit Kaur, and Damanpreet Singh. "A Survey and Taxonomy on Energy Management Schemes in Wireless Sensor Networks". In: *Journal of Systems Architecture* (2020), p. 101782.

[83] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*. Vol. 199. 1. Prentice hall Englewood Cliffs, NJ, 1991.

[84] S Solb o and R Storvold. "Mapping svalbard glaciers with the cryowing UAS". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 1 (2013), W2.

[85] Pei-Cheng Song, Jeng-Shyang Pan, and Shu-Chuan Chu. "A parallel compact cuckoo search algorithm for three-dimensional path planning". In: *Applied Soft Computing* (2020), p. 106443.

[86] Liling Sun et al. "A modified Surrogate-Assisted Multi-swarm Artificial Bee Colony for Complex Numerical Optimization Problems". In: *Microprocessors and Microsystems* (2020), p. 103050.

[87] Nesrine Talbi. "Design of fuzzy controller rule base using bat algorithm". In: *Energy Procedia* 162 (2019), pp. 241–250.

[88] Markus Tilg et al. "LIFE: laser induced fluorescence emission, a non-invasive tool to detect photosynthetic pigments in glacial ecosystems". In: *Instruments, Methods, and Missions for Astrobiology XIV*. Vol. 8152. International Society for Optics and Photonics. 2011, p. 81520I.

[89] Bas Vergouw et al. "Drone technology: Types, payloads, applications, frequency spectrum issues and future developments". In: *The future of drone use*. Springer, 2016, pp. 21–45.

[90] Haoping Wang et al. "Model-free–based terminal SMC of quadrotor attitude and position". In: *IEEE Transactions on Aerospace and Electronic Systems* 52.5 (2016), pp. 2519–2528.

[91] Dwa Desa Warnana et al. "Black hole algorithm for determining model parameter in self-potential data". In: *Journal of Applied Geophysics* 148 (2018), pp. 189–200.

[92] Jing-Jing Xiong and Guo-Bao Zhang. "Global fast dynamic terminal sliding mode control for a quadrotor UAV". In: *ISA transactions* 66 (2017), pp. 233–240.

[93] Jing-Jing Xiong and En-Hui Zheng. "Position and attitude tracking control for a quadrotor UAV". In: *ISA transactions* 53.3 (2014), pp. 725–731.

[94] Shuhui Xu and Yong Wang. "Parameter estimation of photovoltaic modules using a hybrid flower pollination algorithm". In: *Energy Conversion and Management* 144 (2017), pp. 53–68.

[95] Bo Yang et al. "Modified cuckoo search algorithm for the optimal placement of actuators problem". In: *Applied Soft Computing* 67 (2018), pp. 48–60.

[96] Xin-She Yang. "A new metaheuristic bat-inspired algorithm". In: *Nature inspired cooperative strategies for optimization (NICSO 2010)*. Springer, 2010, pp. 65–74.

[97] Xin-She Yang. "Flower pollination algorithm for global optimization". In: *International conference on unconventional computing and natural computation*. Springer. 2012, pp. 240–249.

[98] Xin-She Yang. *Nature-inspired optimization algorithms*. Elsevier, 2014.

[99] Xin-She Yang and Suash Deb. "Cuckoo search via Levy flights". In: *2009 World congress on nature and biologically inspired computing (NaBIC)*. IEEE. 2009, pp. 210–214.

[100] Younes Al Younes et al. "Quadrotor position Control using cascaded adaptive integral backstepping controllers". In: *Applied Mechanics and Materials*. Vol. 565. Trans Tech Publ. 2014, pp. 98–106.

[101] Dalia Yousri and Seyedali Mirjalili. “Fractional-order cuckoo search algorithm for parameter identification of the fractional-order chaotic, chaotic with noise and hyper-chaotic financial systems”. In: *Engineering Applications of Artificial Intelligence* 92 (2020), p. 103662.

[102] Jia Wei Zhang and Gai Ge Wang. “Image matching using a bat algorithm with mutation”. In: *Applied Mechanics and Materials*. Vol. 203. Trans Tech Publ. 2012, pp. 88–93.

[103] Qixun Zhang et al. “IoT enabled UAV: Network architecture and routing algorithm”. In: *IEEE Internet of Things Journal* 6.2 (2019), pp. 3727–3742.