

Towards aerial robotic workers : design, control and guidance for unmanned aerial manipulators

Imanberdiyev, Nursultan

2020

Imanberdiyev, N. (2020). Towards aerial robotic workers : design, control and guidance for unmanned aerial manipulators. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/144236>

<https://doi.org/10.32657/10356/144236>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 09 Apr 2024 14:32:11 SGT



TOWARDS AERIAL ROBOTIC WORKERS: DESIGN, CONTROL AND GUIDANCE FOR UNMANNED AERIAL MANIPULATORS

NURSULTAN IMANBERDIYEV

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

12/09/2020

.....

Date



.....

NURSULTAN IMANBERDIYEV

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

16/09/2020

.....

Date



.....

Assoc Prof Pham Quang Cuong

Co-Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

12/09/2020

.....

Date



.....

Assoc Prof Erdal Kayacan

Authorship Attribution Statement

This thesis contains material from 5 papers published or under review in the following peer-reviewed journals and from papers accepted or under review at conferences in which I am listed as an author.

Chapter 2 and 4 have material from the paper published as Andriy Sarabakha, Nursultan Imanberdiyev, Erdal Kayacan, Mojtaba Ahmadi Khansar, and Hani Hagaras, “Novel Levenberg–Marquardt Based Learning Algorithm for Unmanned Aerial Vehicles”, *Information Sciences*, 417 (2017), pp. 361-380. DOI: [10.1016/j.ins.2017.07.020](https://doi.org/10.1016/j.ins.2017.07.020).

The contributions of the co-authors are as follows:

- I performed the experiments and prepared the manuscript draft.
- Andriy Sarabakha coded the control algorithms and prepared the manuscript draft.
- Dr. Erdal Kayacan has provided supervision and editorial comments while writing the paper.
- Dr. Mojtaba Ahmadi Khansar provided the theoretical framework.
- Dr. Hani Hagaras revised the manuscript.

Chapter 3 has material from the paper under review as Nursultan Imanberdiyev, Sumil Sood, Dogan Kircali, and Erdal Kayacan, “Design, Development and Experimental Validation of a Lightweight Dual-arm Aerial Manipulator with a COG Balancing Mechanism”, *Mechatronics* (submitted).

The contributions of the co-authors are as follows:

- I have developed the proposed design. I have performed all experimental flight tests to validate the proposed design. I have also prepared the manuscript drafts.
- Sumil Sood has helped with the design process and preparation of the experimental setup.
- Dogan Kircali has helped with performing the experimental flight tests.
- Dr. Erdal Kayacan has provided supervision and editorial comments while writing the paper.

Chapter 4 has material from the paper published as Nursultan Imanberdiyev and Erdal Kayacan, “A Fast Learning Control Strategy for Unmanned Aerial Manipulators”, *Journal of Intelligent & Robotic Systems*, 94 (2019), no. 3, pp. 805-824. DOI: [10.1007/s10846-018-0884-7](https://doi.org/10.1007/s10846-018-0884-7).

The contributions of the co-authors are as follows:

- Since I am the only author besides my supervisor, I have conducted the whole research. I have developed and tested the proposed control strategy both in simulations and in real flights. I have also prepared the manuscript drafts.
- Dr. Erdal Kayacan has provided supervision and editorial comments while writing the paper.

Chapter 5 has material from the paper published as Nursultan Imanberdiyev, Josephine Monica, and Erdal Kayacan, “A Multi-Task Velocity-based Redundancy Resolution Strategy for Unmanned Aerial Manipulators”, *2019 18th European Control Conference (ECC)*, 2019, pp. 1130-1135. DOI: [10.23919/ECC.2019.8796272](https://doi.org/10.23919/ECC.2019.8796272).

The contributions of the co-authors are as follows:

- I have developed the proposed redundancy resolution strategy. I have performed all simulations to validate the proposed framework. I have also prepared the manuscript drafts.
- Josephine Monica has helped with performing the simulation studies and provided editorial comments.
- Dr. Erdal Kayacan has provided supervision and editorial comments while writing the paper.

Chapter 5 has material from the paper accepted as Nursultan Imanberdiyev and Erdal Kayacan, “Redundancy Resolution based Trajectory Generation for Dual-Arm Aerial Manipulators via Online Model Predictive Control”, *IECON 2020 - 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020 (Accepted).

The contributions of the co-authors are as follows:

- Since I am the only author besides my supervisor, I have conducted the whole research. I have developed and tested the proposed trajectory generation with redundancy resolution strategy both in simulations and in real flights. I have also prepared the manuscript drafts.
- Dr. Erdal Kayacan has provided supervision and editorial comments while writing the paper.

12/09/2020

.....

Date



.....

NURSULTAN IMANBERDIYEV

Acknowledgements

The past years have been one of the craziest, toughest, and at the same time, one of the most wonderful years of my life, and therefore I would like to thank all the amazing people that made this journey possible. First of all, I want to thank my parents Zheñis and Klara, and my sister Azhara, for their unconditional love and encouragement during my entire life. Besides, I wish to express my deepest gratitude to my supervisors, Prof. Erdal Kayacan and Prof. Pham Quang Cuong, for their time, effort and guidance during my doctoral study. This thesis would not be possible without their constant support.

Last but not least, I would like to thank the people below and all those who have inspired and assisted me during my time at NTU. Thanks a lot!



Dedicated to my parents and sister

Abstract

This thesis focuses on a novel aerial robot system composed of an unmanned aerial vehicle (UAV) and one or more robotic arms, which is employed for aerial manipulation tasks. In particular, this research tries to transform UAVs from a predominantly data gathering platform into a flying robot, frequently called as unmanned aerial manipulators (UAMs), that can physically interact with the object or surrounding environment. This is challenging due to the existing design limitations imposed by the attachment of robotic arms to floating platforms. Furthermore, adding on to the cross-coupled dynamics of the UAV platform and arms, the presence of strong nonlinearities, external disturbances (e.g., wind), and parameter variations further complicates the deployment of UAMs in day-to-day life. Moreover, having such redundant aerial robots, especially when employing two manipulators, imposes additional challenges to successfully perform UAM missions. Motivated by these challenges, the main contributions of this thesis are threefold.

Firstly, we propose a novel design of the dual-arm manipulator specifically developed for UAM missions. The dual-arm system is designed such that it can be attached to different multirotors with minimum design modifications. In addition, the proposed dual-arm design employs prismatic joints to introduce three distinctive features: 1) ability of each arm to dynamically adjust its COG for better flight performance; 2) extended workspace and reach of the arms for enhancing operational capability and improving safety during UAM missions; 3) fully independent control of each arm for performing different tasks simultaneously.

Secondly, we present a learning-based intelligent control approach, the fusion of artificial neural networks and type-2 fuzzy logic controllers, namely type-2 fuzzy neural networks, for the control of UAMs under time-varying working conditions. The proposed control strategy eliminates the need for precise tuning of conventional controllers, and it can compensate for the internal and external disturbances caused by the motion of the arms and unforeseen environmental changes, opening the door for the widespread use of UAMs in daily life.

Thirdly, we propose two different trajectory generation with redundancy resolution strategies for UAMs. The first strategy is based on a weighted damped least-squares method, while the second approach employs a nonlinear model predictive control-based technique. The proposed approaches are capable to exploit the system redundancy by modifying the UAM configuration and simultaneously generate the feasible trajectories during the execution of the assigned tasks while enforcing the system constraints.

The contributions mentioned above are evaluated throughout extensive simulation studies and experimental flight tests with real UAM platforms. Overall, these contributions aim for extending the autonomous functionalities and operational capabilities of UAMs in the transportation and assembly type of scenarios, with the ambition to assist lessening the current gaps that hinder the widespread deployment of aerial robotic workers in real-world applications.

Contents

Acknowledgements	xi
Abstract	xv
List of Figures	xxi
List of Tables	xxv
List of Symbols	xxvii
List of Acronyms	xxix
1 Introduction	1
1.1 Motivation and Objectives	1
1.2 Contributions of the Thesis	4
1.3 Thesis Outline	6
2 Literature Review	9
2.1 Overview of the Current Development in Aerial Manipulation	9
2.1.1 Contact Inspection	10
2.1.2 Slung Load Transportation	12
2.1.3 Aerial Single-DOF Grippers	12
2.1.4 Aerial Multi-DOF Manipulators	13
2.1.5 Discussions and Research Gaps	16
2.2 Overview of Control Approaches	17
2.2.1 Conventional Approaches	18
2.2.2 Contemporary Approaches	18
2.2.3 Intelligent Control Approaches	21
2.2.4 Discussions and Research Gaps	23
2.3 Overview of Guidance Approaches	24
2.3.1 Guidance Approaches for UAMs	24
2.3.2 Discussions and Research Gaps	25
2.4 Overview of State Estimation	26

3	Design and Manufacturing of Unmanned Aerial Manipulators	29
3.1	Introduction	29
3.2	Design Requirements and Considerations	31
3.2.1	Weight, Inertia, and Reach Considerations of the Robotic arms	32
3.2.2	Motors and Materials	33
3.2.3	Kinematic Configuration of the Robotic Arms	34
3.2.4	Compensation of the COG Displacement	35
3.3	Dual-Arm Aerial Manipulator	38
3.3.1	Developed Dual-Arm System	38
3.3.2	Hardware/Software Architecture	44
3.3.3	Incorporating with Aerial Platform	45
3.3.4	Kinematic Parameters	46
3.4	Single-Arm Aerial Manipulator	49
3.4.1	Developed Single-Arm System	50
3.4.2	Experimental Setup	51
3.5	Experimental Validation	53
3.5.1	Experimental Results	54
3.6	Summary	61
4	Learning Control Strategy for Unmanned Aerial Manipulators	65
4.1	Introduction	65
4.2	Fuzzy-Neuro Control Approach	66
4.2.1	Control Architecture	66
4.2.2	Fuzzy Neural Network Structure	68
4.2.3	Sliding Mode Control Theory-Based Learning Algorithm	72
4.3	Simulation Studies	75
4.3.1	Intrinsic Parameters of the System and Control Variables	76
4.3.2	Simulation Results	79
4.4	Experimental Tests	83
4.4.1	Experimental Results	85
4.5	Summary	89
5	Trajectory Generation-based Guidance Approach for Unmanned Aerial Manipulators	91
5.1	Introduction	91
5.2	UAM State	93
5.3	Multi-task Velocity-based Redundancy Resolution Framework	95
5.3.1	General Approach	95
5.3.2	UAM Tasks	96
5.3.3	Simulation Studies	102
5.4	NMPC-based Trajectory Generation	106
5.4.1	General Formulation	107

5.4.2	UAM Tasks and Constraints	107
5.4.3	Simulation and Experimental Studies	112
5.5	Summary	122
6	Conclusions	125
6.1	Conclusions	125
6.1.1	Limitations	128
6.2	Future Work	128
A	Derivation of the UAM model	131
A.1	Dynamic model of UAM in 2D	131
A.2	Dynamic model of UAM in 3D	134
	Video Links	137
	List of Publications	139
	Bibliography	141

List of Figures

1.1	Examples of UAV applications.	2
1.2	Examples of UAM applications.	3
2.1	Examples of contact inspection with aerial robots.	11
2.2	Examples of the slung load transportation.	12
2.3	Examples of UAV(s) carrying a payload via a gripper.	13
2.4	Examples of UAMs: coaxial quadrotors equipped with commercially available 7-DOF manipulator [56] (right) and specifically designed 6-DOF robotic arm [24] (left).	14
2.5	Examples of dual-arm setups for aerial manipulation.	15
3.1	Three distinctive features of the proposed design: ability to dynamically adjust its COG (left), extended workspace and reach of the arms (middle), <i>decoupled</i> configuration of the arms (right).	30
3.2	Developed dual-arm manipulator mounted on a hexarotor platform: the shoulders of dual-arm in the <i>coupled</i> (left) and <i>decoupled</i> (right) configurations.	31
3.3	Examples of material property charts (Ashby plots): Strength - Density (left) and Young's modulus - Relative cost per unit volume (right) [129].	34
3.4	Displacement and dynamic alignment of the dual-arm COG.	36
3.5	3D model of the developed dual-arm system.	38
3.6	FEM analysis of the main frame parts.	41
3.7	Workspace of the developed dual-arm manipulator. Due to the propeller constraints, workspace only covers the region under the dual-arm base.	42
3.8	Workspace of the developed dual-arm manipulator beyond the area delimited by the propellers.	43
3.9	Architecture and components of the proposed dual-arm system.	44
3.10	Developed dual-arm system mounted on the hexarotor platform.	45
3.11	Sequence of images depicting the state change of the robotic arms from compact configuration (left) to the operational state (right). Landing gear down (left) and landing gear up (middle and right).	46
3.12	Kinematic parameters and reference frames for the dual-arm system.	47
3.13	Develop single-arm aerial manipulator.	50
3.14	Architecture and components of the developed single-arm system.	52

3.15	Single-arm aerial manipulator.	52
3.16	Block diagram of the control architecture.	54
3.17	Sequence of images during the indoor experiments corresponding to the first scenario. Arms in the operational state (1), performing different motion trajectories (2-8).	55
3.18	Experimental results for the first scenario.	56
3.19	Sequence of images during the experiments corresponding to the second scenario.	57
3.20	Experimental results for the second scenario.	58
3.21	Sequence of images during the experiments corresponding to the third scenario.	59
3.22	Experimental results for the third scenario.	60
3.23	Developed dual-arm system during execution of the eight-shaped trajectory by the end-effectors in the fourth scenario. Notice that reflective markers are attached to the end-effectors and C_b frame to provide visual feedback to the OptiTrack system.	61
3.24	Experimental results for the fourth scenario.	62
4.1	Block diagram of the control scheme.	67
4.2	Structure of the PD+FNN block.	68
4.3	Structure of T2FNN with two inputs.	69
4.4	Shapes of elliptic type-2 MFs with different values for a_1 and a_2	71
4.5	A sequence of snapshots during the unfolding of the robotic arm: (1) joint angles at $\{0, 0, 0, 0\}$ radians; (2)-(3) intermediate state; (4) joint angles at $\{0, -\pi, \pi, 0\}$ radians.	75
4.6	Performance of the proposed controller in the absence of wind, where the grey shaded regions represent the time periods during the unfolding and folding the arm.	79
4.7	Performance of the proposed controller in the presence of wind (for $\sigma_v = 0$ m/s), where the grey shaded regions represent the time periods during the unfolding and folding the arm.	81
4.8	Performance of the proposed controller in the presence of wind.	82
4.9	Indoor experimental setup.	83
4.10	Real-time implementation of the overall control scheme.	84
4.11	Joint angles of the robotic arm during the trajectory tracking flight.	85
4.12	Experimental results of the proposed controller for the hover case, where the grey shaded regions represent the time periods during the unfolding and folding the arm.	86
4.13	Experimental results of the proposed controller for the trajectory tracking case.	87
4.14	Euclidean RMSE for different controllers for the hover and trajectory tracking case.	88
5.1	UAM state and reference frames.	94
5.2	Constraint stiffness profile example for $g_e(t)$	101

5.3	Position and yaw angle of the UAV platform during scenario 1. The gray shaded regions represent the time period when the OA task is active. Notice how the y position (blue line) is increased to satisfy the OA task.	102
5.4	Arm joint values during scenario 1. The gray shaded regions represent the time period when the OA task is active. The cyan and magenta horizontal dashed lines indicate the start of influence region and joint limits, respectively.	103
5.5	Distance to the obstacle and normalized manipulability measure evolution during scenario 1. The gray shaded region represents the time period when the OA task is active in scenario 1.	104
5.6	Distance to the obstacle and normalized manipulability measure evolution during scenario 2. The gray shaded region represents the time period when the OA task is active in scenario 2.	105
5.7	Trajectory of the end-effector while implementing the trajectory relaxation in scenario 2. The gray regions represent the activation of the OA task in scenario 2.	105
5.8	A sequence of snapshots of the UAM motion in scenario 2: (1) UAM starts its motion; (2)-(3) intermediate states; (4) UAM reaches the goal.	106
5.9	Graphical representation of the aerial robot enclosed by 5 ellipsoids, E_l , with centers at \mathbf{c}_l , $l = 1, \dots, 5$	111
5.10	Block diagram of the dual-arm aerial manipulator architecture for the experimental flight tests.	114
5.11	A sequence of images of the dual-arm aerial manipulator motion during the simulations in scenario 1. The orange box represents the obstacle, which is enclosed by white ellipsoid.	114
5.12	Reference and executed trajectories of the left (left column) and right (right column) end-effectors in scenario 1.	115
5.13	Position of the UAV platform and dual-arm end-effectors during the simulations in scenario 1. Notice that the end-effectors modified their trajectories to avoid the obstacle.	115
5.14	Joint angles of the dual-arm during the simulations in scenario 1. The corresponding upper and lower bounds of the joint limit constraints are represented as solid magenta lines. Since the values of ψ and $q_2^{1,2}$ are very small and close to zero, they are not shown in this figure.	116
5.15	Constraint variables during the simulations in scenario 1: singularity-free motion (top left), safety distances between the UAV base and dual-arm end-effectors (top right), and value of ν related to the avoidance of obstacle (bottom). The corresponding bounds are represented as solid magenta lines.	116
5.16	A sequence of images of the dual-arm aerial manipulator motion during simulations in scenario 2. The orange boxes represent the obstacles, which are enclosed by white ellipsoids.	117

5.17	Reference and executed trajectories of the left (left column) and right (right column) end-effectors in scenario 2	117
5.18	Position of the UAV platform and dual-arm end-effectors during the simulations in scenario 2	118
5.19	Joint angles of the dual-arm during the simulations in scenario 2. The corresponding upper and lower bounds of the joint limit constraints are represented as solid magenta lines.	118
5.20	Constraint variables during simulations in scenario 2: singularity-free motion (top left), safety distances between the end-effectors and UAV base (top middle), virtual wall between the arms (top right), and value of ν related to the avoidance of the first obstacle (middle) and the second obstacle (bottom). The corresponding bounds are represented as solid magenta lines.	119
5.21	A sequence of images of the dual-arm aerial manipulator motion during the experimental flight tests in scenario 1. The metal frame represents the obstacle, which is also enclosed by imaginary ellipsoid. 0.35 m long bar is attached to the end-effectors.	120
5.22	Reference and executed trajectories of the left (left column) and right (right column) end-effectors in scenario 1.	120
5.23	Position of the UAV platform and dual-arm end-effectors during the experimental flight tests in scenario 1. Notice that the end-effectors modified their trajectories to avoid the obstacle.	121
5.24	Joint angles of the dual-arm during the experimental flight tests in scenario 1. The corresponding upper and lower bounds of the joint limit constraints are represented as solid magenta lines.	121
5.25	Constraint variables during the experimental flight tests in scenario 1: singularity-free motion (top left), safety distances between the UAV base and dual-arm end-effectors (top right), and value of ν related to the avoidance of obstacle (bottom). The corresponding bounds are represented as solid magenta lines.	122
A.1	Planar UAM system with relevant frames.	132
A.2	Dual-arm aerial manipulator with relevant frames.	134

List of Tables

2.1	Examples of aerial manipulation applications performed by conventional methods.	19
2.2	Examples of aerial manipulation applications performed by contemporary methods.	20
2.3	Examples of sensing systems for the state estimation of UAM (adopted from [127]).	26
3.1	Comparison of different dual-arm manipulators.	30
3.2	Main parameters of the arm joints.	39
3.3	DH parameters of the arm.	48
3.4	Parameters of the manipulator motors.	51
3.5	Euclidean RMSE for the first scenario (unit: cm).	55
3.6	Euclidean RMSE for the second and third scenario (unit: cm). . . .	58
4.1	Coaxial tricopter intrinsic parameters.	77
4.2	Moments of inertia of each link of the robotic arm.	78
4.3	Euclidean RMSE for different controllers in the absence of wind (unit: cm).	80
4.4	Euclidean RMSE for different controllers in the presence of wind. .	82
4.5	Euclidean RMSE for different controllers (unit: cm).	88

List of Symbols

Here I list some important symbols used in this thesis:

\mathbf{r}_b	position of UAV in world frame
Φ_b	Euler angles of UAV
ϕ, θ, ψ	roll, pitch and yaw angles
\mathbf{q}^*, Θ	joint position vector of the arm
m_*	mass
\mathbf{I}_*	moment of inertia
\mathbf{B}^*	mapping matrix from the joint space to the motor space
C_W	world-fixed reference frame
C_b	body-fixed reference frame
\mathbf{R}_b	rotation matrix describing the orientation of C_b relative to C_W
$\det(*)$	determinant of $*$
\mathbf{e}_p	position error
\mathbf{v}	linear velocity vector
F_z	total thrust force
$\tau_\phi, \tau_\theta, \tau_\psi$	roll, pitch and yaw torques in body frame
u_c	PD control signal
u_f	FNN control signal
\widetilde{M}	type-2 fuzzy sets
$\overline{W}, \underline{W}$	firing strength
$\overline{\mu}, \underline{\mu}$	upper and lower MFs
ζ	configuration state of the aerial manipulator
\mathbf{r}_e	position of the end-effector in world frame
\mathbf{J}_*	Jacobian matrix
C	cost function
$\mathbb{I}_{a \times b}$	$(a \times b)$ identity matrix

$\mathbb{O}_{a \times b}$	$(a \times b)$ zero matrix
β	slack variable
Γ_*	cost function corresponding to a task
$\mathbf{Q}_*, \mathbf{W}_*$	weight matrices
\mathcal{L}	Lagrangian
\mathcal{K}	kinetic energy
\mathcal{U}	potential energy
$\mathbf{r}_{L_i}, \mathbf{r}_{L_i}^b$	position of the center of mass of the link i described in C_W and C_b
$\dot{\mathbf{r}}_{L_i}, \boldsymbol{\omega}_{L_i}$	linear and angular velocity of the link i in C_W
$\boldsymbol{\omega}_b^b$	angular velocity of UAV in body frame
\mathbf{Q}	mapping matrix that relates $\dot{\boldsymbol{\Phi}}_b$ with $\boldsymbol{\omega}_b^b$
$\mathbf{R}_{L_i}^b$	rotation matrix between the link i and C_b
$\boldsymbol{\tau}_*$	joint torque vector

List of Acronyms

ABS	Acrylonitrile Butadiene Styrene
ACADO	Automatic Control and Dynamic Optimization
ANN	Artificial Neural Network
COG	Center Of Gravity
DH	Denavit-Hartenberg
DOF	Degree Of Freedom
FEL	Feedback-error-learning
FEM	Finite Element Method
FLC	Fuzzy Logic Control
FNN	Fuzzy Neural Network
GA	Genetic Algorithms
GD	Gradient Descent
GPS	Global Positioning System
IMU	Inertial Measurement Unit
JLA	Joint Limit Avoidance
LF	Linear Function
MF	Membership Function
MPC	Model Predictive Control
MRAC	Model Reference Adaptive Control
MVEE	Minimum Volume Enclosing Ellipsoid
NMPC	Nonlinear MPC
NSB	Null-Space-based Behavioral
OA	Obstacle Avoidance
PD	Proportional-Derivative
PETG	Polyethylene Terephthalate Glycol
PID	Proportional-Integral-Derivative
PLA	Polylactic Acid

RMSE	Root Mean-Square Error
ROS	Robot Operating System
RRT	Rapidly exploring Random Tree
SITL	Software-In-The-Loop
SMC	Sliding Mode Control
T1FNN	Type-1 FNN
T2FNN	Type-2 FNN
TSK	Takagi-Sugeno-Kang
UAM	Unmanned Aerial Manipulator
UAV	Unmanned Aerial Vehicle
UWB	Ultra-WideBand
VTOL	Vertical Take-Off and Landing

Chapter 1

Introduction

1.1 Motivation and Objectives

In recent years, many exciting developments in the area of unmanned aerial vehicles (UAVs) have appeared, resulting in the widespread usage of UAVs in day-to-day life. They have been utilized in performing a wide variety of tasks in both military and civilian applications such as search and rescue [1], surveillance [2], and damage evaluation after natural disasters [3] as shown in Fig. 1.1. In particular, vertical take-off and landing (VTOL) UAVs have been the main interest of active research among the class of versatile aerial robots due to their remarkable maneuverability, relatively low cost and small size as well as their ability to hover and work in cluttered environments. However, previous research on UAVs has typically been focused on the applications where UAVs act as *flying cameras* without any direct physical interaction or manipulation with the surrounding environment. This trend has recently changed after advancement in electronics and materials, which made VTOL UAVs the promising candidate to perform aerial manipulation tasks leading to the paradigm shift from *flying camera* to *flying hand* concept [4, 5]. Unlike ground vehicles, aerial vehicles equipped with dexterous robotic arm(s) can clearly expand the applicability of mobile manipulation to the three-dimensional space and open the door to new applications such as construction or assembly [6], inspection and maintenance of bridges [7], industrial plants [8] and the regions out of the reach for the ground-based systems [9–11] as shown in Fig. 1.2. In this work, such aerial manipulations systems are referred to as unmanned aerial manipulators (UAMs).



FIGURE 1.1: Examples of UAV applications.

The successful implementation of the above missions is usually affected by the dexterity of arms integrated into UAV platforms. Thus, many research groups have worked on the development of manipulators ranging from a single degree of freedom (DOF) gripper [15] to multi-DOFs robotic arm [16–20]. The aforementioned works mostly concentrate on the development of aerial robots with a single manipulator. However, integrating a dual-arm manipulator with aerial vehicles further extends the capability and range of tasks of the aerial manipulation systems compared to the single-arm case. Specifically, the dual-arm manipulator allows performing certain tasks such as simultaneous manipulation with two objects, grasping bulky items and some actions that cannot be done with a single-arm, such as opening a valve in an offshore oil industrial plant. Nevertheless, only a few works have considered the design and development of the dual-arm manipulators specifically designed for aerial manipulation missions with UAV platforms [21–23]. Therefore, currently, the development of the dual-arm aerial manipulators is quite an active research area with several unresolved issues. In particular, in the existing designs of the dual-arm aerial manipulators, the shoulder joints of two arms are fixed and coupled to each other. This considerably decreases the dexterity and the range of tasks done by the dual-arm aerial manipulators. In fact, if the shoulders of the dual-arm manipulator are separated from each other, then fully independent control of each arm can be achieved to perform different tasks simultaneously. In addition, in the current designs, the dual-arm manipulator does not have the ability to dynamically align its center of gravity (COG). It is known that such ability can lead to better flight performance during the transportation and assembly type of scenarios [24]. Therefore, the first objective of this thesis is the development of a novel dual-arm manipulator specifically designed for aerial manipulation missions, which addresses the aforementioned abilities.

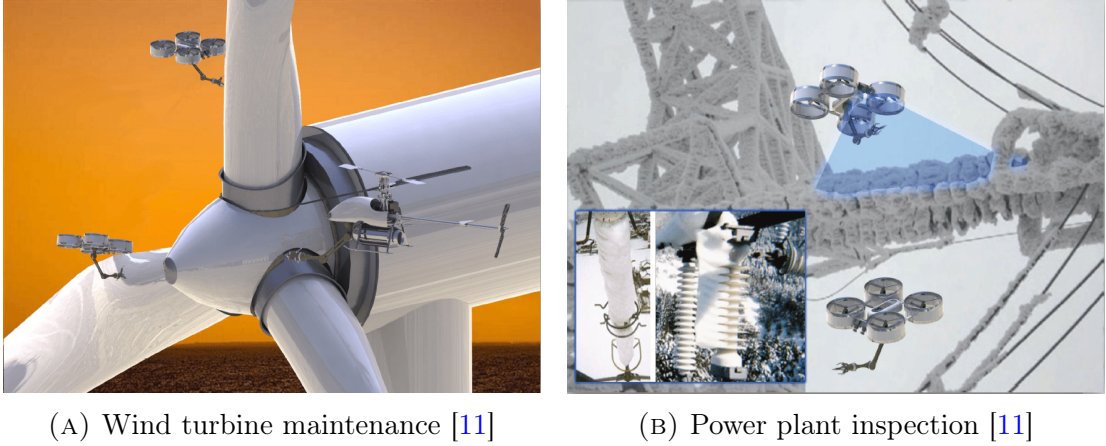


FIGURE 1.2: Examples of UAM applications.

The control of aerial robots performing the complex interaction and manipulation tasks with the environment is also quite challenging because UAVs themselves are inherently unstable, and their dynamics are highly nonlinear and underactuated [25]. Thus, the ability of UAVs to directly manipulate or interact with the external environment further increases the complexity of the controller design. In addition, the coupled dynamics of the UAV platform and the interaction tool, especially when employing the robotic arm(s), is another essential factor that should be carefully considered during the design of the controller. Furthermore, such aerial vehicles need to tolerate the reaction forces from the interaction with the objects or surrounding environment. Moreover, the case of having unforeseen working conditions (e.g., external disturbances like wind gusts), lack of modeling (i.e., internal uncertainties), and the parameter variations caused by adding/releasing the payload impose additional difficulties in the design of overall control architecture. Therefore, it is crucial to design efficient, robust and advanced control methods that can address these issues and improve the control performance and stability during the flight.

In this regard, the existing control approaches can be categorized into three major groups: (1) linear flight control methods, (2) model-based nonlinear approaches, and (3) model-free or intelligent control techniques [26]. Although the first two control approaches can provide a decent balance between control performance, implementation cost and operational complexities, linear control methods are usually restricted by operating regions where non-linearity is negligible, while in the case of model-based approaches, the precision of the representative mathematical models is important. Moreover, they can be quite sensitive to external disturbances and

modeling errors. Therefore, the performance of these methods tends to deteriorate when they are operating under uncertain working conditions. On the other hand, model-free methods have caught proper attention from the research community since they do not require a precise model of the system to be controlled. They have also yielded promising results to deal with uncertainties in the system [27]. Furthermore, they have become popular solutions to control the systems with non-trivial dynamics [28], which is the case for UAVs equipped with the robotic arm(s). In addition, over the last few years, intelligent control techniques endowed with learning capabilities have been developed to further improve the control performance [29]. Hence, the need for the learning-based intelligent control approaches to achieve the robust and stable control of UAM under uncertain working conditions is perceptible. Therefore, the second objective of this thesis is the development of the learning-based intelligent control strategy for UAMs that can compensate for both internal and external disturbances generated during time-varying working conditions.

Due to the structure of UAMs, they can provide a remarkable level of dexterity on performing complex tasks thanks to their high number of DOFs. Having such redundant aerial robots, especially when employing dual-arm aerial manipulators, imposes additional challenges to successfully perform UAM missions. In particular, exploiting the entire system redundancy and simultaneously generating the feasible trajectories for the UAV platform and robotic arm(s) are essential in performing aerial manipulation tasks and achieving dexterous behavior. In the literature, there are several proposed methods (e.g., [30]), which have addressed this issue. Nevertheless, this topic is still an active research area with a promising future. Therefore, the third objective of this thesis is the development of the trajectory generation with redundancy resolution strategy that can successfully guide UAMs by taking advantage of their redundancy.

1.2 Contributions of the Thesis

This thesis focuses on resolving the aforementioned challenges and achieving the objectives given in Section 1.1. The following contributions are realized and reflected in this work:

- **The development of the novel dual-arm manipulator for UAMs.** The design, fabrication, and experimental validation of a lightweight, low inertia dual-arm manipulator are proposed with the following distinctive features: (1) ability of each arm to dynamically adjust its COG for better flight performance; (2) extended workspace and reach of the arms for enhancing operational capability and improving safety during aerial manipulation missions; (3) fully independent control of each arm for performing different tasks simultaneously. The proposed design considerably improves the flight performance and operational capabilities of UAM.
- **The development of the learning-based intelligent control strategy for UAMs.** The artificial intelligence-based control approach, the fusion of artificial neural networks and type-2 fuzzy logic controllers, namely type-2 fuzzy neural networks, is developed for the control of UAMs under time-varying working conditions. The proposed strategy eliminates the need for precise tuning of conventional controllers, and it is capable to compensate the disturbances caused by the robotic arm motion and wind gust, and thus, opening the door for the wider implementation of aerial manipulation tasks in daily life. Due to the adaptive learning capabilities of type-2 fuzzy neural networks, the developed control approach can be employed by both single and dual-arm aerial manipulators.
- **The development of two trajectory generation with redundancy resolution strategies for UAMs.** Two different trajectory generation with redundancy resolution strategies are proposed for UAMs. The first strategy is based on the weighted damped least-squares method, and its performance is evaluated with the single-arm aerial manipulator. The second approach employs the nonlinear model predictive control-based technique, and it is tested with the dual-arm aerial manipulator. The proposed methods are capable of modifying the UAM configuration to perform the assigned tasks by exploiting the system redundancy and satisfying several constraints such as self-collision and obstacle avoidance.
- **The real-time implementations of the proposed contributions.** The extensive simulation studies and experimental flight tests are performed to validate the proposed approaches.

1.3 Thesis Outline

This thesis is composed of six chapters, and it is structured according to the objectives defined in Section 1.1. The motivation of the research, the thesis objectives, and the main contributions are described in Chapter 1. The remainder of this thesis is organized as follows:

- Chapter 2 consists of a comprehensive literature review on the current state-of-the-art in aerial manipulation. First, it starts with describing four major research developments in aerial manipulation, namely contact inspection, slung load transportation, aerial single-DOF grippers and multi-DOF manipulators. Second, by categorizing the current control methods into conventional, contemporary and intelligent ones, a rigorous review of the control approaches is presented. Then, a general overview of guidance approaches for UAMs is provided. Finally, a brief overview of state estimation for UAMs is discussed.
- Chapter 3 describes the design, fabrication, and experimental validation of the proposed novel dual-arm manipulator. This chapter discusses specific design consideration and requirements which are important to address during the design of UAMs. After that, the detailed explanation and advantages of the proposed design of the dual-arm manipulator are presented. The developed single-arm manipulator is also briefly described since it will be employed in the next chapters. Finally, extensive experimental studies are conducted to validate the performance of the proposed dual-arm design with a hexarotor platform equipped with a standard autopilot.
- Chapter 4 presents the adaptive fuzzy-neuro control approach for the control of UAMs. First, the structure of type-2 fuzzy neural networks and sliding mode control theory-based learning algorithm are thoroughly discussed. Then, extensive simulations and experimental flight tests are performed to evaluate and verify the performance of the proposed controller under time-varying working conditions. Finally, the performance comparison of the developed controller with the conventional controllers and with its type-1 counterpart is presented to further validate the proposed approach.

- Chapter 5 discusses two different trajectory generation with redundancy resolution strategies specially developed for UAMs. This chapter provides a description of the UAM states for the trajectory generation strategies. As the first strategy, the multi-task velocity-based redundancy resolution framework is described. It is then followed by simulation studies that evaluate and verify the performance of this framework with the single-arm aerial manipulator. As the second strategy, the nonlinear model predictive control-based technique is presented, and it is tested with the dual-arm aerial manipulator both in simulations and in real flights.
- Chapter 6 concludes the thesis by summarizing and discussing the main contributions, the remaining challenges, and future research directions arising from this work.

In addition, the derivation of the UAM model is reported in Appendix A.

Chapter 2

Literature Review

This chapter is organized as follows. In Section 2.1, the literature review and current development in aerial manipulation are presented. By categorizing the current control methods into conventional, contemporary and intelligent ones, a comprehensive review of the control approaches is given in Section 2.2. A general overview of guidance approaches for UAMs is presented in Section 2.3. It should be noted that discussions and research gaps are provided at the end of each mentioned section. Finally, a brief overview of state estimation for UAMs is discussed in Section 2.4.

2.1 Overview of the Current Development in Aerial Manipulation

Over the past decades, mobile manipulation of the robotic systems has become highly active field of research and had tremendous significance for the development of military and civilian applications due to their capability to physically interact with the surrounding environment. Historically, the focus of mobile manipulation was concentrated on developing ground vehicles due to the essential stability of ground platforms during the manipulation tasks. However, the ground vehicles are limited to move in particular terrains and environments as well as they are constrained to interact with objects located in sites accessible only from the ground. In

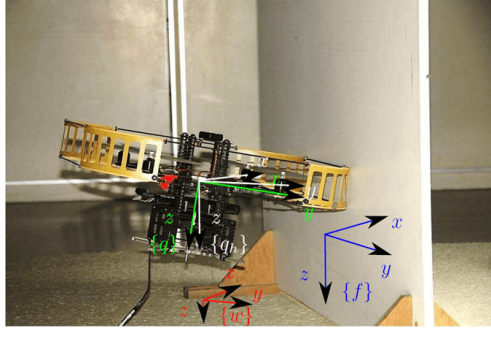
this regard, aerial vehicles are considered as a potential approach to overcome these limitations by accessing the sites that are inaccessible to ground-based systems.

There are two broad classifications of aerial vehicles, namely rotary-wing and fixed-wing UAVs. The former one has the crucial features such as VTOL capability, remarkable maneuverability, low cost, and small size, which allow the rotary-wing UAVs to be used in various civilian applications. However, previous research on rotary-wing UAVs has typically been concentrated on surveillance and monitoring applications where UAVs act as *flying camera* without any direct physical interaction with the surrounding environment. This trend has changed after advances in electronics and materials, which made the rotary-wing UAVs a candidate to perform aerial manipulation in hard-to-reach and dangerous regions. In particular, recent EU funded projects such as AIRobot [31], ARCAS [9], AEROBI [32], AEROWORKS [11] and AEROARMS [10] are examples of the fast-growing research areas in aerial manipulation, which have the goal to develop aerial robots with enhanced manipulation abilities for autonomous inspection and maintenance operations. In general, the current state-of-the-art in aerial manipulation has focused mostly on the following areas:

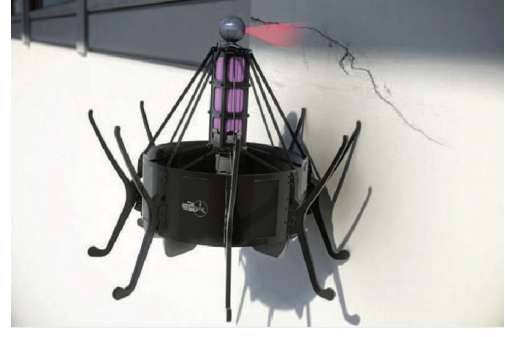
- Contact inspection;
- Slung load transportation;
- Aerial single-DOF grippers;
- Aerial multi-DOF manipulators.

2.1.1 Contact Inspection

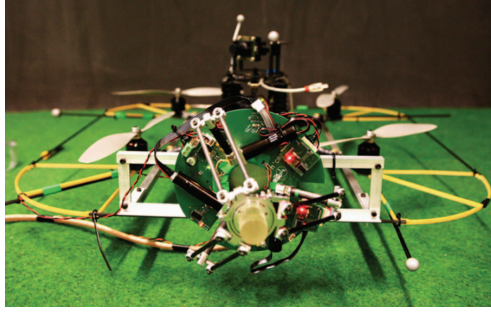
In the literature, there are different forms of aerial manipulation. One of the approaches is the inspection by contact. In recent years, several research groups have investigated the usage of UAVs in contact inspections (see Fig. 2.1). The main issue during contact inspection of UAVs is achieving a stable motion while remaining in contact with the environment. In order to address this issue, the wrench (i.e., force/torque) control is introduced in [33]. In this approach, a proposed hybrid pose/wrench framework can switch between pure pose and pose/wrench control, although the switching is done by a human operator. Besides, to avoid utilizing



(A) Quadrotor executing pose/wrench control [33]



(B) Ducted fan inspecting a structure by contact (the AIRobot prototype) [34]



(C) Quadrotor equipped with a mechanism based on the delta structure and Cardan gimbal [38]



(D) Multi-directional thrust octorotor with the inspection tool [10]

FIGURE 2.1: Examples of contact inspection with aerial robots.

additional force and torque sensors, the wrench estimator is implemented by using pose measurements and inputs of UAV. In [34], a state feedback based hybrid force/position control concept is proposed to accomplish a wall inspection task by employing a ducted fan system, while in [35, 36], a typical quadrotor is utilized to achieve the physical interaction with non-planar surfaces by using prediction control paradigm. Another approach in contact inspection is presented in [37, 38], where the authors equip the UAV platform with a mechanism based on the delta structure and Cardan gimbal as shown in Fig. 2.1c, while in [10], the multi-directional thrust octorotor with the inspection tool (see Fig. 2.1d) is developed for the contact-based inspections. However, in general, the aforementioned works are limited to use only in contact inspection without performing complex interaction tasks such as actual grasping, assembling objects, and removing obstacles. Furthermore, in order to achieve stable motion during contact inspection, UAVs are constrained to perform non-agile movements which may limit the usage of aerial robots.



(A) Illustration of a quadrotor carrying a (B) Quadrotor with suspended load performing a swing-free trajectory tracking [41]

FIGURE 2.2: Examples of the slung load transportation.

2.1.2 Slung Load Transportation

In order to maintain the agility of UAVs during the transportation of external objects, another approach in aerial manipulation is to transport the object by suspending it via a cable (see Fig. 2.2). Considerable research is conducted to improve the performance of cable-suspended load transportation. For example, the problem of lifting the object with unknown mass, which is suspended through the cable, is investigated in [39], while in [40], its experimental validation is presented. In [41] and [42], the trajectory optimization techniques are employed to suppress the movement of the suspended load, while in [43] and [44], the swing excitation is reduced by utilizing adaptive controllers. In addition, the problem of collaborative manipulation and transportation of the object through cables is studied in [45–48]. Nevertheless, aerial manipulation via cable suspension is still inefficient and limited, in the sense that the direct regulation of the motion of the suspended load cannot be usually achieved due to the nature of cable which cannot fully drive the motion of the payload as desired.

2.1.3 Aerial Single-DOF Grippers

Another approach is inspired from aerial hunting by birds of prey, where a gripper is installed at the bottom of the UAV platform to grasp and hold the object [15, 49–51] as shown in Fig. 2.3. Especially in [51], the problem of high-speed grasping and transportation of objects is investigated to imitate the aerial hunting by raptors.



(A) Yale Aerial Manipulator grasping a 1.5 kg object in hover [52] (B) Four quadrotors lifting a payload with a non-trivial shape [53]

FIGURE 2.3: Examples of UAV(s) carrying a payload via a gripper.

The analysis of hover control, load stability, and grasping by utilizing single DOF gripper, which is fixed to a helicopter UAV, is described in [15]. In that paper, the relationship between the stability of UAV, added weight of the payload and the amount of offset from the COG of UAV is determined. This relationship shows that the available offset region decreases for the payloads with the large mass. In addition, the authors show that for certain offsets and weights the perturbation from grasped payload can be coped with the classical proportional-integral-derivative (PID) flight controller [52]. In [50], the stability of quadrotor UAV in the presence of load with unknown mass is studied. In particular, the algorithm is developed to improve trajectory tracking performance by estimating unknown payload parameters such as inertia and mass of the payload as well as the location of its center of mass. In addition, a group of quadcopter UAVs with a similar gripper is used for collaboratively picking up a relatively large load in [53]. Furthermore, the experimental results of a team of quadrotors accomplishing automated construction of cubic type structures are presented in [54]. On the other side, there exists a considerable limitation. Specifically, the available grasping range is usually constrained due to the fixed configuration of the gripper related to the UAV blades and body. Therefore, dexterous manipulation and complex interaction with the environment are challenging to achieve by using this approach.

2.1.4 Aerial Multi-DOF Manipulators

The above limitation can be surmounted by introducing a multi-DOF robotic arm or manipulator which can dynamically interact with the environment as a result of actuated joints. In this way, the efficiency of mobile manipulation is increased

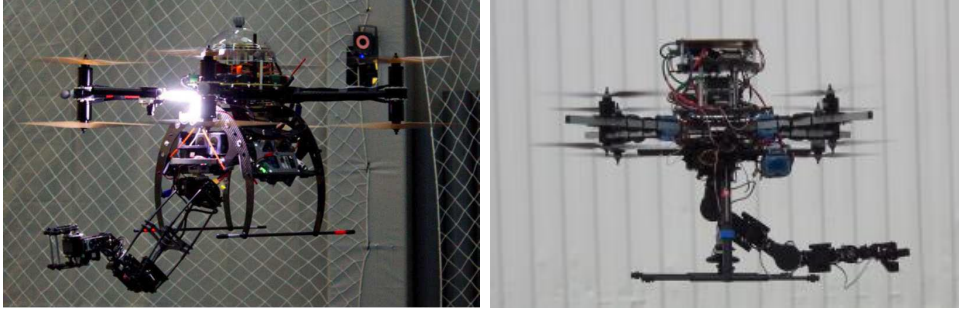


FIGURE 2.4: Examples of UAMs: coaxial quadrotors equipped with commercially available 7-DOF manipulator [56] (right) and specifically designed 6-DOF robotic arm [24] (left).

by integrating the flexibility of the robotic manipulator and the maneuverability of UAVs. In addition, the presence of multiple joints can compensate for the positioning error and oscillations that usually occur during hovering of the UAV platform, which, in its turn, can keep the end-effector on the target point. For instance, the test platform consisting of two 4-DOF robotic manipulators and 6-DOF gantry crane simulating the motion of UAV is presented in [55]. In general, two main approaches in the development of multi-DOF aerial manipulators can be found in the literature, namely utilizing commercially available manipulators [56–58] and designing robotic arms explicitly developed for aerial vehicles [16–20]. Many research groups have worked on the development of the latter approach to satisfy different design requirements imposed by aerial manipulation applications. For instance, the mechanical design of the 6-DOF robotic arm is described in [16], which includes a mechanism to compensate for the COG displacement by moving the battery [24]. However, since the motors are directly integrated into the arm joints, the variation of the inertia terms during the arm motion can negatively affect the UAV stability. In order to decrease this variation, in [17] and [18], the authors propose the arm design where its actuators are placed near the UAV platform by employing a timing belt-based transmission mechanism. Nevertheless, in [17] and [18], the COG compensation mechanism is not considered. Therefore, when there is a grasped object, the COG displacement caused by the arm motion can adversely affect the UAV stability. In addition to serial manipulators, the 6-DOF and 3-DOF parallel manipulators are developed in [19] and [20], respectively. However, parallel manipulators have usually a smaller workspace volume [59], which can limit the application range of the aerial manipulation systems. Figure 2.4 shows examples of



(A) Prototype of the commercial dual-arm setup (Prodrone PD6B-AW-ARM) [21] (B) Research prototype of the dual-arm system incorporated in the DJI Matrice 600 [22]

FIGURE 2.5: Examples of dual-arm setups for aerial manipulation.

UAMs where coaxial quadrotors are equipped with commercially available 7-DOF manipulator [56] and specifically designed 6-DOF robotic arm [24].

The aforementioned works mostly concentrate on the development of aerial robots with a single manipulator. However, integrating the dual-arm manipulator with aerial vehicles further extends the capability and range of tasks of the aerial manipulation systems compared to the single-arm case. Specifically, the dual-arm manipulator allows to perform certain tasks such as simultaneous manipulation with two objects, grasping bulky items and also some actions that cannot be done with a single-arm, such as opening a valve in an offshore oil industrial plant. There are several successful applications of dual-arm for aerial manipulation [21–23, 60]. Particularly in [60], the dual-arm aerial manipulator is used for the valve-turning task during the flight, while in [21], the first commercial implementation of the dual-arm manipulator with UAV is presented for the transportation/grasping purposes. In [22], two 5-DOF human-size arms are developed for aerial manipulation tasks, whereas in [23], mechanical joint compliance for the dual-arm case is addressed. In both works, the estimation of the arms torque is considered to mitigate the influence of the motion of the arms. However, the torque estimation requires a precise dynamic model of the system, which is tedious to obtain. In [61], the dual-arm setup developed in [22] is mounted at the tip of a long flexible link instead of directly attaching it under the UAV platform to increase the reach of the dual-arm manipulator. Nevertheless, it can be noted that the oscillation of the flexible link requires special consideration, which, in its turn, complicates the overall control problem. In addition, mounting the dual-arm at the tip of the long flexible link can hinder the deployment of dual-arm aerial manipulators in confined environments.

It is noteworthy that apart from the aforementioned designs, there are also research works in the development of hybrid aerial vehicle (e.g., HYFLIERS [62]), which can fly and move through another medium, and re-configurable aerial platforms, which can behave as a gripper themselves by transforming their configuration [63, 64]. However, in the former case, employing hybrid robots in aerial manipulation may lead to the reduction of the payload and overall flight time due to the increased mass and power consumption caused by the additional mechanisms in design, which are needed to travel in various mediums. As for the re-configurable aerial platforms, they are usually application-dependent. For more examples in aerial manipulation, interested readers can refer to the recent survey papers [5, 65–67].

2.1.5 Discussions and Research Gaps

The review on the interaction modalities reveals that it is preferred to equip the UAV platforms with the multi-DOF robotic arm(s) in order to achieve dexterous manipulation and complex interaction with the environment. Such systems have two possible configurations for the robotic arms. The first one is the employment of industrial manipulators. In this way, all the benefits coming from commercially available manipulators can be exploited during aerial manipulation missions. However, at the same time, such configuration may lead to the reduction of the final payload of the UAV platforms due to the excessive weight and overall flight time due to increased power consumption. The second configuration is employing robotic arms that are explicitly developed for aerial vehicles. In this way, specific design considerations and requirements for aerial manipulation can be addressed during the development stage. In this regard, it is decided to focus on designing the robotic arms instead of utilizing commercially available examples. In addition, as reviewed in the previous subsection, many research groups have worked on the development of the single-arm aerial manipulators, while only a few works have considered the design and development of the dual-arm aerial manipulators. Therefore, currently, the development of the dual-arm aerial manipulators is quite an active research area with several unresolved issues. In particular, the existing dual-arm aerial manipulators only rely on revolute joints in their designs. As a result, the workspace and reach of the arms are limited and cannot be modified after attaching them to the UAV platform. Another limitation in the current designs of the dual-arm aerial manipulators is that the shoulder joints of two arms are

fixed and coupled to each other. Consequently, fully independent control of each arm cannot be achieved to perform different tasks simultaneously at opposite sides of UAM. Furthermore, the dual-arm aerial manipulators found in the literature do not have the ability to dynamically align their COG. Having such ability can considerably enhance the flight stability and performance of UAMs. Therefore, in an attempt to close the possible research gaps, the first focus in this thesis is the development of the novel dual-arm manipulator designed explicitly for aerial manipulation missions with multirotor platforms that addresses the aforementioned unresolved issues. The proposed design is investigated in Chapter 3.

2.2 Overview of Control Approaches

The control of aerial robots performing the advanced manipulation and interaction tasks with the environment is quite challenging because multirotor platforms themselves are inherently unstable and have strong nonlinearities [25]. Furthermore, they are quite sensitive to parameter variations caused by adding/releasing the payload and the reaction forces from the interaction/manipulation with the object or surrounding environment. In addition, one should not forget about the external disturbances like wind gusts and the internal uncertainties like a lack of modeling. Also, the coupled dynamics of the UAV platform and the interaction tool is another essential factor that should be considered during the design of the controller. In general, there are two control strategies found in the literature that focus on alleviating the aforementioned issues, namely decentralized and centralized control strategies [5]. The former considers the UAV body and robotic manipulator(s) as separate modules and treats coupling between them as disturbances, while in the latter case, the UAV body and robotic manipulator(s) are considered as a combined system, and thus the complete dynamics of the system is taken into account during the controller design. By categorizing the above-mentioned control strategies, the current control methods are presented as conventional, contemporary and intelligent approaches, and they are discussed below.

2.2.1 Conventional Approaches

In this subsection, conventional approaches, which are mainly based on the classical PID and proportional-derivative (PD) controllers, are reviewed. Due to their well matured and examined structure, the PID-based controllers are the most common among the existed conventional approaches. The proportional k_p , integral k_i and derivative k_d terms are the PID controller parameters, which are used to calculate the control input by utilizing the error value between the reference and feedback signal.

As being a simple and practical controller, PID controllers have been used in various types of aerial manipulation tasks, which are summarized in Table 2.1. As can be seen from this table, there are examples when the PID/PD controllers are successfully implemented for the aerial manipulation systems. However, in the mentioned examples, the design of conventional controllers is mostly based on nominal conditions without explicitly considering external disturbances, coupling, and nonlinearities caused by an interaction tool. In addition, the performance of the PID/PD controllers is constrained by operating regions where nonlinearities are negligible, and thus internal uncertainties or external disturbances can violate the approximated linearity, leading to the destabilization of the overall system. Finally, it is evident that the conventional controllers can be tuned more aggressively to obtain better results in each specific scenario. Nevertheless, this may not be practical since aggressive tuning tends to be case dependent, and hence, it cannot give a comparable performance in different conditions. Therefore, advanced methods such as contemporary and intelligent approaches can be more preferable to achieve better flight performance.

2.2.2 Contemporary Approaches

Instead of using conventional control methods such as the PID controllers, more sophisticated control techniques can be employed to increase the overall stability of the system as well as to enhance the performance of the aerial manipulation tasks. Improved flight stability and performance can be obtained by using nonlinear control techniques [72]. For instance, a feedback linearization is considered as a good example of nonlinear control methods, which is capable to expand the flight envelope over linear counterparts [73]. However, in order to find the inverse dynamics

TABLE 2.1: Examples of aerial manipulation applications performed by conventional methods.

UAV platform & Ref	Interaction Tool	Purpose	Controller Type
Quadrotor [68]	Rigid arm	Push and pull the object	PD
Quadrotor [54]	Gripper	Construction	PID
Helicopter [15, 52]	Gripper	Grasp and move objects	PID
Quadrotor [69]	Two 4-DOF robotic arms	Grasp manipulation	PID
Tri-TiltRotor [70]	Rod	Push the object	PID
Quadrotor [48]	Cable	Collaborative transportation	PD
Quadrotor [71]	2-DOF robotic arm	Grasp manipulation	PD

of the system, it is assumed that the precise knowledge of the model is available, while it is not usually true for real-time applications. In addition, due to the differentiation in dynamic equations, the feedback linearization-based controllers are vulnerable to sensor noise and internal/external uncertainties [74]. Sliding mode control (SMC) [75, 76] is another example of nonlinear control methods which can overcome the mentioned problems due to its high robustness to uncertainties such as external disturbance, sensor noise, and modeling errors. On the other hand, due to the compensation of uncertainties, SMC theory-based methods have the inclination to induce large input gains, which, in its turn, can be dangerous for real-world applications. Moreover, SMC theory-based controllers suffer from a chattering phenomenon [77]. Nevertheless, adaptive [79, 80] and second-order [81] SMC techniques can be utilized to alleviate the above-mentioned issues. Similarly, advanced control methods like a model predictive control (MPC) [84, 85] can be utilized to improve the control performance during aerial manipulation tasks. However, in general, model-based control methods require precise knowledge of the system, which is tedious to obtain.

So far, in the aforementioned control methods, the use of force is generally not considered during the controller design. As a result, these methods may not be sufficient to handle undesired effects which are caused by the reaction forces. Therefore,

TABLE 2.2: Examples of aerial manipulation applications performed by contemporary methods.

UAV platform & Ref	Interaction Tool	Purpose	Controller Type
Quadrotor [78]	Two 4-DOF robotic arms	Grasp manipulation	MRAC
Quadrotor [79] Hexacopter [80]	2-DOF robotic arm	Pick and place	Adaptive SMC
Quadrotor team [81]	Gripper	Grasp manipulation	Second-order SMC
Quadrotor [82]	Cable	Suspended payload transportation	Passivity-based control
Quadrotor [83]	Delta arm	Contact inspection	Passivity-based control
Quadrotor [84]	2-DOF robotic arm	Pick and place	Nonlinear MPC
Tri-TiltRotor [85]	4-contact-point surface	Force exertion	Explicit MPC
Quadrotor [86]	Delta arm	Contact inspection	Impedance control
Quadrotor [87]	3-DOF robotic arm	Force exertion	Cartesian impedance control
Quadrotor team [88]	Rod	Aerial tele-manipulation	Force stiffness control
Quadrotor [89]	Contact-point surface	Force exertion	Variable impedance control

the inclusion of force information in the control loop can be essential to increase the performance of aerial manipulation tasks. In particular, it may prevent the destruction of the object/environment during the physical interaction. Impedance/admittance control is one of the control techniques which allow us to impose a desired dynamic relation between the interaction force and position [90]. For instance, in [86, 87, 89], different variations of impedance control are employed for the force exertion and contact inspections with aerial robots, while in [88], the force stiffness control is investigated for cooperative aerial tele-manipulation. Nevertheless, it should be noted that the explicit/implicit inclusion of force information in the control loop requires the measurement and/or estimation of the interaction forces

and/or torques. Although the force and torque sensors can be used to obtain these measurements, it is usually not practical due to the payload limitations of the UAV platform and the high cost of sensors. In the case of force estimation, advanced and robust methods are required to get the precise values, which can be difficult to achieve considering the existing internal uncertainties and unforeseen external disturbances. Overall, examples of aerial manipulation applications performed by contemporary methods are presented in Table 2.2.

2.2.3 Intelligent Control Approaches

Although the aforementioned control methods deliver a balance between implementation cost, control performance and operational complexities, they are usually constrained by the precision of the representative mathematical models. However, in many UAV applications, *a priori* knowledge about the process is generally approximated. Therefore, the performance of these methods can notably deteriorate in the presence of internal uncertainties caused by a lack of modeling. Furthermore, the above-mentioned controllers can be sensitive to sensor noise and external disturbances such as wind gusts. Moreover, obtaining the precise model of the nonlinear systems like UAVs is a time-consuming and challenging task. Therefore, as an alternative approach, model-free or intelligent control methods can be considered which provide an intelligent control scheme. Unlike the model-based controllers, the model-free group does not require precise information either about the vehicle dynamics or about the changing environmental conditions because they usually have adaptive learning capabilities and/or intelligent structure. Among the existing model-free approaches, the control methods including different kinds of intelligence are considered to be prominent. For instance, fuzzy logic has an exceptional ability to handle uncertainties in the system [27]. Therefore, fuzzy logic controllers (FLCs) have become popular solutions to control the nonlinear systems when their precise mathematical model is challenging to obtain [28, 91, 92]. FLCs have been successfully designed and implemented to control mobile robots [93–96], especially UAVs [97, 98]. However, in order to cope with uncertainties, FLCs usually require manual tuning which is a time-consuming and troublesome task. To handle this weakness of FLCs, they have been combined together with artificial neural networks (ANNs), which have exceptional ability to learn from input-output data [29, 99]. As a result, the combination of FLC and ANN, called fuzzy neural networks (FNNs), fuses the

reasoning ability of FLC to handle uncertain information with the training capability of ANN to learn from the controlled process [100]. Therefore, over the last few years, FNN has been successfully applied for control [101, 102] and identification purposes [103, 104]. In addition, in order to further improve the noise handling capabilities of FNN, type-2 fuzzy neural networks (T2FNNs) have been extensively developed over the last two decades [105–107]. In contrast to conventional type-1 fuzzy systems, the membership functions (MFs) of type-2 counterparts are themselves fuzzy, which can provide an extra degree of freedom to deal with uncertainty [107]. Therefore, at higher noise level the structure of T2FNN demonstrates better noise handling capabilities compared to its type-1 counterpart (T1FNN) [108, 109].

The performance of FNN highly depends on training methods. There are two main classes of learning/training algorithms for tuning the FNN parameters, namely derivative-based (computational approaches) and derivative-free (heuristic techniques) methods. The former utilize the derivative information of the objective function to update the parameters of FNN, while the latter do not require functional derivative information to tune the parameters of FNN. Among the existing derivative-based and derivative-free methods, gradient descent (GD) [110] and genetic algorithms (GAs) [111] are one of the most widely used techniques. However, since GD learning algorithms are based on the first order derivatives, it has mediocre efficacy and low learning speed [112]. In particular, the performance of GD based learning algorithms notably deteriorates when the solution is near to the local minimum or when the search space is complex. In addition, since GAs are based on a random search, they are non-deterministic, computationally intensive and slow-converging as well as they do not guarantee to find global maxima. Therefore, GA and GD based learning algorithms are not the best options when it comes to UAM applications due to the aforementioned reasons. Hence, more advanced techniques are needed in order to obtain better performance. Among such techniques, SMC theory-based algorithm has demonstrated faster convergence speed and higher robustness to uncertainties [113, 114]. Furthermore, FNNs trained with SMC theory-based learning algorithms have been successfully applied in robotics such as controlling spherical rolling robot [115], robotic arm [116] and the control of agricultural vehicles [117–119].

2.2.4 Discussions and Research Gaps

Although some of the aerial manipulation tasks can be fairly performed even with the PID/PD-based conventional controllers, the review on control approaches reveals that there is a tendency towards having the nonlinear and intelligent control methods for the control of aerial robots. The nonlinear control methods can provide satisfactory control performance, but they are usually constrained by the precision of the representative mathematical models and the sensitivity to modeling errors, noisy measurements as well as external disturbances. Therefore, in the presence of the aforementioned conditions, the intelligent control methods may be preferred over nonlinear control approaches, especially when the uncertainty is high. In particular, they have become popular solutions to control the vehicles with nontrivial dynamics, which is the case for UAVs equipped with robotic manipulators. In addition, the intelligent control approaches for aerial manipulation are still at an early stage. As can be seen in [120], there is a recent attempt in this research direction. Thus, more effort should be put in this area, in order to fully exploit the advantages of intelligent control methods, especially the ones endowed with learning capabilities. Hence, the need for the learning-based intelligent control approaches to achieve the robust and stable control of UAMs under uncertain working conditions is perceptible. Therefore, in an attempt to close the possible research gaps, the second focus in this thesis is the development of the learning-based intelligent control strategy for UAMs. In fact, the proposed control strategy addresses the following challenges that hinder the wider implementation of aerial manipulation tasks in daily life. Firstly, since obtaining the accurate model of the nonlinear systems like UAMs is a time-consuming and difficult task, the proposed approach does not require a precise model of the system to be controlled. Moreover, unlike conventional model-based controllers, the developed control strategy performs well in the presence of internal/external uncertainties and under time-varying working conditions due to the learning capabilities of its structure. Finally, since the structure of the proposed method is endowed with learning capabilities, it eliminates the need for precise tuning of conventional controllers, allowing it to be employed by both single and dual-arm aerial manipulators. The proposed control strategy is investigated in Chapter 4.

2.3 Overview of Guidance Approaches

The guidance approaches can be considered as one of the most important factors in achieving a high autonomy level in the performance of aerial robots. According to [26], the guidance system for aerial robots consists of several main components, namely trajectory generation, path planning, mission planning and high-level decision making. Depending on the required autonomy level of aerial vehicles, these components can be combined together or they can be performed separately. However, regardless of active research on the guidance systems for UAVs, current research mostly concentrates on guidance systems (e.g., trajectory generation and path planning), which are usually restricted with waypoint navigation. In this context, the trajectory generation algorithms are at an even more immature stage for UAMs which are characterized by exceptional kinematic redundancy. Hence, the current development in the trajectory generation-based guidance approaches for the single and dual-arm aerial manipulators is discussed below.

2.3.1 Guidance Approaches for UAMs

Over the last few years, the trajectory generation and redundancy resolution strategies have been studied for single-arm aerial manipulators in [30, 121, 122]. Specifically, in [30], the trajectory generation approach for UAM via quadratic programming is considered. In [121], a null-space based behavioral (NSB) controller is presented, while a task priority controller combined with visual servoing is developed in [122] to perform multiple tasks during aerial manipulation missions. Nevertheless, very few works consider the trajectory generation strategies for UAMs with more than one arm like dual-arm aerial manipulators [123, 124]. For instance, in [123], motion planning strategy based on the RRT* algorithm is proposed for the aerial long-reach manipulator with two arms, while in [124], set-based inverse kinematics algorithm is developed for the dual-arm aerial manipulator.

In comparison to the aforementioned trajectory generation approaches, utilizing MPC-based methods for UAMs may lead to a better long-term motion strategy due to the fact that MPC considers a longer horizon for the evolution of states during the optimization. However, it should be noted that this approach is known to be computationally demanding because of its iterative nature. There are several

MPC implementations for the UAM missions in the literature. In [125], a direct multiple shooting method-based MPC approach is considered for a pick-and-place task with UAM. However, the results are only presented in simulations and real-time implementation is left for future work. In [84], a pick-and-place operation is achieved by nonlinear MPC (NMPC) with a real robot, but both in [125] and [84], the trajectory generation is performed in an offline manner and no obstacles are considered to simplify the problem. In addition, both works do not aim to exploit the redundancy of the system to perform multi-task missions. Finally, in [126], NMPC based trajectory generation and redundancy resolution strategy for UAM is described to accomplish multiple tasks. Nevertheless, due to the representation of the system in the velocity domain, the generated trajectory values can present discontinuities in the velocity profile with unbounded accelerations. Besides, obstacle avoidance is not considered in the above work. It should be noted that MPC approaches mentioned above only consider the cases when UAVs are equipped with a single-arm.

2.3.2 Discussions and Research Gaps

The literature review on the trajectory generation-based guidance approaches for UAMs has shown that it is still at the young stage, especially this is true for UAMs with more than one arm. In particular, in the case of dual-arm aerial manipulators, very few works consider the redundancy resolution strategies to perform multi-task missions. In addition, obstacles are not considered in most of the works to simplify the problem. Such simplifications can hinder the deployment of aerial robots in real-world applications, where the presence of obstacles is inevitable. Furthermore, during aerial manipulation missions, there are cases when the redundancy of UAM cannot be fully exploited due to antagonistic tasks. In such cases, it is important to have the task relaxation approach that helps to satisfy the robot's physical constraints and ensure the motion smoothness. Therefore, in an attempt to close the possible research gaps, the third focus in this thesis is the development of trajectory generation-based guidance approaches with redundancy resolution strategies for the single and dual-arm aerial manipulators that address the aforementioned cases. The proposed approaches are investigated in Chapter 5.

2.4 Overview of State Estimation

The accurate estimation of the UAM state is usually challenging due to the existing power and weight restrictions of aerial robots. In addition, considering the application requirements and deployment area, the correct choice of sensors and corresponding fusion algorithms can be a difficult task. For instance, sensing systems designed for outdoor localization may not be applicable in an indoor environment. Furthermore, the cost of sensors is an additional constraint that should be carefully considered during the selection process. Therefore, this section covers a brief discussion on the existing sensing systems which are used to acquire system states. Then, it is worth to specify the complete state of the aerial manipulator as follows

$$\begin{cases} \boldsymbol{\sigma} = [\mathbf{r}_b^T \ \boldsymbol{\Phi}_b^T \ \mathbf{q}^T]^T, \\ \dot{\boldsymbol{\sigma}} = [\dot{\mathbf{r}}_b^T \ \dot{\boldsymbol{\Phi}}_b^T \ \dot{\mathbf{q}}^T]^T. \end{cases} \quad (2.1)$$

in which $\mathbf{r}_b = [x_b \ y_b \ z_b]^T \in \mathbb{R}^3$ is the position of the UAV platform, $\boldsymbol{\Phi}_b = [\phi \ \theta \ \psi]^T \in \mathbb{R}^3$ is the UAV orientation represented by the Euler angles, and $\mathbf{q} = [q_1, \dots, q_n]^T \in \mathbb{R}^n$ is the joint position vector of the arm. It should be noted that the state estimation methods for single and dual-arm cases are the same. Therefore, only single-arm case is considered in this section.

In general, various types of sensors can be utilized to acquire reliable system states. In Table 2.3, examples of sensing systems for the state estimation of UAM are presented. As can be seen from this table, among the system states the arm joint values and velocities are the easiest to obtain, since most of the existing motors

TABLE 2.3: Examples of sensing systems for the state estimation of UAM (adopted from [127]).

Sensor Type	States
GPS, Optical flow, IMU, Sonar, Wireless localization techniques, Motion capture systems, Vision/Lidar-based methods	$\mathbf{r}_b, \dot{\mathbf{r}}_b$
Motion capture systems, IMU, Wireless localization techniques, Vision/Lidar-based methods	$\boldsymbol{\Phi}_b$
IMU	$\dot{\boldsymbol{\Phi}}_b$
Motor encoders	$\mathbf{q}, \dot{\mathbf{q}}$

come with embedded encoders. In contrast, acquiring reliable system states corresponding to the UAV platform requires more attention. In particular, although an Inertial Measurement Unit (IMU) and Global Positioning System (GPS) can be used to measure the UAV states, they are usually noisy, and their accuracy is dependent on the deployment area. In addition, GPS sensors cannot be employed in an indoor environment. It should be noted that it is assumed that IMUs are equipped with magnetometers. As for optical flow and sonars, depending on the deployment area, they can provide some system states with satisfactory accuracy. However, they have similar limitations as previously mentioned GPS and IMU sensors. Therefore, other approaches can be utilized in combination with the aforementioned sensing systems or standalone to acquire reliable system states. For instance, from Table 2.3, three classes of sensing systems can be mentioned, namely motion capture systems, wireless localization techniques, and vision/lidar-based methods. Motion capture systems are characterized by the highest accuracy and rate among them. However, they are usually restricted to the indoor environment with the fixed deployment area. In addition, the cost of equipment is high compared to other sensing systems. As for wireless localization techniques (e.g., ultra-wideband (UWB)), they are generally treated as a less expensive approach compared to the motion capture systems [127]. However, due to the lower cost of implementation, its accuracy may not be sufficient for aerial manipulation tasks. In the case of vision/lidar-based methods, they are the most versatile option that can be employed both in the indoor and outdoor environments. Nevertheless, due to the existing computational power, battery, and weight restrictions of aerial robots, the full potential of this approach is still not realized. Besides, the cost of the sensors that can provide reliable system states is still high. It should be noted that the state estimation is not the scope of this thesis, and therefore, in this work, the motion capture system is utilized to obtain reliable system states.

Chapter 3

Design and Manufacturing of Unmanned Aerial Manipulators

3.1 Introduction

As reviewed in the previous chapter, many research groups have worked on the development of the single-arm aerial manipulators, while only a few works have considered the design and development of the dual-arm aerial manipulators. Therefore, in an attempt to close the possible research gaps, this chapter presents the design, fabrication, and experimental validation of a lightweight, low inertia dual-arm manipulator with the COG balancing mechanism, specifically developed for aerial manipulation missions.

Unlike previously developed dual-arm aerial manipulators that rely only on revolute joints in their designs, the proposed dual-arm design employs the prismatic joint as the first DOF in each arm to introduce the linear sliding motion that results in several distinctive features as shown in Fig. 3.1. In particular, each arm has the ability to dynamically align its COG with the symmetry line of the UAV platform (see Fig. 3.1 (left)), and thus the arms can be configured to compensate for the COG displacement of the overall system, allowing to mitigate the influence of the arms motion over the multirotor stability. Furthermore, employing the prismatic joints increases the operation volume and reach of the arms whose workspace is usually restrained by the landing gear and propellers. Thus, the presented novel design allows the UAV platform to be kept at a safer distance from the interacted

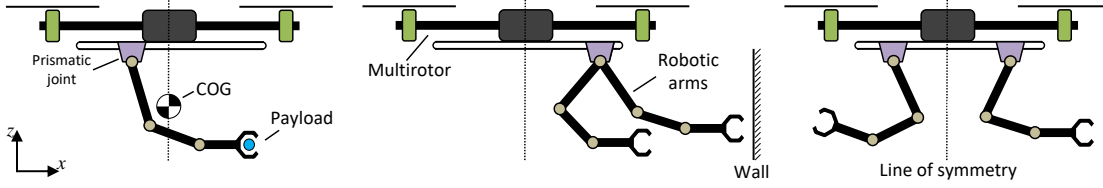


FIGURE 3.1: Three distinctive features of the proposed design: ability to dynamically adjust its COG (left), extended workspace and reach of the arms (middle), *decoupled* configuration of the arms (right).

objects or environment such as walls as shown in Fig. 3.1 (middle), improving safety during aerial manipulation missions. Moreover, while dual-arm aerial manipulators found in the literature have the fixed and coupled shoulders, the proposed novel design enables to achieve fully independent control of each arm by decoupling the shoulders as shown in Figs. 3.1 (right) and 3.2 (right), allowing to perform different tasks at the same time. In Table 3.1, the proposed design and existed dual-arm manipulators are compared. It can be noted that the developed design has the maximum reach and payload as well as the largest number of DOFs.

In addition to the above features, the timing belt-based transmission mechanism is employed to place the actuators near the UAV base, allowing to decrease the inertia of the arms. In order to save the actuators from the excessive axial and radial loads, thrust and roller bearings are used. To ensure the robustness of the mechanical structure, the FEM analysis of the main frame parts is conducted. Finally, extensive experimental flight tests are performed to evaluate the proposed dual-arm design with a hexarotor platform equipped with a standard autopilot. To the best of the author's knowledge, this is the first time when the COG balancing mechanism and decoupled shoulders are proposed in the design of dual-arm aerial manipulators. The video of the experimental flight tests can be accessed via this link (<https://youtu.be/1cUYWqkMCIM>). It should be noted that for the sake of completeness, the developed single-arm manipulator is also briefly discussed in this chapter.

TABLE 3.1: Comparison of different dual-arm manipulators.

DOF	Reach [m]	Weight & Max payload [kg]	COG balancing mechanism	Decoupled shoulders	Ref
12	0.67	2.5/1.0	Yes	Yes	Proposed
4	NA	NA/NA	No	No	[60]
NA	NA	NA/NA	No	No	[21]
10	0.5	1.8/0.75	No	No	[22]
8	0.5	1.3/0.2	No	No	[23]

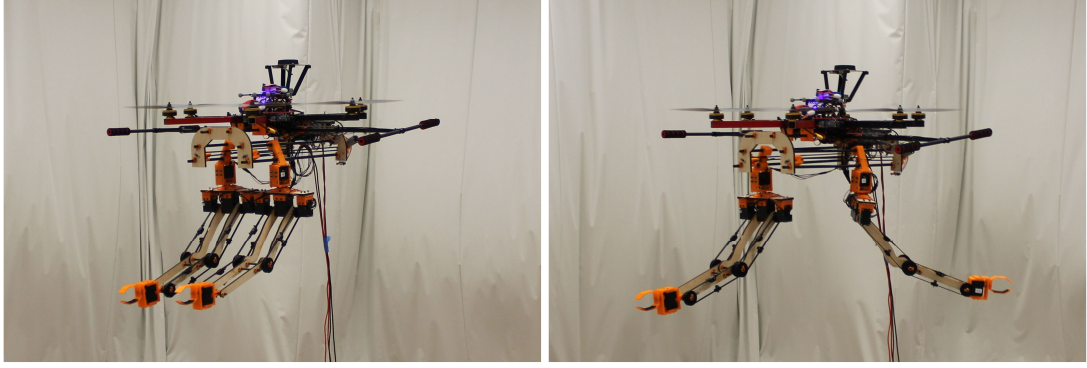


FIGURE 3.2: Developed dual-arm manipulator mounted on a hexarotor platform: the shoulders of dual-arm in the *coupled* (left) and *decoupled* (right) configurations.

The remainder of this chapter is organized as follows. In Section 3.2, specific design considerations and requirements for aerial manipulation are discussed. In Section 3.3, the developed novel dual-arm system is explained in detail. In Section 3.4, the single-arm manipulator is briefly described. In Section 3.5, experimental studies are conducted in order to validate the performance of the proposed dual-arm design. Lastly, conclusions and a summary of this chapter are provided in Section 3.6.

3.2 Design Requirements and Considerations

In general, the successful performance of aerial manipulation missions is conditioned by the fulfillment of several design requirements. In the context of aerial manipulation, some of these requirements are specific due to the attachment to floating platforms, while others are generic, which are common for any type of robotic arms. In addition, some of the design requirements and considerations can be motivated by specific applications or tasks that could be performed by these type of aerial robots. For instance, for aerial manipulation missions like aerial hammering and throwing, the elastic-joint types are more suitable to obtain high-speed link velocities compared to the rigid-joint types [128]. The advantage of the elastic joint is that potential energy stored in the elastic elements can be released in the form of kinetic energy to perform explosive/fast motions by boosting the speed of the actuated joints. On the other hand, for aerial manipulation missions like aerial grasping, high-speed link velocities are not crucial. Therefore, the rigid-joint types are more favorable for such kind of applications. Besides, the utilization of

the elastic joints imposes additional design and control challenges arising from the employment of the elastic elements [128]. Thus, the scope of this work is focused on designing the dual-arm aerial manipulator with the rigid-joint type that is aimed to perform the assembling task in a cyclic manner, while advanced UAM missions like aerial hammering, which require the elastic-joint types, will be considered in future work. It should be noted that the aforementioned assembling task generally consists of three phases, namely *grasping*, *transporting*, and *placing* an object at the assigned location [16]. Each phase can impose further design requirements. In particular, during *grasping* and *placing* the object, it is important that robotic arms have a wide workspace beyond the area delimited by the propellers to ensure the safety of UAMs during interactions with the environment. As for the *transporting* phase, adding the ability to compensate for the COG displacement caused by the grasped object is beneficial to lessen the control burden and improve overall flight behavior. These and other specific design considerations and requirements for aerial manipulation are discussed below [16, 23].

3.2.1 Weight, Inertia, and Reach Considerations of the Robotic arms

The design of the robotic arms specifically developed for aerial manipulation purposes is quite challenging considering the design requirements imposed by multirotor UAVs in terms of weight, inertia, and the reach of the arms. Firstly, due to the nature of multirotor UAVs, their maximum payload capacity is restricted. This capacity is further decreased when the mass of the object is located far from the COG of the UAV base. In addition, if the multirotor is overloaded, its motors may undergo overheating, resulting in the poor response to the control signals. Therefore, it is crucial to design a lightweight dual-arm manipulator for aerial manipulation tasks. Regarding the inertia of the arms, in order to decrease the influence of the motion of the arms over the UAV stability, it is beneficial to have dual-arm with low inertia feature. This can be achieved by placing the actuators near the UAV base employing various transmission mechanisms. In this work, the timing belt-based transmission mechanism is employed for this purpose. Finally, the reach of the arms should be long enough in order to have a wide workspace beyond the area delimited by the propellers. This will ensure that the robotic arms will have an appropriate operation volume and reach, improving the safety of UAMs during

interactions with the environment. In order to satisfy this criterion, in this work, not only are the arms designed to be sufficiently long but also the prismatic joints are introduced to further increase the range of the arms.

3.2.2 Motors and Materials

The critical factors during the selection of motors are their weight, size, torque, and power consumption. Other important aspects are the high accuracy, control modes, available feedback and communications. In this regard, actuators like Dynamixel and Herkulex can be considered among the existing motors for constructing lightweight robotic arms since they are relatively lightweight, small and have a good balance between provided torque and power consumption. Nevertheless, these actuators are constrained by their embedded controller and control rate, which is typically less than 100Hz [22]. In addition, it should be noted that the stall torque values of the selected motors should be at least two or three times larger than the required dynamic torque values. By considering all the aspects mentioned above, we have chosen the latest version of actuator modules from Robotis, namely Dynamixel X-series, since they can provide various control modes with improved torque to weight ratio and size.

The materials for constructing the frame structure should be selected in a way that the final design can have a good combination between cost, mechanical robustness, and weight. Therefore, material property charts presented in [129] can be employed to quickly exclude large numbers of materials and to determine the possible candidates that satisfy the required properties. Figure 3.3 illustrates two material property charts, namely strength against density and Young's modulus against relative cost per unit volume. Based on this figure, most of the materials can be eliminated from consideration. For instance, most of the metal alloys are heavy and expensive, and thus they cannot be considered for designing lightweight manipulators. Similarly, foams and elastomers are not suitable due to their weak and flexible structure. Carbon fiber and aluminium can be considered for designing manipulators. Still, the material and fabrication cost is higher compared to natural materials (e.g., wood) and polymers, especially when the frame parts with complex geometry are required. Therefore, in this work, the combination of the

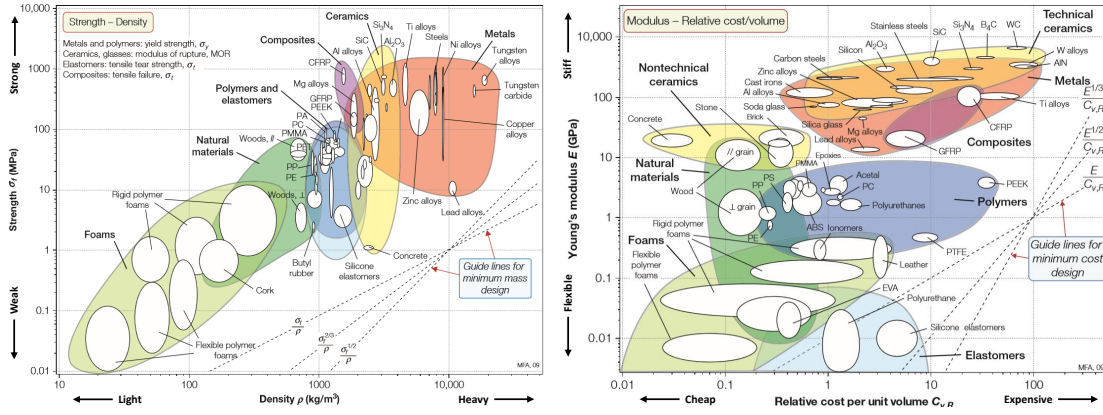


FIGURE 3.3: Examples of material property charts (Ashby plots): Strength - Density (left) and Young's modulus - Relative cost per unit volume (right) [129].

materials is considered to obtain the best possible design in terms of weight, mechanical structure, and cost. In particular, for our application, we prefer to use birch plywood for the simple frame parts over aluminium and polyethylene terephthalate glycol (PETG) plastics for the complex parts over polylactic acid (PLA) and acrylonitrile butadiene styrene (ABS) plastics due to its high flexural strength and better elongation before breaking. The main reason for choosing birch plywood over aluminium is the weight and cost advantages. In addition, it is characterized by decent stiffness and strength properties [130], and thus, birch plywood behaves sufficiently in the applications demanding high rigidity and strength. Finally, carbon fiber tubes/rods are used for the parts subjected to maximum stress.

3.2.3 Kinematic Configuration of the Robotic Arms

The selection of the joint types and the number of DOFs of the robotic arms is usually dictated by the tasks to perform during aerial manipulation missions. In general, it is preferable to have at least 3 DOFs per each arm to improve the dexterity and applicability of UAMs for various scenarios. In the case of the assembling task, the choice of joint type and its sequence can be motivated by the three phases mentioned above (*grasping*, *transporting*, and *placing*). In particular, the proposed design provides 6 DOFs for the end-effector positioning and orientation in each arm where the first joint is prismatic followed by five revolute joints with the *yaw-pitch-pitch-pitch-roll* sequence as illustrated in Fig. 3.5. Drawing inspiration from [16], the details of the identification process are discussed below. During *grasping* and *placing* the object, in order to have accurate linear movements and decrease

power consumption, it is desirable to move robotic arms instead of moving the UAV platform. Therefore, the third, fourth, and fifth joints of each arm are selected to be a pitch-type. Besides, having these joints as the pitch-type enables the arms to achieve compact configuration by folding onto themselves. Such configuration can protect the arms from the unexpected collision with obstacles during the contactless flight or the transportation of objects. Furthermore, the compact configuration achieved by folding the arms onto themselves allows the UAM to take-off and land without the need for additional design solutions enabling easier integration of dual-arm with the aerial platforms. Moreover, sometimes it is required to change the orientation of the end-effector due to the shape/state of the object. This can be achieved by employing a roll-type joint. In order to avoid reorientation of the other links of the robotic arm, this joint should be the closest one to the end-effector. Hence, the sixth joint is selected as the roll-type joint. As for the second joint, it is selected as the yaw-type joint to increase the operation volume and have the redundancy in the yaw motion while positioning the end-effectors. Finally, the first joint is selected to be the prismatic-type joint. Employing the prismatic joint at the beginning of the kinematic chain not only allows increasing the range of the arms but also it can be used to compensate for the COG displacement caused by the grasped object, which will be discussed in more detail in the next section.

3.2.4 Compensation of the COG Displacement

When the UAV platform is internally balanced, it is assumed that the line of symmetry goes through the geometric center of the UAV platform. When the dual-arm COG is aligned with the line of symmetry, the multirotor maintains its attitude and position as shown in Fig. 3.4a (left). However, the motion of the robotic arms causes the displacement in COG, d_x , which, in its turn, produces a reaction torque at the UAV base. As a result, this generates the inclination of the propellers plane leading to the pose variation of the whole aerial robot as shown in Fig. 3.4a (top right). Especially, it is evident in the presence of the arms payload. In order to avoid these perturbations in the UAV platform, it is desirable to keep the dual-arm COG as close as possible to the line of symmetry, i.e., $d_x = 0$. This can be achieved by employing several approaches. For instance, in [16] and [24], the linear sliding mechanism is designed to compensate for the COG displacement by moving the battery as a counterweight, while, in [131], the

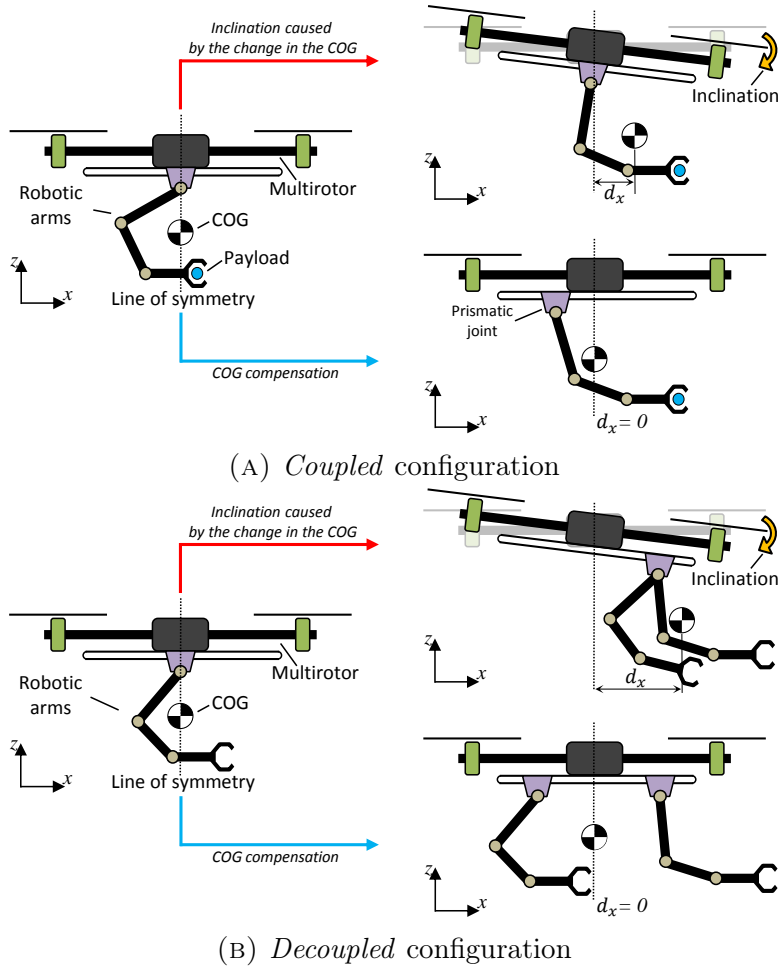


FIGURE 3.4: Displacement and dynamic alignment of the dual-arm COG.

linear sliding motion, which adjusts the arm COG, is obtained by adding additional DOF to the arm. In this work, the latter approach is adopted for the dual-arm case where the prismatic joint is implemented as the first DOF in each arm to dynamically align the dual-arm COG with the line of symmetry as shown in Fig. 3.4a (bottom right). It is worth noting that although both approaches aim to compensate for the COG displacement of the overall system, moving the battery as a counterweight is usually considered as a passive approach because it is only utilized during the compensation of the COG displacement. On the other hand, in the proposed design, the prismatic joints are not only used to perform COG balancing, but they also can be used to increase the operation volume and reach of the arms by moving them to the forward or backward direction. This can enhance safety during aerial manipulation missions by keeping the UAV platform at a safer distance from the interacted objects. Moreover, employing the prismatic joints allows achieving fully independent control of each arm and performing different tasks at the opposite sides of UAM by decoupling the shoulders of the dual-arm.

Besides, the ability to decouple the shoulders of the dual-arm can be employed for further COG compensation. This can be explained by examining Fig. 3.4b. When the shoulders of the arms are in the *coupled* configuration, their linear motion causes the inclination and pose variation of the UAV platform as shown in Fig. 3.4b (top right). In this case, the *decoupled* configuration of the proposed design can be exploited by moving one of the arms in the opposite direction to align the dual-arm COG with the line of symmetry as depicted in Fig. 3.4b (bottom right), especially if only one arm is required to perform a given task. The aforementioned cases show that introducing the prismatic joints at the beginning of the kinematic chain of each arm significantly increases the range of tasks and operational capabilities of the dual-arm aerial manipulator compared to the case when only battery movement is utilized as the compensation for the COG displacement. It should be noted that the inclination of UAV in the roll direction is not considered in this work.

The above-mentioned COG balancing mechanisms are mainly focused on the static compensation of the overall system. However, the fast motion of the arms can produce further reaction forces/torques that create the additional dynamic impact on the UAV platform. There are several approaches that attempt to address this case. In [131], the reduction of the dynamic effects of the robotic arm is considered from the design perspective, where the authors focus on keeping the sum of the angular momentum of the rotating parts to zero. On the other side, such design solution requires an increased number of parts and complex transmission systems, which decrease the payload capability of the UAV platform, especially when employing the robotic arms with more than 3 DOFs. Another approach is using the force/torque sensors to measure reaction forces acting on the UAV platform. This approach can also be beneficial to measure interaction forces during contact with the environment [128]. However, utilizing such sensors is usually not practical due to the payload limitations of the UAV platform and the high cost of sensors. Instead of employing the force/torque sensors, the forces acting on the UAV platform can be estimated and fed back to the control algorithm [24]. Nevertheless, the precise estimation can be difficult to achieve, considering the existing internal uncertainties and unforeseen external disturbances. Finally, intelligent control approaches can be used to compensate for the internal disturbances caused by the motion of the arm(s) and reduce the dynamic impact on the UAV platform. This work focuses on the last approach, and it will be discussed in more detail in Chapter 4.

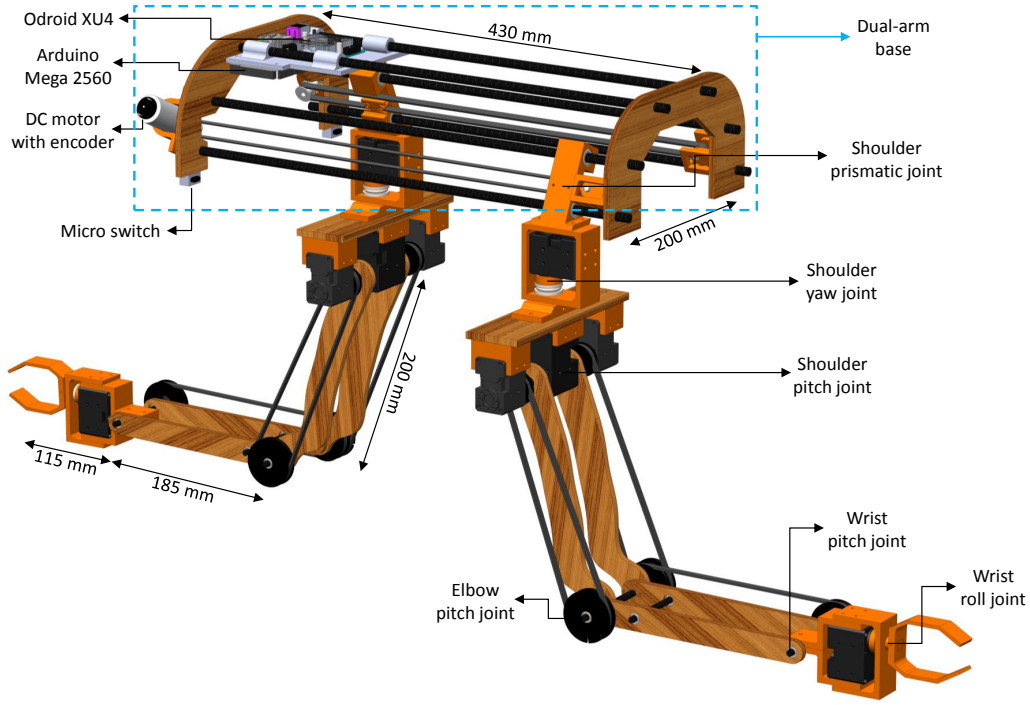


FIGURE 3.5: 3D model of the developed dual-arm system.

3.3 Dual-Arm Aerial Manipulator

3.3.1 Developed Dual-Arm System

Based on the aforementioned design requirements and considerations, a 3D model of the dual-arm system is developed as can be seen in Fig. 3.5, consisting of the dual-arm base and two human-size arms. Each arm provides 6 DOFs for the end-effector positioning and orientation with the following joint types: shoulder prismatic, shoulder yaw, shoulder pitch, elbow pitch, wrist pitch, and wrist roll. A DC motor with an encoder from DFRobot is employed for the shoulder prismatic joint, while Dynamixel X-series actuators are utilized for the rest of the joints. The main motor parameters and specifications associated with each joint are shown in Table 3.2. The total mass of the dual-arm system is 2.5 kg with the maximum payload per arm of 1.0 kg, which is obtained when the arm is in fully horizontally stretched configuration. In order to decrease the effect of the motion of the arms over the UAV stability, most of the actuators are placed as close as possible to the dual-arm base by employing 2GT 6 mm wide timing belts for the motion transmission. The belts are made from high tensile steel cord and wear-resistant polyurethane to provide durability and strength. The gear ratio between

the driving gear and the driven one is kept as 1.5 to increase joint torque at elbow pitch and wrist pitch joints. In addition, employing timing belts rather than other motion transmission mechanisms allows us to achieve the compact configuration by folding the arms onto themselves. In the developed design, thrust and roller bearings are used for reducing the axial and radial loads exerted over shoulder yaw and wrist roll actuators by distributing the loads over the frame parts and isolating the actuators from overloads. Due to the space limitations, such mechanisms are not considered for other joints.

The shoulder prismatic joints are actuated by DC motors, which are fixed to the wooden part of the dual-arm base. A geared pulley is attached to the DC motors, which rotate 2GT 6 mm wide timing belt to obtain the rotary to linear motion transmission. The timing belt goes to the other side of the dual-arm base and passes over the idler pulley to complete the loop. In order to adjust the tension of the belt, the position of the idler pulleys can be easily changed by using the nut and bolt mechanism. In addition, micro switches are used to implement the initializing and homing procedure for prismatic joints. In general, thanks to the shoulder prismatic joints, the shoulders of the arms can be separated from each other to achieve *decoupled* configuration. Thus, the arms can move to two extremes of the dual-arm base as shown in Fig. 3.5 and simultaneously perform different tasks, which is not possible with the *coupled* configuration.

TABLE 3.2: Main parameters of the arm joints.

Joint	Motion range	Motor model	Stall torque	Motor weight [g]
Shoulder prismatic	[-17cm, 17cm]	6V DC Motor w/Encoder	10.0 [kg · cm]	96
Shoulder yaw	[-180°, 180°]	Dynamixel XM430-W350-R	4.1 [N · m]	82
Shoulder pitch	[-90°, 90°]	Dynamixel XM540-W270-R	10.6 [N · m]	165
Elbow pitch	[-80°, 160°]	Dynamixel XM430-W350-R	4.1 [N · m]	82
Wrist pitch	[-120°, 120°]	Dynamixel XM430-W350-R	4.1 [N · m]	82
Wrist roll	[-180°, 180°]	Dynamixel XL430-W250-T	1.5 [N · m]	57.2

The topmost part of the developed system is the dual-arm base, which is attached to the multirotor platform by using two \varnothing 8 mm carbon fiber rods. Odroid XU4 and Arduino Mega 2560 are both supported by these carbon fiber rods, and they are placed at the rear part of the dual-arm base. During the flight, batteries for the dual-arm system can be placed at the front part to balance the dual-arm base internally. There are four \varnothing 8 mm carbon fiber tubes that are used for attaching the arms (two per arms) to the dual-arm base. These tubes are kept tilted because of the space constraint imposed by the landing gear. In order to slide along these tubes, the polymer linear bearings are used. All carbon fiber tubes and rods are fixed at both ends with two 6 mm thickness birch plywood parts, which are obtained by laser cutting. Since most of the hardware is attached to the dual-arm base, the proposed dual-arm design is not peculiar to a specific multirotor. Therefore, such a design makes our system more versatile as the dual-arm system can be attached to different multirotors without making major modifications in the design.

The design of the frame parts is developed in such a way that the best possible design can be obtained in terms of weight, mechanical structure, fabrication cost, and complexity. For our application, we prefer to use birch plywood for the simple frame parts and PETG plastics for the parts with complex geometries. Hence, the arm links in this design are made of 6 mm thick birch plywood.

3.3.1.1 FEM analysis of the main frame parts

In order to check the robustness of the mechanical structure, the FEM analysis of the main frame parts has been carried out as shown in Fig. 3.6. In particular, two FEM analyses have been performed. The former considers the main parts of the robotic arm as depicted in Fig. 3.6a. These main parts include the upper arm link (from shoulder to elbow), the forearm link (from elbow to wrist), and the carbon fiber rod connecting these links. These parts support the load originating from the payload carried by the end-effector and their own weight. The latter (see Fig. 3.6b) has been executed for the cuboid-shaped frame part and the shaft, which connects the shoulder yaw actuator with the rest of the arm. These parts carry the weight of the payload and the arm itself. Due to the symmetry of the dual-arm system, the FEM analysis has only been done for the left arm.

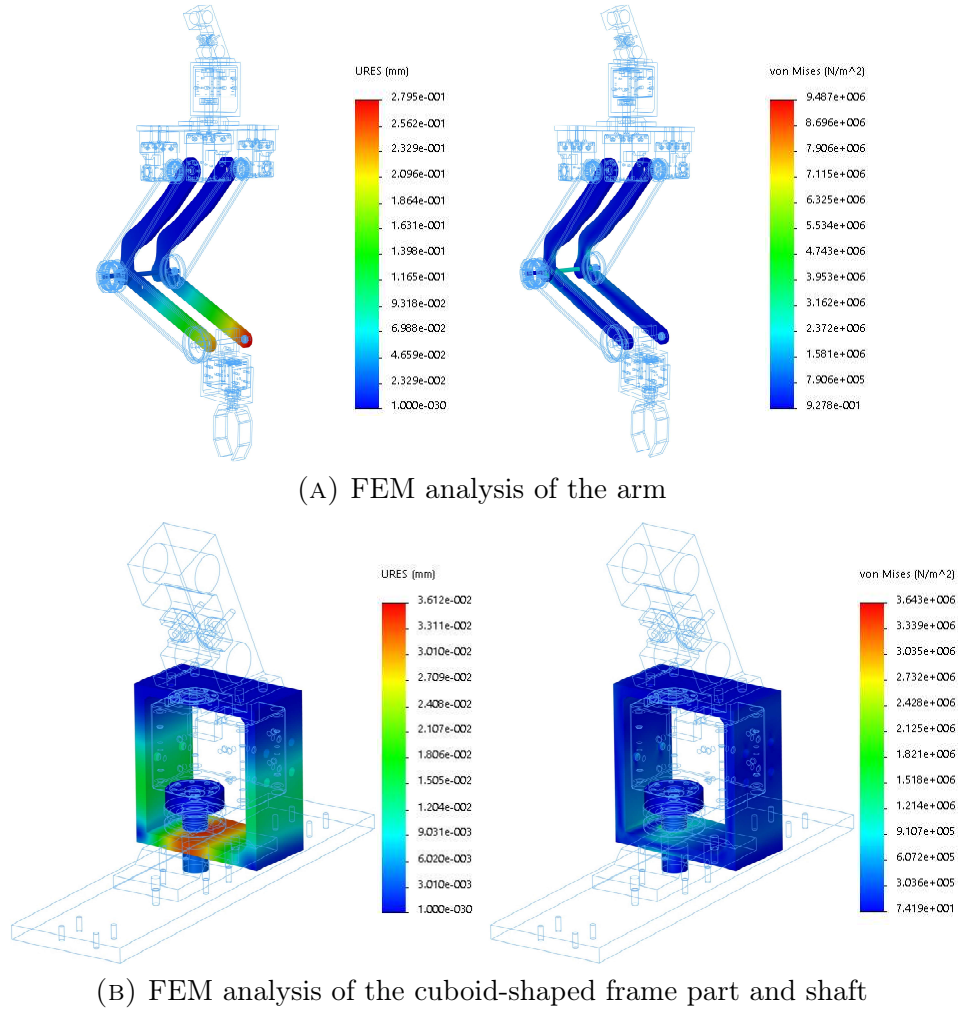


FIGURE 3.6: FEM analysis of the main frame parts.

Figure 3.6 describes the results in terms of stress and displacement. In Fig. 3.6a, a vertical-down load of 1 kg is applied at the end of the forearm link near the wrist joint. It represents the load coming from the maximum payload of the arm. The maximum displacement (0.28 mm) originates at the end of the forearm link near the wrist joint, and it can be considered as negligible, taking into account the forearm link size. Regarding the stresses, the maximum occurred stress is 9.49 MPa, which is less than the bending strength of the 6 mm birch plywood (50 MPa). In Fig. 3.6b, a vertical-down load of 2 kg is applied to the shaft. The thrust bearing placed in between the shaft and the cuboid-shaped frame distributes the load over the frame. The maximum displacement is 0.036 mm, which is again negligible compared to the size of the structure. The maximum occurred stress is 3.64 MPa, which is well below the bending strength of PETG plastics (45 MPa). Based on the above discussions, the system has a safety factor of 5:1. With these results, the robustness of the mechanical structure is ensured.

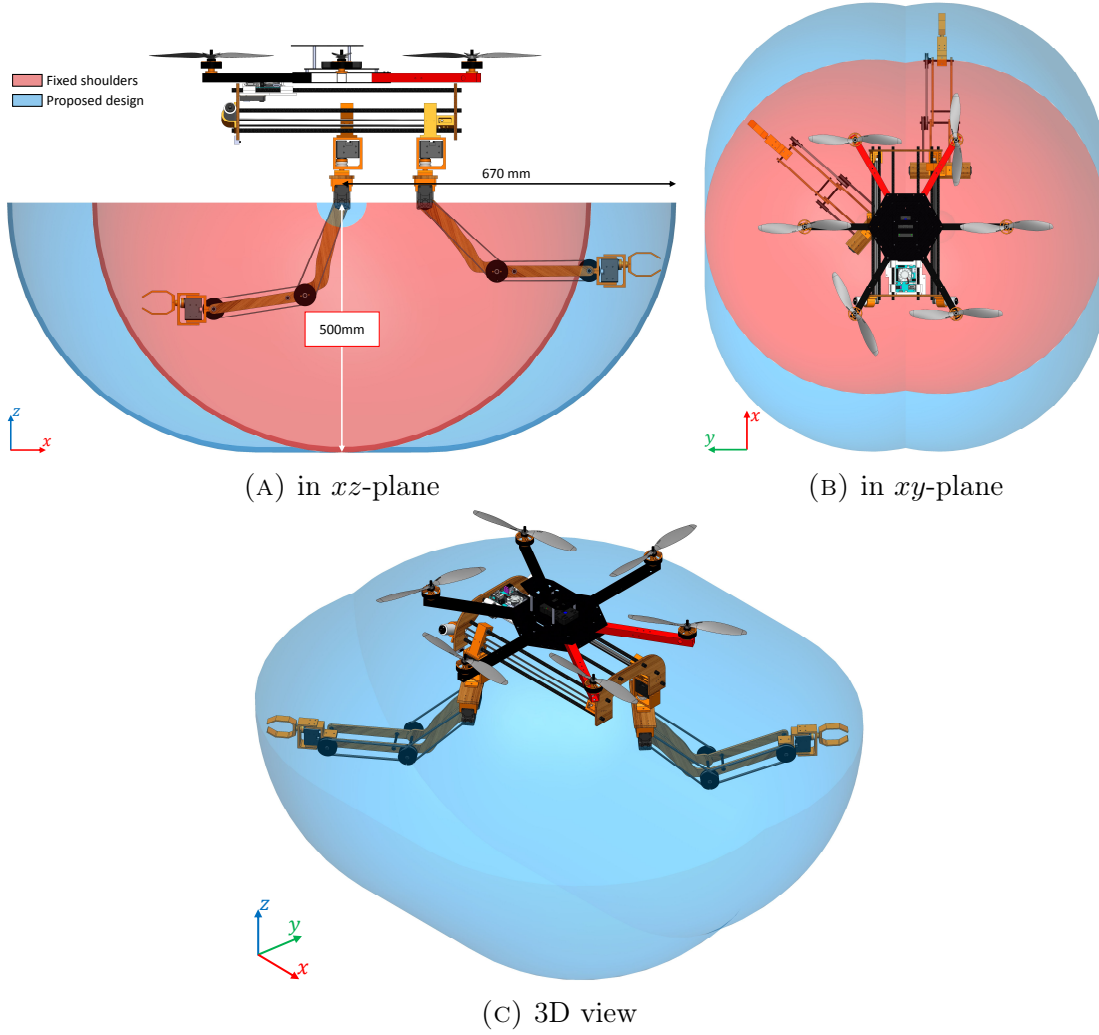


FIGURE 3.7: Workspace of the developed dual-arm manipulator. Due to the propeller constraints, workspace only covers the region under the dual-arm base.

3.3.1.2 Workspace of the Dual-Arm Manipulator

The workspace of the developed dual-arm manipulator is depicted in Fig. 3.7. In the proposed design, the prismatic joints considerably increase the operation volume and reach of the arms as shown in Figs. 3.7a and 3.7b compared with the existing designs of the dual-arm manipulators, where the shoulder joints of two arms are fixed and coupled to each other. In particular, from the section view of the workspace in xz -plane depicted in Fig. 3.7a, it can be seen that the reach of the arms is increased from 0.5 m to 0.67 m, while the area covered by each arm motion represents a semi-stadium geometric shape (blue colored region) in xz -plane. Consequently, the rotation around the shoulder yaw axis produces the semi-capsule shaped workspace for each arm with the operation volume of

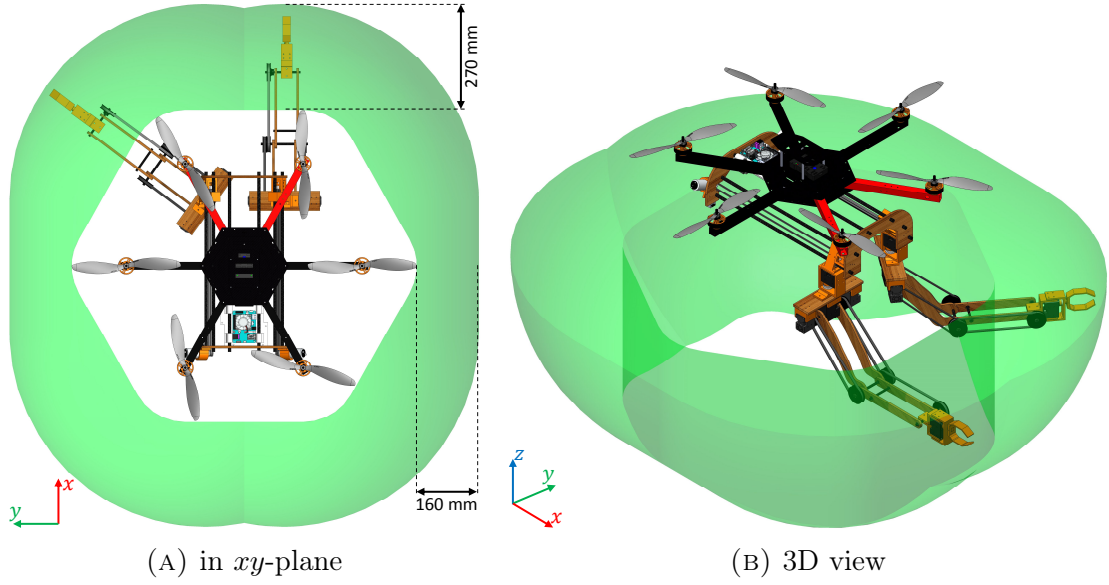


FIGURE 3.8: Workspace of the developed dual-arm manipulator beyond the area delimited by the propellers.

$V_a = 0.395 \text{ m}^3$. Figure 3.7c presents the overall workspace for both arms with the operation volume of $V = 0.507 \text{ m}^3$, considering the common operation region. In contrast, when the shoulder joints of two arms are fixed like in the existing designs, the workspace of each arm resembles a hollow semi-sphere (red colored region) with the volume of $V_a^f = 0.262 \text{ m}^3$, while the volume of the overall workspace for both arms is $V^f = 0.339 \text{ m}^3$. Hence, it can be noted that around 67% increase in the volume of the workspace is achieved with the proposed design. This, in its turn, allows the UAV platform to be kept at a safer distance from the interacted environment such as walls, improving safety during aerial manipulation missions. Specifically, in Fig. 3.8, the workspace of the dual-arm manipulator beyond the area delimited by the propellers is shown. As can be seen from Fig. 3.8a, if one of the arms is equipped with a tool to operate in front of a wall, the longitudinal and lateral distances between the tip of the closest propeller and the wall are 0.27 m and 0.16 m, respectively. The overall operation volume of the workspace depicted in Fig. 3.8b is equal to $V_w = 0.247 \text{ m}^3$.

It should be noted that the workspace shown in Figs. 3.7 and 3.8 represents the case when the multirotor is on flight with the retracted landing gear. When the landing gear is not retracted, the linear motion by the prismatic joints is restricted and the workspace of the arms is considerably limited. Therefore, before the take-off and landing phases, the arms are commanded to have the compact configuration

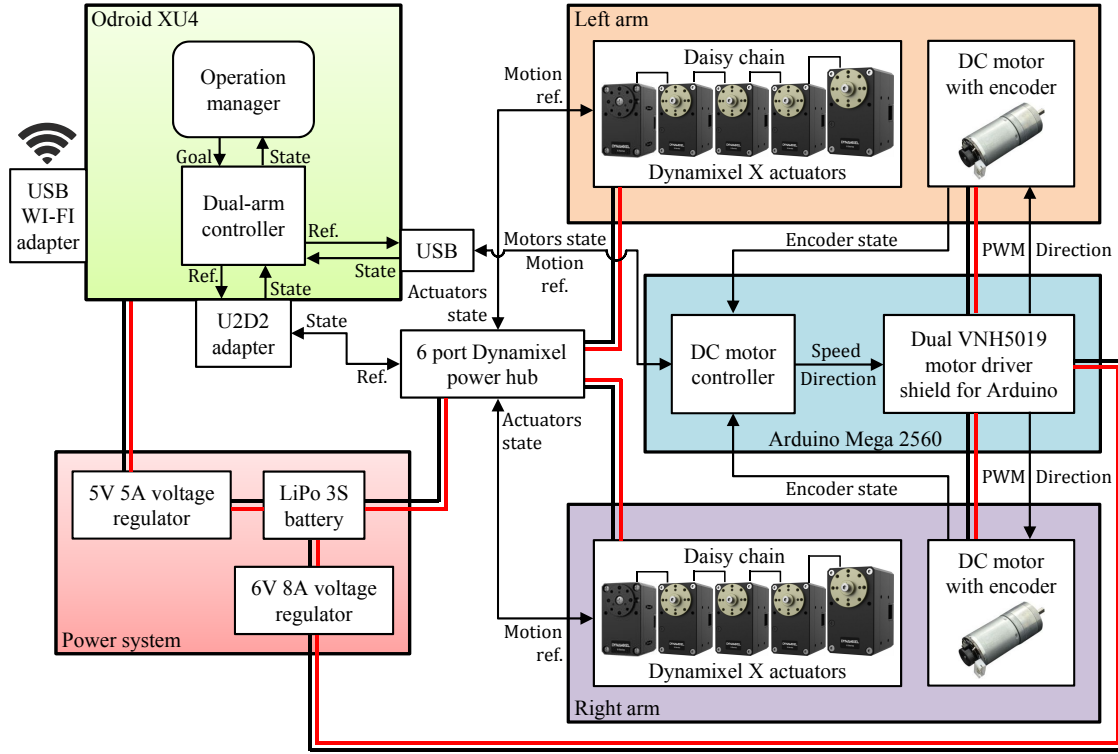


FIGURE 3.9: Architecture and components of the proposed dual-arm system.

by folding onto themselves to avoid collision with the ground. In addition, the motion of the arms during the flight should be carefully planned in order to avoid self-collisions of the arms. In general, the trajectory generation and redundancy resolution strategies developed in Chapter 5 can be utilized to cope with the problem above.

3.3.2 Hardware/Software Architecture

The block diagram of the architecture and components related to the developed dual-arm system is shown in Fig. 3.9. Dynamixel X-series actuators are connected in daisy chain on each arm. They are further connected to 6 Port Dynamixel Power Hub, which connects them to the Odroid XU4 single board computer via U2D2 communication adapter. Dual VN5019 motor driver shield from Pololu together with Arduino Mega 2560 are used to control the rotation direction and speed of the DC motors based on the encoder readings. Arduino Mega 2560 is programmed in C using the Arduino IDE and it is connected to the Odroid XU4 via USB cable. The Dynamixel X-series actuators are powered from 3S LiPo battery by connecting them to 6 Port Dynamixel Power Hub. 6V 8A voltage regulator is used to transfer

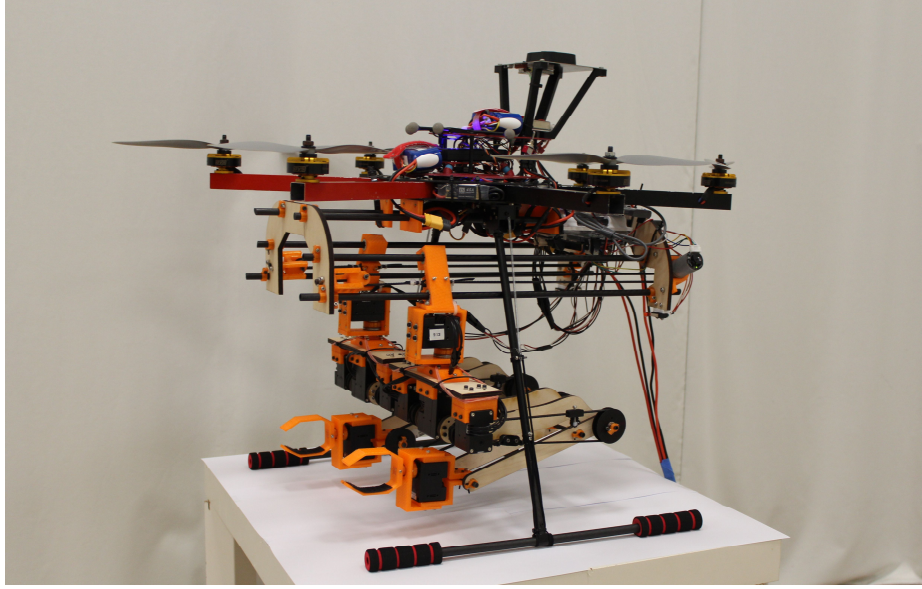


FIGURE 3.10: Developed dual-arm system mounted on the hexarotor platform.

required power from 3S LiPo battery to the dual VNH5019 motor driver shield and DC motors, while Odroid XU4 computer is powered through 5V 5A voltage regulator.

All control programs running on the Odroid XU4 single board computer (with Ubuntu 16.04OS) are developed in C/C++ using robot operating system (ROS) environment for software portability between different onboard computers. The operation manager can be considered as the high-level supervisor that performs the tasks chosen by the user from the ground control station. It collects the arms state and provides the desired goal to the dual-arm controller. The dual-arm controller generates the motion references and sends them back to the Dynamixel actuators and DC motor controller. The DC motor controller derives the desired speed and direction of the motors for the dual VNH5019 motor driver that produces required PWM signals to control DC motors.

3.3.3 Incorporating with Aerial Platform

In Fig. 3.10, the developed dual-arm system is mounted on the hexarotor platform, which was designed and fabricated by ourselves in the Flight Mechanics and Control Laboratory at Nanyang Technological University, Singapore. The dual-arm base is fixed between the landing gear legs right under the central hub to align the line of

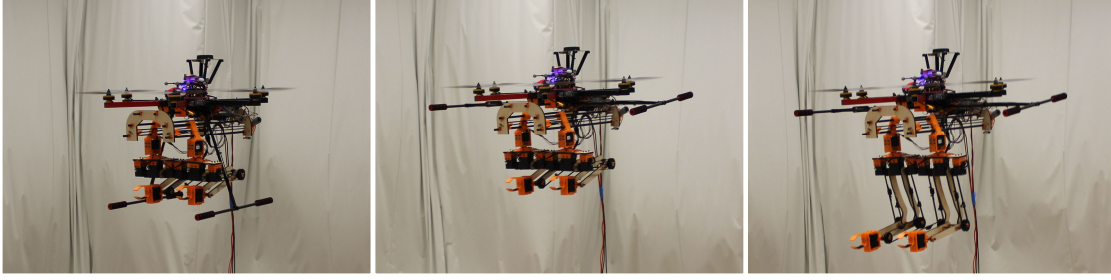


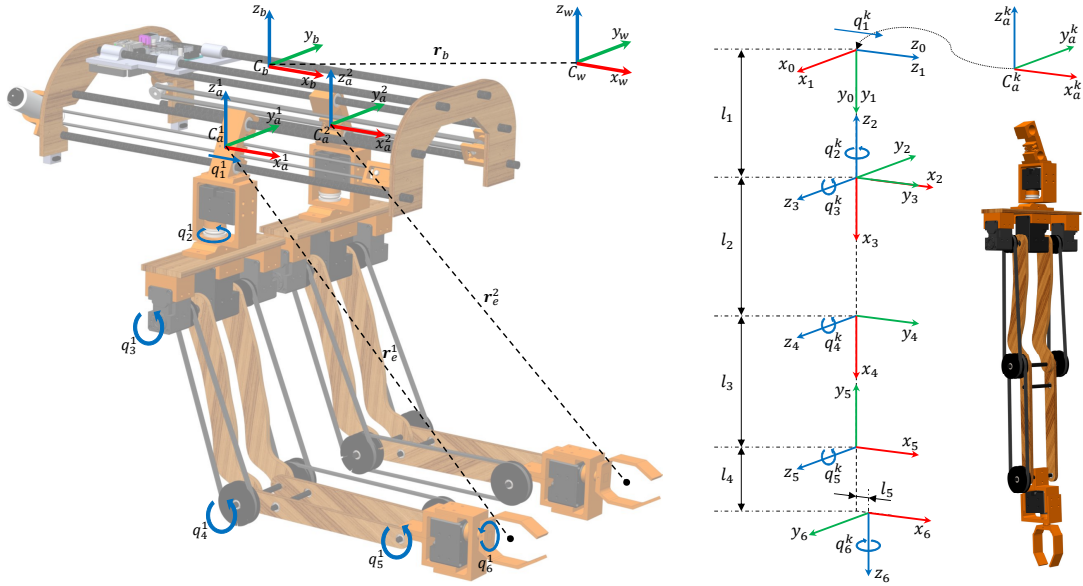
FIGURE 3.11: Sequence of images depicting the state change of the robotic arms from compact configuration (left) to the operational state (right). Landing gear down (left) and landing gear up (middle and right).

symmetry of the dual-arm system with the COG of the hexarotor platform. In order to internally balance the whole system, the position of the shoulder prismatic joints are moved forward as shown in Fig. 3.10. As it can be seen from Fig. 3.10, the arms are commanded to have the compact configuration by folding onto themselves to decrease the occupied space and avoid collision with the ground during the take-off and landing operations. The retractable landing gear is employed to increase the workspace of the arms during the flight. In Fig. 3.11, the sequence of images is shown where the robotic arms undergo the state change from compact configuration to the operational state. In order to evaluate the low weight and inertia design of the proposed dual-arm system, the common off-the-shelf autopilot (Pixhawk) is used to control the hexarotor platform, and no feedback from the robotic arms is provided to the autopilot.

3.3.4 Kinematic Parameters

3.3.4.1 Reference frames and kinematic coupling

The kinematic parameters and reference frames for the dual-arm system are shown in Fig. 3.12. The position of the joints is described as the vector $\mathbf{q}^k = [q_1^k, \dots, q_6^k]^T$, with $k = \{1, 2\}$ for the right and left arms, respectively. The reference frames of the arm joints are assigned according to the Denavit-Hartenberg (DH) convention as illustrated in Fig. 3.12b, in which the motion of the joints is shown in the positive direction. The DH parameters are presented in Table 3.3, in which $l_1 = 16.5$ cm, $l_2 = 20$ cm, $l_3 = 18.5$ cm, $l_4 = 11.5$ cm, and $l_5 = 1.5$ cm. The reference frame



(A) Kinematic parameters of the dual-arm system (B) Reference frames of the arm joints assigned according to DH convention. The arm is depicted in $\mathbf{q}^k = 0$ configuration

FIGURE 3.12: Kinematic parameters and reference frames for the dual-arm system.

C_a^k is fixed to the base of the k^{th} arm, while the reference frame C_b is the body-fixed frame with origin at the hexarotor COG. Due to the motion transmission mechanism and placement of the motors, there is a kinematic coupling among the arm joints that can be defined as follows:

$$\mathbf{q}_m^k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & (-1)^k & (-1)^k g_r & 0 & 0 \\ 0 & 0 & (-1)^{k+1} & (-1)^{k+1} g_r & (-1)^{k+1} g_r & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{q}^k = \mathbf{B}^k \mathbf{q}^k, \quad (3.1)$$

in which $\mathbf{q}_m^k \in \mathbb{R}^6$ describes the joint values in the motor space, $g_r = 1.5$ is the gear ratio and \mathbf{B}^k is the matrix that maps the joint values from the joint space to the motor space. The above relation is always invertible, since $\det(\mathbf{B}^k) = g_r^2 > 0$. It should be noted that since the DC motors are used to actuate the first joints on both arms, $q_{m_1}^k$ represents the linear position that these motors need to achieve.

3.3.4.2 Forward and inverse kinematics

The forward kinematic model of the k^{th} arm, which specifies the position and orientation of the end-effector, can be presented by the following transformation matrix:

$${}^a\mathbf{T}^k(\mathbf{q}^k) = {}^a_0\mathbf{T}^k\left(\prod_{i=1}^6 {}^{i-1}_i\mathbf{T}^k(q_i^k)\right) = \begin{bmatrix} {}^a_e\mathbf{R}^k(\mathbf{q}^k) & \mathbf{r}_e^k(\mathbf{q}^k) \\ 0 & 1 \end{bmatrix}, \quad (3.2)$$

in which ${}^a_e\mathbf{R}^k(\mathbf{q}^k) \in \mathbb{R}^{3 \times 3}$ and $\mathbf{r}_e^k(\mathbf{q}^k) \in \mathbb{R}^3$ describe the rotation matrix and the position vector of the end-effector of the k^{th} arm with respect to the C_a^k frame, ${}^{i-1}_i\mathbf{T}^k \in \mathbb{R}^{4 \times 4}$ is the transformation matrix corresponding to each joint and ${}^a_0\mathbf{T}^k \in \mathbb{R}^{4 \times 4}$ is the transformation matrix that defines the frame $\{0\}$ of the k^{th} arm relative to the frame C_a^k .

The closed-form analytical solution to the inverse kinematics is utilized to obtain the position of the joints of the k^{th} arm, given the desired position of the end-effector, $\mathbf{r}_e^k = [x_k, y_k, z_k]^T$, within the workspace of the arm, and the end-effector orientation, φ_k , which describes the angle with respect to the plane parallel to the ground plane. It should be noted that the position of the shoulder prismatic joint is considered as the parameter, $q_1^k = \rho$, which would be defined depending on the task. For instance, if the COG compensation is required as indicated in Section 3.2.4, the position of the shoulder prismatic joint can be selected in such a way that the dual-arm COG is aligned with the line of symmetry, while if it is required to increase the reach of the arm, this value can be selected to slide the arm in the forward or backward direction. In addition, since the wrist roll joint of the arm is only used to reorient the end-effector, it is kept zero throughout all experiments for simplicity, i.e., $q_6^k = 0$. The shoulder yaw joint defines the pointing direction of

TABLE 3.3: DH parameters of the arm.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	q_1	0
2	$\pi/2$	0	$-l_1$	$q_2 + \pi/2$
3	$\pi/2$	0	0	$q_3 - \pi/2$
4	0	l_2	0	q_4
5	0	l_3	0	$q_5 + \pi/2$
6	$\pi/2$	l_5	l_4	q_6

the arm, and it is equal to

$$q_2^k = \text{atan2}(y_k, x_k - q_1^k), \quad (3.3)$$

The elbow pitch joint can be obtained by considering the following equations

$$\begin{aligned} \cos(q_4^k) &= \frac{\Delta x^2 + \Delta z^2 - l_2^2 - l_3^2}{2l_2l_3}, \\ \sin(q_4^k) &= \pm \sqrt{1 - \cos^2(q_4^k)}, \end{aligned} \quad (3.4)$$

in which

$$\begin{aligned} \Delta x &= R - l_4 \cos(\varphi_k) + l_5 \sin(\varphi_k), \\ \Delta z &= |z_k| - l_1 + l_4 \sin(\varphi_k) + l_5 \cos(\varphi_k), \\ R &= \sqrt{(x_k - q_1^k)^2 + y_k^2}, \end{aligned}$$

and selecting $q_4^k \geq 0$, we can find the angle of the elbow pitch joint as follows

$$q_4^k = \text{atan2}(\sin(q_4^k), \cos(q_4^k)). \quad (3.5)$$

Knowing q_4^k , we can obtain the shoulder pitch joint in the following way

$$q_3^k = \pi/2 - (\gamma + \beta). \quad (3.6)$$

where

$$\begin{aligned} \gamma &= \text{atan2}(l_3 \sin(q_4^k), l_2 + l_3 \cos(q_4^k)), \\ \beta &= \text{atan2}(\Delta z, \Delta x). \end{aligned}$$

Finally, the wrist pitch joint angle is defined as follows

$$q_5^k = \varphi_k - (q_3^k + q_4^k) + \pi/2. \quad (3.7)$$

3.4 Single-Arm Aerial Manipulator

Since the single-arm aerial manipulator will be employed in Chapters 4 and 5 during simulation and experimental studies, it was decided to provide the brief description of the developed single-arm manipulator in this section.

3.4.1 Developed Single-Arm System

The schematic design of the single-arm aerial manipulator consists of three parts: a UAV, 4-DOF robotic arm, and an end-effector as displayed in Fig. 3.13a. As can be seen, Y6 coaxial tricopter is chosen as the UAV platform due to its payload capability (750g) and lightweight/customizable structure. As for manipulator, 4-DOF robotic arm is selected such that it can perform basic *pick* and *place* tasks. The robotic arm has four revolute joints with the *yaw-pitch-pitch-roll* sequence as illustrated in Fig. 3.13b. As for the end-effector, two jaws are combined together by a set of spur gears on each jaw which can be operated by a standard micro-servo motor. Such design can provide reliable strength and actuation as well as the capability to grasp the objects with basic shape.

3.4.1.1 Evaluation of Required Motorization

In this section, the assessment of motorization of the single-arm is presented. The provided analysis of motors is the simplified evaluation of motorization, but at the same time, it can be considered as a useful initial approach during the design process.

As mentioned earlier, the main critical factors during the selection of motors are their weight, size, torque, and power consumption. Hence, similarly as in the dual-arm case, Dynamixel servo actuators are employed for the single-arm. Dynamixel MX-64T and MX-106T are selected as the motors for the first and second joint, respectively. Although the combined mass of this two motors is around 55% of the entire mass of robotic arm, these two motors are selected to ensure that provided

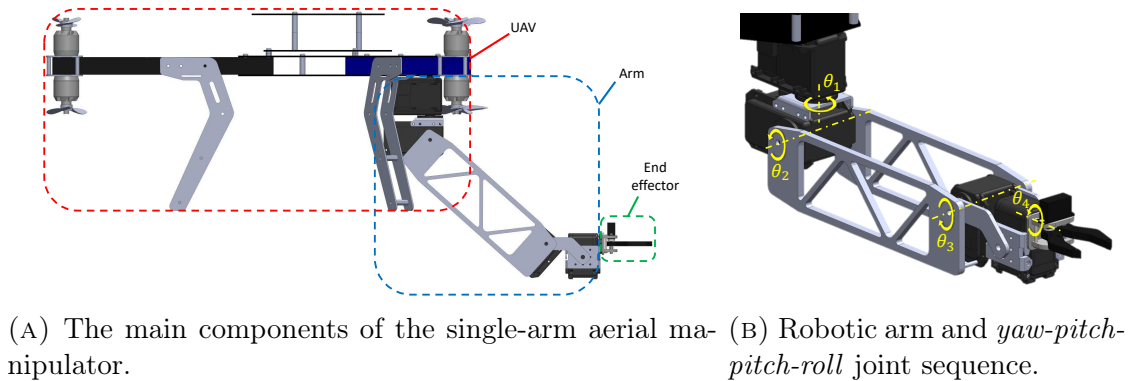


FIGURE 3.13: Develop single-arm aerial manipulator.

torques are enough to safely activate the manipulator. As for the third and fourth joint, the lighter versions of Dynamixel motors are chosen, namely MX-28T and AX-18A respectively. Selected motors should be able to provide required torques to each joint in the most unfavorable configuration of the robotic arm, e.g. the horizontally extended configuration. This can be written in the following form

$$\tau_{motor} \geq \tau_{dyn} + \tau_{grav} + \tau_{oth}, \quad (3.8)$$

where τ_{motor} is the motor torques, τ_{dyn} is the dynamic torques caused by accelerations of the manipulator and Coriolis effects, τ_{grav} is the torques originating with regard to the gravitational force caused by both mass of the arm and maximum payload, and τ_{oth} denotes the torques generated by contact forces, friction and other disturbances. Generally, the influences of τ_{dyn} and τ_{oth} are relatively smaller compared to τ_{grav} when the speeds involved are small. In addition, τ_{dyn} and τ_{oth} usually quite difficult to estimate, and therefore, it is assumed that if the operating torque of the motor is two times larger than τ_{grav} , then such motor is suitable for the proposed design. The parameters of motors can be seen in Table 3.4. The combined mass of the single-arm with all components, motors and the end-effector is 600g which is below the payload limitation of the UAV platform. The length of robotic arm from the first joint to the end-effector is 260mm when it is fully stretched. In addition, the first joint of the arm is fastened about 110mm far from the center of the UAV platform, and thus, increasing the total reach of the arm up to 370mm.

3.4.2 Experimental Setup

The architecture and components related to the single-arm system is shown in Fig. 3.14. Dynamixel actuators are connected in daisy chain. They are further

TABLE 3.4: Parameters of the manipulator motors.

Joint	Type	Stall torque	Motor	Motor weight [g]
1	Yaw	6.0 [N · m]	MX-64T	165
2	Pitch	8.4 [N · m]	MX-106T	165
3	Pitch	2.5 [N · m]	MX-28T	72
4	Roll	1.8 [N · m]	AX-18A	55.9

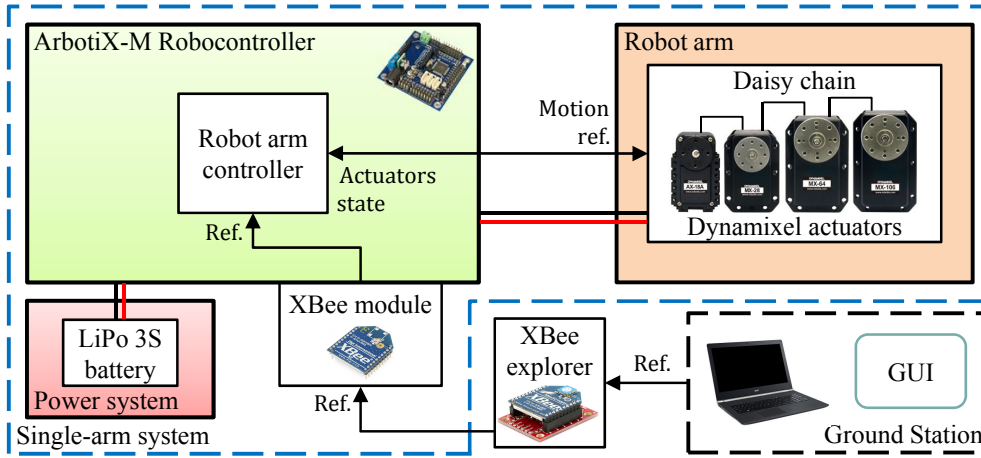


FIGURE 3.14: Architecture and components of the developed single-arm system.

connected to ArbotiX-M robocontroller, which is an Arduino compatible micro-controller. The communication between ground computer and ArbotiX-M robocontroller is realized via XBee wireless modules. The Dynamixel actuators and ArbotiX-M are powered from 3S LiPo battery. In Fig. 3.15, the real-time implementation of the single-arm aerial manipulator is shown. The coaxial tricopter is equipped with a Pixhawk autopilot that serves as the low-level controller, while the robotic arm is actuated by four Dynamixel servomotors, which are controlled by an ArbotiX-M robocontroller. The performance of the developed single-arm manipulator will be displayed during the experimental flight tests in Chapter 4.

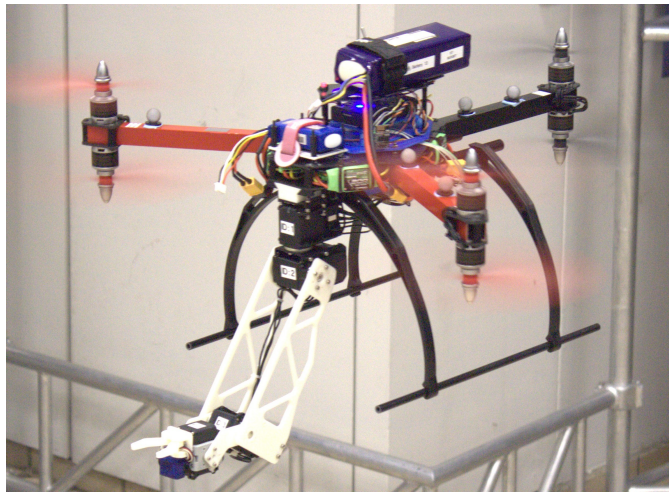


FIGURE 3.15: Single-arm aerial manipulator.

3.5 Experimental Validation

Extensive experimental studies are performed in an indoor environment and evaluated in the OptiTrack motion capture system laboratory to validate the performance of the proposed dual-arm system. Four scenarios are considered for the experimental tests. In the first scenario, in order to evaluate the low weight and inertia design of the proposed dual-arm system, the hexarotor platform is commanded to remain in hover (at $x = 0$ m, $y = 0$ m, $z = 2$ m), while the robotic arms are performing different motion sequences. The main goal of the first scenario is to show that the impact from the motion of the arms on the UAV stability is minor in contactless situation due to the low inertia and weight design. On the other hand, this influence is more evident when there is a grasped object that increases the variation of COG and inertia during the arms motion. Therefore, in the second scenario, the ability of the proposed design to dynamically align the dual-arm COG with the symmetry line of the UAV platform is validated. In this scenario, 0.25 kg aluminium rod is attached to the end-effectors, which are commanded to move forward and backward in the xz -plane by 25 cm three times, while the hexarotor platform is required to hover at the same position as in the first scenario. Note that the shoulders of the arm are kept in the *coupled* configuration in this scenario. In the third scenario, the positioning stability of UAM is evaluated when the arms are extended in the *coupled* and *decoupled* configurations (i.e., when the shoulder prismatic joints of the arms are coupled and decoupled, respectively). As we mentioned before, utilizing the prismatic joints can increase the reach of the arms by moving them to the front or back of the dual-arm base. However, when the arms are in the *coupled* configuration, their linear motion can result in the inclination of the UAV platform as indicated in Section 3.2.4 leading to the deterioration of the positioning stability. In particular, this can be the case when the arms are extended to operate in front of the wall. In order to address this case, the *decoupled* configuration of the proposed design can be exploited by moving one of the arms in the opposite direction to align the dual-arm COG with the line of symmetry, especially if only one arm is required to perform a task. Finally, the positioning accuracy of the developed dual-arm itself is discussed in the fourth scenario. The end-effectors of both arms are commanded to track the eight-shaped trajectory in the yz -plane, while their positions in the x -axis are kept constant. In order to evaluate accuracy

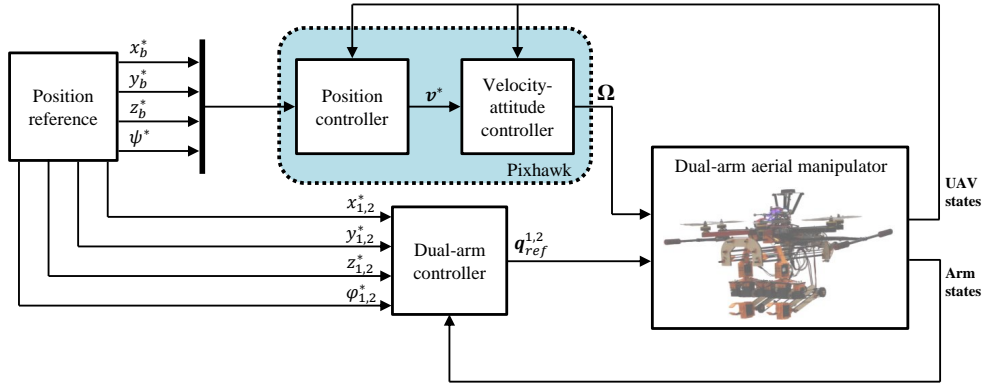


FIGURE 3.16: Block diagram of the control architecture.

and repeatability in the positioning of the end-effectors, the eight-shaped trajectory is performed three times by both arms. The following eight-shaped time-based trajectory is defined to coordinate the motion of the end-effectors:

$$\begin{aligned} x(t) &= 0.29 \text{ m}, \quad \varphi = 0, \\ y(t) &= \frac{\sqrt{2} \cos(t/4 + \pi/2)}{5(3 - \cos(t/2 + \pi))} \mp 0.1 \text{ m}, \\ z(t) &= \frac{\sin(-(t/2 + \pi))}{4(3 - \cos(t/2 + \pi))} - 0.35 \text{ m}, \end{aligned} \quad (3.9)$$

where ∓ 0.1 term is added to have the separate trajectory for each end-effector. The aforementioned trajectory is defined with respect to the body-fixed reference frame C_b . The block diagram of the overall control architecture during experimental studies is shown in Fig. 3.16. The video of the experimental flight tests can be accessed via this link (<https://youtu.be/1cUYWqkMCIM>).

3.5.1 Experimental Results

3.5.1.1 First scenario

In order to qualitatively assess the influence of the arms motion over the hexarotor stability, two cases are considered in this scenario. In the first case, the arms are fixed and kept in the operational state as shown in Fig. 3.17 (frame 1), while in the second case, the arms are moving and performing different motion sequences as depicted in Fig. 3.17 (frames 2-8). The experimental results corresponding to the first scenario are summarized in Fig. 3.18. In Fig. 3.18a and 3.18b, the hovering

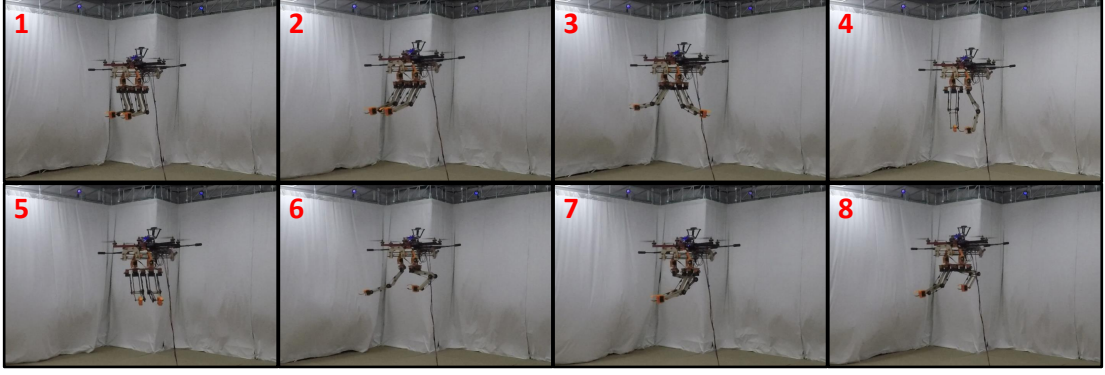


FIGURE 3.17: Sequence of images during the indoor experiments corresponding to the first scenario. Arms in the operational state (1), performing different motion trajectories (2-8).

performance of the hexarotor is illustrated, while the position (x , y , and z) response of the hexarotor COG is shown in Fig. 3.18c. It can be seen that the impact from the motion of the arms on the hexarotor stability is minor in contactless flight due to the low inertia and weight of the proposed dual-arm design. In particular, the motion of the arms does not cause significant pose variations of the hexarotor as depicted in Fig. 3.18c, since most of the dual-arm mass is concentrated near the hexarotor base. This can be also seen from the Euclidean error values presented in Fig. 3.18d and Table 3.5, which shows the root-mean-square error (RMSE) values. It is observed that the Euclidean error and RMSE values for both cases are similar, which validates the low weight and inertia features of the proposed dual-arm design. In addition, it can be seen that the well-tuned standard autopilot is capable to control the hexarotor in contactless situations regardless of the change in the COG and inertia terms caused by the motion of the arms. Finally, it should be noted that the effect of the arms motion over the hexarotor stability depends on the arms speed. Thus, this effect can be reduced at the expense of increasing the execution time of aerial manipulation missions.

TABLE 3.5: Euclidean RMSE for the first scenario (unit: cm).

Case	Fixed arms	Moving arms
RMSE	8.67	11.55

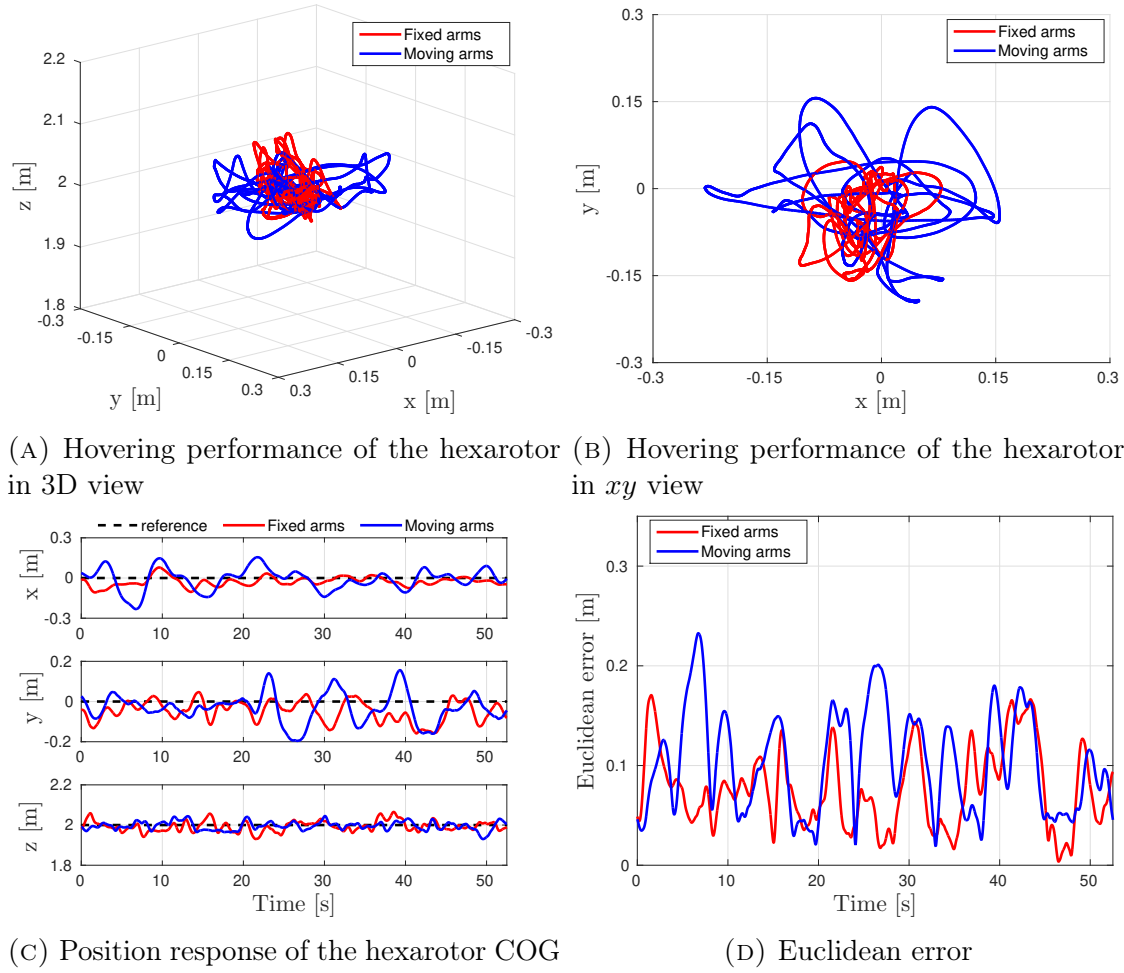


FIGURE 3.18: Experimental results for the first scenario.

3.5.1.2 Second scenario

In the first scenario, we have shown that the influence of the motion of the arms over the hexarotor stability in contactless situations is not significant due to the employed design. However, when the end-effectors grasp the object, the variation of the COG and inertia terms during the arms motion can cause notable deviations in the position of the hexarotor. Hence, in the second scenario, 0.25 kg aluminium rod is attached to the end-effectors to increase the variation of the COG and inertia terms during the arms motion. The sequence of images corresponding to the second scenario is shown in Fig. 3.19. The left column ($t = 0s$) of Fig. 3.19 displays the initial configuration of the robotic arms. The end-effectors are moved forward in the xz -plane by 25 cm as depicted in the middle column ($t = 3s$) of Fig. 3.19. Finally, the end-effectors are moved back to the initial position as shown in the right column ($t = 8s$) of Fig. 3.19. The above sequence is repeated two more times.

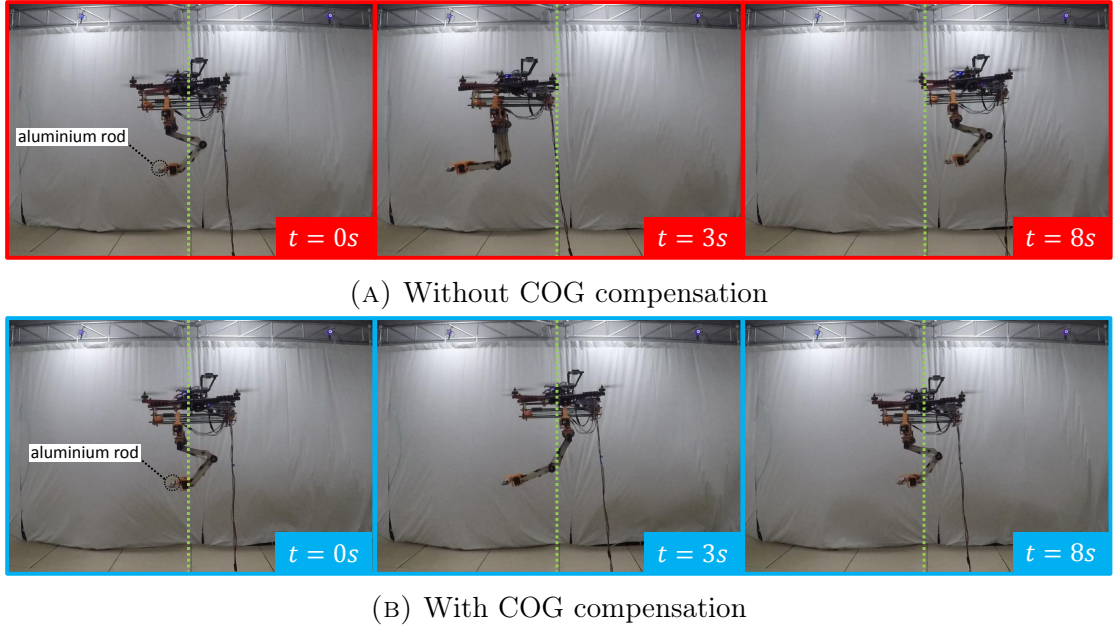


FIGURE 3.19: Sequence of images during the experiments corresponding to the second scenario.

There is also 2s waiting time between forward and backward motion of the end-effectors. Two cases are considered for this scenario as illustrated in Fig. 3.19. In particular, Fig. 3.19a shows the first case when the COG compensation mechanism of the proposed design is not activated, while the second case with the active COG compensation mechanism is presented in Fig. 3.19b. The experimental results corresponding to the second scenario are summarized in Fig. 3.20. The hovering performance of the hexarotor for the aforementioned cases is shown in Figs. 3.20a and 3.20b. It can be observed that the COG compensation mechanism of the proposed design yields better hovering performance compared to the case when the COG compensation mechanism is not activated. It is clearly visible from x -axis response of the hexarotor COG in Fig. 3.20c that the ability of the proposed design to dynamically align the dual-arm COG with the symmetry line of the hexarotor notably decreases the pose variation of the aerial robot. This can be also noted from Fig. 3.20d, which shows the Euclidean error. Finally, it is found that the RMSE value (see Table 3.6) is decreased by around 64% for the case when the COG compensation mechanism is active.

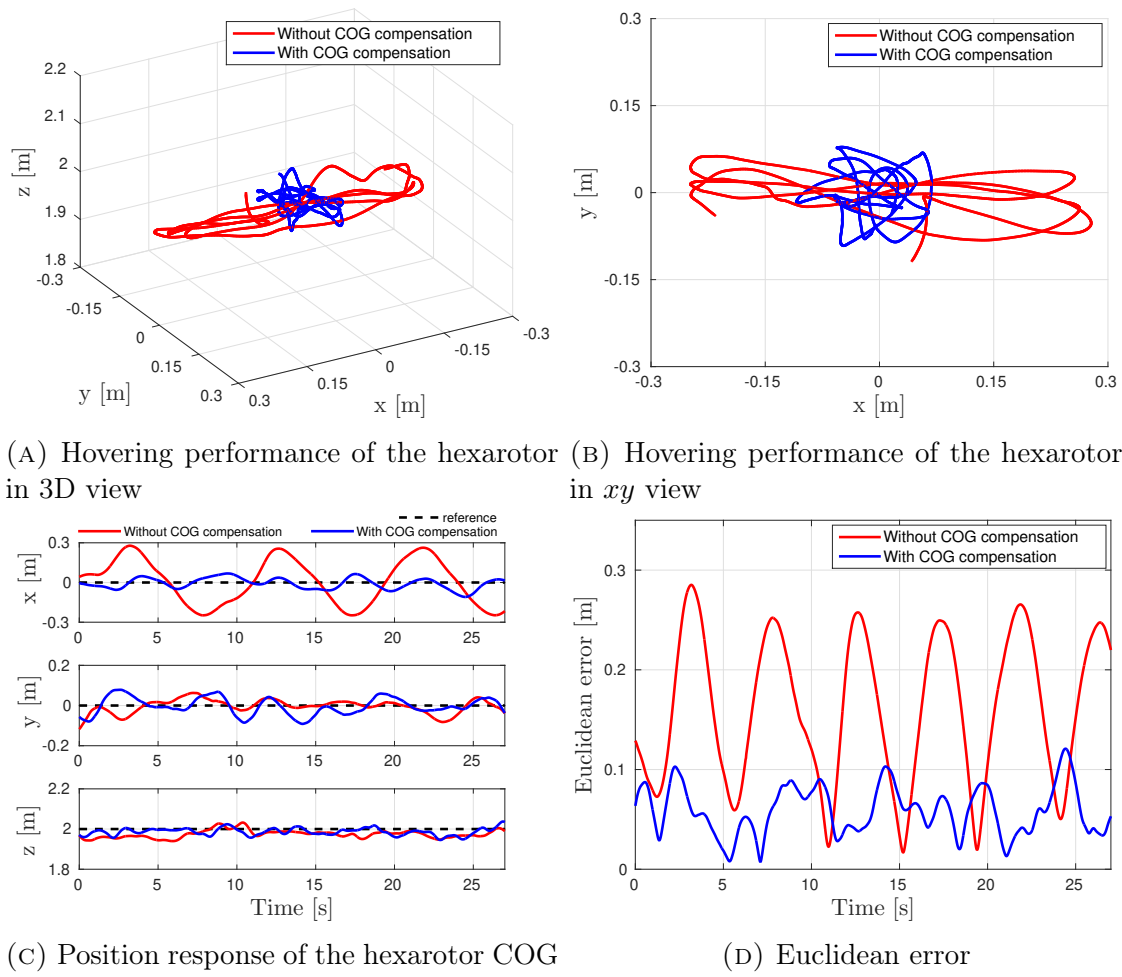


FIGURE 3.20: Experimental results for the second scenario.

3.5.1.3 Third scenario

The sequence of images corresponding to the third scenario is shown in Fig. 3.21. The top row (Fig. 3.21a) corresponds to the case when the arms are kept in the

TABLE 3.6: Euclidean RMSE for the second and third scenario (unit: cm).

Scenario	Case	RMSE
Second	Without COG compensation	17.74
	With COG compensation	6.40
Third	<i>Coupled</i> configuration	18.59
	<i>Decoupled</i> configuration	7.40

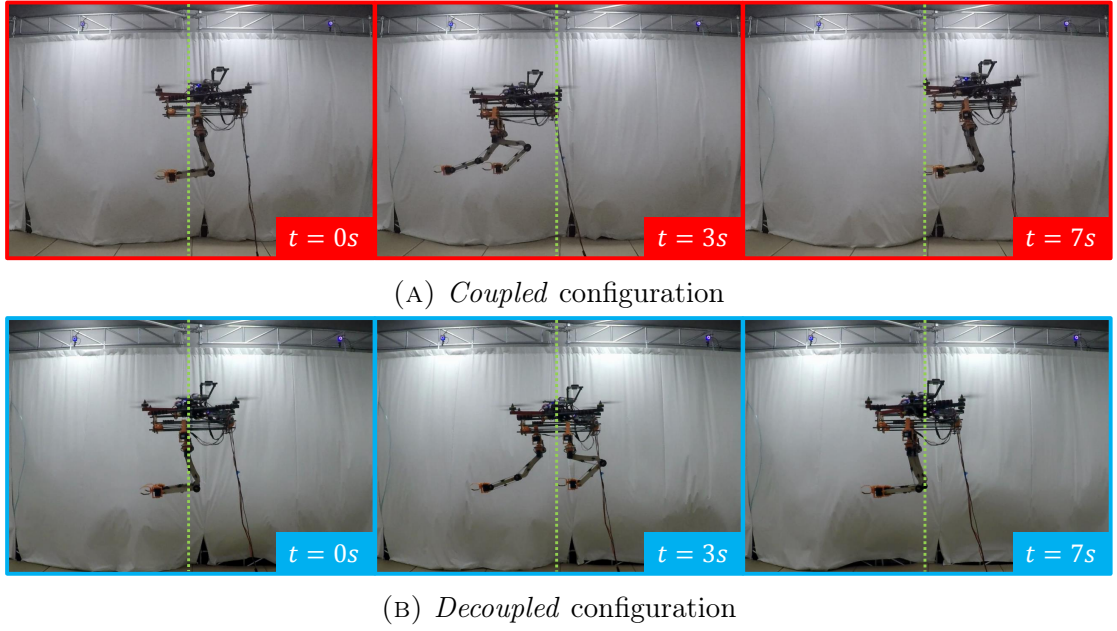


FIGURE 3.21: Sequence of images during the experiments corresponding to the third scenario.

coupled configuration, while the bottom row (Fig. 3.21b) displays the arms in the *decoupled* configuration. From the visual inspection of the images, it can be seen that when the arms are kept in the *coupled* configuration, their forward motion causes the pose variation of the hexarotor (see Fig. 3.21a). On the other hand, as depicted in Fig. 3.21b, the *decoupled* configuration of the proposed design allows to compensate the COG displacement by moving one of the arms in the opposite direction. The experimental results corresponding to the third scenario are summarized in Fig. 3.22. In Fig. 3.22a and 3.22b, the hovering performance of the hexarotor is presented, while Fig. 3.22c shows the position (x , y , and z) response of the hexarotor COG. It can be noted that the deviation in the position of the hexarotor is significantly reduced when the left arm is used to compensate the COG displacement caused by the motion of the right arm. This can be also seen from Fig. 3.22d and Table 3.6, which show the Euclidean error and RMSE values, respectively. It is observed that the RMSE value is decreased by around 60% when the *decoupled* configuration is used. In general, this scenario resembles the case when the arms are extended to operate in front of the wall. The results show that when only one arm is required to perform the given task, the *decoupled* configuration of the proposed design allows to use the second arm as the counterweight to align the dual-arm COG with the line of symmetry improving the positioning stability of UAM.

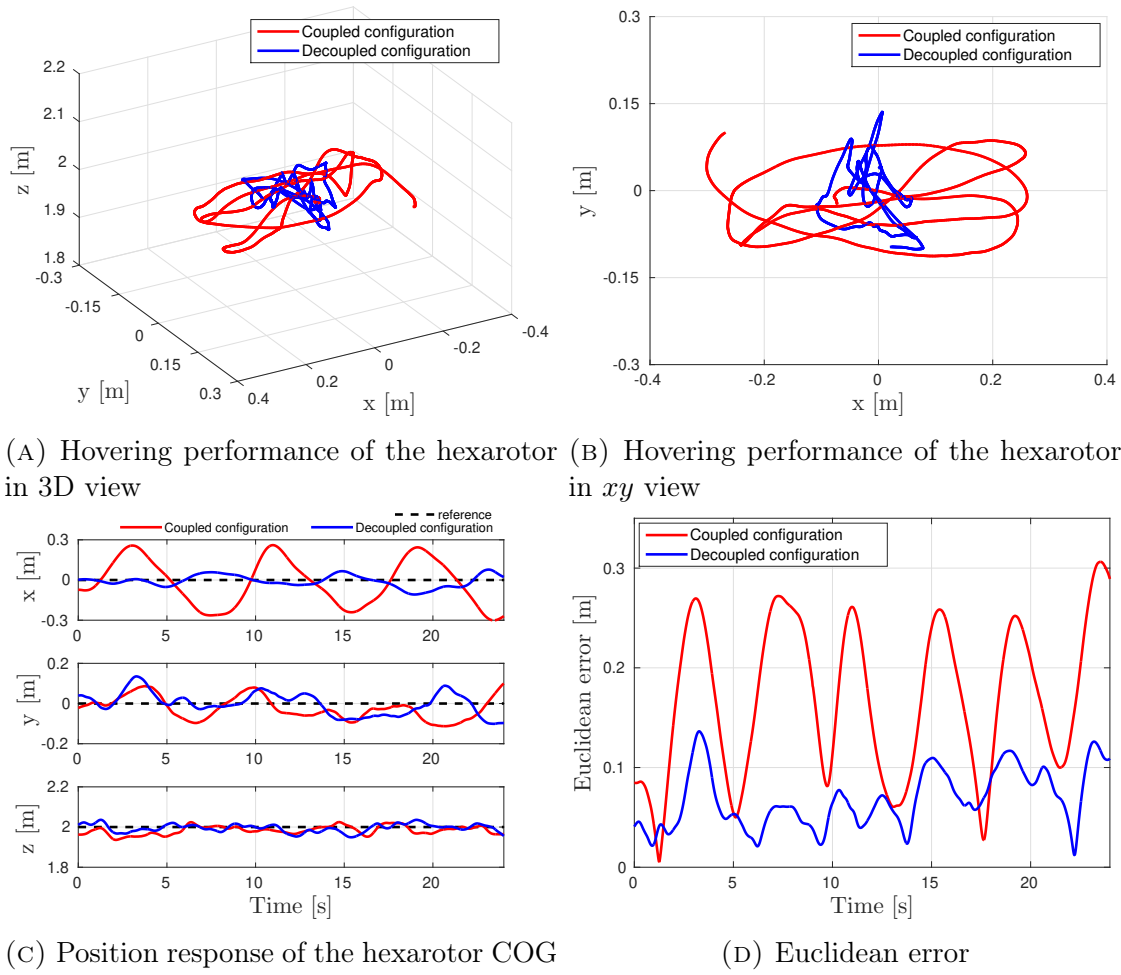


FIGURE 3.22: Experimental results for the third scenario.

3.5.1.4 Fourth scenario

The main goal of the fourth scenario is verifying the accuracy and repeatability in the positioning of the end-effectors of the developed dual-arm setup. As we mentioned before, the end-effectors of both arms are commanded to track the eight-shaped time-based trajectory defined in equation (3.9). The position of the end-effectors is obtained by utilizing the OptiTrack motion capture system as shown in Fig. 3.23. The reflective markers are attached to the dual-arm setup such that the obtained results can be presented with respect to the body-fixed reference frame C_b . The experimental results corresponding to the fourth scenario are summarized in Fig. 3.24. The trajectory tracking performance of the end-effectors in the yz -plane and xz -plane is presented in Figs. 3.24a and 3.24b, respectively, while the position (x , y , and z) responses of the end-effectors are given in Fig. 3.24c. It can be observed that the deviation is within 0.01 m. This is clearly visible from the

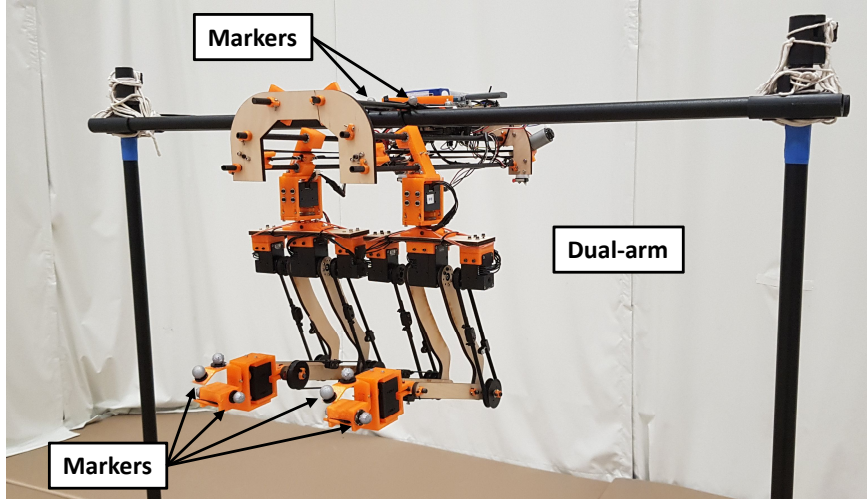


FIGURE 3.23: Developed dual-arm system during execution of the eight-shaped trajectory by the end-effectors in the fourth scenario. Notice that reflective markers are attached to the end-effectors and C_b frame to provide visual feedback to the OptiTrack system.

x -axis response in Fig. 3.24b and from the position error values depicted in Fig. 3.24d. Hence, it can be noted that the developed dual-arm setup has satisfactory positioning accuracy itself.

3.6 Summary

In this chapter, we have presented the design, fabrication and experimental validation of the novel, lightweight, low inertia dual-arm manipulator with the COG balancing mechanism specifically designed for aerial manipulation missions. The main goal of this study is to design the dual-arm manipulator that can be easily integrated with the multirotors, which utilize standard autopilots, opening the door for the widespread use of dual-arm manipulators with the commercially available UAV platforms. The developed system consists of two arms with 6 DOFs each and weighs 2.5 kg in total with a maximum payload of 1.0 kg per arm. The timing belt-based transmission mechanism is utilized to decrease the inertia of the arms by placing the actuators near the UAV base. The proposed design of the dual-arm system employs the shoulder prismatic joints to introduce the linear sliding motion that produces several distinctive features. In particular, each arm has the ability to adjust its COG that allows to notably mitigate the influence of the arm motions over the multirotor stability. Furthermore, introducing the prismatic joints

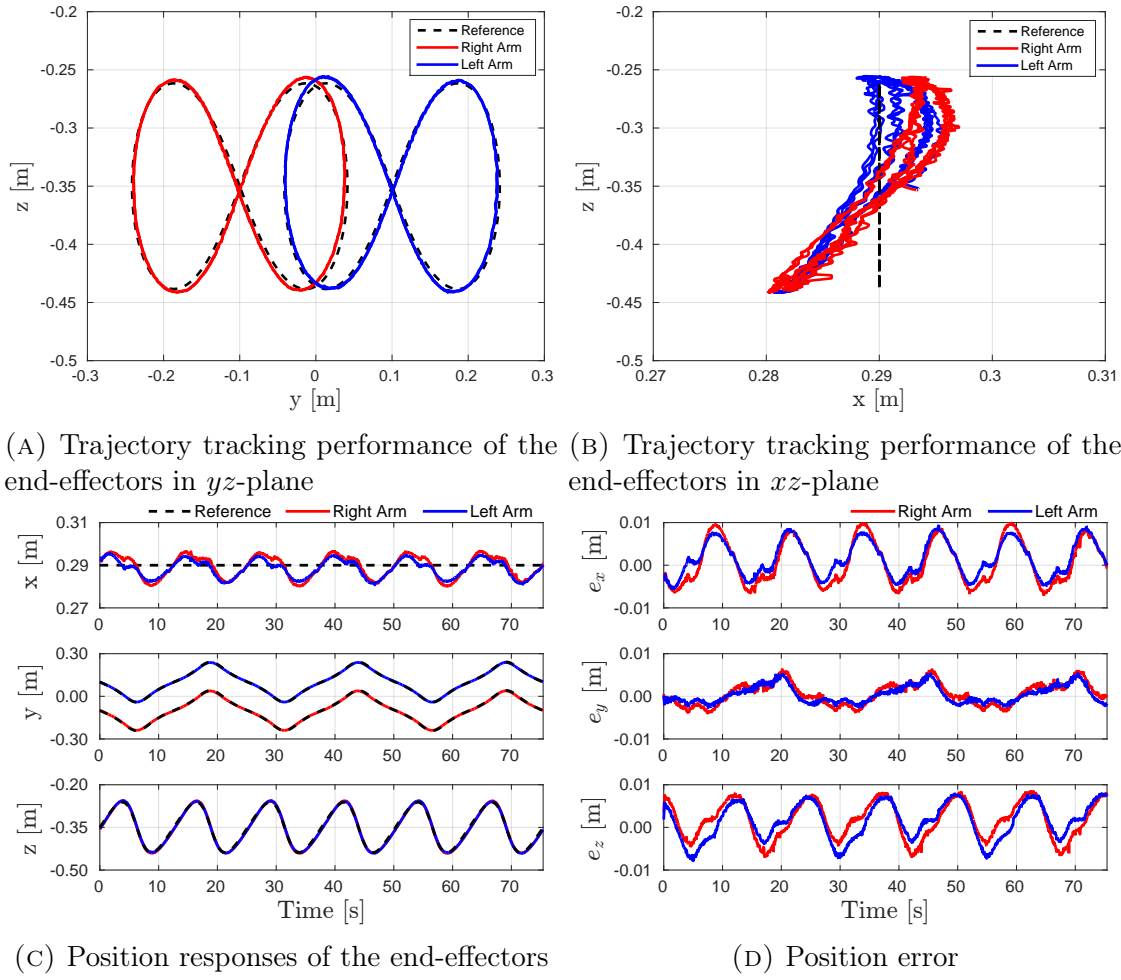


FIGURE 3.24: Experimental results for the fourth scenario.

increases the operation volume and reach of the arms. Moreover, the *decoupled* configuration of the arms enables to simultaneously perform different tasks. Extensive experimental flight tests have been performed to evaluate the proposed dual-arm design with the hexarotor equipped with the off-the-shelf autopilot. The results have illustrated that the influence of the motion of the arms over the hexarotor stability is minor in contactless flight due to the low inertia and weight of the proposed dual-arm design. In addition, the COG compensation mechanism has been validated to reduce the influence of the arms motion even when there is a grasped object, which increases the variation of COG and inertia during the arms motion.

Although the results demonstrate the efficacy of the proposed dual-arm design, it is essential to provide the limitation of the design. In particular, the proposed COG balancing mechanism does not address the complete stabilization of UAM, since the mechanical limits of the proposed design during COG compensation can

be reached depending on the desired configuration of the arms and grasped load, making infeasible to fully compensate the COG displacement. Moreover, there is the velocity limitation of utilized DC motors, which hinders the COG compensation during the fast motion of the arms. Therefore, in addition to the proposed COG balancing mechanism, a control approach that can compensate the internal disturbances caused by the motion of the arm(s) and external disturbances such as wind gust is desired. Hence, such control strategy will be proposed and discussed in the next chapter.

Chapter 4

Learning Control Strategy for Unmanned Aerial Manipulators

4.1 Introduction

This chapter presents an artificial intelligence-based control approach, the fusion of artificial neural networks and type-2 fuzzy logic controllers, namely type-2 fuzzy neural networks, for the outer adaptive position controller of UAMs. In particular, T2FNN with novel elliptic type-2 MFs, which is trained by the SMC theory-based learning algorithm, is developed for the control of UAMs under time-varying working conditions. The feedback-error-learning (FEL) scheme [132] is employed in which the proposed T2FNN controller works in parallel with a conventional PD controller. In this scheme, the PD controller's output is utilized as a learning error to train T2FNN so that it can become a nonlinear regulator which assists a conventional parallel controller (in our case the PD controller) to obtain the desired system response. The presented strategy eliminates the need for precise tuning of conventional controllers by learning from system dynamics and compensating disturbances online. The proposed control approach can be employed by both single and dual-arm aerial manipulators due to the adaptive learning capabilities of the T2FNN structure, which does not require the precise model of the system under control. In this work, the developed control strategy is tested for the control of the coaxial tricopter equipped with the 4-DOF robotic arm in the absence and presence of wind gust conditions.

Extensive simulations in ROS and Gazebo environment are conducted to evaluate and verify the performance of the proposed controller with the conventional PD controller as well as with its type-1 counterpart (T1FNN). In addition, in order to further validate the proposed controller, the experimental flight tests are performed in the indoor environment, and the performance comparison of the developed controller with PD/PID controllers is presented. The results demonstrate the efficacy and efficiency of the proposed control strategy to compensate for the disturbances caused by the robotic arm motion and wind gust, and thus leading to better flight behavior. The video of the experimental flight tests can be accessed via this link (<https://youtu.be/qTdNbm0Xy44>).

The remainder of this chapter is organized as follows. In Section 4.2, the learning control strategy for UAMs is described. Section 4.3 contains simulation results that evaluate and verify the performance of the proposed control solution. In Section 4.4, experimental flight tests are presented to validate the proposed controller. Lastly, conclusions and a summary of this chapter are provided in Section 4.5.

4.2 Fuzzy-Neuro Control Approach

4.2.1 Control Architecture

The overall structure of the control scheme for the UAM is depicted in Fig. 4.1. It is based on two interconnected control loops which track the corresponding motion references. The outer loop (position and manipulator control) is responsible for the position control of UAV platform and manipulator joints, respectively, while the inner loop (velocity-attitude control) is in charge of the velocity tracking and attitude stabilization of UAV platform. In this work, the desired position of UAV $\mathbf{r}_b^* = [x^* \ y^* \ z^*]^T$ and the manipulator joints $\Theta^* = [\theta_1^* \ \theta_2^* \ \theta_3^* \ \theta_4^*]^T$ are selected as the reference to control the overall system as shown in Fig. 4.1. The explicit representation of the PD+FNN block in Fig. 4.1 is shown in Fig. 4.2.

The position control block incorporates three independent sub-controllers to track the position reference of the coaxial tricopter on x , y , and z axes. The input to this control block is the position error measurements $\mathbf{e}_p = [e_x \ e_y \ e_z]^T$ which are

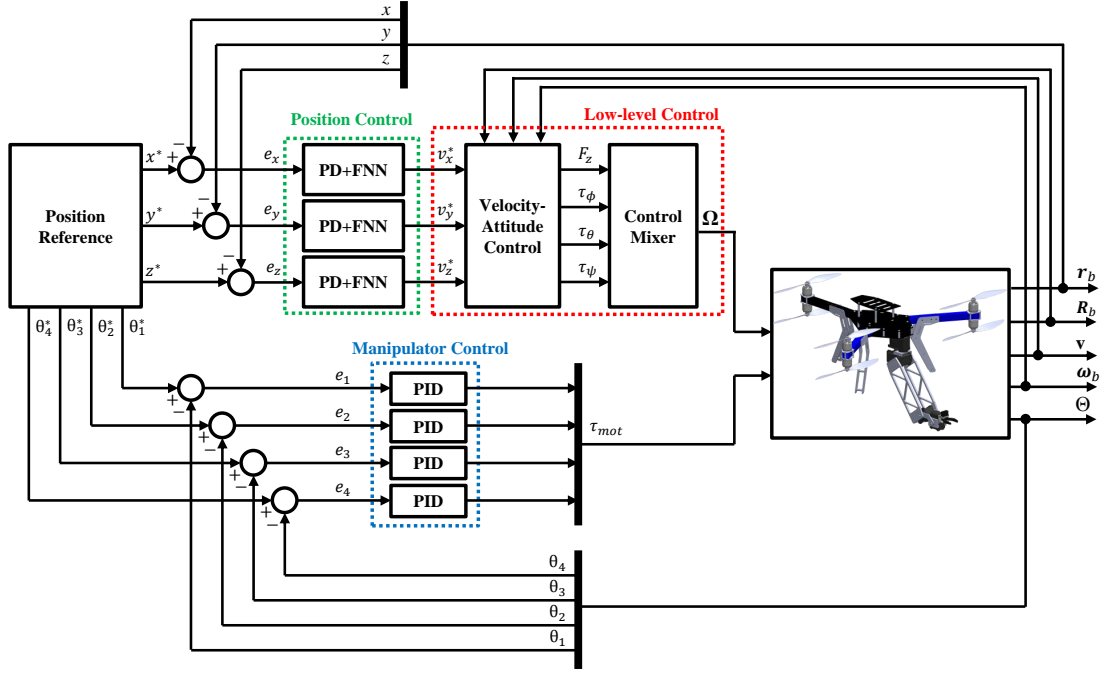


FIGURE 4.1: Block diagram of the control scheme.

defined as

$$\begin{cases} e_x = x^* - x \\ e_y = y^* - y \\ e_z = z^* - z, \end{cases} \quad (4.1)$$

while the output of this block is the desired linear velocity $\mathbf{v}^* = [v_x^* \ v_y^* \ v_z^*]^T$. Intelligent control approach based on T2FNN for position control of the aerial robot will be described in detail in the Section 4.2.2. It should be noted that the proposed intelligent controller treats the couplings between the axes as a disturbance, and tries to compensate them without knowing the interaction model.

Manipulator control consists of four independent joint control loops based on the conventional PID controller. The desired angular displacement θ_i^* ($i = 1, 2, 3, 4$) for each joint is used as reference signals for deriving the torques acting on the manipulator joints.

For the velocity-attitude control, the nonlinear geometric controller on the special Euclidean group $SE(3)$ can be used [133]. The input to this control block is the desired velocity \mathbf{v}^* and the output of this block is the control variables $[F_z \ \tau_\phi \ \tau_\theta \ \tau_\psi]^T$.

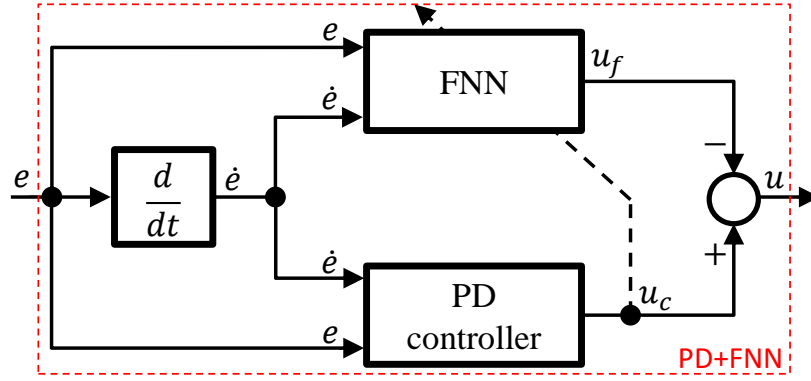


FIGURE 4.2: Structure of the PD+FNN block.

4.2.2 Fuzzy Neural Network Structure

In the proposed method which is shown in Fig. 4.2, the conventional PD controller is working in parallel with the FNN controller (FNN block on Fig. 4.2), which can represent T2FNN or T1FNN. In this work, the structure of the FNN block implements a Takagi–Sugeno–Kang (TSK) fuzzy model for the T2FNN case as illustrated in Fig. 4.3. The PD controller is utilized as an ordinary feedback controller to guarantee the stability of the system and provide enough time for the initialization of the learning process of T2FNN without going into the unstable working region. In this way, T2FNN is supposed to learn online from the system dynamics and take over the control responsibilities of the system. The PD control law can be described in the following form

$$u_c = k_p e + k_d \dot{e}. \quad (4.2)$$

in which k_d and k_p are some positive constants corresponding the PD controller gains, while e is the feedback error. It is to be noted that the output of the PD controller u_c is used as the learning error to train T2FNN.

4.2.2.1 Fuzzy-Neuro Inference System

In this work, the proposed structure of T2FNN includes two inputs, $x_1 = e$ and $x_2 = \dot{e}$, and one output u_f . TSK A2-C0 fuzzy model is utilized in which the antecedent/premise part is the type-2 fuzzy set and the consequent part is composed of only crisp numbers [134]. Representing the consequent part as crisp numbers notably reduces the computational burden of type-2 fuzzy systems. The structure

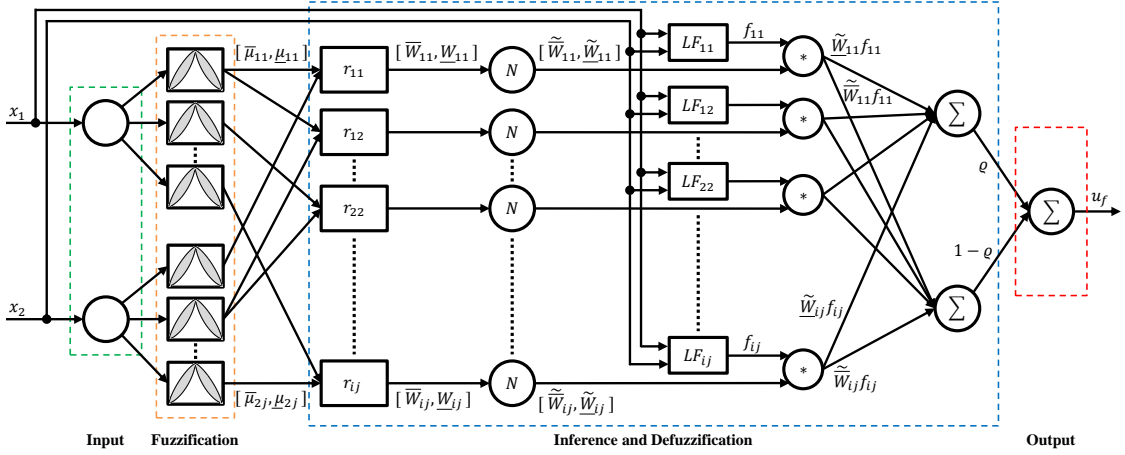


FIGURE 4.3: Structure of T2FNN with two inputs.

of A2-C0 fuzzy system is shown in Fig. 4.3. The r_{ij} rule of a first-order TSK model with x_1 and x_2 input variables can be described in the following way ($i = 1, \dots, I$ and $j = 1, \dots, J$)

$$r_{ij} \text{ Rule: IF } x_1 \text{ is } \widetilde{M}_{1i} \text{ and } x_2 \text{ is } \widetilde{M}_{2j}, \text{ THEN } f_{ij} = l_i x_1 + m_j x_2 + g_{ij},$$

where f_{ij} is a linear function (LF) of the input variables representing the consequent part with J and I corresponding to the number of MFs for x_2 and x_1 , respectively. The parameters l_i , m_j , and g_{ij} are the coefficients of the linear function in the consequent part for r_{ij} rule, while \widetilde{M}_{1i} and \widetilde{M}_{2j} are type-2 fuzzy sets. For the first input, we have upper and lower MFs represented as $\bar{\mu}_{1i}$ and $\underline{\mu}_{1i}$, respectively. Likewise, for the second input, upper and lower MFs are defined as $\bar{\mu}_{2j}$ and $\underline{\mu}_{2j}$, respectively. In this study, the coefficients l_i and m_j in the r_{ij} rule are assumed to be equal to zero, which is a widely-utilized simplification that leads to a zeroth-order TSK model, i.e., $f_{ij} = g_{ij}$.

The firing strength of the r_{ij} rule is determined by using the T -norm (multiplication) of the MFs in the antecedent/premise part

$$\begin{cases} \bar{W}_{ij} = \bar{\mu}_{1i}(x_1)\bar{\mu}_{2j}(x_2) \\ \underline{W}_{ij} = \underline{\mu}_{1i}(x_1)\underline{\mu}_{2j}(x_2), \end{cases} \quad (4.3)$$

in which elliptic type-2 fuzzy MFs are employed and will be described in detail in the Section 4.2.2.2. In addition, the main reason that the T -norm is chosen

as multiplication is to ensure that the mathematical model is derivable, which is required by the majority of neural network learning algorithms [135].

Finally, the inference engine proposed in [134] is employed to acquire a crisp output for a practical implementation. Hence, the output signal u_f is computed by utilizing the normalized values of the firing strength, $\widetilde{\overline{W}}_{ij}$ and $\widetilde{\underline{W}}_{ij}$, in the following form

$$u_f = \varrho \sum_{i=1}^I \sum_{j=1}^J f_{ij} \widetilde{\overline{W}}_{ij} + (1 - \varrho) \sum_{i=1}^I \sum_{j=1}^J f_{ij} \widetilde{\underline{W}}_{ij}, \quad (4.4)$$

where ϱ is the parameter that defines the sharing contribution of the lower and upper MFs [134], and it is selected to be equal to 0.5, while $\widetilde{\overline{W}}_{ij}$ and $\widetilde{\underline{W}}_{ij}$ are expressed as follows

$$\widetilde{\overline{W}}_{ij} = \frac{\overline{W}_{ij}}{\sum_{i=1}^I \sum_{j=1}^J \overline{W}_{ij}}, \quad \widetilde{\underline{W}}_{ij} = \frac{\underline{W}_{ij}}{\sum_{i=1}^I \sum_{j=1}^J \underline{W}_{ij}}. \quad (4.5)$$

Thus, the control input u to the system to be controlled is determined as follows

$$u = u_c - u_f, \quad (4.6)$$

where u_f and u_c are the control signals produced by the T2FNN controller and the PD controller, respectively.

4.2.2.2 Elliptic Type-2 Fuzzy MFs

In this study, novel elliptic type-2 fuzzy MFs are selected to obtain better control performance and noise handling capability. These type-2 MFs have already demonstrated satisfactory results for identification [112] and control [136] purposes. The upper and lower MFs are computed in the similar manner as the triangular MFs and they are expressed as follows

$$\overline{\mu}(x) = \begin{cases} \left(1 - \left|\frac{x-c}{d}\right|^{a_1}\right)^{1/a_1} & \text{if } c - d < x < c + d \\ 0 & \text{otherwise} \end{cases} \quad (4.7)$$

$$\underline{\mu}(x) = \begin{cases} \left(1 - \left|\frac{x-c}{d}\right|^{a_2}\right)^{1/a_2} & \text{if } c - d < x < c + d \\ 0 & \text{otherwise} \end{cases} \quad (4.8)$$

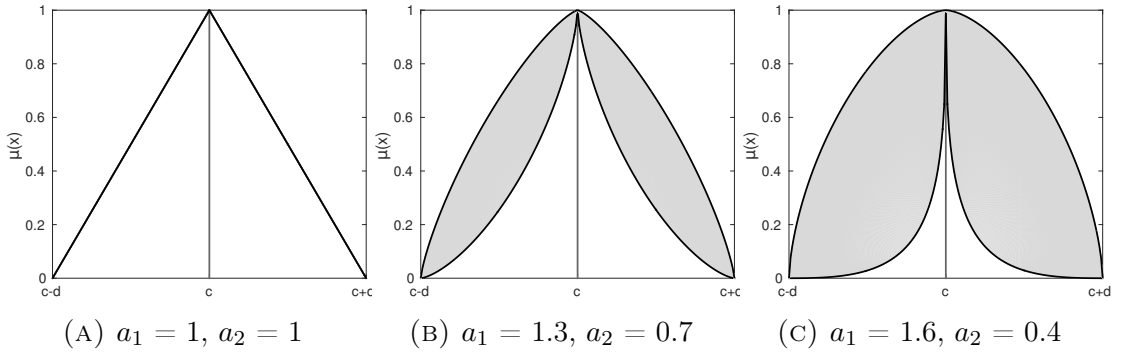


FIGURE 4.4: Shapes of elliptic type-2 MFs with different values for a_1 and a_2 .

in which x is the input variable, d and c are the width and the center of the MF. The uncertainty width of the elliptic MFs are determined by a_1 and a_2 , which should be chosen in the following way

$$\begin{cases} a_1 > 1, \\ 0 < a_2 < 1. \end{cases} \quad (4.9)$$

In Fig. 4.4 the different shapes of the proposed MFs are shown. When the parameters of the MFs are chosen as $(a_1, a_2) = (1, 1)$ the shape of type-2 MF is transformed to a type-1 triangular MF as shown in Fig. 4.4(a). In addition to (4.9), in [29], it is advised to select a_1 and a_2 as $a_1 < 2$ and $a_2 > 0.5$. The above mentioned parameters c , d , a_1 and a_2 can be tuned in an adaptive manner or they can be chosen as some constants.

In order to explain the uncertainty representation of elliptic MFs, let us consider Fig. 4.4b. It can be seen that the expert, who decides the center values of the MFs, is very certain that the particular input around the center belongs to the particular fuzzy set. This is something that can be expected from the expert since he/she is the person who selects the center values of the MFs. However, as we are moving from center towards the right ($c + d$) or left ($c - d$) end of support, the uncertainty in the expert's viewpoint increases. It should be noted that after the halfway, the experts again starts to be more certain that a particular input does not belong to this fuzzy set. The main reason for such modeling is that we believe that this way of uncertainty representation is closer to the human way of thinking compared to triangular or Gaussian MFs.

According to the control scheme in Fig. 4.2, where the PD controller is responsible

to ensure stability of the system in compact space, it is assumed that the two input signals, x_1 and x_2 , and their respective time derivatives, \dot{x}_1 and \dot{x}_2 , cannot have infinite values [136]. Hence, they can be considered bounded as follows

$$|x_1| < B_x, \quad |x_2| < B_x, \quad |\dot{x}_1| < B_{\dot{x}}, \quad |\dot{x}_2| < B_{\dot{x}}, \quad (4.10)$$

where $B_x, B_{\dot{x}} > 0$ are assumed to be some real constants.

Based on the same arguments, the parameters c and d of the elliptic type-2 fuzzy MFs are considered bounded too

$$B_{d,\min} < \|\mathbf{d}_1\|, \quad B_{d,\min} < \|\mathbf{d}_2\|, \quad \|\mathbf{c}_1\| < B_c, \quad \|\mathbf{c}_2\| < B_c, \quad (4.11)$$

where $\mathbf{d}_1 = [d_{11}, \dots, d_{1i}]^T$, $\mathbf{d}_2 = [d_{21}, \dots, d_{2j}]^T$, $\mathbf{c}_1 = [c_{11}, \dots, c_{1i}]^T$ and $\mathbf{c}_2 = [c_{21}, \dots, c_{2j}]^T$, and $B_{d,\min}, B_c > 0$ are some real constants. Similarly, due to physical constraints, the value of the linear function of the input variables representing the consequent part, f_{ij} , can be considered bounded as well

$$|f_{ij}| < B_f \quad (4.12)$$

for some real constant $B_f > 0$.

In addition, it should be noted that $0 < \widetilde{\overline{W}}_{ij} \leq 1$ and $0 < \widetilde{\underline{W}}_{ij} \leq 1$. It is also evident that $\sum_{i=1}^I \sum_{j=1}^J \widetilde{\overline{W}}_{ij} = 1$ and $\sum_{i=1}^I \sum_{j=1}^J \widetilde{\underline{W}}_{ij} = 1$. Finally, it is also considered that u and \dot{u} will also be bounded signals

$$|u| < B_u, \quad |\dot{u}| < B_{\dot{u}}, \quad (4.13)$$

where $B_u, B_{\dot{u}} > 0$ are some real constants.

4.2.3 Sliding Mode Control Theory-Based Learning Algorithm

In this section, the SMC theory-based parameter update rules for T2FNN with elliptic type-2 fuzzy MFs are considered. The SMC theory-based approaches can ensure the robustness of the system to external disturbances and parameter uncertainties, and thus they are widely-utilized for nonlinear systems applications

[137]. Moreover, apart from making the system more robust, the SMC theory-based learning algorithms can also provide faster convergence speed than the conventional learning techniques (e.g., the gradient-based methods) in online training of FNNs [29, 138]. The SMC theory-based framework is designed by choosing a suitable sliding manifold and restricting the motion of the system on it so that desired system response can be obtained [137]. Hence, by employing the principles of the SMC theory [139], the zero value of the learning error $u_c(t)$ can be represented as a time-varying sliding surface S_c in the following way

$$S_c(u_f, u) = u_c(t) = u_f(t) + u(t) = 0. \quad (4.14)$$

The above equation represents the sliding manifold, which acts as a guideline to train the T2FNN parameters. In particular, it works as the condition that the T2FNN structure is trained to become the nonlinear regulator which assists the PD controller so that desired performance of the system can be obtained. Hence, the sliding surface for the nonlinear system under control is given by

$$S_p(e, \dot{e}) = \dot{e} + \lambda e, \quad (4.15)$$

where λ is a parameter which defines the slope of the sliding surface.

In order to obtain the sliding motion on the sliding surface $S_c(u_f, u) = u_c(t) = 0$ after a finite time t_h , the condition $S_c(t)\dot{S}_c(t) = u_c(t)\dot{u}_c(t) < 0$ should be fulfilled for all t in some nontrivial semi-open sub-interval of time $[t, t_h) \subset (0, t_h)$ [138]. Therefore, it is desirable to employ the online learning algorithm or adaptation law for consequent and premise parts of T2FNN to enforce the aforementioned sliding mode condition. It should be noted that in different tests, it is observed that the relative improvement in the performance for the case where both consequent and premise parts of T2FNN are updated is small, while the corresponding computation time is considerably larger than the case when only consequent part of T2FNN is updated [140]. In addition, in [29], it is noticed that the output of FNN is quite sensitive to the changes in the parameters of the consequent part. Therefore, in this work, it is decided to update only the consequent part of T2FNN. Hence, similarly as in [29], the following adaptation law is used to update the time-varying

parameter of the consequent part f_{ij}

$$\dot{f}_{ij} = -\frac{\varrho \widetilde{W}_{ij} + (1 - \varrho) \widetilde{\overline{W}}_{ij}}{\Pi^T \Pi} \alpha \text{sgn}(u_c), \quad (4.16)$$

where

$$\Pi = \left(\sum_{i=1}^I \sum_{j=1}^J (\varrho \widetilde{W}_{ij} + (1 - \varrho) \widetilde{\overline{W}}_{ij}) \right). \quad (4.17)$$

In the above, the adaptive learning rate $\alpha > 0$ is updated as follows

$$\dot{\alpha} = 2\gamma |u_c| - \nu \gamma \alpha, \quad (4.18)$$

where γ and ν need to be selected as positive. As a result, the considered learning algorithm creates a sliding motion in terms of the T2FNN parameters, driving the learning error to zero. The stability proof of the above adaptation law can be found in [29]. It should be noted that the parameter γ can be considered as the learning rate for α , while ν can be interpreted as the parameter that helps to avoid a possible bursting in α . Note that the parameter ν should be carefully selected to prevent it from interrupting the training process. In order to prevent division by zero in the above equations, the denominator is set to be equal to 0.001 when its computed value is smaller than this number. In addition, it is known that SMC theory-based methods may suffer from high-frequency oscillations, which are commonly called chattering. Although high-order SMC techniques can be used to decrease the chattering effect, there are generally simpler approaches to alleviate this problem [138]. In particular, the sigmoid or saturation function can be used to replace the *sgn* function. Therefore, the following equation is utilized instead of the signum function in (4.16)

$$\text{sgn}(u_c) := \frac{u_c}{|u_c| + 0.001}. \quad (4.19)$$

Finally, in (4.15), if the parameter λ is selected as $\lambda = \frac{k_p}{k_d}$, then there exists a relation between the sliding manifold S_p and u_c in the following form

$$S_p = \dot{e} + \lambda e = \dot{e} + \frac{k_p}{k_d} e = \frac{1}{k_d} (k_d \dot{e} + k_p e) = \frac{u_c}{k_d} = \frac{S_c}{k_d}. \quad (4.20)$$

This relation shows that the convergence of S_c to zero ensures the convergence of S_p to zero and there is a sliding motion in the system states.

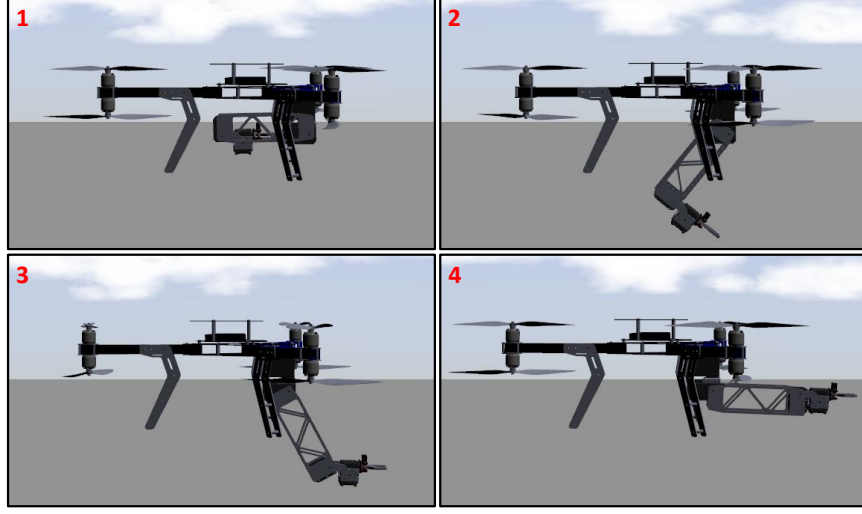


FIGURE 4.5: A sequence of snapshots during the unfolding of the robotic arm: (1) joint angles at $\{0, 0, 0, 0\}$ radians; (2)-(3) intermediate state; (4) joint angles at $\{0, -\pi, \pi, 0\}$ radians.

4.3 Simulation Studies

Typical aerial manipulation tasks such as performing assembly, inspection or maintenance operations require constant hover of the aerial robot while simultaneously manipulating the object which involves dexterous arm motion. Such arm motion results in the COG displacement and the change of the moments of inertia of the overall system, which, in its turn, creates additional torques and forces on UAV body causing the displacement of the UAV and losing the hover state. Hence, in our scenario, SMC theory based intelligent T2FNN controller is used to maintain a constant hover (at $x = 0$ m, $y = 0$ m, $z = 1.5$ m) and stabilize the UAV platform when the motion of the robotic arm is commanded in the xz -plane. The two motions that are considered for our scenario are unfolding and folding of the robotic arm. The former is commanded starting from initial joint angles $\{0, 0, 0, 0\}$ radians to final joint angles $\{0, -\pi, \pi, 0\}$ radians, as shown in Fig. 4.5, while the latter is the motion of the arm in opposite direction. The simulation scenario can be summarized as follows:

- UAV hovers with the folded arm for the first 10 s;
- The robotic arm is unfolded in the next 10 s followed by the folding of the arm for another 10 s;
- The above two steps are repeated one more time;
- Finally, UAV hovers with the folded arm for the final 10 s.

Two different cases are considered for the aforementioned scenario. In the first case, the performance of the proposed T2FNN controller is investigated when there are no external disturbances in the control system. In the second case, in order to show the superior noise reduction property of T2FNN compared to its type-1 counterpart ($a_1 = 1$ and $a_2 = 1$) and the conventional PD controller at higher noise level, the wind is added to our simulation scenario. For generating the wind with different speeds v_w , we use the Gaussian distribution

$$v_w = \mathcal{N}(\mu_v, \sigma_v^2), \quad (4.21)$$

where σ_v and μ_v are the standard deviation and the mean value of the wind speed, respectively. The wind gust begins to blow with a speed v_w and a constant direction $[0, 1, 0]$ (along y -axis) from $t = 0$ s. In this case, the efficacy of the proposed control strategy is evaluated not only in the presence of internal uncertainties caused by arm motion, but also in the presence of external disturbances such as wind gust. Throughout the simulation, the maximum operating speed of the end-effector is restricted to 100.0 mm/s. The mean value of the wind speed is selected as $\mu_v = 2.5$ m/s. In addition, in order to produce different noise levels, five different standard deviation of the wind speed are selected as $\sigma_v = \{0, 0.1, 0.3, 0.5, 1.0\}$ [m/s].

4.3.1 Intrinsic Parameters of the System and Control Variables

For the dynamical simulations, the coaxial tricopter equipped with the 4-DOF manipulator with all revolute joints is implemented in Gazebo simulator as shown in Fig. 4.5. In order to achieve the realistic behavior of aerial robots during the flight, the software-in-the-loop (SITL) configuration of the PX4 autopilot is utilized. The PX4 flight controller is responsible for the low-level control and attitude stabilization of the UAV platform. High-level commands (e.g., position/velocity set-points) should be sent to the PX4 flight controller to control the UAV platform. In addition, the Gazebo simulator provides the simulated sensor data to the PX4 flight controller, which utilizes this information together with set-points to compute actuator commands. The communication between all components is handled by the ROS environment. This ensures the software portability and integration of the developed codes with the real UAV platform. The coaxial tricopter intrinsic parameters are listed in Table 4.1. For the link masses (including the mass of

motors), the following values have been considered $m_{L_1} = 170$ g, $m_{L_2} = 125$ g, $m_{L_3} = 80$ g and $m_{L_4} = 64$ g. Note that since the position of the first motor is static, its mass is contributed to the UAV body mass. The inertia moments, which are measured at the center of gravity of their corresponding rigid bodies, are given in Table 4.2. While parameters such as mass ($m_b, m_{L_1}, m_{L_2}, m_{L_3}, m_{L_4}$) and the UAV arm length (l) can be directly measured, the moment of inertia values can be obtained from their SolidWorks CAD model. In addition, the drag-moment (k_d) and thrust (k_t) coefficients can be evaluated based on a simple experimentation, details of which are given in [141], or by analyzing the motor-propeller combination using RCbenchmark test setup. The joint positions have been obtained by using Gazebo API. It should be noted, while masses and the UAV arm length can be directly measured from the real UAM, the certain intrinsic properties like moments of inertia, drag and thrust coefficients are always a close approximation of the real UAM due to the estimation accuracy in the existing experimental and analytical methods.

The control gains for the PD controller are selected as follows

$$k_p = 1.5, \quad k_d = 0.1$$

while PID gains for each joint are as follows

$$\begin{aligned} k_{p_1} &= 10, & k_{i_1} &= 0.05, & k_{d_1} &= 0.1 \\ k_{p_2} &= 10, & k_{i_2} &= 0.05, & k_{d_2} &= 0.5 \\ k_{p_3} &= 1.0, & k_{i_3} &= 0.05, & k_{d_3} &= 0.005 \\ k_{p_4} &= 0.08, & k_{i_4} &= 0.05, & k_{d_4} &= 0.003 \end{aligned}$$

TABLE 4.1: Coaxial tricopter intrinsic parameters.

Parameter	Value	Unit
m_b	2.0	[kg]
l	0.285	[m]
k_t	8.55×10^{-6}	[N · s ²]
k_d	1.66×10^{-7}	[N · m · s ²]
I_{xx}	1.8977×10^{-2}	[kg · m ²]
I_{yy}	2.1915×10^{-2}	[kg · m ²]
I_{zz}	3.6685×10^{-2}	[kg · m ²]

In order to address the well-known issue with the PID controllers, *i.e.*, integral windup [142], the integral upper (i_{max}) and lower (i_{min}) limits are set to be $i_{max} = 0.1$ and $i_{min} = -0.1$. As for the antecedent parameters of the T2FNN, they are set to (subindexes x, y and z correspond to position channels x, y, z)

$$\begin{aligned} \mathbf{c}_{1x} &= \mathbf{c}_{1y} = \mathbf{c}_{1z} = [-0.15, 0.0, 0.15]^T \\ \mathbf{c}_{2x} &= \mathbf{c}_{2y} = \mathbf{c}_{2z} = [-0.25, 0.0, 0.25]^T \\ \mathbf{d}_{1x} &= \mathbf{d}_{1y} = \mathbf{d}_{1z} = [0.15, 0.15, 0.15]^T \\ \mathbf{d}_{2x} &= \mathbf{d}_{2y} = \mathbf{d}_{2z} = [0.25, 0.25, 0.25]^T \end{aligned}$$

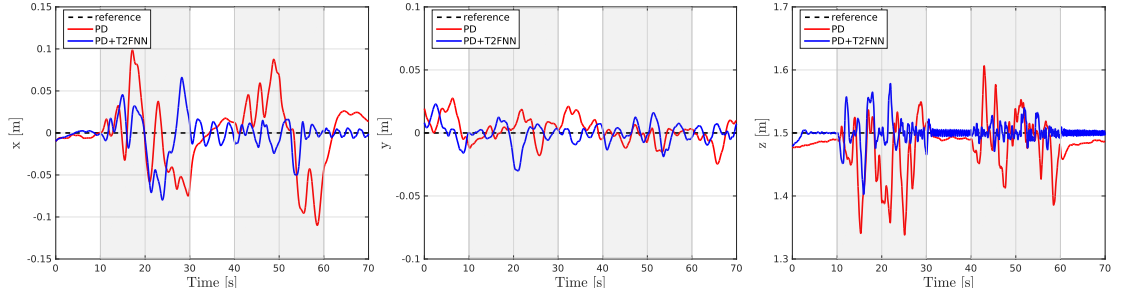
The initial condition for time-varying weight coefficient, $f_{ij}(0)$, is chosen to be a sufficiently small positive number, *i.e.*, $f_{ij}(0) \in [0, 0.001]$, while the initial values for learning rate $\alpha(0)$ are also selected to be a small positive number, *i.e.*, $\alpha_x = \alpha_y = \alpha_z = 0.001$. The elliptic MF parameters are set to be $a_1 = 1.25$ and $a_2 = 0.8$. The adaptive learning parameters for the consequent part of FNN are chosen as

$$\begin{aligned} \gamma_x &= 0.1 & \gamma_y &= 0.045 & \gamma_z &= 0.5 \\ \nu_x &= 0.08 & \nu_y &= 0.005 & \nu_z &= 0.045 \end{aligned}$$

The total simulation time is equal to 70s, with the time step of $dt = 0.01$ s. Trial-and-error method is used to select the gains for the PD controller. The width (d), the center (c) and the uncertainty width (a_1, a_2) values of the MF are selected by using expert knowledge. In addition, the MFs are equally placed on the universe of discourse. The initial values for time-varying weight coefficient $f_{ij}(0)$ and the learning rate $\alpha(0)$ are selected randomly, although they are selected to be a sufficiently small positive number, *i.e.*, $f_{ij}(0)$ and $\alpha(0) \in [0, 0.001]$. Finally, the adaptive learning parameters (γ and ν) for the consequent part of FNN are chosen by using trial-and-error method.

TABLE 4.2: Moments of inertia of each link of the robotic arm.

Link	I_{xx}	I_{yy}	I_{zz}	Unit
1	4.4563×10^{-5}	6.2332×10^{-5}	7.6670×10^{-5}	$[\text{kg} \cdot \text{m}^2]$
2	6.2812×10^{-5}	1.2517×10^{-4}	1.4761×10^{-4}	$[\text{kg} \cdot \text{m}^2]$
3	1.7075×10^{-5}	1.8862×10^{-5}	1.6000×10^{-5}	$[\text{kg} \cdot \text{m}^2]$
4	7.8370×10^{-6}	0.8868×10^{-5}	0.9909×10^{-5}	$[\text{kg} \cdot \text{m}^2]$



(A) Position response of the UAV center of gravity.

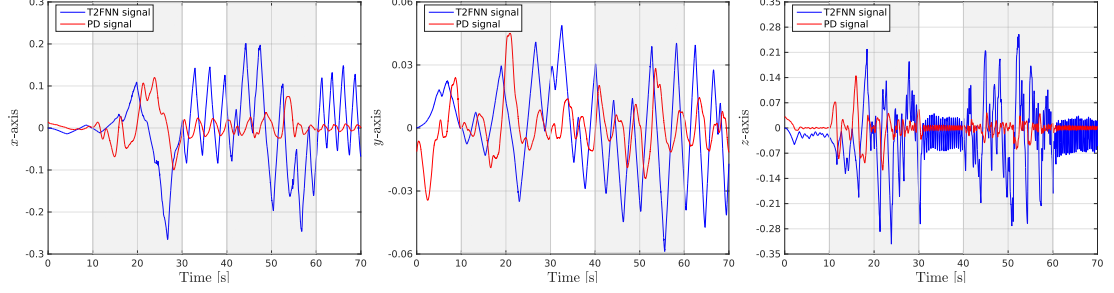
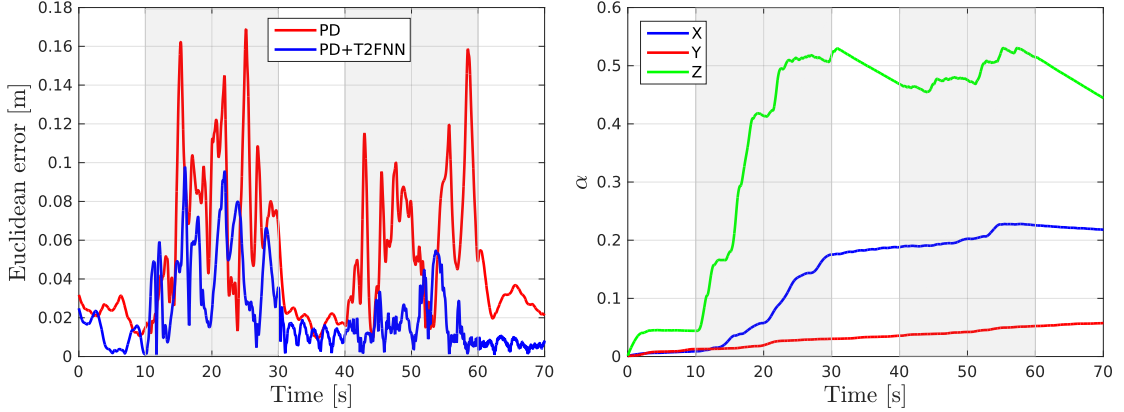
(B) Control signals on the x , y , and z axes in the case when the PD controller is working in parallel with the T2FNN controller.(C) Euclidean error for different controllers. (D) Evolution of adaptive learning rate (α) over time.

FIGURE 4.6: Performance of the proposed controller in the absence of wind, where the grey shaded regions represent the time periods during the unfolding and folding the arm.

4.3.2 Simulation Results

The simulation results are summarized in Figs. 4.6 and 4.7. In Fig. 4.6a, the position (x , y , and z) response of the UAV center of gravity is shown during the arm motion for the PD controller only and T2FNN working in parallel with the PD controller for the case when there are no external disturbances, i.e., no wind. It can be seen that when the PD controller is working alone, it cannot deal with the disturbances caused by the arm motion. Furthermore, it can be noticed

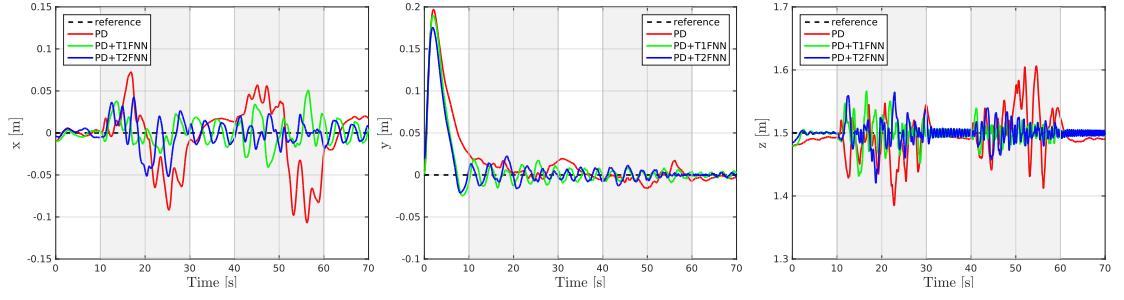
that the PD controller has a significant steady-state error occurred from internal uncertainties such as lack of modeling. Moreover, when the PD controller works alone, it cannot compensate the induced torque caused by the arm motion, and therefore, the existed error persists during the whole simulation. However, in case of the proposed T2FNN controller, the steady-state error is notably decreased with the help of its adaptive learning capabilities. Hence, it can be seen that the T2FNN controller achieves better tracking performance in x , y , and z axes of the UAV center of gravity. It is evident that the PD controller can be tuned more aggressively to obtain better results in each specific scenario, although this is not practical due to the lack of modeling and the existed uncertainties in the environment such as wind gust. In addition, aggressive tuning tends to be case dependent, and hence, it cannot give a comparable performance in different conditions. On the other hand, adaptive learning capabilities of T2FNN structure do not have such limitations, and therefore, they are more suitable for real-time applications.

The control signals on x , y and z axes in the case when the PD controller is working in parallel with the T2FNN controller are shown in Fig. 4.6b. It can be observed that the intelligent compensator, the T2FNN controller, is taking over the control responsibilities from PD, and hence, it gradually substitutes the PD controller. This causes the output control signal from the PD controller to approach to zero neighborhood after some time, and then, only T2FNN dominates the system. It should be noted that the T2FNN controller cannot completely replace the PD controller since when the trajectory sequence of arm motion changes the output control signal from the PD controller becomes nonzero. In such cases, T2FNN restarts the learning process and takes over control responsibility of the system again as illustrated in Fig. 4.6b. In addition, the evolution of the adaptive learning rate (α) over time is shown in Fig. 4.6d.

Figure 4.6c presents the Euclidean error values for both controllers. It is observed that the combination of PD and T2FNN controller (PD+T2FNN) results in a superior performance than that of the conventional PD controller when it works

TABLE 4.3: Euclidean RMSE for different controllers in the absence of wind (unit: cm).

Controller Types	PD	PD+T2FNN
RMSE	6.08	2.70



(A) Position response of the UAV center of gravity.

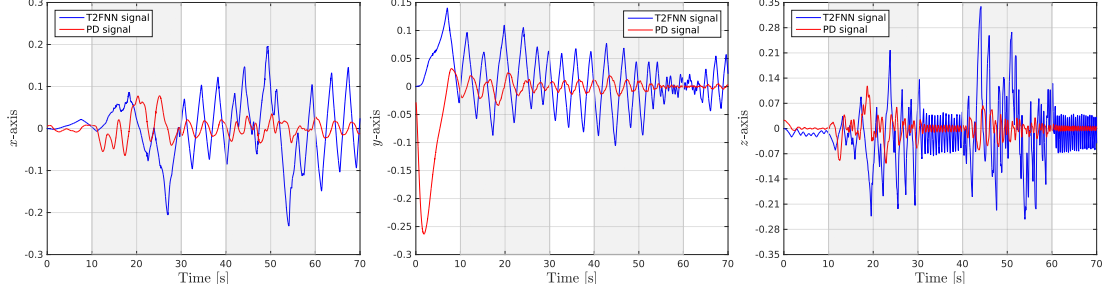
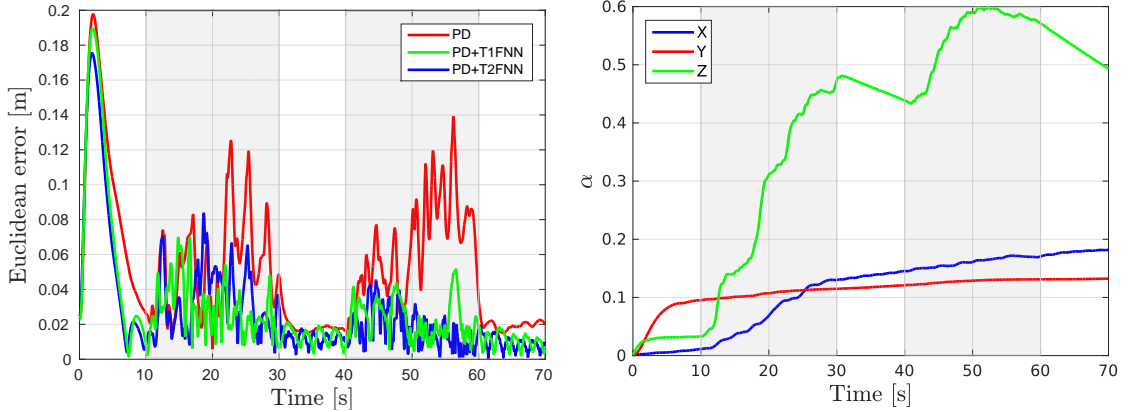
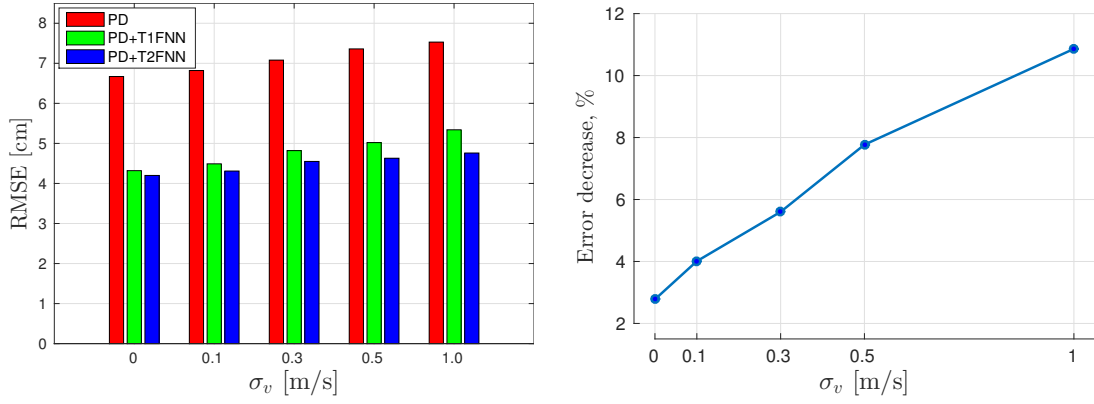
(B) Control signals on the x , y , and z axes in the case when the PD controller is working in parallel with the T2FNN controller.(C) Euclidean error for different controllers. (D) Evolution of adaptive learning rate (α) over time.

FIGURE 4.7: Performance of the proposed controller in the presence of wind (for $\sigma_v = 0$ m/s), where the grey shaded regions represent the time periods during the unfolding and folding the arm.

alone. It should be noted that the error of proposed controller is higher than the PD controller at the beginning of the arm motion. However, once it tunes its parameters, it yields a significantly smaller error than the PD controller. This can be also seen from Table 4.3 which shows the root-mean-square error (RMSE) values. It can be noticed that T2FNN controller decreases the PD controller RMSE value by around 56%. As a result, hovering performance of coaxial tricopter which uses intelligent T2FNN structure becomes significantly better compared to the case when the PD controller is only utilized.



(A) Euclidean RMSE for different controllers in the presence of wind. (B) RMSE error decrease (in percentage) of T2FNN compared to T1FNN with different wind variances.

FIGURE 4.8: Performance of the proposed controller in the presence of wind.

For the second case, where the wind is present, the performance of the different controllers when $\sigma_v = 0$ m/s is shown in Fig. 4.7. It is evident that the intelligent T2FNN and T1FNN structures yield the superior performance compared to the PD controller when it is operating alone. On the other hand, the performances of T2FNN and T1FNN are more or less the same when $\sigma_v = 0$ m/s as shown in Figs. 4.7a and 4.7c. It is observed that the benefit of the use of T2FNN is more apparent at higher noise level. In order to make a quantitative comparison between T2FNN and T1FNN structures, let us consider Fig. 4.8 and Table 4.4. It is seen that at higher noise level, e.g., $\sigma_v = 0.5$ m/s and $\sigma_v = 1.0$ m/s, the T2FNN controller demonstrates better noise handling capability compared to its type-1 counterpart. It can be explained by noting that type-2 MFs are themselves fuzzy, which, in its turn, provides an extra degree of freedom to deal with uncertainty.

TABLE 4.4: Euclidean RMSE for different controllers in the presence of wind.

Controller Types	RMSE, [cm]				
	σ_v , [m/s]				
	0	0.1	0.3	0.5	1.0
PD	6.67	6.82	7.08	7.36	7.53
PD+T1FNN $(a_1, a_2) = (1,1)$	4.32	4.49	4.82	5.02	5.34
PD+T2FNN $(a_1, a_2) = (1.25,0.8)$	4.20	4.31	4.55	4.63	4.76

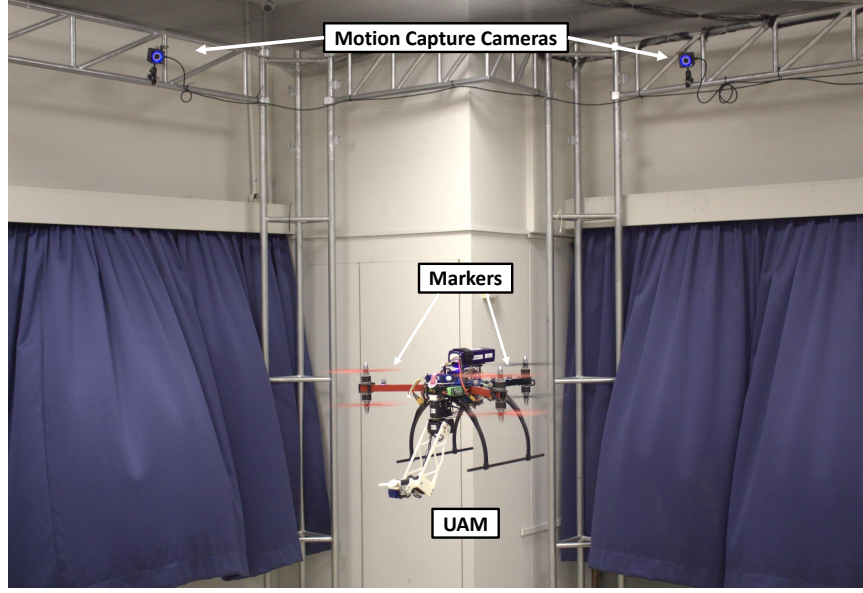


FIGURE 4.9: Indoor experimental setup.

In addition, it is observed that in the presence of wind, T2FNN controller again decreases the PD controller RMSE errors by more than 35% for all σ_v values as illustrated in Fig. 4.8a. Moreover, due to the structure of T2FNN, it can further reduce noise effects by around 3% to 11% as compared to the T1FNN controller as shown in Fig. 4.8b.

4.4 Experimental Tests

The experimental flight tests to validate the superior control performance of the proposed controller were conducted in the indoor environment and evaluated by the motion capture system, consisting of eight OptiTrack Prime 13 cameras, which provides the real-time pose (position and attitude) measurements with an update rate of 100Hz and accuracy around 0.5mm. The indoor experimental setup including UAM and cameras is depicted in Fig. 4.9. Reflective markers were fixed on the UAM's frame in a unique pattern to provide visual feedback of UAM to the OptiTrack cameras. The aerial robot used for experiments consists of the Y6 coaxial tricopter from 3D Robotics and the 4-DOF robotic arm. The coaxial tricopter is equipped with a Pixhawk autopilot that serves as the low-level controller, while the robotic arm is actuated by four Dynamixel servomotors (MX series) with internal PID position control loop and controlled by an ArbotiX-M robocontroller.

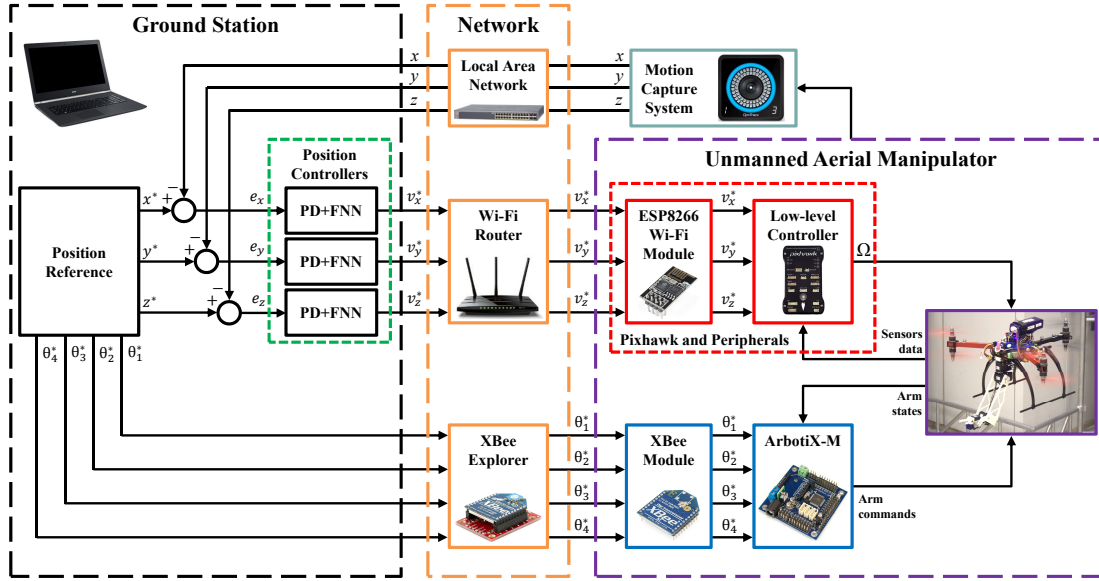


FIGURE 4.10: Real-time implementation of the overall control scheme.

The proposed control strategy presented in Sec. 4.2, implemented in ROS and C++ environment, runs on Linux based ground computer and computes the required control commands which are fed to the low-level controller (Pixhawk) over a wireless network at 50Hz. The communication between ground computer and ArbotiX-M robocontroller is realized via XBee which transmits the desired joint angles of the robotic arm. The real-time implementation of the overall control scheme is shown in Fig. 4.10.

In order to evaluate the efficacy of the proposed control strategy, two scenarios are considered for the experimental tests. The first scenario for the experimental tests is chosen to be the same as in the simulation studies (no wind case) in order to verify the results obtained in Section 4.3. In particular, SMC theory based intelligent T2FNN controller is used to maintain a constant hover (at $x = 0$ m, $y = 0$ m, $z = 1.5$ m) and stabilize Y6 coaxial tricopter when the motion of the robotic arm is commanded in the xz -plane to perform unfolding and folding motions. In the second scenario, in order to resemble the actual UAV flight, Y6 coaxial tricopter is commanded to track the figure-eight shaped time based trajectory twice for 50 seconds. While Y6 coaxial tricopter tracks the given trajectory, the robotic arm is configured to move in particular sequence. The motion of the robotic arm is shown in Fig. 4.11. Due to the space limitation of our Motion Capture Lab, the maximum speed of UAV along trajectory is kept to be 1m/s. In addition, the proposed controller is compared not only with PD, but with PID controller as well, where PID

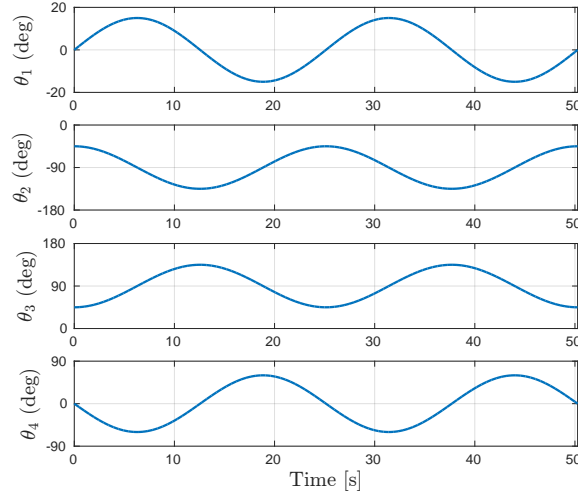
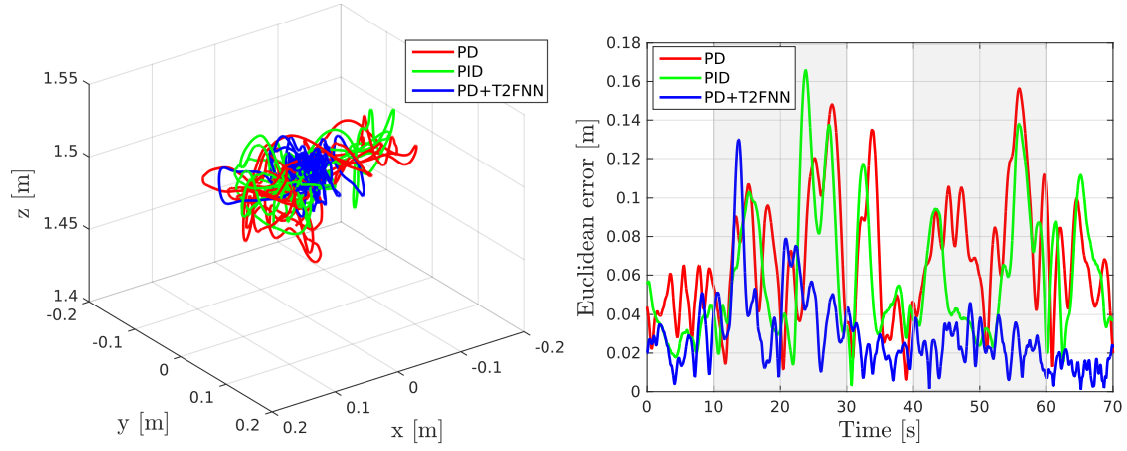


FIGURE 4.11: Joint angles of the robotic arm during the trajectory tracking flight.

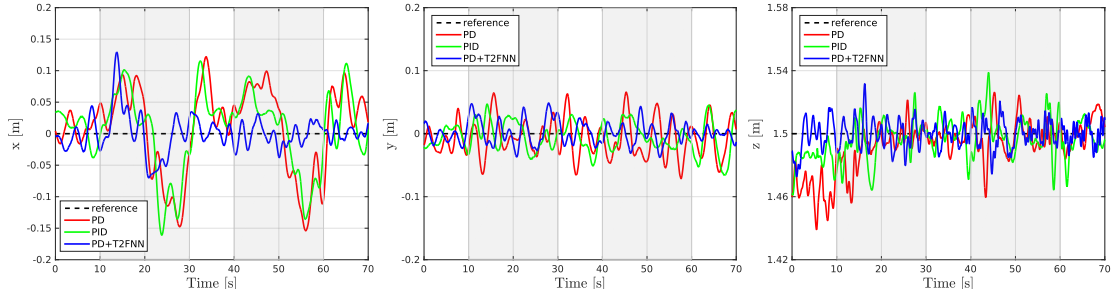
gains are equal to $k_p = 1.5$, $k_i = 0.2$ and $k_d = 0.1$. The video of the experimental flight tests can be seen via this link (<https://youtu.be/qTdNbmOXy44>).

4.4.1 Experimental Results

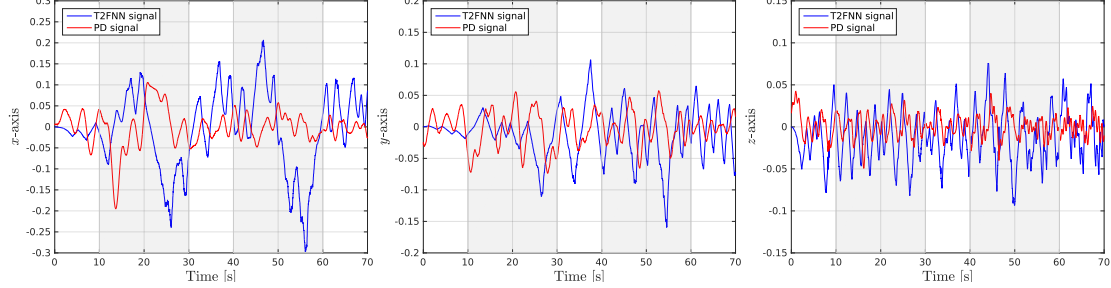
The experimental results for the first scenario (hover case) are summarized in Fig. 4.12. In Fig. 4.12a, the hovering performance of the UAV platform is shown for the PD and PID controllers only and T2FNN working in parallel with the PD controller. In addition, the position (x , y , and z) response of the UAV center of gravity is depicted in Fig. 4.12c. It is evident that the PD and PID controllers have a significant error occurred from internal uncertainties such as lack of modeling, although the performance of the PID controller is slightly better compared to the PD controller. In addition, both controllers cannot compensate the induced torque caused by the arm motion when they work alone. As a result, the existed error persists, and it increases during the arm motion. It is clearly visible from x response of the UAV center of gravity as shown in Fig 4.12c. The performance of the PD and PID controllers can be improved by precisely tuning their parameters, although this may yield the substandard results due to the unforeseen working conditions. On the other hand, the proposed T2FNN-based controller eliminates the need for precise tuning of the conventional controller by learning from system dynamics and compensating disturbances online. Hence, it can be seen that the T2FNN controller achieves better tracking performance in x , y , and z axes of the UAV center of gravity.



(A) Hovering performance of UAV base in (B) Euclidean error for different controllers. 3D view.



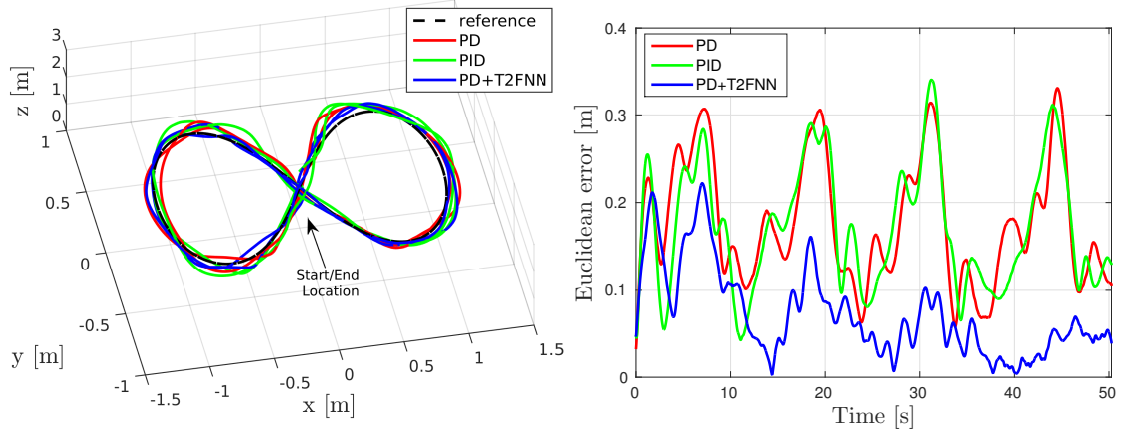
(C) Position response of the UAV center of gravity.



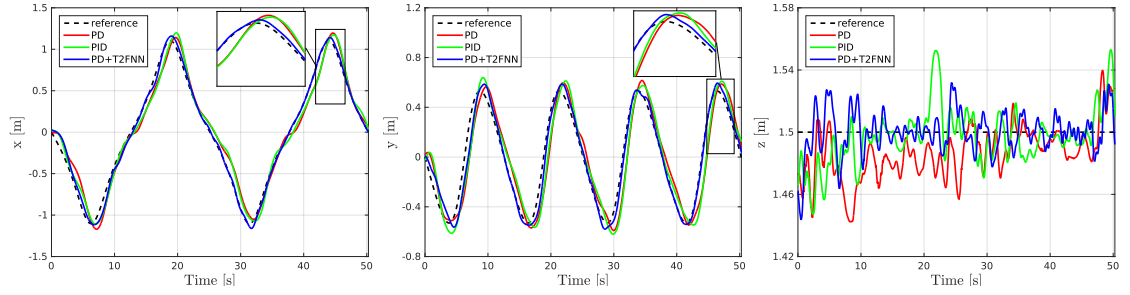
(D) Control signals on the x , y , and z axes in the case when the PD controller is working in parallel with the T2FNN controller.

FIGURE 4.12: Experimental results of the proposed controller for the hover case, where the grey shaded regions represent the time periods during the unfolding and folding the arm.

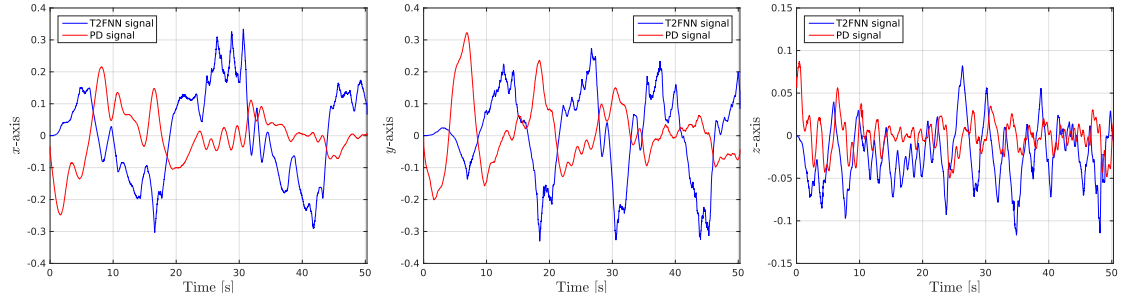
Figure 4.12d presents the control signals on x , y and z axes in the case when the PD controller is working in parallel with the T2FNN controller. It is observed that the T2FNN controller takes over the control responsibilities from PD and gradually substitutes the PD controller. This results in the output control signal from the PD controller to approach to zero neighborhood after some time. It should be noted that when the trajectory sequence of the arm motion changes, the output control signal from the PD controller becomes nonzero. In such cases, T2FNN restarts the



(A) Trajectory tracking performance of UAV (B) Euclidean error for different controllers. base in 3D view.



(C) Position response of the UAV center of gravity.



(D) Control signals on the x , y , and z axes in the case when the PD controller is working in parallel with the T2FNN controller.

FIGURE 4.13: Experimental results of the proposed controller for the trajectory tracking case.

learning process and takes over control responsibility of the system again.

The Euclidean error values for different controllers are shown in Fig. 4.12b. It can be seen that the combination of PD and T2FNN controller (PD+T2FNN) yields a superior performance than that of the conventional PD and PID controllers when they work alone. This can be also seen from Table 4.5 and Fig. 4.14 which show the RMSE values. It is observed that the T2FNN controller decreases the PD and PID controllers RMSE error by around 54% and 48%, respectively. Finally, as

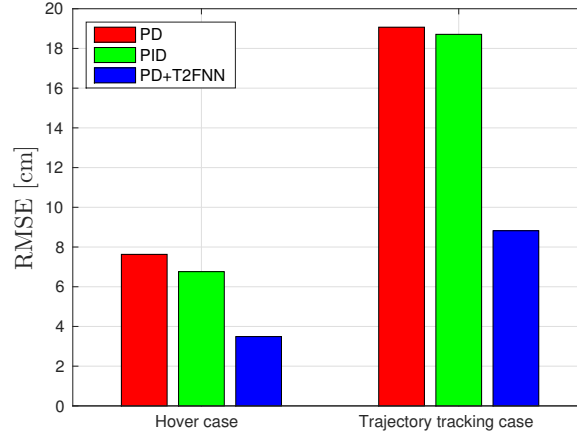


FIGURE 4.14: Euclidean RMSE for different controllers for the hover and trajectory tracking case.

in simulation studies the error of proposed controller is higher than the PD and PID controllers at the beginning of the arm motion (see Fig. 4.12b). However, once it tunes its parameters, the proposed controller gives the significantly better performance than the case when the conventional controllers act alone during the second time of the unfolding and folding of the arm.

The experimental results for the second scenario, where the Y6 coaxial tricopter tracks the figure-eight shaped time based trajectory twice while the robotic arm is in continuous motion, are shown in Fig. 4.13. The trajectory tracking performance of the UAV platform for the different controllers is depicted in Fig. 4.13a. It can be seen that the proposed controller yields the notably better trajectory tracking performance compared to the case when the PD and PID controllers are only utilized. This can be also seen from Fig. 4.13c which shows the position (x , y , and z) responses of the UAV center of gravity. In addition, it can be noticed that the conventional controllers cannot deal with the disturbances caused by the arm motion. Therefore, as shown in Fig. 4.13b the Euclidean error of the proposed

TABLE 4.5: Euclidean RMSE for different controllers (unit: cm).

Scenario	Controller Types	RMSE
Hover	PD	7.63
	PID	6.76
	PD+T2FNN	3.49
Eight shaped trajectory	PD	19.07
	PID	18.71
	PD+T2FNN	8.82

controller is a significantly smaller than that of the conventional controllers due to the adaptive learning capabilities of T2FNN structure. Finally, Table 4.5 and Fig. 4.14 shows the performance difference between the proposed and conventional controllers. It is observed that the T2FNN controller decreases the conventional controllers RMSE error by around 53%.

As for the control signals, Fig. 4.13d presents the control signals on the x , y , and z axes for the case when the PD and T2FNN controllers are working in parallel. It is seen that at the beginning, the dominating control signal is the one coming from the PD controller. However, after some time, the T2FNN controller takes over the control responsibilities from PD and gradually becomes the leading controller.

4.5 Summary

In this chapter, the SMC theory-based intelligent T2FNN controller has been proposed for the control of the coaxial tricopter equipped with the 4-DOF robotic arm in the absence and presence of wind gust conditions. The main goal has been to develop the control approach that can learn and compensate for the disturbances caused by the robotic arm motion and wind gust, and thus can open the door for the wider implementation of aerial manipulation tasks in daily life. Extensive simulations in ROS and Gazebo environment have been performed to evaluate the efficacy of this learning-based framework. The experimental flight tests to further validate the performance of the proposed control approach have been also conducted by using OptiTrack Motion Capture System. Both simulation and experimental results have illustrated that the proposed controller, thanks to its learning capability, is capable to significantly reduce the steady state-error and overcome the disturbances caused by the arm motion and wind gust as compared to the conventional PD and PID controllers working alone. On account of learning, the Euclidean RMSE improvements of around 56% and 52% have been observed for the simulation and real-time tests, respectively. Moreover, for the case, where the wind is present, the T2FNN controller has demonstrated better noise handling capability compared to its type-1 counterpart due to the implementation of elliptic type-2 MFs.

Chapter 5

Trajectory Generation-based Guidance Approach for Unmanned Aerial Manipulators

5.1 Introduction

UAMs provide a remarkable level of dexterity on performing complex aerial manipulation tasks thanks to their high number of DOFs. Having such redundant aerial robots, especially when employing dual-arm aerial manipulators, imposes additional challenges to successfully perform UAM missions. In particular, exploiting the entire system redundancy and simultaneously generating the feasible trajectories for the UAV platform and robotic arm(s) are essential in performing complex tasks and achieving dexterous behavior. In this regard, in this chapter, two different approaches are proposed to address these challenges. In both cases, redundancy resolution and trajectory generation methods are incorporated into one framework.

In the first approach, a trajectory generation with a multi-task redundancy resolution strategy is proposed to accomplish multiple UAM tasks, while satisfying additional constraints such as joint angle limit and obstacle avoidance. This strategy is based on the weighted damped least-squares method [143], which works at a velocity level. Although this method is well-known for robotic arms, its implementation for aerial manipulation tasks is novel. The proposed approach is

computationally inexpensive to run online, allowing UAM to dynamically react to unforeseen events by modifying its configuration during the execution of the primary task (e.g., end-effector positioning). In addition, in order to avoid the singular configurations of the robot arm, the maximization of the manipulability measure is considered. Specifically, the strategy for integration of the manipulability measure into the objective function of the redundancy resolution framework by using the approximated derivatives is described. Furthermore, in the cases when the redundancy of UAM cannot be fully exploited due to antagonistic tasks, the task relaxation approach for aerial manipulation is presented to satisfy the robot physical constraints and ensure the motion smoothness. Extensive simulation studies in ROS-Gazebo environment are performed to validate the performance of the proposed approach with the single-arm aerial manipulator. To the best of the author's knowledge, this is the first work where the task relaxation framework and the maximization of the manipulability measure represented by the first-order approximation are implemented for UAMs while satisfying other constraints.

In the second approach, the nonlinear model predictive control-based trajectory generation with redundancy resolution strategy is developed for the dual-arm aerial manipulator. The proposed approach continuously re-plans the aerial robot motion to perform the assigned task (e.g., 3D trajectory tracking with the end-effectors of dual-arm), while satisfying several constraints such as avoiding the obstacles and limits of the motion variables, i.e. the position and yaw angle of a multirotor as well as the dual-arm joint variables. In addition, the minimum manipulability measure is imposed to ensure a smooth motion and avoid singular configurations. Furthermore, by defining the MPC problem in the acceleration domain, velocity and position profiles without discontinuities can be obtained to have a smooth behavior of the aerial robot. The performance of the developed framework is shown via simulations in ROS-Gazebo environment and real-time experimental flight tests, involving the dual-arm aerial manipulator performing non-trivial multi-task missions (e.g., the transportation of the long bar), which require the coordination of both arms, while keeping the configuration of the aerial robot within imposed bounds and preventing collisions with obstacles. To the best of the author's knowledge, this is the first work where the NMPC-based trajectory generation with redundancy resolution strategy is implemented for the dual-arm aerial manipulator with all computations done online and onboard. The video of the experimental flight tests can be accessed via this link (<https://youtu.be/8n-Vz4ECr2U>).

The remainder of this chapter is organized as follows. In Section 5.2, UAM state is presented. In Section 5.3, the multi-task velocity-based redundancy resolution framework is described, while Section 5.3.3 contains simulation results that evaluate and verify the performance of this framework. In Section 5.4, NMPC-based trajectory generation method is explained, while simulation and experimental studies are presented in Section 5.4.3 to validate the proposed method. Lastly, the comparison between two proposed methods and a summary of this chapter are provided in Section 5.5.

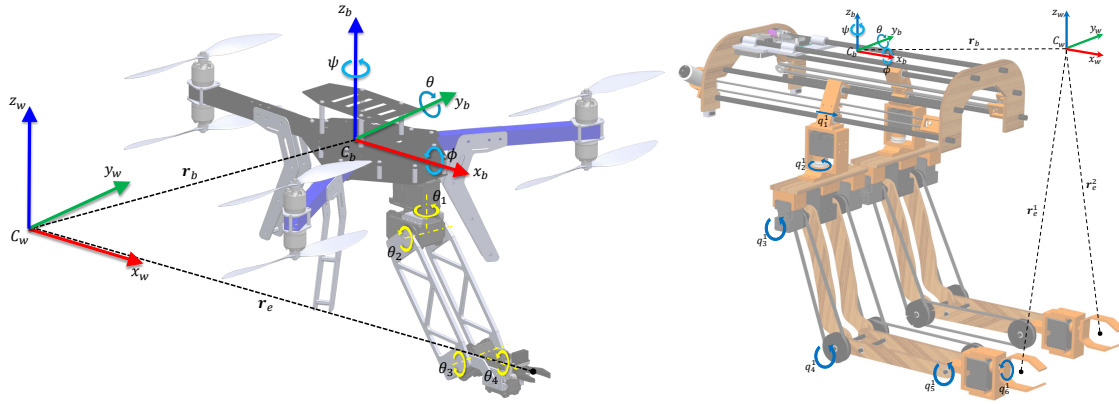
5.2 UAM State

Multirotor UAVs are generally under-actuated systems since they have only 4 independent controllable variables for the 6 DOFs system, i.e., the position and yaw angle of UAV, while the roll and pitch angles are considered as the non-controllable variables. Hence, attaching multi-DOFs robotic arm(s) to multirotor UAVs leads to the whole new systems with exceptional redundancy. Then, it is worth to specify the complete state of the aerial manipulator which includes the position ($\mathbf{r}_b = [x_b \ y_b \ z_b]^T \in \mathbb{R}^3$) and UAV orientation represented by the Euler angles ($\mathbf{\Phi}_b = [\phi \ \theta \ \psi]^T \in \mathbb{R}^3$) with respect to the world-fixed reference frame C_W , and joint position vectors of the single-arm $\mathbf{\Theta} = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4]^T \in \mathbb{R}^4$ or dual-arm $\mathbf{q}^m = [q_1^m, \dots, q_6^m]^T \in \mathbb{R}^6$, with $m = \{1, 2\}$ for the right and left arms, respectively (see Fig. 5.1). Thus, the complete states of the aerial manipulators can be defined as follows

$$\begin{cases} \boldsymbol{\sigma}_s = [\mathbf{r}_b^T \ \mathbf{\Phi}_b^T \ \mathbf{\Theta}^T]^T \in \mathbb{R}^{10}, \\ \boldsymbol{\sigma}_d = [\mathbf{r}_b^T \ \mathbf{\Phi}_b^T \ \mathbf{q}^{1,T} \ \mathbf{q}^{2,T}]^T \in \mathbb{R}^{18}. \end{cases} \quad (5.1)$$

where $\boldsymbol{\sigma}_s$ and $\boldsymbol{\sigma}_d$ correspond to the single-arm and dual-arm aerial manipulator, respectively.

In general, aerial robots employed for aerial manipulation tasks are not intended for aggressive and agile maneuvers. Therefore, the multirotor tilt values (roll (ϕ) and pitch (θ) angles are assumed to be negligible) are not considered in the proposed trajectory generation algorithms. Moreover, having such high-dimensional states, especially in the case of the dual-arm aerial manipulator, increase the computational burden, which, in its turn, can hinder the real-time implementation of



(A) Joint angles and reference frames for the single-arm aerial manipulator. (B) Joint positions and reference frames for the dual-arm aerial manipulator. Note that the hexarotor image is omitted here for better illustration purposes.

FIGURE 5.1: UAM state and reference frames.

the proposed trajectory generation algorithms, considering the available onboard computational power. Hence, it was decided to exclude q_1^1 , q_6^1 and q_1^2 , q_6^2 from consideration in the proposed framework, in order to decrease the computational load. Then, by considering all the aspects mentioned above, the final configuration states of the aerial manipulators for the trajectory generation algorithms can be re-written in the following way

$$\begin{cases} \zeta_s = [\mathbf{r}_b^T \ \psi \ \theta_1 \ \theta_2 \ \theta_3 \ \theta_4]^T \in \mathbb{R}^{n_s}, \\ \zeta_d = [\mathbf{r}_b^T \ \psi \ q_2^1 \ q_3^1 \ q_4^1 \ q_5^1 \ q_2^2 \ q_3^2 \ q_4^2 \ q_5^2]^T \in \mathbb{R}^{n_d}. \end{cases} \quad (5.2)$$

where ζ_s and ζ_d correspond to the single-arm and dual-arm aerial manipulator, respectively, while $n_s = 8$ and $n_d = 12$.

It should be noted that the disturbances on the UAV platform caused by the motion of the robotic arm(s) are assumed to be negligible due to the slow arm(s) movements, and they are mostly compensated by the inner loop controller. This assumption can be further justified by preventing high configuration rates in trajectory generation stage, and by employing controllers with adaptive learning capabilities, such as the one presented in Chapter 4, that can compensate disturbances caused by the motion of the arm(s). With the assistance of such controller and by considering the slow arm(s) movements, the UAV platform can be considered as a nominal multirotor, regardless of the motion of the arm(s) [144].

5.3 Multi-task Velocity-based Redundancy Resolution Framework

5.3.1 General Approach

In typical aerial manipulation missions the main interaction with the environment is performed by the end-effector of the robotic arm, thus it is crucial to precisely position the end-effector on the desired trajectory ($\mathbf{r}_e^* \in \mathbb{R}^3$). The configuration vector $\boldsymbol{\zeta}_s$ of the single-arm aerial manipulator describes the position of the end-effector \mathbf{r}_e by the forward kinematics model as follows

$$\mathbf{r}_e = f(\boldsymbol{\zeta}_s), \quad (5.3)$$

and by the differential kinematics relation as

$$\dot{\mathbf{r}}_e = \mathbf{J}_e \dot{\boldsymbol{\zeta}}_s, \quad (5.4)$$

where $\mathbf{J}_e \in \mathbb{R}^{3 \times n_s}$ is the end-effector Jacobian matrix. The pseudo-inverse techniques [145] can be used to find the desired $\dot{\boldsymbol{\zeta}}_s^*$ which reconstructs the desired end-effector velocity $\dot{\mathbf{r}}_e^*$. However, when the UAM configuration is near to a singular posture, the pseudo-inverse techniques can suffer from high velocity components which can exceed the physical limits of the aerial robot. In order to avoid such situations, in this study, the weighted damped least-squares method [143] is employed. Specifically, with the given desired end-effector velocity $\dot{\mathbf{r}}_e^*$ as the primary task, the redundancy resolution strategy can be formulated as finding the desired $\dot{\boldsymbol{\zeta}}_s^*$ that approximates (5.4) by minimizing the following cost function [143]

$$C = \|\mathbf{J}_e \dot{\boldsymbol{\zeta}}_s - \dot{\mathbf{r}}_e^*\|^2 + \|\nu \dot{\boldsymbol{\zeta}}_s\|^2, \quad (5.5)$$

where $\nu > 0$ is the damping factor which prevents the system from reaching singular postures by penalizing high configuration rates. In order to perform additional tasks (e.g., obstacle avoidance), (5.5) can be modified to account the additional tasks in the following way [143]

$$C = \|\mathbf{J}_e \dot{\boldsymbol{\zeta}}_s - \dot{\mathbf{r}}_e^*\|_{\mathbf{W}_e}^2 + \sum_{j=1}^M \|\mathbf{J}_{\eta_j} \dot{\boldsymbol{\zeta}}_s - \dot{\boldsymbol{\eta}}_j^*\|_{\mathbf{W}_{\eta_j}}^2 + \|\dot{\boldsymbol{\zeta}}_s\|_{\mathbf{W}_\nu}^2, \quad (5.6)$$

where $\dot{\boldsymbol{\eta}}_j^*$ is the desired additional task velocity for $j = 1, 2, \dots, M$ in which M is the number of additional tasks. \mathbf{J}_{η_j} is the Jacobian matrix associated with

the additional task. \mathbf{W}_e , \mathbf{W}_{η_j} , and \mathbf{W}_ν are diagonal positive-definite weighting matrices which assign the importance between the tasks. The cost function (5.6) can be expanded as follows

$$C = (\mathbf{J}_e \dot{\boldsymbol{\zeta}}_s - \dot{\mathbf{r}}_e^*)^T \mathbf{W}_e (\mathbf{J}_e \dot{\boldsymbol{\zeta}}_s - \dot{\mathbf{r}}_e^*) + \sum_{j=1}^M (\mathbf{J}_{\eta_j} \dot{\boldsymbol{\zeta}}_s - \dot{\boldsymbol{\eta}}_j^*)^T \mathbf{W}_{\eta_j} (\mathbf{J}_{\eta_j} \dot{\boldsymbol{\zeta}}_s - \dot{\boldsymbol{\eta}}_j^*) + \dot{\boldsymbol{\zeta}}_s^T \mathbf{W}_\nu \dot{\boldsymbol{\zeta}}_s. \quad (5.7)$$

Then, the desired $\dot{\boldsymbol{\zeta}}_s^*$ which minimizes the cost function (5.7) can be derived by differentiating C with respect to $\dot{\boldsymbol{\zeta}}_s$ and equating it to zero [143]

$$\dot{\boldsymbol{\zeta}}^d = [\mathbf{J}_e^T \mathbf{W}_e \mathbf{J}_e + \sum_{j=1}^M \mathbf{J}_{\eta_j}^T \mathbf{W}_{\eta_j} \mathbf{J}_{\eta_j} + \mathbf{W}_\nu]^{-1} (\mathbf{J}_e^T \mathbf{W}_e \dot{\mathbf{r}}_e^* + \sum_{j=1}^M \mathbf{J}_{\eta_j}^T \mathbf{W}_{\eta_j} \dot{\boldsymbol{\eta}}_j^*). \quad (5.8)$$

The UAM configuration vector $\boldsymbol{\zeta}_s^*$ is computed by integrating the velocities obtained from (5.8) for every time step.

5.3.2 UAM Tasks

This section presents and discusses a number of additional tasks, which will be included in (5.7) to improve flight performance during UAM missions.

5.3.2.1 Incorporating Joint Limit Avoidance

In order to incorporate joint limit avoidance (JLA) into the redundancy resolution framework, we define it as the additional task by using part-time constraints, meaning that the JLA task is only activated when the joint position is close to the actual joint limit of the robotic arm. The advantage of implementing JLA as a part-time task is that when the joint positions are far from their limits, the JLA task is deactivated allowing to utilize the redundancy of the system for other tasks. The task function for JLA can be defined as a one-to-one mapping of the UAM configuration vector $\boldsymbol{\eta}_{JL} = \boldsymbol{\zeta}_s$ [146]. Then, the first-order differential relation can be used to find the Jacobian matrix for the JLA task, $\mathbf{J}_{JL} \in \mathbb{R}^{n_s \times n_s}$

$$\mathbf{J}_{JL} = \mathbf{J}_\eta = \frac{\partial \boldsymbol{\eta}_{JL}}{\partial \boldsymbol{\zeta}_s} = \mathbb{I}_{n_s \times n_s}, \quad (5.9)$$

where $\mathbb{I}_{n_s \times n_s}$ is a $(n_s \times n_s)$ identity matrix. When the JLA task is active the desired JLA task velocities should be zero, i.e., $\dot{\mathbf{r}}_{JL}^* = 0$, as the joint velocities have to vanish when the system approaches the joint limit. Then, equation (5.7) can be rewritten as follows

$$C = (\mathbf{J}_e \dot{\boldsymbol{\zeta}}_s - \dot{\mathbf{r}}_e^*)^T \mathbf{W}_e (\mathbf{J}_e \dot{\boldsymbol{\zeta}}_s - \dot{\mathbf{r}}_e^*) + \dot{\boldsymbol{\zeta}}_s^T \mathbf{W}_{JL} \dot{\boldsymbol{\zeta}}_s + \dot{\boldsymbol{\zeta}}_s^T \mathbf{W}_\nu \dot{\boldsymbol{\zeta}}_s. \quad (5.10)$$

Solving the derivative of (5.10) for the unknown $\dot{\boldsymbol{\zeta}}_s$ gives

$$\dot{\boldsymbol{\zeta}}_s^* = [\mathbf{J}_e^T \mathbf{W}_e \mathbf{J}_e + \mathbf{W}_{JL} + \mathbf{W}_\nu]^{-1} \mathbf{J}_e^T \mathbf{W}_e \dot{\mathbf{r}}_e^*, \quad (5.11)$$

where $\mathbf{W}_{JL} \in \mathbb{R}^{n_s \times n_s}$ is the weighting matrix which activates and deactivates the JLA task. In addition, in order to have smooth joint trajectories, \mathbf{W}_{JL} should be selected as follows

$$\mathbf{W}_{JL} = \begin{bmatrix} \mathbb{O}_{4 \times 4} & & \mathbb{O}_{4 \times 4} \\ & \begin{bmatrix} w_1 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 \\ 0 & 0 & w_3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & \\ \mathbb{O}_{4 \times 4} & & \end{bmatrix}, \quad (5.12)$$

in which $\mathbb{O}_{a \times b}$ is $(a \times b)$ zero matrix and w_i ($i = 1, \dots, 3$) corresponds to the first three joints of the robotic arm and it is defined as [146]

$$w_i = \begin{cases} w_{jl} & \text{if } \theta_i < \theta_i^- \\ \frac{w_{jl}}{2} [1 + \cos(\pi \frac{\theta_i - \theta_i^-}{\tau_i})] & \text{if } \theta_i^- \leq \theta_i \leq \theta_i^- + \tau_i \\ 0 & \text{if } \theta_i^- + \tau_i < \theta_i < \theta_i^+ - \tau_i \\ \frac{w_{jl}}{2} [1 + \cos(\pi \frac{\theta_i^+ - \theta_i}{\tau_i})] & \text{if } \theta_i^+ - \tau_i \leq \theta_i \leq \theta_i^+ \\ w_{jl} & \text{if } \theta_i > \theta_i^+, \end{cases} \quad (5.13)$$

where θ_i^- and θ_i^+ are the lower and upper limit of the joint, τ_i is the influence region, and w_{jl} is a user-defined constant representing the weighting importance. It should be noted that when the joint value enters the influence region, the JLA task will be activated, preventing self-collision.

5.3.2.2 Obstacle Avoidance

Another important constraint, while performing aerial manipulation tasks, is the obstacle avoidance (OA). Similar to JLA, the OA task can be implemented as a part-time task, meaning that it is only activated in the vicinity of the obstacle allowing UAM to fly around the obstacle without collision. Given the position of the obstacle $\mathbf{r}_{ob} \in \mathbb{R}^3$ and the position of the UAV platform $\mathbf{r}_b \in \mathbb{R}^3$ the OA task function can be defined as

$$\eta_{OB} = d = \|\mathbf{r}_b - \mathbf{r}_{ob}\|, \quad (5.14)$$

where d is the distance between the center of the obstacle and UAM. The Jacobian matrix for the OA task is defined as $\mathbf{J}_{OA} = \frac{(\mathbf{r}_b - \mathbf{r}_{ob})^T}{d} [\mathbb{I}_{3 \times 3} \quad \mathbf{0}_{3 \times 5}] \in \mathbb{R}^{1 \times n_s}$.

Similar to the JLA task, when the OA task is activated, the desired OA task velocities should vanish to prevent collision between UAM and the obstacle, i.e., $\dot{\eta}_{OB}^* = 0$. Then, equation (5.10) is modified to accommodate the OA task

$$\begin{aligned} C = & (\mathbf{J}_e \dot{\boldsymbol{\zeta}}_s - \dot{\mathbf{r}}_e^*)^T \mathbf{W}_e (\mathbf{J}_e \dot{\boldsymbol{\zeta}}_s - \dot{\mathbf{r}}_e^*) + \dot{\boldsymbol{\zeta}}_s^T \mathbf{W}_{JL} \dot{\boldsymbol{\zeta}}_s + \\ & (\mathbf{J}_{OA} \dot{\boldsymbol{\zeta}}_s)^T W_{OA} (\mathbf{J}_{OA} \dot{\boldsymbol{\zeta}}_s) + \dot{\boldsymbol{\zeta}}_s^T \mathbf{W}_\nu \dot{\boldsymbol{\zeta}}_s, \end{aligned} \quad (5.15)$$

resulting to

$$\begin{aligned} \dot{\boldsymbol{\zeta}}_s^* = & [\mathbf{J}_e^T \mathbf{W}_e \mathbf{J}_e + \mathbf{W}_{JL} + \mathbf{J}_{OA}^T W_{OA} \mathbf{J}_{OA} + \\ & \mathbf{W}_\nu]^{-1} \mathbf{J}_e^T \mathbf{W}_e \dot{\mathbf{r}}_e^*, \end{aligned} \quad (5.16)$$

where $W_{OA} \in \mathbb{R}^1$ is the weighting constant which activates and deactivates the OA task. In order to have a smooth flight behavior near the obstacle, the weighting constant is defined as follows

$$W_{OA} = \begin{cases} w_{obs} & \text{if } d < d_{saf} \\ \frac{w_{obs}}{2} [1 + \cos(\pi \frac{d - d_{saf}}{d_{in} - d_{saf}})] & \text{if } d_{saf} \leq d \leq d_{in} \\ 0 & \text{if } d > d_{in}, \end{cases} \quad (5.17)$$

where w_{obs} is a user-defined constant representing the weighting importance, while d_{saf} and d_{in} are the safety and influence distances, respectively. It is decided that the obstacle and aerial robot are enclosed in spheres with the following radii r_{obs} and r_{rob} , respectively. Hence, the safety distance d_{saf} is selected as $d_{saf} = r_{obs} + r_{rob}$ to prevent collision, while d_{in} is defined such that it increases the safety factor during the OA task.

5.3.2.3 Manipulability Maximization

Manipulability is an important feature which describes the dexterity of the robotic arm at the given posture, and it is defined as follows [145]

$$w(\zeta_s) = \sqrt{\det(\mathbf{J}_a \mathbf{J}_a^T)}, \quad (5.18)$$

where $\mathbf{J}_a \in \mathbb{R}^{3 \times 4}$ is the Jacobian matrix corresponding to the robotic arm. The manipulability of the system is close to zero near singular configurations. Therefore, it is desirable to have high manipulability so that the system can have enough admissible motions in case of unforeseen events. In order to maximize the manipulability measure, the cost function (5.15) is modified as follows

$$C = (\mathbf{J}_e \dot{\zeta}_s - \dot{\mathbf{r}}_e^*)^T \mathbf{W}_e (\mathbf{J}_e \dot{\zeta}_s - \dot{\mathbf{r}}_e^*) + \dot{\zeta}_s^T \mathbf{W}_{JL} \dot{\zeta}_s + (\mathbf{J}_{OA} \dot{\zeta}_s)^T \mathbf{W}_{OA} (\mathbf{J}_{OA} \dot{\zeta}_s) + \dot{\zeta}_s^T \mathbf{W}_\nu \dot{\zeta}_s - \mu w(\zeta_s), \quad (5.19)$$

where $\mu > 0$ is the weighting importance, preceded by a negative sign to maximize $w(\zeta_s)$. However, since the manipulability $w(\zeta_s)$ is a nonlinear and complex function, finding its derivative is a tedious and difficult task. Therefore, $w(\zeta_s)$ can be linearized by the first-order approximation as follows [147]

$$w(\zeta_{s,t}) = w(\zeta_{s,t-1}) + \delta t \nabla w^T \dot{\zeta}_s, \quad (5.20)$$

where δt is the time step and ∇w is the gradient of $w(\zeta_s)$. Similarly as in [147], the i^{th} entry of vector ∇w can be approximated numerically in the following way

$$(\nabla w)_i = \frac{\partial w}{\partial \zeta_i} = \frac{w(\zeta_s + \delta \zeta_i \mathbf{m}_i) - w(\zeta_s - \delta \zeta_i \mathbf{m}_i)}{2\delta \zeta_i}, \quad (5.21)$$

where $\mathbf{m}_i \in \mathbb{R}^n$ is a vector whose entries are zero except the i^{th} entry which equals to one, while $\delta \zeta_i$ is a small posture displacement. Substituting (5.20) into (5.19) and differentiating it with respect to $\dot{\zeta}_s$, we can find $\dot{\zeta}_s^*$

$$\dot{\zeta}_s^* = [\mathbf{J}_e^T \mathbf{W}_e \mathbf{J}_e + \mathbf{W}_{JL} + \mathbf{J}_{OA}^T \mathbf{W}_{OA} \mathbf{J}_{OA} + \mathbf{W}_\nu]^{-1} (\mathbf{J}_e^T \mathbf{W}_e \dot{\mathbf{r}}_e^* + \frac{\mu}{2} \delta t \nabla w). \quad (5.22)$$

5.3.2.4 Relaxed Task Formulation

In cases when the redundancy of UAM cannot be fully exploited due to antagonistic tasks, the relaxation of the task (e.g., end-effector positioning) is important to satisfy the robot physical constraints and ensure the motion smoothness. Specifically, it is appealing to let the end-effector to deviate from the desired trajectory for a period of time when multiple opposing tasks are present. In this way, it is possible to enhance the flight behavior while satisfying the antagonistic tasks in a more efficient way. For instance, in case of a pick-and-place task, it is beneficial to slightly deviate from the planned trajectory near the obstacle for a period of time, and then recover to the desired goal afterward.

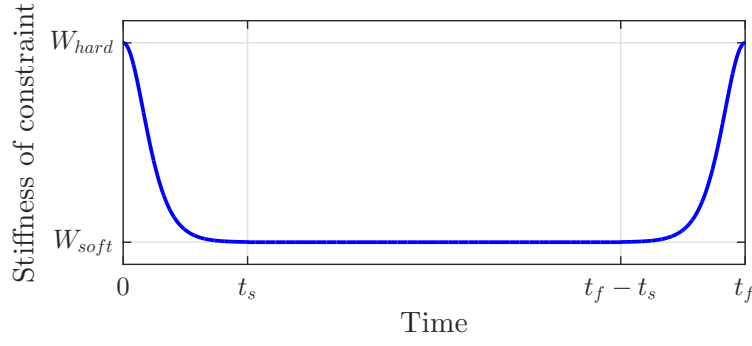
Similarly as in [148], the trajectory relaxation can be implemented by introducing the slack variable $\beta \in \mathbb{R}^3$ into the primary task of the redundancy resolution strategy. Specifically, (5.4) can be modified in the following way, $\dot{\mathbf{r}}_e = \mathbf{J}_e \dot{\zeta}_s + \beta$. Hence, (5.4) can be rewritten as follows

$$\dot{\mathbf{r}}_e = \mathbf{J}_e^\chi \chi, \quad (5.23)$$

where $\chi = [\dot{\zeta}_s^T \beta^T]^T$ and $\mathbf{J}_e^\chi = [\mathbf{J}_e \mathbb{I}_{3 \times 3}]$. Then, the end-effector performs the modified trajectory which is defined in the velocity terms $\dot{\mathbf{r}}_e - \beta$, where β is the deviation from the desired trajectory in the velocity domain. Through similar procedures as in the previous subsections, (5.22) can be modified to account for the trajectory relaxation

$$\begin{aligned} \chi^* = & [(\mathbf{J}_e^\chi)^T \mathbf{W}_e \mathbf{J}_e^\chi + (\mathbf{J}_{OA}^\chi)^T \mathbf{W}_{OA} \mathbf{J}_{OA}^\chi + \\ & (\mathbf{J}_{JL}^\chi)^T \mathbf{W}_{JL} \mathbf{J}_{JL}^\chi + \mathbf{W}_\nu^\chi]^{-1} ((\mathbf{J}_e^\chi)^T \mathbf{W}_e \dot{\mathbf{r}}_e^* + \frac{\mu}{2} \mathbf{W}_m), \end{aligned} \quad (5.24)$$

where $\mathbf{W}_m = [(\delta t \nabla w)^T \mathbb{O}_{1 \times 3}]^T \in \mathbb{R}^{(n_s+3)}$, $\mathbf{J}_{OA}^\chi = [\mathbf{J}_{OA} \mathbb{O}_{1 \times 3}] \in \mathbb{R}^{1 \times (n_s+3)}$, $\mathbf{J}_{JL}^\chi = [\mathbb{I}_{n_s \times n_s} \mathbb{O}_{n_s \times 3}] \in \mathbb{R}^{n_s \times (n_s+3)}$ and $\mathbf{W}_\nu^\chi = \begin{bmatrix} \mathbf{W}_v & \mathbb{O}_{n_s \times 3} \\ \mathbb{O}_{3 \times n_s} & \mathbf{W}_\beta \end{bmatrix} \in \mathbb{R}^{(n_s+3) \times (n_s+3)}$.

FIGURE 5.2: Constraint stiffness profile example for $g_e(t)$.

The weighting matrix $\mathbf{W}_\beta \in \mathbb{R}^{3 \times 3}$ modifies the stiffness of the task relaxation, and it is defined as $\mathbf{W}_\beta = g_e(t) \mathbb{I}_{3 \times 3}$, where $g_e(t)$ is described in the following way [148]

$$g_e(t) = \begin{cases} g_1(t) & \text{if } 0 \leq t \leq t_s \\ W_{soft} & \text{if } t_s \leq t \leq t_f - t_s \\ g_2(t) & \text{if } t_f - t_s \leq t \leq t_f, \end{cases} \quad (5.25)$$

in which $g_1(t)$ and $g_2(t)$ are time-varying continuous functions which ensure a continuous and smooth transition for the slack variable β which leads to the smooth behavior for $\dot{\xi}_s$ during the operation. It should be noted that $g_e(t)$ varies between W_{soft} (in the middle of the task) and W_{hard} (at the beginning and the end of the task) allowing to start and recover to the desired trajectory, while deviating from the desired trajectory in the middle of the task (see Fig. 5.2). In this work, derivative hyperbolic tangent transition [149] is used to define the $g_1(t)$ and $g_2(t)$ functions in the following way

$$\begin{cases} g_1(t) = -(W_{hard} - W_{soft}) \tanh^2 \left(\frac{\mathcal{Q}}{t_2 - t_1} (t - t_1) \right) + W_{hard} \\ g_2(t) = -(W_{hard} - W_{soft}) \tanh^2 \left(\frac{\mathcal{Q}}{t_4 - t_3} (t - t_3) - \mathcal{Q} \right) + W_{hard} \end{cases} \quad (5.26)$$

where $t_1 = 0$, $t_2 = t_s$, $t_3 = t_f - t_s$, $t_4 = t_f$ and $\mathcal{Q} = 4.15$. Please kindly refer to [149] for the detailed explanation of choosing the $g_1(t)$ and $g_2(t)$ functions.

5.3.3 Simulation Studies

Two scenarios are considered to validate the performance of the proposed approach. In the first scenario, the primary task consists of positioning the end-effector along the straight line trajectory while performing the additional tasks such as joint angle limit and obstacle avoidance. In addition to the aforementioned tasks, the maximization of the manipulability measure is also considered in this scenario. Finally, the mission is considered accomplished when the end-effector reaches the goal position $[3.0 \ 0.0 \ 1.37]^T$ m. The second scenario is similar as the first scenario with an addition of the task relaxation. In the second scenario, it is demonstrated that the task relaxation approach can increase the manipulability measure when the redundancy of the system cannot be fully exploited due to the physical constraints of UAM.

For the dynamical simulations, the coaxial tricopter equipped with the 4-DOF manipulator with all revolute joints is implemented in ROS-Gazebo environment. The intrinsic parameters and link masses are the same as in Section 4.3.1. In the beginning of the simulation, the aerial robot is hovering with the following initial configuration, $\zeta_s = [0 \ 0 \ 1.5 \ 0 \ 0 \ -\frac{2\pi}{3} \ \frac{\pi}{4} \ 0]^T$. The obstacle is located at the position

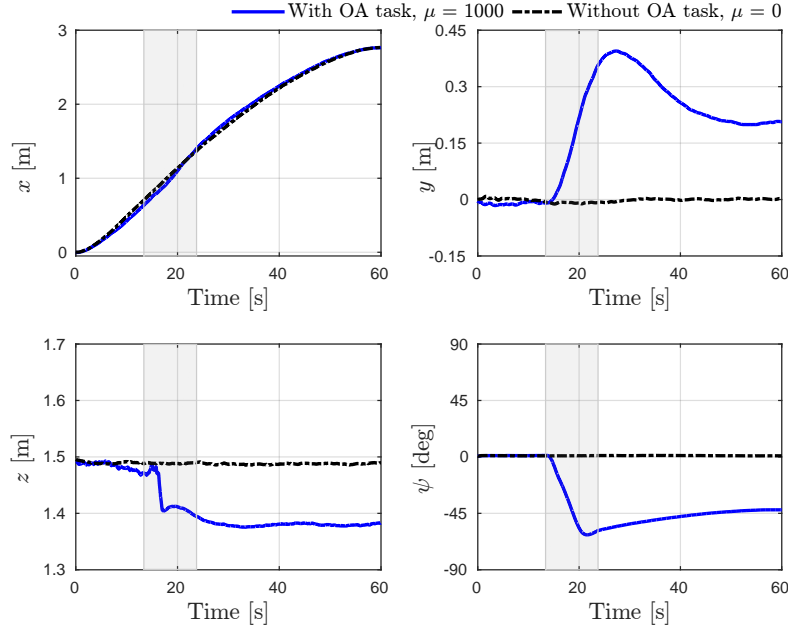


FIGURE 5.3: Position and yaw angle of the UAV platform during scenario 1. The gray shaded regions represent the time period when the OA task is active. Notice how the y position (blue line) is increased to satisfy the OA task.

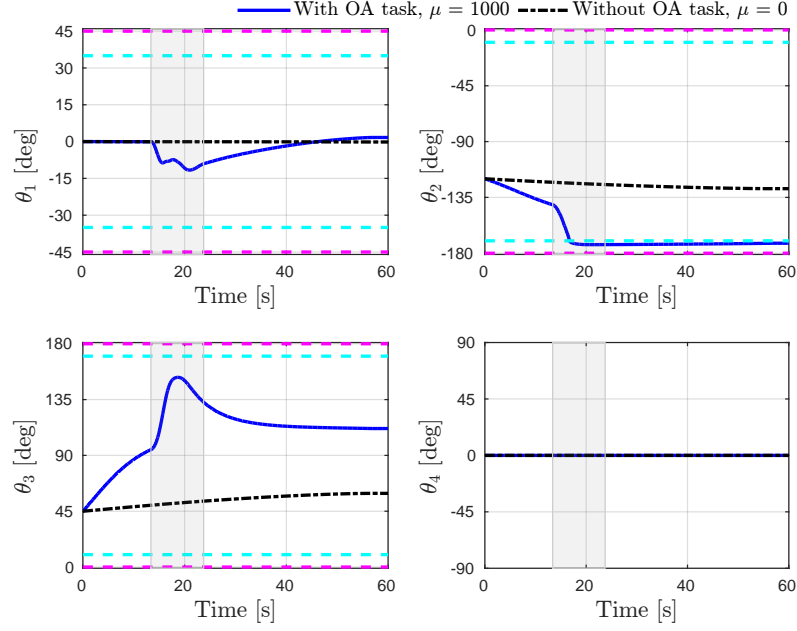


FIGURE 5.4: Arm joint values during scenario 1. The gray shaded regions represent the time period when the OA task is active. The cyan and magenta horizontal dashed lines indicate the start of influence region and joint limits, respectively.

$\mathbf{r}_{ob} = [1.2 \ -0.2 \ 1.5]^T \text{m}$. Trial-and-error method is used to select the parameters for the simulations and they are equal to: $\mathbf{W}_e = 500\mathbb{I}_{3 \times 3}$, $w_{obs} = 150$, $w_{jl} = 100$, $\mathbf{W}_v = 0.3\mathbb{I}_{8 \times 8}$, $W_{hard} = 10^6$ and $W_{soft} = 10$. It should be noted that the JLA task is considered for all cases, and it is only activated when the joint values enter the corresponding influence regions. The total simulation time is equal to 60s, with the time step of $dt = 0.02\text{s}$.

The simulation results for scenario 1 are summarized in Figs. 5.3-5.5. In Fig. 5.3, the position (x , y , and z) and yaw angle (ψ) of the UAV platform are shown, while Fig. 5.4 shows the robotic arm joint values during scenario 1. It can be seen that the proposed algorithm generates the feasible UAM configuration vector ζ to achieve the primary task while satisfying the additional task requirements. In particular, as can be seen in Fig. 5.3, when the OA task is active, the y position of UAV increases to avoid the obstacle, while the yaw angle changes to continue performing the primary task of the end-effector. In addition, from the evolution of θ_2 in Fig. 5.4, it can be noted that the JLA task is activated when θ_2 enters the influence region. The JLA task successfully keeps the joint angle away from the joint limit, preventing self-collision.

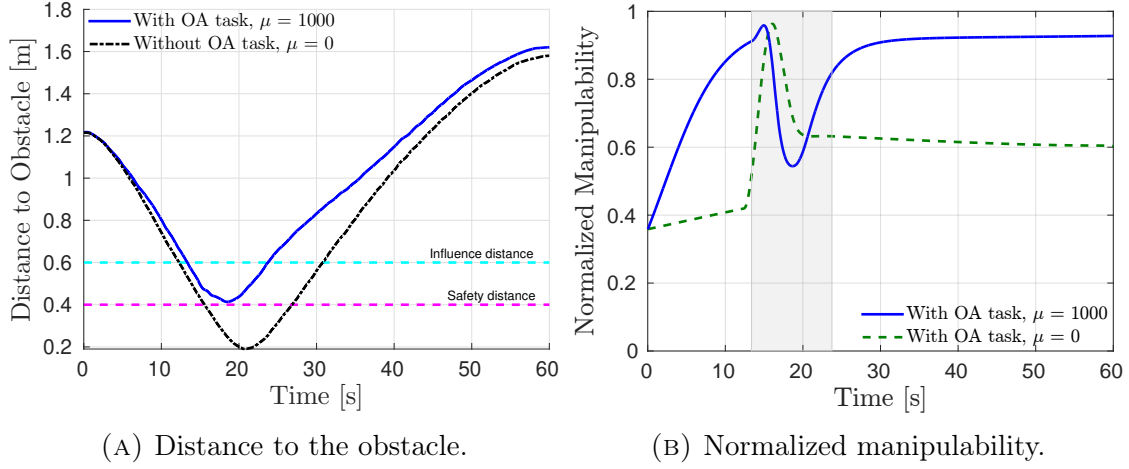


FIGURE 5.5: Distance to the obstacle and normalized manipulability measure evolution during scenario 1. The gray shaded region represents the time period when the OA task is active in scenario 1.

The distance between UAM and the obstacle is presented in Fig. 5.5a for various cases. It can be seen that when the OA task is incorporated as the additional task to the redundancy resolution framework, UAM can successfully avoid the obstacle and does not go beyond the safety distance while performing the primary task. When the OA task is not included in the redundancy resolution framework, UAM crosses the safety distance and cannot avoid collision as shown in Fig. 5.5a by the dashed-dot black line. The evolution of the manipulability measure during scenario 1 is shown in Fig. 5.5b. When the maximization of the manipulability measure is considered ($\mu = 1000$), the manipulability of the arm is improved compared to the case when the maximization of the manipulability measure is neglected ($\mu = 0$) as can be seen in Fig. 5.5b. In addition, Fig. 5.5b shows that the manipulability measure maximization and the OA task are counteracting tasks when UAM is in the vicinity of the obstacle. During that time the manipulability measure approaches to the similar value as the non-maximized case. In this case, the enhancement of the manipulability measure is constrained due to the exploitation of the redundancy and the physical limits of the aerial robot. Nevertheless, this can be overcome by implementing the trajectory relaxation as described in scenario 2.

In the second scenario, the trajectory relaxation for the end-effector's primary task is considered with the same requirements as in scenario 1. In Fig. 5.6a, the distance between UAM and the obstacle is shown. It can be seen that the trajectory relaxation for the primary task does not negatively affect the OA task, allowing UAM to successfully avoid the obstacle. In addition, Fig. 5.6b presents a significant

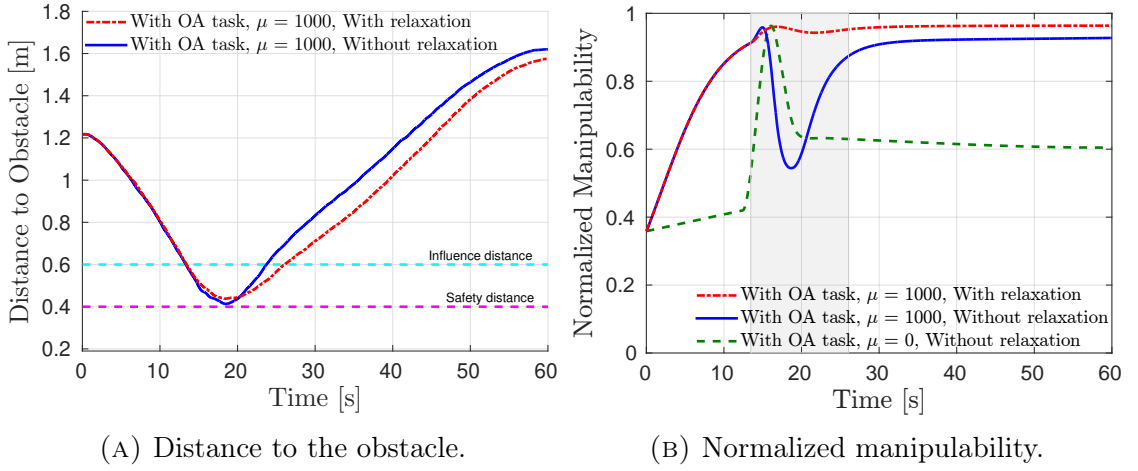


FIGURE 5.6: Distance to the obstacle and normalized manipulability measure evolution during scenario 2. The gray shaded region represents the time period when the OA task is active in scenario 2.

improvement of the manipulability thanks to the trajectory relaxation. Figure 5.7 shows the influence of the trajectory relaxation on the end-effector trajectory. It can be seen that the trajectory relaxation causes the reasonable deviations in the end-effector trajectory in order to increase the manipulability measure in the vicinity of the obstacle. However, after passing the obstacle the end-effector resumes the task and recovers towards the desired goal.

Figure 5.8 shows a sequence of snapshots of the UAM motion during scenario 2. In frame 1, UAM starts its motion towards the goal position represented by a small

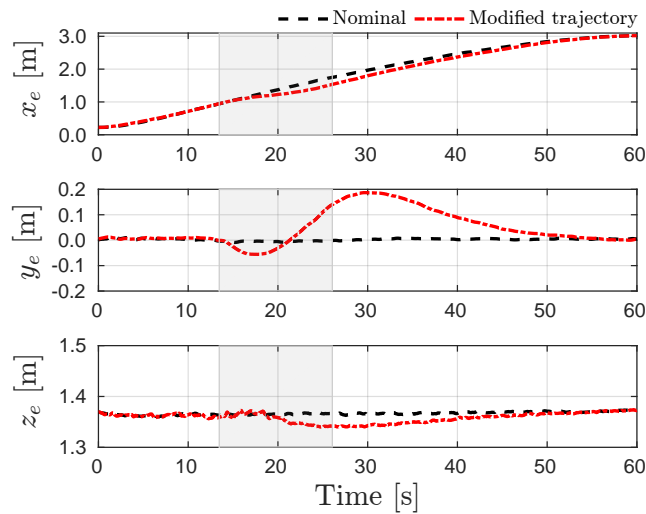


FIGURE 5.7: Trajectory of the end-effector while implementing the trajectory relaxation in scenario 2. The gray regions represent the activation of the OA task in scenario 2.

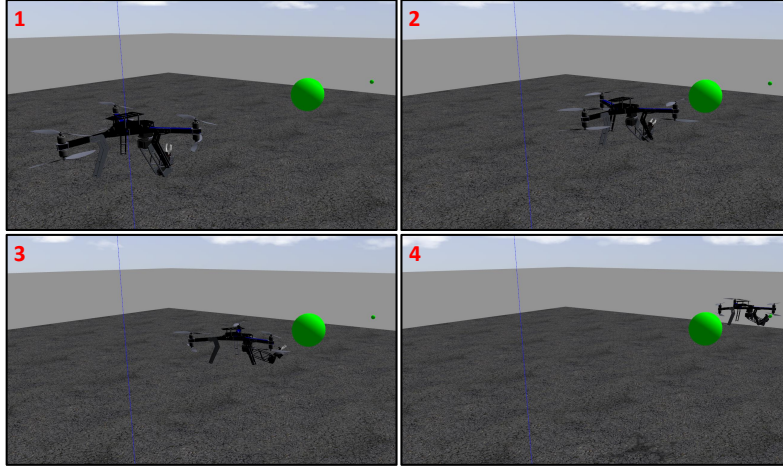


FIGURE 5.8: A sequence of snapshots of the UAM motion in scenario 2: (1) UAM starts its motion; (2)-(3) intermediate states; (4) UAM reaches the goal.

green sphere. When UAM approaches to the obstacle (big green sphere) as shown in frame 2, it is pushed sideways to fulfill the OA task as shown in frame 3. Finally, the mission is accomplished as the end-effector reaches the goal position (frame 4).

5.4 NMPC-based Trajectory Generation

MPC is a control method which solves optimization problem numerically over a finite window, commonly called as the prediction horizon (N_p), while satisfying some constraints. Nonlinear MPC (NMPC) corresponds to particular MPC cases where a nonlinear process model, non-quadratic cost functions or nonlinear constraints are utilized during the formulation of MPC problem.

In comparison to the previous approach, NMPC-based method allows to handle system constraints explicitly without defining additional tasks as it was done in Section 5.3. In addition, utilizing NMPC-based trajectory generation for UAMs may lead to a better long-term motion strategy due to the fact that MPC has the ability to predict the future state evolution within the prediction horizon. However, it should be noted that this approach is known to be computationally demanding because of its iterative nature. Therefore, in order to account for the real-time implementation of this approach, especially for UAMs, the prediction horizon length should be wisely selected, considering the available onboard computational power.

5.4.1 General Formulation

NMPC minimizes the cost function over the prediction horizon $t_i \leq t \leq t_{i+N_p}$, where N_p is a horizon length, at each sampling instant (t_i) and finds optimal control inputs for each time interval. In this work, the NMPC cost function is formulated as follows

$$\min_{\Lambda_k, \mathbf{u}_k} \sum_{j=1}^{N_\Gamma} \Gamma_j + \sum_{k=i}^{i+N_p-1} \|\mathbf{u}_k\|_{\mathbf{Q}_u}^2 \quad (5.27a)$$

$$\text{subj. to } \Lambda_{k+1} = f(\Lambda_k, \mathbf{u}_k) \quad (5.27b)$$

$$\Lambda_{min} \leq \Lambda_k \leq \Lambda_{max} \quad (5.27c)$$

$$\mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \quad (5.27d)$$

$$\mathbf{y}_{min} \leq g(\Lambda_k, \mathbf{u}_k) \leq \mathbf{y}_{max} \quad (5.27e)$$

where $\Lambda_k = [\zeta_{d,k}, \dot{\zeta}_{d,k}]^T \in \mathbb{R}^{24}$ is the differential state vector, and it is bounded by Λ_{min} and Λ_{max} as defined in (5.27c), while $\mathbf{u}_k \in \mathbb{R}^{12}$ is the control input, which is bounded by \mathbf{u}_{min} and \mathbf{u}_{max} as defined in (5.27d), $\mathbf{Q}_u \in \mathbb{R}^{12 \times 12}$ is the weight matrix on control actions. The robot's state-space dynamics is represented by $f(\Lambda_k, \mathbf{u}_k)$ as shown in (5.27b). Γ_j is the cost function corresponding to a generic task, and N_Γ is the number of tasks. As for (5.27e), it specifies bounds of a generic constraint $g(\Lambda_k, \mathbf{u}_k)$. Notice that the cost function of NMPC (5.27) includes the term $\|\mathbf{u}_k\|_{\mathbf{Q}_u}^2$ which penalizes control actions. By selecting appropriate \mathbf{Q}_u , one can favor either UAV base or arms' motion depending on the task.

5.4.2 UAM Tasks and Constraints

This sections presents and discusses several cost functions, which correspond to the tasks done by UAMs. In addition, unlike in Section 5.3.2, system constraints are explicitly imposed without defining them as additional tasks.

5.4.2.1 End-effector tracking

As we mentioned earlier in Section 5.3.1, the main interaction during UAM missions is performed by the end-effectors of the aerial manipulator, and thus it is essential

to be able to follow the desired end-effector trajectories. Hence, we define two main tasks Γ_1 and Γ_2 . The goal of these tasks is to track desired trajectories $\mathbf{r}_*^1(t)$ and $\mathbf{r}_*^2(t)$ for both arms. Then, the cost function, Γ_1 , can be formulated as follows

$$\Gamma_1 = \sum_{k=i}^{i+N_p-1} \|\mathbf{r}_e^1(\zeta_{d,k}) - \mathbf{r}_*^1(t_k)\|_{\mathbf{Q}_1}^2 \quad (5.28a)$$

$$+ \|\mathbf{r}_e^1(\zeta_{d,i+N_p}) - \mathbf{r}_*^1(t_{i+N_p})\|_{\mathbf{W}_1}^2, \quad (5.28b)$$

where the first term (5.28a) minimizes the tracking error within prediction horizon length N_p , while the second term (5.28b) optimizes the terminal cost at the end of prediction horizon. \mathbf{Q}_1 and \mathbf{W}_1 are weight matrices for the task and terminal cost. Analogously, the cost function Γ_2 for the second arm is defined as

$$\Gamma_2 = \sum_{k=i}^{i+N_p-1} \|\mathbf{r}_e^2(\zeta_{d,k}) - \mathbf{r}_*^2(t_k)\|_{\mathbf{Q}_2}^2 \quad (5.29a)$$

$$+ \|\mathbf{r}_e^2(\zeta_{d,i+N_p}) - \mathbf{r}_*^2(t_{i+N_p})\|_{\mathbf{W}_2}^2. \quad (5.29b)$$

where \mathbf{Q}_2 and \mathbf{W}_2 are weight matrices for the task and terminal cost corresponding to the second arm. Notice that having separate cost functions allows to obtain independent movement for each arm, thus increasing the dexterity of the overall system.

5.4.2.2 Velocity minimization

In order to achieve a smooth flight behavior, reduce energy consumption and wear of the mechanical components, it is important to keep velocities low, when it is possible. This can be achieved by formulating the cost function Γ_3 in the following form

$$\Gamma_3 = \sum_{k=i}^{i+N_p-1} \|\dot{\zeta}_{d,k}\|_{\mathbf{Q}_3}^2 + \|\dot{\zeta}_{d,i+N_p}\|_{\mathbf{W}_3}^2, \quad (5.30)$$

where \mathbf{Q}_3 and \mathbf{W}_3 are weight matrices for velocity minimization task and terminal cost.

5.4.2.3 Minimization of the COG displacement

As we mentioned earlier, the motion of the robotic arms causes the displacement in COG, which, in its turn, can lead to the pose variation of the whole aerial robot. Therefore, the minimization of the COG displacement can be implemented to improve the flight behavior. Similarly as in [126], this can be achieved by formulating the cost function Γ_4 in the following form

$$\Gamma_4 = \sum_{k=i}^{i+N_p-1} \|\mathbf{r}_{COG}^b(\boldsymbol{\zeta}_{d,k})\|_{\mathbf{Q}_4}^2 + \|\mathbf{r}_{COG}^b(\boldsymbol{\zeta}_{d,i+N_p})\|_{\mathbf{W}_4}^2, \quad (5.31)$$

where \mathbf{Q}_4 and \mathbf{W}_4 are weight matrices for the COG displacement minimization task and terminal cost, while \mathbf{r}_{COG}^b represents the projection of the dual-arm COG on xy -plane defined in the body-fixed frame C_b .

5.4.2.4 Singularity-free motion

The singularity-free regions can be achieved by considering the manipulability measure. The manipulability measures for both arms, ω_1 and ω_2 , are defined as

$$\omega_m(\mathbf{q}^m) = \sqrt{\det(J(\mathbf{q}^m)J(\mathbf{q}^m)^T)}, \quad (5.32)$$

where $J(\mathbf{q}^m)$ is Jacobian matrix of the m -th arm. Having higher manipulability yields better maneuverability for the robotic arm so that the arm can have enough admissible motions in case of unforeseen events, while when the manipulability measure is very small, the arm can be near singular configuration. Therefore, it is desirable to have high manipulability. However, in Section 5.3, we learned that the manipulability measure maximization and the OA task can be antagonistic tasks when UAM is in the vicinity of the obstacle. Therefore, instead of directly maximizing the manipulability measure, the singularity-free motion is imposed by specifying a minimum value of manipulability measure ($\bar{\omega}$) via following constraint

$$\omega_m(\mathbf{q}^m) \geq \bar{\omega} \quad (5.33)$$

This can decrease the occurrence of sudden peaks in joint velocities and accelerations.

5.4.2.5 Collision avoidance

In this section, we define the collision avoidance constraints for the aerial robot during trajectory generation.

Self-collision avoidance: To prevent self-collision of the aerial robot, two methods are implemented. The first method is explicitly imposing the constraints on joint limits of \mathbf{q}^1 and \mathbf{q}^2 as defined in (5.27c), while the second method is to create the virtual walls between the arms and UAV base as well as between the arms themselves to delimit their working regions. The virtual wall between the UAV base and arms can be created by imposing a safety distance between the position of the dual-arm end-effectors and UAV base in z -axis

$$\begin{cases} z_b - z_e^1 \geq \bar{z}, \\ z_b - z_e^2 \geq \bar{z} \end{cases} \quad (5.34)$$

where z_e^1 and z_e^2 are the position of the end-effectors in the z -direction of the world-fixed reference frame, while \bar{z} is a imposed safety distance. The virtual wall between the arms can be imposed by separating the working spaces of the arms along the y -direction of the body-fixed frame C_b

$$\begin{cases} y_e^{1,b} \leq 0, & y_{elb}^{1,b} \leq 0 \\ y_e^{2,b} \geq 0, & y_{elb}^{2,b} \geq 0 \end{cases} \quad (5.35)$$

where $y_e^{1,2,b}$ and $y_{elb}^{1,2,b}$ are the position of the end-effectors and elbow joints in the y -direction, respectively, defined in the body-fixed frame C_b . Hence, considering the aforementioned conditions, it is ensured that the self-collision will never occur.

Obstacle avoidance: In order to implement obstacle avoidance, the obstacles are modeled as ellipsoids using minimum volume enclosing ellipsoid (MVEE) technique [150]. In addition, we represent the aerial robot as the union of the different ellipsoids E_l with centers at \mathbf{c}_l , $l = 1, \dots, 5$, as shown in Fig. 5.9. Since E_l ellipsoids fully enclose the aerial robot, these ellipsoids and ellipsoids representing the obstacles should not intersect. Similarly as in [151], this can be achieved by imposing the following constraints

$$\nu_{d,l}(\mathbf{c}_l) \geq 1 \quad (5.36)$$

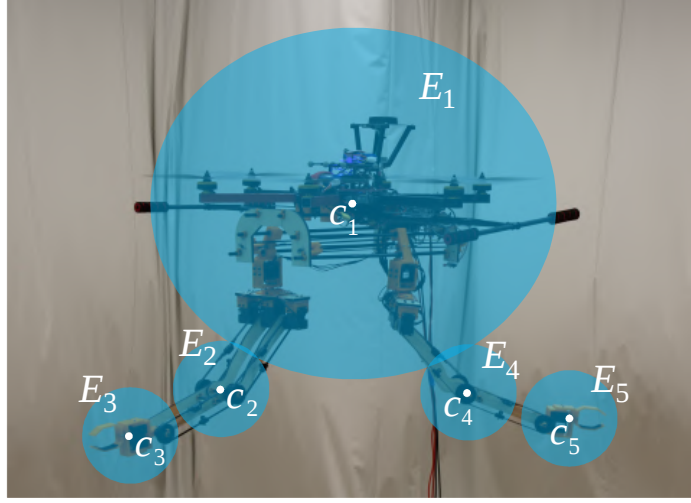


FIGURE 5.9: Graphical representation of the aerial robot enclosed by 5 ellipsoids, E_l , with centers at \mathbf{c}_l , $l = 1, \dots, 5$.

where $\nu_{d,l}(\mathbf{c}_l)$ is defined as

$$\nu_{d,l}(\mathbf{c}_l) = (\mathbf{c}_l - \mathbf{o}_d)^T \mathbf{A}_d (\mathbf{c}_l - \mathbf{o}_d) \quad (5.37)$$

in which \mathbf{o}_d is center of MVEE for d -th obstacle, $d = 1, \dots, N_d$ (N_d is the number of obstacles). The diagonal matrix \mathbf{A}_d defines ellipsoid's radii, and it composes of radii of MVEE of d -th obstacle and corresponding E_l ellipsoid. A safety margin is added to the radii of obstacle's ellipsoid, so that the slight violations of the constraints are considered.

After defining aforementioned constraints, the final form of NMPC cost function is formulated as follows

$$\begin{aligned} \min_{\mathbf{\Lambda}_k, \mathbf{u}_k} \quad & \sum_{j=1}^{N_\Gamma} \Gamma_j + \sum_{k=i}^{i+N_p-1} \|\mathbf{u}_k\|_{Q_u}^2 \\ \text{subj. to} \quad & \mathbf{\Lambda}_{k+1} = f(\mathbf{\Lambda}_k, \mathbf{u}_k) \\ & \mathbf{\Lambda}_{min} \leq \mathbf{\Lambda}_k \leq \mathbf{\Lambda}_{max}, \quad \mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \\ & \omega_m(\mathbf{q}^m) \geq \bar{\omega}, \quad z_b - z_e^{1,2} \geq \bar{z}, \quad \nu_{d,l}(\mathbf{c}_l) \geq 1 \\ & y_e^{1,b} \leq 0, \quad y_{elb}^{1,b} \leq 0, \quad y_e^{2,b} \geq 0, \quad y_{elb}^{2,b} \geq 0 \end{aligned} \quad (5.38)$$

The values of $\mathbf{\Lambda}$ and \mathbf{u} at time t_i is obtained as the solution to the above MPC problem. As indicated in Chapter 3, the dual-arm system has the pre-configured

internal controllers for each Dynamixel actuators, while the hexarotor employs Pixhawk as the low-level controller. These controllers are relatively fast and can be used to track the positional set-points, allowing us to place the NMPC-based trajectory generator at the top level as a high-level controller, which generates the desired configuration state (ζ_d^*) for UAM as shown in Fig. 5.10.

5.4.3 Simulation and Experimental Studies

To validate the performance of the proposed NMPC-based approach presented in the previous subsections, two scenarios are considered. In both scenarios, the dual-arm end-defectors are commanded to track the desired straight line trajectories while considering tasks and constraints introduced in Section 5.4.2. For scenario 1, tasks defined in (5.28)–(5.30) and constraints defined in (5.33), (5.34), and (5.36) are considered. Instead of creating the virtual wall between the arms (5.35), we set $q_2^{1,2} = 0$ to prevent the self-collision between the arms. Besides, only one obstacle is considered for this scenario, and it is located at $\mathbf{o}_1 = [0 \ 0 \ 0.75]^T \text{m}$. For scenario 2, tasks defined in (5.28)–(5.30) and all constraints defined in (5.33)–(5.36) are considered. This time the virtual wall between the arms is explicitly imposed without constraining $q_2^{1,2}$. In addition, two obstacles are considered for scenario 2. The first obstacle is located at $\mathbf{o}_1 = [0 \ -0.5 \ 0.9]^T \text{m}$, while the second obstacle is located at $\mathbf{o}_2 = [0 \ 0.5 \ 0.9]^T \text{m}$. In both scenarios, the following constraints are imposed for motion variables: $z_b \leq 1.6 \text{ m}$, $|q_3^{1,2}| \leq \frac{\pi}{2}$ and $q_{4,5}^{1,2} \in [0, \pi]$. The manipulability threshold is set to $\bar{\omega} = 10^{-8}$ for both arms, while the safety distance is set to $\bar{z} = 0.2 \text{ m}$. The elliptic radii of ellipsoid E_1 are set to be $[0.45 \ 0.45 \ 0.35] \text{ m}$, while the radii of other ellipsoids are equal, and they are set to be 0.1 m . Extensive simulation and experimental studies have been performed to evaluate the performance of the proposed framework. In particular, for the simulation studies, the hexarotor equipped with the dual-arm system is implemented in ROS-Gazebo environment as shown in Fig. 5.11, while the experimental flight tests are performed in the OptiTrack motion capture system laboratory, which provides the real-time pose measurements of UAV with 100Hz and accuracy around 0.5 mm . The UAV's velocities are obtained via numerical differentiation, which is based on the moving average with 10 samples. The joint positions and velocities are provided by the actuators. The end-effector positions are derived from the forward kinematics. Trial-and-error method is used to select the parameters for trajectory

generation algorithm, and they are equal to: $\mathbf{Q}_{1,2} = 10\mathbb{I}_{3 \times 3}$, $\mathbf{W}_{1,2} = 10^{-1}\mathbb{I}_{3 \times 3}$, $\mathbf{Q}_3 = \mathbf{Q}_u = 10^{-5}\mathbb{I}_{12 \times 12}$, and $\mathbf{W}_3 = 10^{-1}\mathbb{I}_{12 \times 12}$, where \mathbb{I}_{**} is the identity matrix. The discrete solutions of NMPC problem are solved by a direct multiple-shooting approach with number of integrator steps 20. The prediction horizon is set to $N_p = 10$ and it divides the horizon time, $T = 1$ s, into equal time intervals with length of 100 ms. The choice of T and N_p comes from trade off between computational complexity and ability to predict future evolution of states. Since the low-level controllers work in higher rates, the NMPC update time 100 ms is enough for high-level control. The formulated NMPC given in (5.38) is solved using ACADO Toolkit [152] which solves through sequential quadratic programming. The solution at each time comes from solving qpOASES 5 times [153] with Gauss-Newton approximation of Hessian matrix.

The block diagram of the dual-arm aerial manipulator architecture for the experimental flight tests is shown in Fig. 5.10. The onboard computer (Intel NUC) performs four main software programs to control the aerial robot: Operation Manager, NMPC-based Trajectory Generator, Dual-Arm and UAV Controllers. The Operation Manager can be considered as the high-level supervisor that helps to execute the tasks chosen by the user from the ground control station. It collects the robot states and provides the motion commands for the dual-arm system (e.g., go to rest and operation positions) and the aerial platform (e.g., take-off and landing commands). In addition, it provides the desired goal (e.g., the desired trajectory for the dual-arm end-effectors) for the NMPC-based Trajectory Generator module, in which the feasible motion variables are generated to perform the assigned task while satisfying different constraints. The Dual-Arm and UAV Controller command the joints and hexarotor to move to the reference values received from Trajectory Generator and Operation Manager, while simultaneously sending back their current states. An off-the-shelf Pixhawk autopilot with PX4 flight stack is utilized as the low-level controller of the aerial platform. As for the dual-arm motors, internal controller of each Dynamixel actuator manages the joint operations, while Dual VNH5019 motor driver shield together with Arduino Mega 2560 are used to control the rotation direction and speed of the DC motors based on the encoder readings and references. All control programs running on the Intel NUC (with Ubuntu 16.04 OS) are developed in C/C++ using ROS environment for the better software portability between different onboard computers. The video of the experimental flight tests can be seen via this link (<https://youtu.be/8n-Vz4ECr2U>).

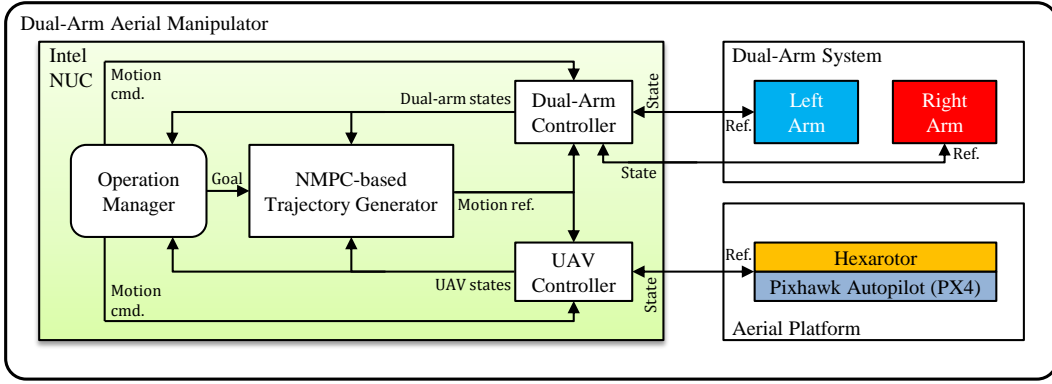


FIGURE 5.10: Block diagram of the dual-arm aerial manipulator architecture for the experimental flight tests.

5.4.3.1 Simulations Results

The simulation results for scenario 1 are summarized in Figs. 5.11-5.15. It can be seen that the presented method generates the feasible configuration state ζ_d to perform the assigned tasks while ensuring the constraints satisfaction. In Fig. 5.11, a sequence of images of the dual-arm aerial manipulator motion during the simulations is shown. As can be seen, the dual-arm aerial manipulator starts its motion by tracking the desired trajectories for both end-effectors. When it approaches to the obstacle (orange box enclosed by white ellipsoid) as shown in frame 2, instead of just flying away, the aerial robot exploits its redundancy and squeezes the arms to avoid the collision with the obstacle (frame 3). It should be noted that this behavior is achieved by constraining the z -axis motion of the UAV platform ($z_b \leq 1.6$ m). After passing the obstacle, the algorithm resumes the tracking the end-effector trajectories (frame 4), and the aerial robot reaches the final destination as shown in frame 5.

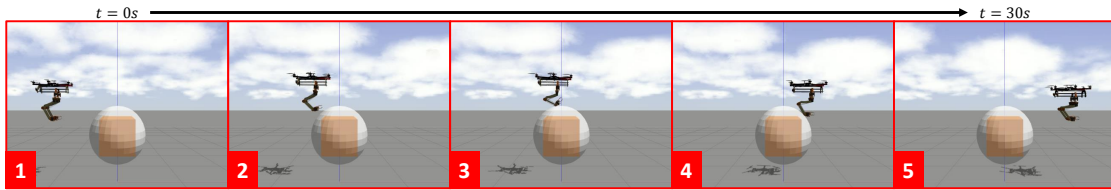


FIGURE 5.11: A sequence of images of the dual-arm aerial manipulator motion during the simulations in scenario 1. The orange box represents the obstacle, which is enclosed by white ellipsoid.

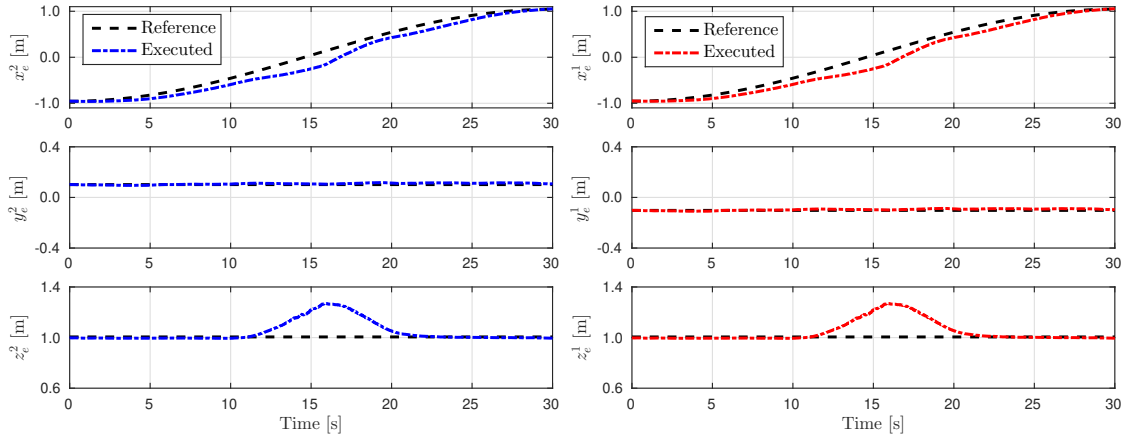


FIGURE 5.12: Reference and executed trajectories of the left (left column) and right (right column) end-effectors in scenario 1.

In Fig. 5.12, the tracking performance of the end-effector trajectories is presented. As we previously mentioned, in the vicinity of the obstacle, the proposed algorithm allows increasing of the tracking error for both end-effectors (this is mainly visible for $z_e^{1,2}$ and $x_e^{1,2}$), in order to achieve the best tracking performance while fulfilling the existed constraints. This can be also seen from Fig. 5.13, where 3D view of the position of the UAV platform and dual-arm end-effectors are shown. As it can be seen, the obstacle is modeled as the ellipsoid which fully encloses it.

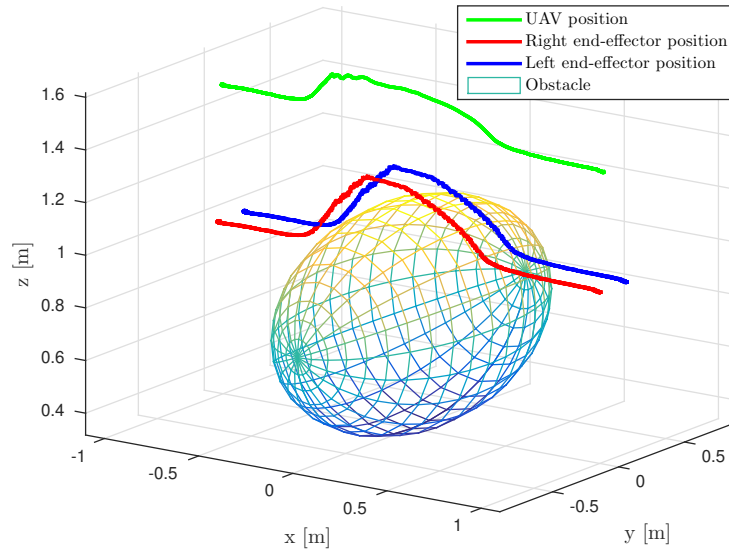


FIGURE 5.13: Position of the UAV platform and dual-arm end-effectors during the simulations in scenario 1. Notice that the end-effectors modified their trajectories to avoid the obstacle.

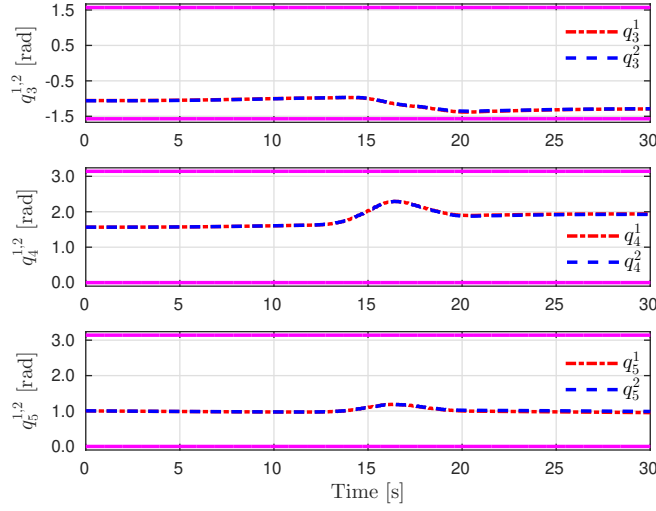


FIGURE 5.14: Joint angles of the dual-arm during the simulations in scenario 1. The corresponding upper and lower bounds of the joint limit constraints are represented as solid magenta lines. Since the values of ψ and $q_2^{1,2}$ are very small and close to zero, they are not shown in this figure.

Figure 5.14 presents the joint angles of the dual-arm with corresponding bounds. It is shown that the joint angles are kept within the imposed limits, preventing self-collision. In Fig. 5.15, the measured time evolution of the constraint variables are shown. In Fig. 5.15 (top left), the manipulability measures for both arms are

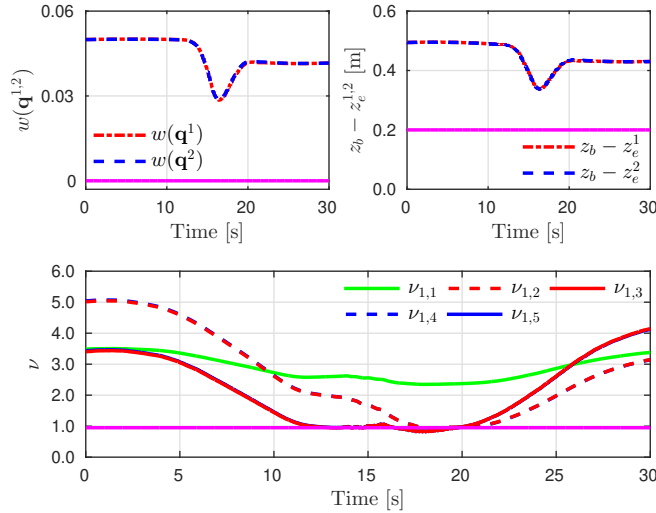


FIGURE 5.15: Constraint variables during the simulations in scenario 1: singularity-free motion (top left), safety distances between the UAV base and dual-arm end-effectors (top right), and value of ν related to the avoidance of obstacle (bottom). The corresponding bounds are represented as solid magenta lines.

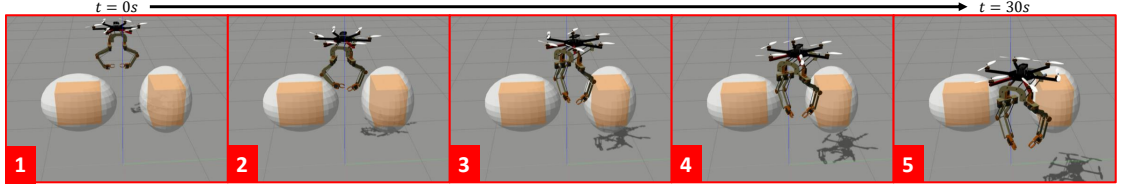


FIGURE 5.16: A sequence of images of the dual-arm aerial manipulator motion during simulations in scenario 2. The orange boxes represent the obstacles, which are enclosed by white ellipsoids.

displayed, and they are always above the imposed threshold. In Fig. 5.15 (top right), the time evolution of the safety distances between the UAV base and dual-arm end-effectors is presented. It is clearly visible that the safety distances are kept above the corresponding threshold. Therefore, considering the bounded joint angles and safety distances, it is ensured that the self-collision will never occur. Finally, values of $\nu_{d,l}$ (defined in (5.37)) related to the avoidance of obstacle are shown in Fig. 5.15 (bottom). As it can be seen, $\nu_{1,1}$, $\nu_{1,2}$ and $\nu_{1,4}$, which correspond to the UAV body and elbow joints of the dual-arm, never violate the imposed threshold. As for $\nu_{1,3}$ and $\nu_{1,5}$, which represent the dual-arm end-effectors, they are on the threshold and slightly below it during some time intervals. However, the presence of the safety margins during the definition of the ellipsoids ensures that the slight violation will not cause the collisions.

The simulation results for scenario 2 are summarized in Figs. 5.16-5.20. As it can be seen from Fig. 5.16, the initial configuration of the arms is different compared

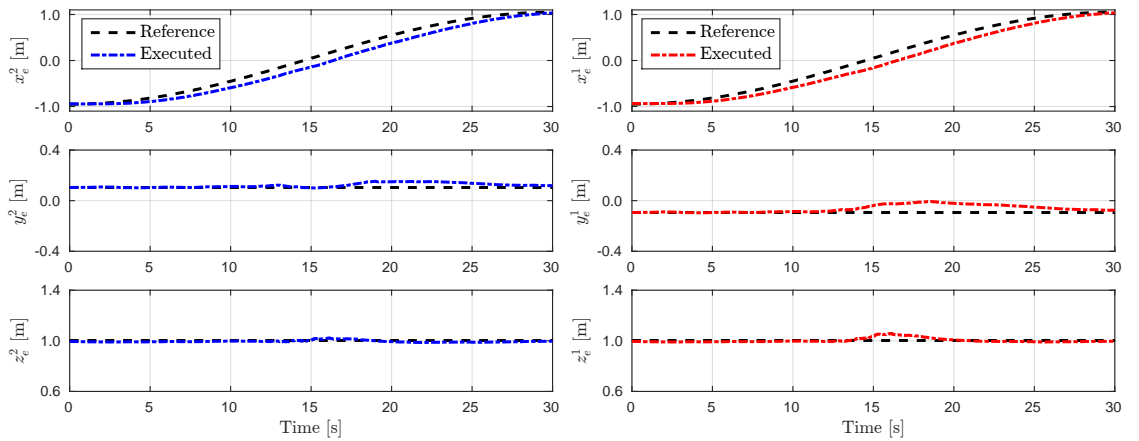


FIGURE 5.17: Reference and executed trajectories of the left (left column) and right (right column) end-effectors in scenario 2

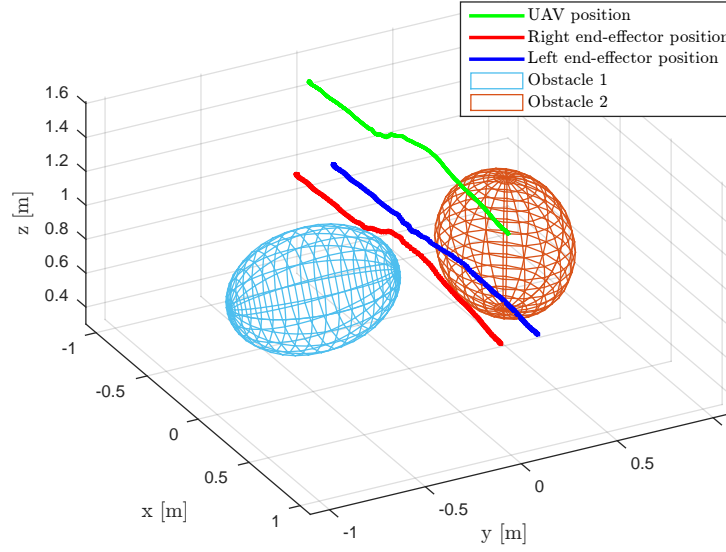


FIGURE 5.18: Position of the UAV platform and dual-arm end-effectors during the simulations in scenario 2

to the previous scenario (frame 1). Besides, this time there are two obstacles along the way of UAM. Therefore, when the aerial robot approaches to the obstacles (frame 2), the proposed approach allows the robot to slide in between them (frame 3), thus successfully avoiding the collision (frame 4 and Fig. 5.18). It should be noted that this behavior can be quite useful when aerial robots work in a confined environment. In Fig. 5.17, we can see that there are still slight deviations between

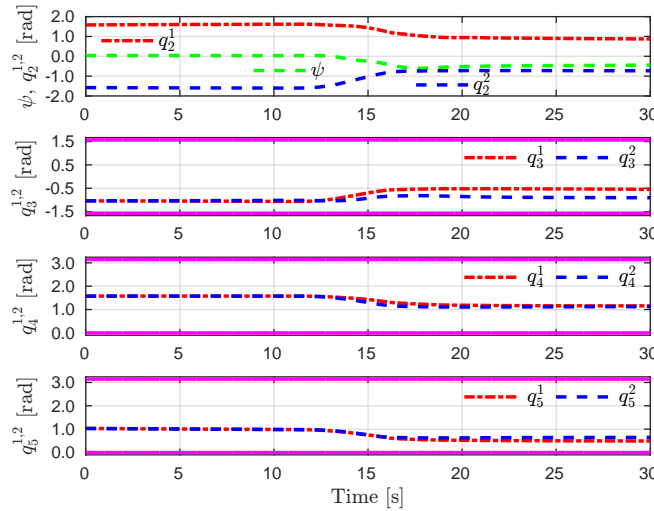


FIGURE 5.19: Joint angles of the dual-arm during the simulations in scenario 2. The corresponding upper and lower bounds of the joint limit constraints are represented as solid magenta lines.

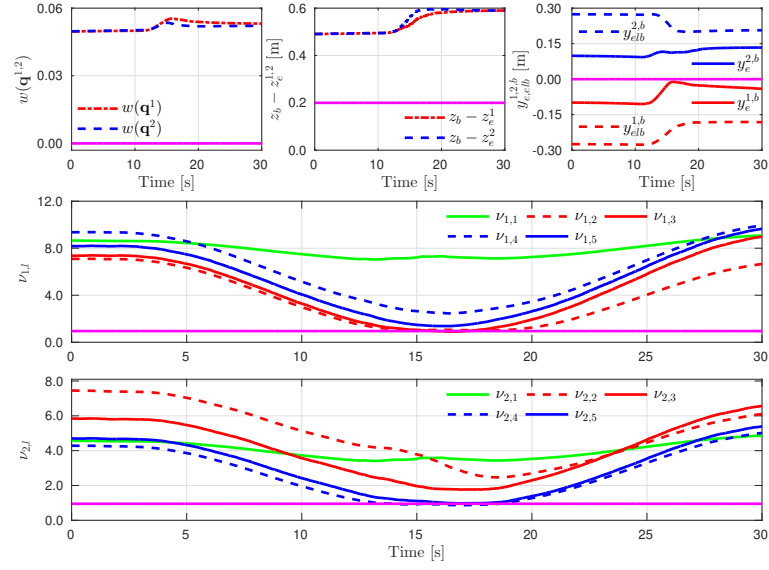


FIGURE 5.20: Constraint variables during simulations in scenario 2: singularity-free motion (top left), safety distances between the end-effectors and UAV base (top middle), virtual wall between the arms (top right), and value of ν related to the avoidance of the first obstacle (middle) and the second obstacle (bottom). The corresponding bounds are represented as solid magenta lines.

the reference and executed trajectories of the end-effectors. This is mainly because the proposed method tries to find the trade-off between the tracking performance and fulfilling the constraints. From Fig. 5.19, it can be noticed that not only the configurations of the arms are modified, but also the proposed method takes advantage of the yaw motion of UAV to obtain the best trade-off mentioned above. Fig. 5.20 shows the measured time evolution of the constraint variables. It can be seen that the performance is similar to the previous scenario. The only difference is the inclusion of the virtual wall between the arms. In Fig. 5.20 (top right), it can be observed that the proposed approach keeps the right arm in its working region, preventing the collision with the left arm.

5.4.3.2 Experimental Flight Tests

In order to verify the real-time implementation of the proposed approach, only scenario 1 is considered for experimental flight tests. The experimental results are summarized in Figs. 5.21-5.25. The obtained results are similar with simulation case studies. In Fig. 5.21, a sequence of images of the dual-arm aerial manipulator motion during the experimental flight tests is displayed. Similarly as in Section

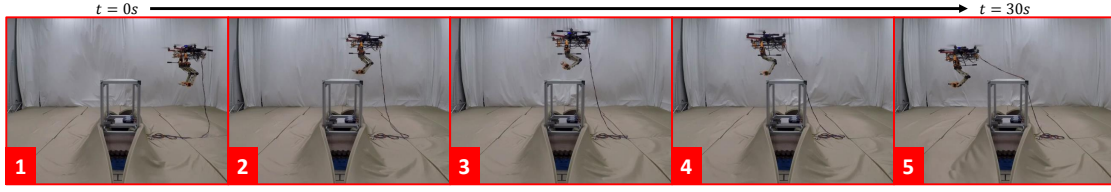


FIGURE 5.21: A sequence of images of the dual-arm aerial manipulator motion during the experimental flight tests in scenario 1. The metal frame represents the obstacle, which is also enclosed by imaginary ellipsoid. 0.35 m long bar is attached to the end-effectors.

5.4.3.1, the actual robot takes advantage of its redundancy, and it is able to avoid the obstacle by squeezing the arms as illustrated in frame 3. After avoiding the obstacle, the aerial robot returns to track the end-effector trajectories (frame 4) and continues its flight towards the final destination as shown in frame 5.

Trajectory tracking performance of the end-effectors shown in Fig. 5.22 demonstrates the similar behavior as in simulations. However, there are oscillations in y -axis response, which show that the finer tuning in Pixhawk low-level controller is required. The offset in x -axis response towards the end of trajectory was caused by the disturbance generated from tethered cable. 3D view of the position of the UAV platform and dual-arm end-effectors are shown in Fig. 5.23.

The joint angles and measured time evolution of the constraint variables during the experimental flight tests are shown in Figs. 5.24 and 5.25, respectively. Similarly as in simulations, the joint angles, manipulability measures and the safety distances

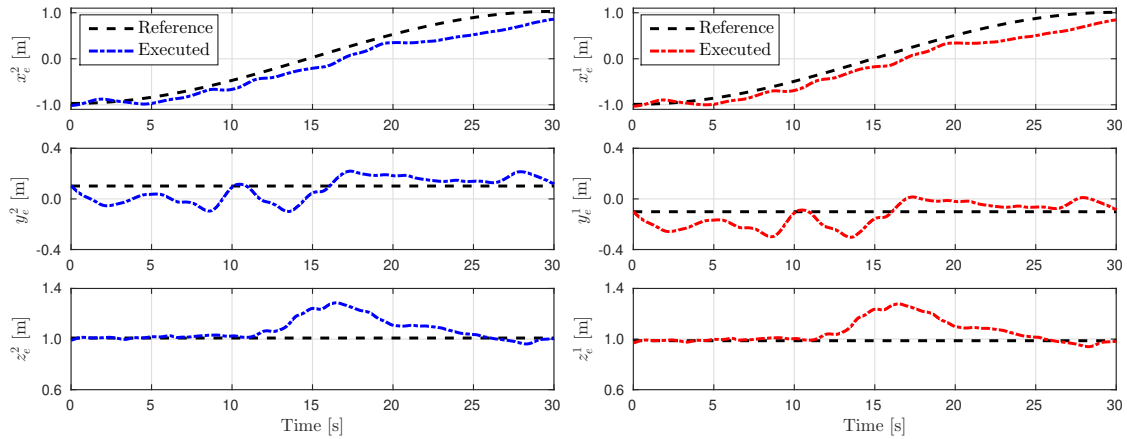


FIGURE 5.22: Reference and executed trajectories of the left (left column) and right (right column) end-effectors in scenario 1.

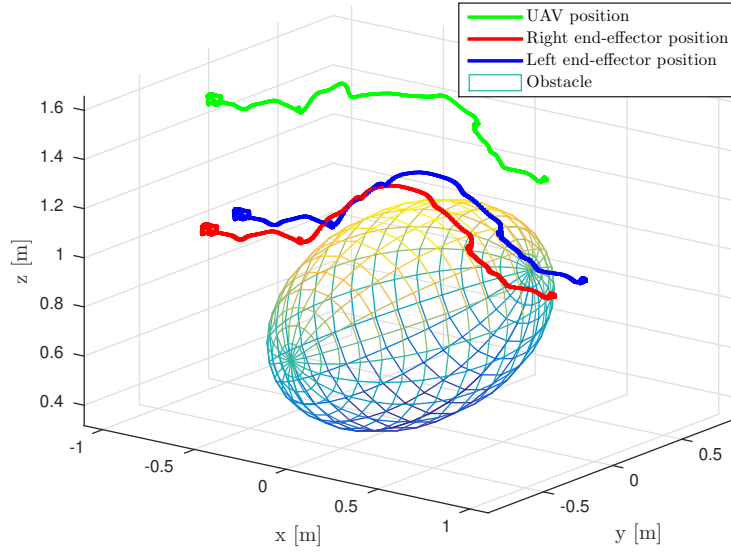


FIGURE 5.23: Position of the UAV platform and dual-arm end-effectors during the experimental flight tests in scenario 1. Notice that the end-effectors modified their trajectories to avoid the obstacle.

are kept within the imposed limits. Hence, it is ensured that the self-collision is avoided. As for values of $\nu_{d,l}$, it can be seen that $\nu_{1,1}$ corresponding to the UAV body never violates the imposed threshold, while the rest of $\nu_{d,l}$ are slightly below the threshold during some time intervals, which are common for the practical applications. However, as we indicated previously, the safety margins will ensure that no collisions occur.

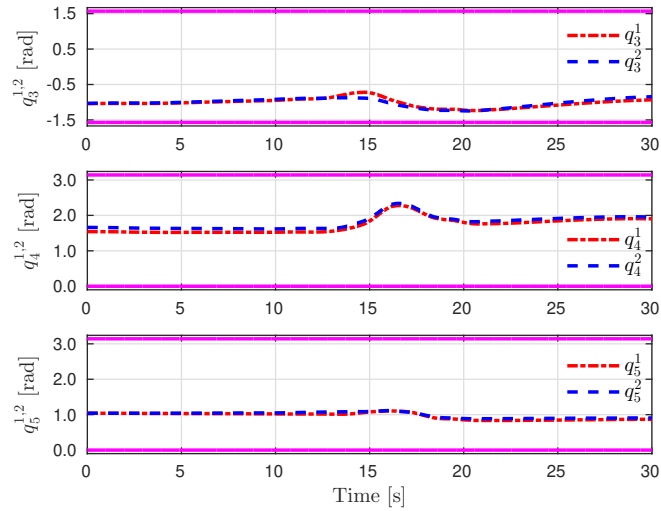


FIGURE 5.24: Joint angles of the dual-arm during the experimental flight tests in scenario 1. The corresponding upper and lower bounds of the joint limit constraints are represented as solid magenta lines.

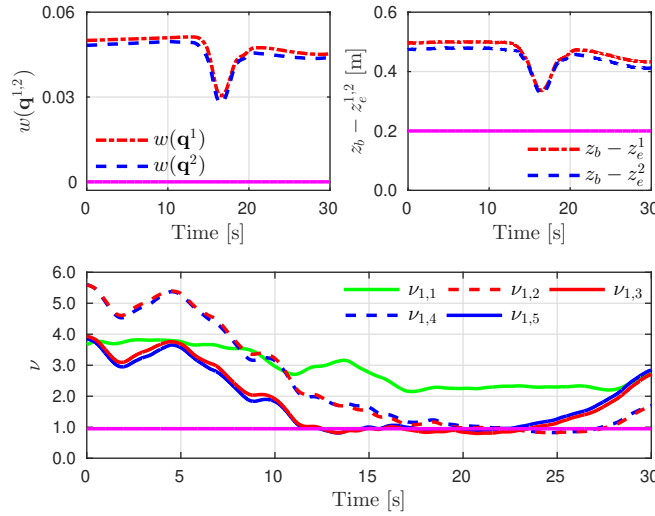


FIGURE 5.25: Constraint variables during the experimental flight tests in scenario 1: singularity-free motion (top left), safety distances between the UAV base and dual-arm end-effectors (top right), and value of ν related to the avoidance of obstacle (bottom). The corresponding bounds are represented as solid magenta lines.

5.5 Summary

In this chapter, two different trajectory generation with redundancy resolution strategies have been proposed for UAMs to take advantage of the entire system redundancy and simultaneously generate the feasible trajectories during aerial manipulation missions. Specifically, the first strategy is based on the weighted damped least-squares method, and its performance is evaluated with the single-arm aerial manipulator. The second approach employs the NMPC-based technique, and it is tested with the dual-arm aerial manipulator. Extensive simulation studies in ROS-Gazebo environment and real-time experimental flight tests have been performed to validate the proposed trajectory generation strategies. Both simulation and experimental results have shown that the proposed approaches are capable to exploit the system redundancy by modifying the UAM configuration during the execution of the assigned tasks while satisfying several constraints such as self-collision and obstacle avoidance.

Although the results demonstrate the efficacy and efficiency of the proposed methods, it is essential to provide a comparison between two approaches. There are several advantages of the NMPC-based method. Firstly, it allows to handle system constraints explicitly without defining them as additional tasks. Secondly,

a better long-term motion strategy can be obtained due to the fact that MPC has the ability to predict the future state evolution within the prediction horizon. Thirdly, due to the nature of MPC, the formulation of cost functions is not restricted to have the derivation of Jacobian matrices representing UAM tasks, while in the first strategy the derivation of these matrices is usually required. Finally, by defining the MPC problem in the acceleration domain, smooth velocity and position profiles can be obtained compared with the first strategy. However, the main drawback of the NMPC-based method is that this approach is known to be computationally demanding, while the first strategy does not have such restriction due to its structure. Therefore, this may hinder the real-time implementation of the second strategy with UAMs which do not have required onboard computational power. Nevertheless, advance onboard computers such as Intel NUC can be used to increase available computational power as we did in our flight tests, where the NMPC-based method has been implemented with a rate of 10 Hz.

Chapter 6

Conclusions

This thesis has focused on the novel aerial robot system composed of the UAV platform and one or more robotic arms. In particular, the performed research work has tried to extend the typical functionality of UAVs from passive observation and sensing to dynamic interaction with the object or surrounding environment. Therefore, three main contributions have been proposed, and they are summarized in Section 6.1, which also includes the limitations that shed light on potential future research directions discussed in Section 6.2.

6.1 Conclusions

In Chapter 3, we have presented the design, fabrication and experimental validation of the novel, lightweight, low inertia dual-arm manipulator specifically developed for UAM missions. The developed system consists of two arms with 6 DOFs each and weighs 2.5 kg in total with the maximum payload of 1.0 kg per arm. Unlike in the existing designs, the proposed dual-arm manipulator is endowed with the COG balancing mechanism and decoupled shoulders, which have notably improved the flight performance and operational capabilities of UAM. In particular, since the shoulder prismatic joints are introduced in the dual-arm design, each arm can adjust its configuration to compensate for the COG displacement, improving the aerial robot stability during the flight. In addition, the presence of the prismatic joints considerably increases the workspace and reach of the arms, while having decoupled shoulders enables to simultaneously execute several tasks, enhancing

the operational capability of the developed UAM. Extensive experimental studies have been conducted to validate the proposed dual-arm design with the hexarotor equipped with the off-the-shelf autopilot. The results have demonstrated that the influence of the arms motion over the hexarotor stability is minor in contactless flight due to the low weight and inertia features of the design. Furthermore, better hovering performance has been achieved by exploiting the ability of the proposed design to compensate its COG displacement. Nevertheless, it should be noted that the proposed COG balancing mechanism does not address the complete stabilization of UAM due to the hardware limitations of utilized DC motors and the mechanical limits, which can be reached depending on the desired configuration of the arms and grasped load. Therefore, in addition to the proposed COG balancing mechanism, the learning-based intelligent control approach has been developed in Chapter 4 to further improve the flight performance of UAMs.

In Chapter 4, the SMC theory-based intelligent T2FNN controller has been proposed for the control of UAMs under time-varying working conditions. The main goal has been to develop the control approach that can learn and compensate both internal and external disturbances caused by the robotic arm(s) motion and wind gust, and thus, leading to the wider implementation of aerial manipulation tasks in daily life. Due to the adaptive learning capabilities of the T2FNN structure, the developed control approach can be employed by both single and dual-arm aerial manipulators. Extensive simulation studies in ROS-Gazebo environment have been conducted to evaluate the efficiency of this learning-based framework by verifying its performance with the conventional controllers and with its type-1 counterpart in the absence and presence of wind gust conditions. Moreover, experimental studies have also been performed in the indoor environment to further validate the proposed control strategy. Both simulation and experimental results have demonstrated that the proposed controller, thanks to its learning capability, is capable to significantly reduce the steady state-error and overcome the disturbances caused by the arm motion and wind gust as compared to the conventional PD and PID controllers working alone, and thus leading to better flight behavior. On account of learning, the Euclidean RMSE improvements of around 56% and 52% have been observed for the simulation and real-time tests, respectively.

In Chapter 5, two different trajectory generation with redundancy resolution strategies have been proposed to take advantage of the entire system redundancy and

simultaneously generate the feasible trajectories during aerial manipulation missions. The first approach is based on the weighted damped least-squares method, which works at a velocity domain. Although this method is well-known for robotic arms, its implementation for aerial manipulation missions is novel. This approach is computationally inexpensive to run online, and it is flexible enough to incorporate different UAM tasks and constraints into its objective function. Specifically, the task relaxation framework and the maximization of the manipulability measure represented by the first-order approximation have been implemented along with other constraints to ensure the motion smoothness and improve flight behavior. Extensive simulation studies in ROS-Gazebo environment are conducted to evaluate the performance of this approach with the single-arm aerial manipulator. The results have shown that UAM can modify its configuration during the flight to satisfy imposed constraints. The second approach employs the NMPC-based technique, which is defined in the acceleration domain, resulting in smooth velocity and position profiles compared with the first approach. In addition, unlike the first method, it can explicitly handle the system constraints without defining them as additional tasks. Moreover, a better long-term motion strategy can be obtained with this approach since it has the ability to predict the future state evolution within the prediction horizon. Furthermore, due to the nature of the second approach, the formulation of cost functions is not constrained with the derivation of Jacobian matrices, which are usually required in the first approach. However, the major disadvantage of the NMPC-based method is that this approach is known to be computationally demanding. As a result, the real-time implementation of the second approach with UAMs, which do not have required onboard computational power, can be problematic to achieve. Nevertheless, with recent advances in technology, onboard computers like Intel NUC can be utilized to increase available computational power. In particular, the performance of the second approach has been illustrated via simulations and experimental flight tests where all computations are done online with Intel NUC onboard computer. Both simulation and experimental results have shown that the dual-arm aerial manipulator can perform non-trivial tasks while keeping its configuration within imposed bounds and avoiding collisions with obstacles.

6.1.1 Limitations

The experimental evaluations of the proposed contributions are conducted in a motion capture environment. This is a laboratory setting with the presence of an accurate localization. However, a practical test might require the system to work in an indoor and/or outdoor environment with an onboard localization system. These conditions bring additional noise and time-varying drift for the feedback. In this context, the validity of the proposed approaches might need to be reassessed under the absence of perfect positioning information.

In the proposed dual-arm design, the focus has been given to rigid extensions for the UAV body. However, in nature, most of the interactions are based on soft mechanisms. While the developed system might be damaged due to a sudden impact, a soft mechanism may compensate and absorb the additional forces for this case. As an intermediate solution, a compliant mechanism based on the elastic components (e.g., springs) can be integrated into the proposed system to avoid potential damages during physical interaction. This may require a thorough investigation to consider additional design and control challenges arising from the employment of the elastic elements.

Finally, one aspect of the proposed control strategy is that it is configured for the position control of aerial robots, where all disturbances are treated in a coupled manner. However, during the interaction with the environment, it could be beneficial to explicitly consider the interaction wrenches (i.e., forces and torques) to increase the robustness and flight performance of the aerial robot. In particular, adding the interaction force and torque information in the control loop can further extend the capability of the proposed control strategy and transform it into the hybrid force/position control of the aerial robot. Therefore, a thorough investigation of such an approach will be considered for future research.

6.2 Future Work

The aforementioned limitations shed light on potential future studies, which are discussed below:

- *Outdoor realization of the proposed contributions integrating onboard localization.* So far, the presented results have been based on a laboratory environment. However, onboard localization can validate the practical aspects of the proposed approaches. Therefore, integrated with the onboard localization system, the proposed methods can be used for various applications, including risky places where it is beyond the safety limitations of human workers.
- *Variable-stiffness compliant mechanism for aerial manipulation.* The elastic elements like springs can be utilized not only to obtain compliance behavior but also to perform explosive/fast motions by boosting the speed of the actuated joints. This can be achieved by controlling or varying the stiffness of the elastic element. In particular, this ability can notably increase the range of tasks and operational capabilities of UAMs, resulting in potentially advanced aerial manipulation missions like aerial repairing and hammering.
- *Collaborative aerial manipulation.* As mentioned in previous chapters, one of the main challenges in aerial manipulation is the limited payload of the UAV platform. Furthermore, some objects can be awkwardly shaped or not graspable close to their center of mass, and thus it is quite challenging to handle them only with one UAM. To address these issues, the collaborative work-task execution can be performed by employing several UAMs. Besides, the utilization of several UAMs instead of a single powerful one delivers additional advantages such as increased reliability and enhanced manipulation capability.
- *Inclusion of wrench information in the control loop.* The implementation of collaborative aerial manipulation comes with a set of issues such as reciprocal interaction caused by multiple UAMs. This creates additional disturbances to each UAM, making the control of aerial robots even more challenging. Hence, wrench/impedance-based control paradigms can be investigated to improve the interactive behavior between UAMs by including the knowledge of the interactive wrenches in the control loop. This will also significantly expand the operational capability of UAMs during the interaction with the environment, particularly for the low impedance cases with a fragile surface/sensor.
- *Exploiting the differential flatness and exact linearizability of the system.* The possibility to exploit the differential flatness properties of the aerial platform

for the control of UAM could be an exciting research direction. In particular, a differentially flat system is represented by the flat outputs that can be exactly linearizable via dynamic feedback. In this case, the flatness property of the system can be exploited to implement widely-used linear control approaches (e.g., dynamic feedback linearization) for aerial manipulation. This may further allow for the wider implementation of aerial manipulation tasks in daily life.

- *Whole-body control of UAMs by utilizing the dual-quaternions.* Another potential extension of the work is a whole-body motion control of the dual-arm aerial manipulators by utilizing the dual-quaternion representation that takes into consideration floating-base and underactuated features of the system. In particular, the cooperative dual task-space can be employed through dual-quaternions in order to exploit the full potential of bi-manual manipulation for aerial robots. In addition, by representing the dual pose, twist, and wrench in dual-quaternion space, a full description of two-arm coordination can be generated. This can allow performing advanced UAM applications that require complex two-arm coordination, such as opening a vane in an offshore oil industrial plant.

Appendix A

Derivation of the UAM model

In this Appendix, dynamic model of UAM is derived. In particular, two cases are considered. For the first case, the manual derivation of the dynamic model for the planar UAM (i.e., in 2D space) equipped with the single-joint arm is presented in [A.1](#). The derivation of the dynamic model for the planar case is considerably more concise compared to the full three-dimensional version. In addition, the study of the planar case can clearly show the coupling between the different parameters and their effect on each other. In contrast, the complete derivation of a dynamic model for UAM equipped with the multi-DOFs arm in 3D space is a cumbersome process, which results in intricate dynamics equations. Therefore, the manual derivation of the complete model could be error-prone and impractical for understanding the relationship between the different parameters. Thus, for the second case, the generic analysis of the dynamic model in 3D space is presented in [A.2](#).

A.1 Dynamic model of UAM in 2D

The derivation of the dynamic model for the planar UAM is covered in this section. The planar UAM consists of two dimensional UAV platform and single-joint arm as illustrated in Fig. [A.1](#). With reference to the system shown in Fig. [A.1](#), we have the world-fixed reference frame C_W , the body-fixed frame C_b with origin at the center of mass of the planar UAV. The position of the planar UAV with respect to C_W is described by the vector $\mathbf{r}_b = [x_b \ z_b]^T$, its attitude is denoted by the pitch angle θ , while the joint angle is defined as θ_1 . Hence, similarly as in [\[154\]](#), the

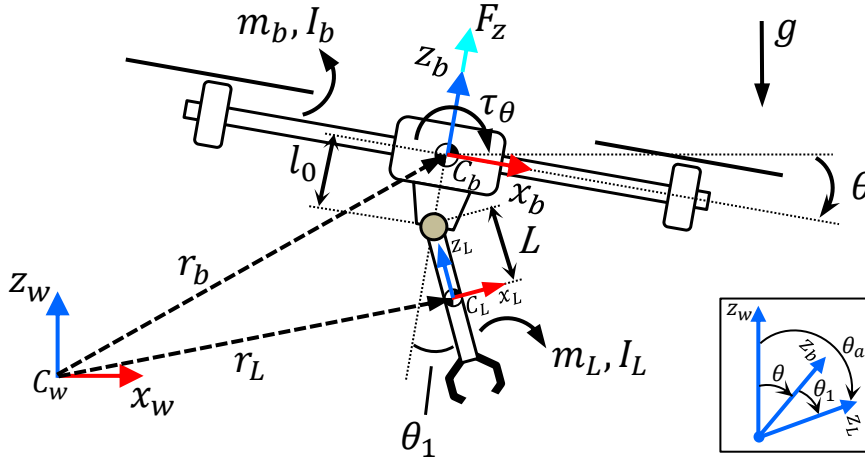


FIGURE A.1: Planar UAM system with relevant frames.

generalized coordinate variables can be expressed by the vector $\boldsymbol{\sigma} = [x_b \ z_b \ \theta \ \theta_1]^T$. The absolute joint angle of the arm is represented by $\theta_a = \theta + \theta_1$ [128]. It should be noted that in order to simplify the equations, it is assumed that the single-joint arm is directly attached at the center of mass of the planar UAV (i.e., $l_0 = 0$). Thus, the position of the center of mass of the arm link \mathbf{r}_L with respect to C_w can be defined as follows

$$\mathbf{r}_L = \mathbf{r}_b + \begin{bmatrix} -L \sin(\theta_a) \\ -L \cos(\theta_a) \end{bmatrix}. \quad (\text{A.1})$$

The velocity of the arm link are obtained by the differentiation of (A.1)

$$\dot{\mathbf{r}}_L = \dot{\mathbf{r}}_b + \begin{bmatrix} -L \cos(\theta_a) \dot{\theta}_a \\ L \sin(\theta_a) \dot{\theta}_a \end{bmatrix}. \quad (\text{A.2})$$

Then, the dynamics of the overall system can be derived by applying the well-known Euler-Lagrange formulation [145], in which the Lagrange equation can be expressed as follows

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{\sigma}}} - \frac{\partial \mathcal{L}}{\partial \boldsymbol{\sigma}} = \boldsymbol{\xi}, \quad (\text{A.3})$$

where $\boldsymbol{\xi}$ is the generalized input forces and $\mathcal{L} = \mathcal{K} - \mathcal{U}$ is the Lagrangian. The total kinetic energy \mathcal{K} and potential energy \mathcal{U} of the system are defined as [154]

$$\begin{cases} \mathcal{K} = \mathcal{K}_b + \mathcal{K}_L = \frac{1}{2} m_b \dot{\mathbf{r}}_b^T \dot{\mathbf{r}}_b + \frac{1}{2} I_b \dot{\theta}^2 + \frac{1}{2} m_L \dot{\mathbf{r}}_L^T \dot{\mathbf{r}}_L + \frac{1}{2} I_L \dot{\theta}_a^2, \\ \mathcal{U} = \mathcal{U}_b + \mathcal{U}_L = m_b g \mathbf{e}_2^T \mathbf{r}_b + m_L g \mathbf{e}_2^T \mathbf{r}_L, \end{cases} \quad (\text{A.4})$$

where $\mathbf{e}_2 = [0 \ 1]^T$, g is the gravity, m_b , I_b and m_L , I_L are the mass and moment of inertia of the planar UAV and the arm link, respectively. Then, similarly as in [154], the equation of motion of the planar UAM can be derived by evaluating (A.3) under the generalized coordinate vector $\boldsymbol{\sigma}$ in the following way

$$\begin{cases} \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{x}_b} - \frac{\partial \mathcal{L}}{\partial x_b} = F_z \sin(\theta), & \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\theta}} - \frac{\partial \mathcal{L}}{\partial \theta} = \tau_\theta, \\ \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{z}_b} - \frac{\partial \mathcal{L}}{\partial z_b} = F_z \cos(\theta), & \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\theta}_1} - \frac{\partial \mathcal{L}}{\partial \theta_1} = \tau_1, \end{cases} \quad (\text{A.5})$$

where F_z is the total thrust directed along the axis $\vec{\mathbf{z}}_b$, τ_θ is the pitch torque of the planar UAV and τ_1 is the joint torque. Then, the dynamics of the system can be rewritten in a compact form as follows

$$\mathbf{M}(\boldsymbol{\sigma})\ddot{\boldsymbol{\sigma}} + \mathbf{C}(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}})\dot{\boldsymbol{\sigma}} + \mathbf{G}(\boldsymbol{\sigma}) = \boldsymbol{\xi}, \quad (\text{A.6})$$

where $\mathbf{M}(\boldsymbol{\sigma}) \in \mathbb{R}^{4 \times 4}$ is the inertia matrix of the system, $\mathbf{G}(\boldsymbol{\sigma}) \in \mathbb{R}^4$ is the vector of gravity effects and $\mathbf{C}(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}}) \in \mathbb{R}^{4 \times 4}$ is the Coriolis/centrifugal matrix. The detailed expressions of the above matrices/vectors are defined as

$$\mathbf{M} = \begin{bmatrix} m_b + m_L & 0 & -m_L L \cos(\theta_a) & -m_L L \cos(\theta_a) \\ 0 & m_b + m_L & m_L L \sin(\theta_a) & m_L L \sin(\theta_a) \\ -m_L L \cos(\theta_a) & m_L L \sin(\theta_a) & I_b + I_L + m_L L^2 & I_L + m_L L^2 \\ -m_L L \cos(\theta_a) & m_L L \sin(\theta_a) & I_L + m_L L^2 & I_L + m_L L^2 \end{bmatrix}, \quad (\text{A.7})$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & m_L L \sin(\theta_a) \dot{\theta}_a & m_L L \sin(\theta_a) \dot{\theta}_a \\ 0 & 0 & m_L L \cos(\theta_a) \dot{\theta}_a & m_L L \cos(\theta_a) \dot{\theta}_a \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (\text{A.8})$$

$$\mathbf{G} = \begin{bmatrix} 0 & (m_b + m_L)g & m_L L g \sin(\theta_a) & m_L L g \sin(\theta_a) \end{bmatrix}^T, \quad (\text{A.9})$$

$$\boldsymbol{\xi} = \begin{bmatrix} F_z \sin(\theta) & F_z \cos(\theta) & \tau_\theta & \tau_1 \end{bmatrix}^T. \quad (\text{A.10})$$

From the above expressions, it can be concluded that the dynamic equations of the UAV-arm system are coupled and nonlinear. It should be noted that interested readers might refer to [128] for the derivation of the dynamic model in the case when $l_0 \neq 0$ and when the planar UAM is equipped with multi-joint arms.

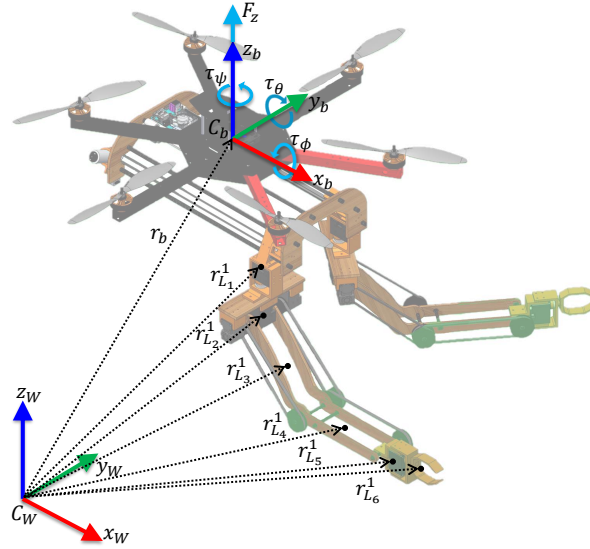


FIGURE A.2: Dual-arm aerial manipulator with relevant frames.

A.2 Dynamic model of UAM in 3D

The generic analysis of the dynamic model of UAM in 3D space is presented in this section. UAM consists of the hexarotor platform equipped with two 6-DOF robotic arms as shown in Fig. A.2. Similarly as in the previous section, the dynamics of UAM is derived by applying the Euler-Lagrange formulation with $\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\sigma}} - \frac{\partial \mathcal{L}}{\partial \sigma} = \xi$, where $\mathcal{L} = \mathcal{K} - \mathcal{U}$. The total kinetic energy \mathcal{K} and potential energy \mathcal{U} are given by

$$\begin{cases} \mathcal{K} = \mathcal{K}_b + \sum_{m=1}^2 \sum_{i=1}^6 \mathcal{K}_{L_i}^m, \\ \mathcal{U} = \mathcal{U}_b + \sum_{m=1}^2 \sum_{i=1}^6 \mathcal{U}_{L_i}^m, \end{cases} \quad (\text{A.11})$$

in which $m = \{1, 2\}$ corresponds to the right and left arms, respectively, \mathcal{K}_b and \mathcal{U}_b denote the kinetic and potential energy associated with the hexarotor, while $\mathcal{K}_{L_i}^m$ and $\mathcal{U}_{L_i}^m$ represent the kinetic and potential energy of each link of m -th arm. The aforementioned values are computed as follows [155]

$$\begin{cases} \mathcal{K}_b = \frac{1}{2} m_b (\dot{\mathbf{r}}_b)^T \dot{\mathbf{r}}_b + \frac{1}{2} (\dot{\Phi}_b)^T \mathbf{Q}^T \mathbf{I}_b \mathbf{Q} \dot{\Phi}_b, \\ \mathcal{U}_b = m_b g \mathbf{e}_3^T \mathbf{r}_b, \\ \mathcal{K}_{L_i}^m = \frac{1}{2} m_{L_i}^m (\dot{\mathbf{r}}_{L_i}^m)^T \dot{\mathbf{r}}_{L_i}^m + \frac{1}{2} (\boldsymbol{\omega}_{L_i}^m)^T (\mathbf{R}_b \mathbf{R}_{L_i}^{b,m}) \mathbf{I}_{L_i}^m (\mathbf{R}_b \mathbf{R}_{L_i}^{b,m})^T \boldsymbol{\omega}_{L_i}^m, \\ \mathcal{U}_{L_i}^m = m_{L_i}^m g \mathbf{e}_3^T (\mathbf{r}_b + \mathbf{R}_b \mathbf{r}_{L_i}^{b,m}), \end{cases} \quad (\text{A.12})$$

where $\mathbf{r}_b = [x_b \ y_b \ z_b]^T \in \mathbb{R}^3$ is the position of the hexarotor with respect to C_W , its attitude is denoted by the Euler angles $\Phi_b = [\phi \ \theta \ \psi]^T \in \mathbb{R}^3$, \mathbf{R}_b is the rotation matrix describing the orientation of C_b relative to C_W , \mathbf{Q} is the mapping matrix that relates the Euler angle rates $\dot{\Phi}_b$ with the hexarotor angular velocity described in the body-fixed frame C_b . The vectors $\dot{\mathbf{r}}_{L_i}^m$ and $\boldsymbol{\omega}_{L_i}^m$ represent the linear and angular velocity of the link i of m -th arm with respect to C_W , respectively, while $\mathbf{R}_{L_i}^{b,m}$ is the rotation matrix between the frame associated to the link i of m -th arm and C_b . As for m_* and \mathbf{I}_* , they represent the corresponding mass and moment of inertia, respectively, g is the gravity and $\mathbf{e}_3 = [0 \ 0 \ 1]^T$, while $\mathbf{r}_{L_i}^{b,m}$ is the position of the center of mass of the link i of m -th arm described in C_b . Finally, the generalized coordinate variables are expressed by the vector $\boldsymbol{\sigma} = [\mathbf{r}_b^T \ \Phi_b^T \ \mathbf{q}^{1,T} \ \mathbf{q}^{2,T}]^T \in \mathbb{R}^{18}$, in which $\mathbf{q}^m = [q_1^m, \dots, q_6^m]^T \in \mathbb{R}^6$ is the joint position vector of the dual-arm. Then, the dynamics of the dual-arm aerial manipulator can be defined as follows

$$\mathbf{M}(\boldsymbol{\sigma})\ddot{\boldsymbol{\sigma}} + \mathbf{C}(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}})\dot{\boldsymbol{\sigma}} + \mathbf{G}(\boldsymbol{\sigma}) = \boldsymbol{\xi}, \quad (\text{A.13})$$

where $\mathbf{M}(\boldsymbol{\sigma}) \in \mathbb{R}^{18 \times 18}$ is the inertia matrix, $\mathbf{G}(\boldsymbol{\sigma}) \in \mathbb{R}^{18}$ is the gravity component and $\mathbf{C}(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}}) \in \mathbb{R}^{18 \times 18}$ is the Coriolis/centrifugal matrix. The generalized input vector $\boldsymbol{\xi}$ can be defined as follows [155]

$$\boldsymbol{\xi} = \begin{bmatrix} \mathbf{R}_b \mathbf{f}_b^b \\ \mathbf{Q}^T \boldsymbol{\tau}_b^b \\ \boldsymbol{\tau}^1 \\ \boldsymbol{\tau}^2 \end{bmatrix}, \quad (\text{A.14})$$

where $\boldsymbol{\tau}^m \in \mathbb{R}^6$ is the joint torques vector of the robotic arms, $\boldsymbol{\tau}_b^b$ and \mathbf{f}_b^b are the torques and forces applied on the hexarotor, described in C_b , and they are defined as in [156]

$$\mathbf{f}_b^b = \begin{bmatrix} 0 \\ 0 \\ F_z \end{bmatrix}, \quad \boldsymbol{\tau}_b^b = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}, \quad (\text{A.15})$$

in which F_z is the total thrust directed along the axis $\bar{\mathbf{z}}_b$, whereas τ_ψ , τ_θ and τ_ϕ are the three rotational torques acting around $\bar{\mathbf{z}}_b$, $\bar{\mathbf{y}}_b$ and $\bar{\mathbf{x}}_b$ axes, respectively.

Video Links

- Experimental flight tests in Chapter 3: <https://youtu.be/1cUYWqkMCIM>
- Experimental flight tests in Chapter 4: <https://youtu.be/qTdNbm0Xy44>
- Experimental flight tests in Chapter 5: <https://youtu.be/8n-Vz4ECr2U>

List of Publications

Journal Papers

- **N. Imanberdiyev**, S. Sood, D. Kircali, and E. Kayacan, “Design, Development and Experimental Validation of a Lightweight Dual-arm Aerial Manipulator with a COG Balancing Mechanism”, *Mechatronics* (submitted). [Video](#).
- **N. Imanberdiyev** and E. Kayacan, “A Fast Learning Control Strategy for Unmanned Aerial Manipulators”, *Journal of Intelligent & Robotic Systems*, 94 (2019), no. 3, pp. 805-824. [Video](#). DOI: [10.1007/s10846-018-0884-7](https://doi.org/10.1007/s10846-018-0884-7).
- A. Sarabakha, **N. Imanberdiyev**, E. Kayacan, M. A. Khanesar, and H. Hagaras, “Novel Levenberg-Marquardt Based Learning Algorithm for Unmanned Aerial Vehicles”, *Information Sciences*, 417 (2017), pp. 361-380. [Video](#). DOI: [10.1016/j.ins.2017.07.020](https://doi.org/10.1016/j.ins.2017.07.020).

Conference Papers

- **N. Imanberdiyev** and E. Kayacan, “Redundancy Resolution based Trajectory Generation for Dual-Arm Aerial Manipulators via Online Model Predictive Control”, *IECON 2020 - 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020 (Accepted). [Video](#).
- **N. Imanberdiyev**, J. Monica, and E. Kayacan, “A Multi-Task Velocity-based Redundancy Resolution Strategy for Unmanned Aerial Manipulators”, *2019 18th European Control Conference (ECC)*, 2019, pp. 1130-1135. DOI: [10.23919/ECC.2019.8796272](https://doi.org/10.23919/ECC.2019.8796272)
- **N. Imanberdiyev**, C. Fu, E. Kayacan, and I. Chen, “Autonomous Navigation of UAV by Using Real-Time Model-Based Reinforcement Learning”, *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 812-817, 2016. [Video](#). DOI: [10.1109/ICARCV.2016.7838739](https://doi.org/10.1109/ICARCV.2016.7838739).
- B. Cowan, **N. Imanberdiyev**, C. Fu, Y. Dong, and E. Kayacan, “A Performance Evaluation of Detectors and Descriptors for UAV Visual Tracking”, *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 292-297, 2016. DOI: [10.1109/ICARCV.2016.7838649](https://doi.org/10.1109/ICARCV.2016.7838649).

Bibliography

- [1] Juntong Qi, Dalei Song, Hong Shang, Nianfa Wang, Chunsheng Hua, Chong Wu, Xin Qi, and Jianda Han. Search and rescue rotary-wing uav and its application to the lushan ms 7.0 earthquake. *Journal of Field Robotics*, 2015. 1
- [2] J. Capitan, L. Merino, and A. Ollero. Cooperative decision-making under uncertainties for multi-target surveillance with multiples uavs. *Journal of Intelligent & Robotic Systems*, 84(1):371–386, Dec 2016. doi: 10.1007/s10846-015-0269-0. URL <https://doi.org/10.1007/s10846-015-0269-0>. 1
- [3] Jinhong Chen, Haoting Liu, Jingchen Zheng, Ming Lv, Beibei Yan, Xin Hu, and Yun Gao. Damage degree evaluation of earthquake area using uav aerial image. *International Journal of Aerospace Engineering*, 2016, 2016. 1
- [4] Matko Orsag, Christopher Korpela, Paul Oh, Stjepan Bogdan, and Anibal Ollero. *Aerial Manipulation*. Springer, 2018. 1
- [5] F. Ruggiero, V. Lippiello, and A. Ollero. Aerial manipulation: A literature review. *IEEE Robotics and Automation Letters*, 3(3):1957–1964, 2018. 1, 16, 17
- [6] N. Staub, D. Bicego, Q. Sablé, V. Arellano, S. Mishra, and A. Franchi. Towards a flying assistant paradigm: the othex. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6997–7002, 2018. 1
- [7] A. E. Jimenez-Cano, J. Braga, G. Heredia, and A. Ollero. Aerial manipulator for structure inspection by contact from the underside. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1879–1884, 2015. 1
- [8] M. Tognon, H. A. T. Chavez, E. Gasparin, Q. Sable, D. Bicego, A. Mallet, M. Lany, G. Santi, B. Revaz, J. Cortes, and A. Franchi. A truly-redundant aerial manipulator system with application to push-and-slide inspection in industrial plants. *IEEE Robotics and Automation Letters*, 4(2):1846–1851, 2019. 1
- [9] ARCAS project home page. <http://www.arcas-project.eu/>. Accessed: 2019-12-28. 1, 10

- [10] A. Ollero, G. Heredia, A. Franchi, G. Antonelli, K. Kondak, A. Sanfeliu, A. Viguria, J. R. Martinez-de Dios, F. Pierri, J. Cortes, A. Santamaria-Navarro, M. A. Trujillo Soto, R. Balachandran, J. Andrade-Cetto, and A. Rodriguez. The aeroarms project: Aerial robots with advanced manipulation capabilities for inspection and maintenance. *IEEE Robotics Automation Magazine*, 25(4):12–23, 2018. 10, 11
- [11] AEROWORKS project: Collaborative aerial workers. <http://www.aeroworks2020.eu/presentations>. Accessed: 2019-12-27. 1, 3, 10
- [12] New Atlas. Pars aerial robot delivers a payload of life preservers to drowning victims. <https://newatlas.com/pars-aerial-robot/26878/>, 2013. [Online; accessed 2019-12-28]. 2
- [13] DesignMind. These drones will save your life. <http://designmind.frogdesign.com/2014/03/drones-will-save-life/>, 2014. [Online; accessed 2019-12-28]. 2
- [14] Drone Nodes. Firefighter drones - how drones are being used for helping fire departments. <http://dronenodes.com/firefighter-drones/>. [Online; accessed 2019-12-29]. 2
- [15] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar. Grasping from the air: Hovering capture and load stability. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2491–2498, 2011. 2, 12, 13, 19
- [16] R Cano, C Pérez, F Pruano, A Ollero, and G Heredia. Mechanical design of a 6-dof aerial manipulator for assembling bar structures using uavs. In *2nd RED-UAS 2013 workshop on research, education and development of unmanned aerial systems*, 2013. 2, 14, 32, 34, 35
- [17] C. D. Bellicoso, L. R. Buonocore, V. Lippiello, and B. Siciliano. Design, modeling and control of a 5-dof light-weight robot arm for aerial manipulation. In *2015 23rd Mediterranean Conference on Control and Automation (MED)*, pages 853–858, 2015. 14
- [18] D. Wuthier, D. Kominiak, C. Kanellakis, G. Andrikopoulos, M. Fumagalli, G. Schipper, and G. Nikolakopoulos. On the design, modeling and control of a novel compact aerial manipulator. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 665–670, 2016. 14
- [19] T. W. Danko, K. P. Chaney, and P. Y. Oh. A parallel manipulator for mobile manipulating uavs. In *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pages 1–6, 2015. 14
- [20] M. Kamel, K. Alexis, and R. Siegwart. Design and modeling of dexterous aerial manipulator. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4870–4876, 2016. 2, 14

- [21] ProDrone home page. <https://www.prodrone.com/products/pd6b-aw-arm/>. Accessed: 2019-12-01. 2, 15, 30
- [22] Alejandro Suarez, Antonio Enrique Jimenez-Cano, Victor Manuel Vega, Guillermo Heredia, Angel Rodriguez-Castano, and Anibal Ollero. Design of a lightweight dual arm system for aerial manipulation. *Mechatronics*, 50: 30 – 44, 2018. 15, 30, 33
- [23] A. Suarez, G. Heredia, and A. Ollero. Design of an anthropomorphic, compliant, and lightweight dual arm for aerial manipulation. *IEEE Access*, 6: 29173–29189, 2018. 2, 15, 30, 32
- [24] F. Ruggiero, M. A. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Perez, V. Lippiello, A. Ollero, and B. Siciliano. A multilayer control for multirotor uavs equipped with a servo robot arm. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4014–4020, 2015. xxi, 2, 14, 15, 35, 37
- [25] Kimon P. Valavanis and George J. Vachtsevanos. *UAV Control: Introduction*, pages 527–528. Springer Netherlands, Dordrecht, 2015. ISBN 978-90-481-9707-1. doi: 10.1007/978-90-481-9707-1_137. URL https://doi.org/10.1007/978-90-481-9707-1_137. 3, 17
- [26] Farid Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378, 2012. 3, 24
- [27] Asli Celiyilmaz and I. Burhan Turksen. *Modeling Uncertainty with Fuzzy Logic: With Recent Theory and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2009. ISBN 3540899235, 9783540899235. 4, 21
- [28] Jerry Mendel, Hani Hagaras, Woei-Wan Tan, William W Melek, and Hao Ying. *Introduction to type-2 fuzzy logic control: theory and applications*. John Wiley & Sons, 2014. 4, 21
- [29] Erdal Kayacan and Mojtaba Ahmadi Khamesar. *Fuzzy neural networks for real time control applications: concepts, modeling and algorithms for fast learning*. Butterworth-Heinemann, 2015. 4, 21, 71, 73, 74
- [30] Roberto Rossi, Angel Santamaria-Navarro, Juan Andrade-Cetto, and Paolo Rocco. Trajectory generation for unmanned aerial manipulators through quadratic programming. *IEEE Robotics and Automation Letters*, 2(2):389–396, 2017. 4, 24
- [31] L. Marconi, F. Basile, G. Caprari, R. Carloni, P. Chiacchio, C. Hurzeler, V. Lippiello, R. Naldi, J. Nikolic, B. Siciliano, S. Stramigioli, and E. Zwicker. Aerial service robotics: The airobots perspective. In *2012 2nd International Conference on Applied Robotics for the Power Industry (CARPI)*, pages 64–69, 2012. 10

- [32] AERial RObotic system for in-depth Bridge Inspection (AEROBI) by contact. <http://www.aerobi.eu/>. Accessed: 2019-12-29. 10
- [33] S. Bellens, J. De Schutter, and H. Bruyninckx. A hybrid pose / wrench control framework for quadrotor helicopters. In *2012 IEEE International Conference on Robotics and Automation*, pages 2269–2274, 2012. 10, 11
- [34] L. Marconi and R. Naldi. Control of aerial robots: Hybrid force and position feedback for a ducted fan. *IEEE Control Systems*, 32(4):43–65, 2012. 11
- [35] G. Darivianakis, K. Alexis, M. Burri, and R. Siegwart. Hybrid predictive control for aerial robotic physical interaction towards inspection operations. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 53–58, 2014. 11
- [36] Kostas Alexis, Georgios Darivianakis, Michael Burri, and Roland Siegwart. Aerial robotic contact-based inspection: planning and control. *Autonomous Robots*, 40(4):631–655, Apr 2016. 11
- [37] A. Q. L. Keemink, M. Fumagalli, S. Stramigioli, and R. Carloni. Mechanical design of a manipulation system for unmanned aerial vehicles. In *2012 IEEE International Conference on Robotics and Automation*, pages 3147–3152, 2012. 11
- [38] M. Fumagalli, R. Naldi, A. Macchelli, F. Forte, A. Q. L. Keemink, S. Stramigioli, R. Carloni, and L. Marconi. Developing an aerial manipulator prototype: Physical interaction with the environment. *IEEE Robotics Automation Magazine*, 21(3):41–50, 2014. 11
- [39] P. Cruz and R. Fierro. Autonomous lift of a cable-suspended load by an unmanned aerial robot. In *2014 IEEE Conference on Control Applications (CCA)*, pages 802–807, 2014. 12
- [40] Patricio J. Cruz and Rafael Fierro. Cable-suspended load lifting by a quadrotor uav: hybrid model, trajectory generation, and control. *Autonomous Robots*, 41(8):1629–1643, Dec 2017. 12
- [41] I. Palunko, R. Fierro, and P. Cruz. Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2691–2697, 2012. 12
- [42] Philipp Foehn, Davide Falanga, Naveen Kuppaswamy, Russ Tedrake, and Davide Scaramuzza. Fast trajectory optimization for agile quadrotor maneuvers with a cable-suspended payload. In *Robotics: Science and Systems*, pages 1–10, 2017. 12
- [43] Morten Bisgaard, Anders la Cour-Harbo, and Jan Dimon Bendtsen. Adaptive control system for autonomous helicopter slung load operations. *Control Engineering Practice*, 18(7):800–811, 2010. 12

- [44] I. Palunko, P. Cruz, and R. Fierro. Agile load transportation : Safe and efficient load manipulation with aerial robots. *IEEE Robotics Automation Magazine*, 19(3):69–79, 2012. [12](#)
- [45] Markus Bernard, Konstantin Kondak, Ivan Maza, and Anibal Ollero. Autonomous transportation and deployment with aerial robots for search and rescue missions. *Journal of Field Robotics*, 28(6):914–931, 2011. doi: 10.1002/rob.20401. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20401>. [12](#)
- [46] Nathan Michael, Jonathan Fink, and Vijay Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, 2011.
- [47] Koushil Sreenath and Vijay Kumar. Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots. In *Proceedings of Robotics: Science and Systems*, 2013.
- [48] F. A. Goodarzi and T. Lee. Dynamics and control of quadrotor uavs transporting a rigid body connected via flexible cables. In *2015 American Control Conference (ACC)*, pages 4677–4682, 2015. [12](#), [19](#)
- [49] V. Ghadiok, J. Goldin, and W. Ren. Autonomous indoor aerial gripping using a quadrotor. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4645–4651, 2011. [12](#)
- [50] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2668–2673, 2011. [13](#)
- [51] Justin Thomas, Joe Polin, Koushil Sreenath, and Vijay Kumar. Avian-inspired grasping for quadrotor micro uavs. In *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2013. [12](#)
- [52] Paul E. I. Pounds, Daniel R. Bersak, and Aaron M. Dollar. Stability of small-scale uav helicopters and quadrotors with added payload mass under pid control. *Autonomous Robots*, 33(1):129–142, 2012. [13](#), [19](#)
- [53] Daniel Mellinger, Michael Shomin, Nathan Michael, and Vijay Kumar. *Co-operative Grasping and Transport Using Multiple Quadrotors*, pages 545–558. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-32723-0. doi: 10.1007/978-3-642-32723-0_39. URL https://doi.org/10.1007/978-3-642-32723-0_39. [13](#)
- [54] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. Construction with quadrotor teams. *Autonomous Robots*, 33(3):323–336, 2012. [13](#), [19](#)

- [55] Christopher M. Korpela, Todd W. Danko, and Paul Y. Oh. Mm-uav: Mobile manipulating unmanned aerial vehicle. *Journal of Intelligent & Robotic Systems*, 65(1):93–101, 2012. 14
- [56] G. Heredia, A. E. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J. A. Acosta, and A. Ollero. Control of a multicopter outdoor aerial manipulator. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3417–3422, 2014. xxi, 14, 15
- [57] F. Huber, K. Kondak, K. Krieger, D. Sommer, M. Schwarzbach, M. Laiacker, I. Kossyk, S. Parusel, S. Haddadin, and A. Albu-Schaffer. First analysis and experiments in aerial manipulation using fully actuated redundant robot arm. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3452–3457, 2013.
- [58] K. Kondak, F. Huber, M. Schwarzbach, M. Laiacker, D. Sommer, M. Bejar, and A. Ollero. Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2107–2112, 2014. 14
- [59] E. E. Komendera, S. Adhikari, S. Glassner, A. Kishen, and A. Quartaro. Structure assembly by a heterogeneous team of robots using state estimation, generalized joints, and mobile parallel manipulators. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4672–4679, 2017. 14
- [60] M. Orsag, C. Korpela, S. Bogdan, and P. Oh. Valve turning using a dual-arm aerial manipulator. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 836–841, 2014. 15, 30
- [61] A. Suarez, A. M. Giordano, K. Kondak, G. Heredia, and A. Ollero. Flexible link long reach manipulator with lightweight dual arm: Soft-collision detection, reaction, and obstacle localization. In *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 406–411, 2018. 15
- [62] HYbrid FLying-rollIng with-snakeE-aRm robot for contact inSpection (HYFLIERS) project. <https://www.oulu.fi/hyfliers/>. Accessed: 2019-12-30. 16
- [63] Raymond Oung and Raffaello D’Andrea. The distributed flight array: Design, implementation, and analysis of a modular vertical take-off and landing vehicle. *The International Journal of Robotics Research*, 33(3):375–400, 2014. doi: 10.1177/0278364913501212. 16
- [64] M. Zhao, K. Kawasaki, X. Chen, S. Noda, K. Okada, and M. Inaba. Whole-body aerial manipulation by transformable multicopter with two-dimensional multilinks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5175–5182, 2017. 16

- [65] Xilun DING, Pin GUO, Kun XU, and Yushu YU. A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems. *Chinese Journal of Aeronautics*, 32(1):200 – 214, 2019. ISSN 1000-9361. 16
- [66] Xiangdong Meng, Yuqing He, and Jianda Han. Survey on aerial manipulator: System, modeling, and control. *Robotica*, pages 1–30. doi: 10.1017/S0263574719001450.
- [67] Abdullah Mohiuddin, Taha Tarek, Yahya Zweiri, and Dongming Gan. A survey of single and multi-uav aerial manipulation. *Unmanned Systems*, 08(02):119–147, 2020. doi: 10.1142/S2301385020500089. URL <https://doi.org/10.1142/S2301385020500089>. 16
- [68] Manohar Srikanth, Albert Soto, Anuradha Annaswamy, Eugene Lavretsky, and Jean-Jacques Slotine. Controlled manipulation with multiple quadrotors. In *AIAA Guidance, Navigation, and Control Conference*, page 6547, 2011. 19
- [69] M. Orsag, C. Korpela, M. Pekala, and P. Oh. Stability control in aerial manipulation. In *2013 American Control Conference*, pages 5581–5586, 2013. 19
- [70] C. Papachristos and A. Tzes. Large object pushing via a direct longitudinally-actuated unmanned tri-tiltrotor. In *21st Mediterranean Conference on Control and Automation*, pages 173–178, 2013. 19
- [71] Y. Stergiopoulos, E. Kontouras, K. Gkountas, K. Giannousakis, and A. Tzes. Modeling and control aspects of a uav with an attached manipulator. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 653–658, 2016. 19
- [72] P. Castillo, R. Lozano, and A. Dzul. Stabilization of a mini rotorcraft with four rotors. *IEEE Control Systems*, 25(6):45–55, 2005. 18
- [73] T. Taniguchi, L. Eciolaza, and M. Sugeno. Quadrotor control using dynamic feedback linearization based on piecewise bilinear models. In *2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, pages 1–7, 2014. 18
- [74] Daewon Lee, H. Jin Kim, and Shankar Sastry. Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter. *International Journal of Control, Automation and Systems*, 7(3):419–428, 2009. 19
- [75] R. Xu and U. Ozguner. Sliding mode control of a quadrotor helicopter. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 4957–4962, 2006. 19
- [76] S. Bouabdallah and R. Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2247–2252, 2005. 19

- [77] En-Hui Zheng, Jing-Jing Xiong, and Ji-Liang Luo. Second order sliding mode control for a quadrotor uav. *ISA transactions*, 53(4):1350–1356, 2014. [19](#)
- [78] M. Orsag, C. Korpela, S. Bogdan, and P. Oh. Lyapunov based model reference adaptive control for aerial manipulation. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 966–973, 2013. [20](#)
- [79] S. Kim, S. Choi, and H. J. Kim. Aerial manipulation using a quadrotor with a two dof robotic arm. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4990–4995, 2013. [19](#), [20](#)
- [80] H. Lee and H. J. Kim. Estimation, control, and planning for autonomous aerial transportation. *IEEE Transactions on Industrial Electronics*, 64(4):3369–3379, 2017. [19](#), [20](#)
- [81] Vicente Parra-Vega, Anand Sanchez, Carlos Izaguirre, Octavio Garcia, and Francisco Ruiz-Sanchez. Toward aerial grasping and manipulation with multiple uavs. *Journal of Intelligent & Robotic Systems*, 70(1):575–593, Apr 2013. [19](#), [20](#)
- [82] M. E. Guerrero, D. A. Mercado, R. Lozano, and C. D. Garcia. Passivity based control for a quadrotor uav transporting a cable-suspended payload with minimum swing. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 6718–6723, 2015. [20](#)
- [83] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi. Modeling and control of a flying robot for contact inspection. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3532–3537, 2012. [20](#)
- [84] G. Garimella and M. Kobilarov. Towards model-predictive control for aerial pick-and-place. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4692–4697, 2015. [19](#), [20](#), [25](#)
- [85] Christos Papachristos, Kostas Alexis, and Anthony Tzes. Technical activities execution with a tiltrotor uas employing explicit model predictive control. *IFAC Proceedings Volumes*, 47(3):11036–11042, 2014. [19](#), [20](#)
- [86] Jasper LJ Scholten, Matteo Fumagalli, Stefano Stramigioli, and Raffaella Carloni. Interaction control of an uav endowed with a manipulator. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4910–4915. IEEE, 2013. [20](#)
- [87] V. Lippiello and F. Ruggiero. Exploiting redundancy in cartesian impedance control of uavs equipped with a robotic arm. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3768–3773, 2012. [20](#)
- [88] G. Gioioso, A. Franchi, G. Salvietti, S. Scheggi, and D. Prattichizzo. The flying hand: A formation of uavs for cooperative aerial tele-manipulation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4335–4341, 2014. [20](#)

- [89] A. Y. Mersha, S. Stramigioli, and R. Carloni. Variable impedance control for aerial interaction. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3435–3440, 2014. [20](#)
- [90] C. Ott, R. Mukherjee, and Y. Nakamura. Unified impedance and admittance control. In *2010 IEEE International Conference on Robotics and Automation*, pages 554–561, 2010. [20](#)
- [91] Oscar Castillo, Leticia Amador-Angulo, Juan R. Castro, and Mario Garcia-Valdez. A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems. *Information Sciences*, 354:257 – 274, 2016. ISSN 0020-0255. doi: <https://doi.org/10.1016/j.ins.2016.03.026>. [21](#)
- [92] Leticia Cervantes and Oscar Castillo. Type-2 fuzzy logic aggregation of multiple fuzzy controllers for airplane flight control. *Information Sciences*, 324: 247 – 256, 2015. ISSN 0020-0255. [21](#)
- [93] H. A. Hagras. A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Transactions on Fuzzy Systems*, 12(4):524–539, 2004. [21](#)
- [94] A. Melendez and O. Castillo. Optimization of type-2 fuzzy reactive controllers for an autonomous mobile robot. In *2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pages 207–211, 2012. Mexico City, Mexico, November 05-09, 2012.
- [95] Mauricio A. Sanchez, Oscar Castillo, and Juan R. Castro. Generalized Type-2 Fuzzy Systems for controlling a mobile robot and a performance comparison with Interval Type-2 and Type-1 Fuzzy Systems. *Expert Systems with Applications*, 42(14):5904 – 5914, 2015. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2015.03.024>.
- [96] Oscar Castillo, Leticia Cervantes, Jose Soria, Mauricio Sanchez, and Juan R. Castro. A generalized type-2 fuzzy granular approach with applications to aerospace. *Information Sciences*, 354:165 – 177, 2016. ISSN 0020-0255. doi: <https://doi.org/10.1016/j.ins.2016.03.001>. [21](#)
- [97] C. Fu, A. Sarabakha, E. Kayacan, C. Wagner, R. John, and J. M. Garibaldi. A comparative study on the control of quadcopter uavs by using singleton and non-singleton fuzzy logic controllers. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1023–1030, 2016. [21](#)
- [98] F. Fakurian, M. B. Menhaj, and A. Mohammadi. Design of a fuzzy controller by minimum controlling inputs for a quadrotor. In *Robotics and Mechatronics (ICRoM), 2014 Second RSI/ISM International Conference on*, pages 619–624, Oct 2014. doi: [10.1109/ICRoM.2014.6990971](https://doi.org/10.1109/ICRoM.2014.6990971). [21](#)

- [99] T. Dierks and S. Jagannathan. Output feedback control of a quadrotor uav using neural networks. *IEEE Transactions on Neural Networks*, 21(1):50–66, 2010. [21](#)
- [100] Fernando Gaxiola, Patricia Melin, Fevrier Valdez, and Oscar Castillo. Generalized type-2 fuzzy weight adjustment for backpropagation neural networks in time series prediction. *Information Sciences*, 325:159 – 174, 2015. ISSN 0020-0255. [22](#)
- [101] Andriy Sarabakha, Nursultan Imanberdiyev, Erdal Kayacan, Mojtaba Ahmadih Khanesar, and Hani Hagraas. Novel Levenberg-Marquardt based learning algorithm for unmanned aerial vehicles. *Information Sciences*, 417: 361 – 380, 2017. [22](#)
- [102] N. Wang, M. J. Er, and M. Han. Dynamic tanker steering control using generalized ellipsoidal-basis-function-based fuzzy neural networks. *IEEE Transactions on Fuzzy Systems*, 23(5):1414–1427, 2015. [22](#)
- [103] Xiucheng Dong, Yunyuan Zhao, Hamid Reza Karimi, and Peng Shi. Adaptive variable structure fuzzy neural identification and control for a class of mimo nonlinear system. *Journal of the Franklin Institute*, 350(5):1221 – 1247, 2013. [22](#)
- [104] Jafar Tavoosi, Amir Abolfazl Suratgar, and Mohammad Bagher Menhaj. Stable anfis2 for nonlinear system identification. *Neurocomputing*, 182:235 – 246, 2016. [22](#)
- [105] C. Kim and D. Chwa. Obstacle avoidance method for wheeled mobile robots using interval type-2 fuzzy neural network. *IEEE Transactions on Fuzzy Systems*, 23(3):677–687, 2015. [22](#)
- [106] Erkan Kayacan, Joshua M. Peschel, and Girish Chowdhary. A self-learning disturbance observer for nonlinear systems in feedback-error learning scheme. *Engineering Applications of Artificial Intelligence*, 62:276–285, 2017.
- [107] Erdal Kayacan and Reinaldo Maslim. Type-2 fuzzy logic trajectory tracking control of quadrotor vtol aircraft with elliptic membership functions. *IEEE/ASME Transactions on Mechatronics*, 22(1):339–348, 2017. [22](#)
- [108] J. M. Mendel and R. I. B. John. Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*, 10(2):117–127, 2002. [22](#)
- [109] Kevin Tai, Abdul-Rahman El-Sayed, Mohammad Biglarbegian, Claudia I. Gonzalez, Oscar Castillo, and Shohel Mahmud. Review of recent type-2 fuzzy controller applications. *Algorithms*, 9(2), 2016. ISSN 1999-4893. URL <https://www.mdpi.com/1999-4893/9/2/39>. [22](#)
- [110] J. M. Mendel. Computing derivatives in interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, 12(1):84–98, 2004. [22](#)

- [111] Oscar Castillo and Patricia Melin. *Overview of Genetic Algorithms Applied in the Optimization of Type-2 Fuzzy Systems*, pages 19–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. doi: 10.1007/978-3-642-28956-9_4. URL https://doi.org/10.1007/978-3-642-28956-9_4. 22
- [112] M. A. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak. Levenberg marquardt algorithm for the training of type-2 fuzzy neuro systems with a novel type-2 fuzzy membership function. In *2011 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*, pages 88–93, 2011. Paris, France, April 11-15, 2011. 22, 70
- [113] Ting Huang, Hossein Javaherian, and Derong Liu. Nonlinear torque and air-to-fuel ratio control of spark ignition engines using neuro-sliding mode techniques. *International journal of neural systems*, 21(03):213–224, 2011. 22
- [114] F. J. Lin, Y. C. Hung, and K. C. Ruan. An intelligent second-order sliding-mode control for an electric power steering system using a wavelet fuzzy neural network. *IEEE Transactions on Fuzzy Systems*, 22(6):1598–1611, 2014. 22
- [115] E. Kayacan, E. Kayacan, H. Ramon, and W. Saeys. Adaptive neuro-fuzzy control of a spherical rolling robot using sliding-mode-control-theory-based online learning algorithm. *IEEE Transactions on Cybernetics*, 43(1):170–179, 2013. 22
- [116] R. J. Wai and R. Muthusamy. Fuzzy-neural-network inherited sliding-mode control for robot manipulator including actuator dynamics. *IEEE Transactions on Neural Networks and Learning Systems*, 24(2):274–287, 2013. 22
- [117] Erdal Kayacan, Erkan Kayacan, Herman Ramon, and Wouter Saeys. Neuro-fuzzy control with a novel training method based-on sliding mode control theory: Application to tractor dynamics. *IFAC Proceedings Volumes*, 45(22):889 – 894, 2012. 10th IFAC Symposium on Robot Control. 22
- [118] E. Kayacan, W. Saeys, E. Kayacan, H. Ramon, and O. Kaynak. Intelligent control of a tractor-implement system using type-2 fuzzy neural networks. In *2012 IEEE International Conference on Fuzzy Systems*, pages 1–8, June 2012. Brisbane, Australia, June 10-15, 2012.
- [119] Erkan Kayacan, Erdal Kayacan, I-Ming Chen, Herman Ramon, and Wouter Saeys. *On the Comparison of Model-Based and Model-Free Controllers in Guidance, Navigation and Control of Agricultural Vehicles*, pages 49–73. Springer International Publishing, Cham, 2018. doi: 10.1007/978-3-319-72892-6_3. 22
- [120] Ahmed Khalifa and Mohamed Fanni. A new quadrotor manipulation system: Modeling and point-to-point task space control. *International Journal of Control, Automation and Systems*, 15(3):1434–1446, Jun 2017. 23

- [121] K. Baizid, G. Giglio, F. Pierri, M. A. Trujillo, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero. Experiments on behavioral coordinated control of an unmanned aerial vehicle manipulator system. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4680–4685, 2015. [24](#)
- [122] V. Lippiello, J. Cacace, A. Santamaria-Navarro, J. Andrade-Cetto, M. Á. Trujillo, Y. R. Esteves, and A. Viguria. Hybrid visual servoing with hierarchical task composition for aerial manipulation. *IEEE Robotics and Automation Letters*, 1(1):259–266, 2016. [24](#)
- [123] A. Caballero, A. Suarez, F. Real, V. M. Vega, M. Bejar, A. Rodriguez-Castaño, and A. Ollero. First experimental results on motion planning for transportation in aerial long-reach manipulators with two arms. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8471–8477, 2018. [24](#)
- [124] E. Cataldi, F. Real, A. Suarez, P. A. Di Lillo, F. Pierri, G. Antonelli, F. Caccavale, G. Heredia, and A. Ollero. Set-based inverse kinematics control of an anthropomorphic dual arm aerial manipulator. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2960–2966, 2019. in press. [24](#)
- [125] M. Geisert and N. Mansard. Trajectory generation for quadrotor based systems using numerical optimal control. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2958–2964, 2016. [25](#)
- [126] D. Lunni, A. Santamaria-Navarro, R. Rossi, P. Rocco, L. Bascetta, and J. Andrade-Cetto. Nonlinear model predictive control for aerial manipulation. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 87–93, 2017. [25](#), [109](#)
- [127] Roberto Rossi. *Control of an unmanned aerial vehicle equipped with a robotic arm*. PhD thesis, Italy, 2017. [xxv](#), [26](#), [27](#)
- [128] Burak Yüksel. *Design, Modeling and Control of Aerial Robots for Physical Interaction and Manipulation*. PhD thesis, 2017. [31](#), [32](#), [37](#), [132](#), [133](#)
- [129] Michael F. Ashby. Chapter 4 - material property charts. In Michael F. Ashby, editor, *Materials Selection in Mechanical Design (Fourth Edition)*, pages 57 – 96. Butterworth-Heinemann, Oxford, fourth edition edition, 2011. ISBN 978-1-85617-663-7. doi: <https://doi.org/10.1016/B978-1-85617-663-7.00004-7>. URL <https://www.sciencedirect.com/science/article/pii/B9781856176637000047>. [xxi](#), [33](#), [34](#)
- [130] FFI Federation. Handbook of finnish plywood. *Kirjapaino Markprint Oy, Lahti, Finland*, 2002. [34](#)

- [131] Y. Ohnishi, T. Takaki, T. Aoyama, and I. Ishii. Development of a 4-joint 3-dof robotic arm with anti-reaction force mechanism for a multicopter. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 985–991, 2017. 35, 37
- [132] Hiroaki Gomi and Mitsuo Kawato. Neural network control for a closed-loop system using feedback-error-learning. *Neural Networks*, 6(7):933 – 946, 1993. 65
- [133] T. Lee, M. Leok, and N. H. McClamroch. Nonlinear robust tracking control of a quadrotor uav on $se(3)$. In *2012 American Control Conference (ACC)*, pages 4649–4654, 2012. Montreal, Canada, June 27-29, 2012. 67
- [134] M. Biglarbegan, W. W. Melek, and J. M. Mendel. On the stability of interval type-2 tsk fuzzy logic control systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(3):798–818, June 2010. 68, 70
- [135] Andon V. Topalov, Yesim Oniz, Erdal Kayacan, and Okyay Kaynak. Neuro-fuzzy control of antilock braking system using sliding mode incremental learning algorithm. *Neurocomputing*, 74(11):1883 – 1893, 2011. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2010.07.035>. 70
- [136] M. A. Khanesar, E. Kayacan, M. Reyhanoglu, and O. Kaynak. Feedback error learning control of magnetic satellites using type-2 fuzzy neural networks with elliptic membership functions. *IEEE Transactions on Cybernetics*, 45(4):858–868, 2015. 70, 72
- [137] Erdal Kayacan. Interval type-2 fuzzy logic systems: Theory and design. *Bogazici University: PhD theses*, 2011. 73
- [138] E. Kayacan, O. Cigdem, and O. Kaynak. Sliding mode control approach for online learning as applied to type-2 fuzzy neural networks and its experimental evaluation. *IEEE Transactions on Industrial Electronics*, 59(9): 3510–3520, 2012. 73, 74
- [139] V. I. Utkin. *Sliding modes in control optimization*. Springer-Verlag, 1992. 73
- [140] Nursultan Imanberdiyev and Erdal Kayacan. A fast learning control strategy for unmanned aerial manipulators. *Journal of Intelligent & Robotic Systems*, 94(3):805–824, Jun 2019. 73
- [141] Boyang Li, Weifeng Zhou, Jingxuan Sun, Chihyung Wen, and Chihkeng Chen. Model predictive control for path tracking of a vtol tailsitter uav in an hil simulation environment. In *2018 AIAA Modeling and Simulation Technologies Conference*, page 1919, 2018. Kissimmee, Florida, January 8-12 2018. 77
- [142] C. Bohn and D. P. Atherton. An analysis package comparing pid anti-windup strategies. *IEEE Control Systems*, 15(2):34–40, April 1995. 78

- [143] H. Seraji and R. Colbaugh. Singularity-robustness and task-prioritization in configuration control of redundant robots. In *29th IEEE Conference on Decision and Control*, pages 3089–3095 vol.6, 1990. [91](#), [95](#), [96](#)
- [144] H. Seo, S. Kim, and H. J. Kim. Locally optimal trajectory planning for aerial manipulation in constrained environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1719–1724, 2017. [94](#)
- [145] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2010. ISBN 1849966346. [95](#), [99](#), [132](#)
- [146] Farbod Fahimi. *Autonomous Robots: Modeling, Path Planning, and Control*. Springer, 2009. [96](#), [97](#)
- [147] K. Dufour and W. Suleiman. On integrating manipulability index into inverse kinematics solver. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6967–6972, 2017. [99](#)
- [148] F. Kanehiro, W. Suleiman, K. Miura, M. Morisawa, and E. Yoshida. Feasible pattern generation method for humanoid robots. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 542–548, 2009. [100](#), [101](#)
- [149] K. Dufour and W. Suleiman. On inverse kinematics with nonlinear criteria: Trajectory relaxation. In *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*, pages 102–107, 2018. [101](#)
- [150] Michael J Todd and E Alper Yildirim. On Khachiyan’s algorithm for the computation of minimum-volume enclosing ellipsoids. *Discrete Applied Mathematics*, 155(13):1731–1744, 2007. [110](#)
- [151] Matteo Rubagotti, Tasbolat Taunyazov, Bukeikhan Omarali, and Almas Shintemirov. Semi-autonomous robot teleoperation with obstacle avoidance via model predictive control. *IEEE Robotics and Automation Letters*, 4(3):2746–2753, 2019. [110](#)
- [152] Boris Houska, Hans Joachim Ferreau, and Moritz Diehl. Acado toolkit-an open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011. [113](#)
- [153] Hans Joachim Ferreau, Christian Kirches, Andreas Potschka, Hans Georg Bock, and Moritz Diehl. qpOases: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014. [113](#)
- [154] Peng Xu. *Unified Dynamics and Control of a Robot Manipulator Mounted on a VTOL Aircraft Platform*. PhD thesis, 2014. [131](#), [132](#), [133](#)

-
- [155] Vincenzo Lippiello and Fabio Ruggiero. Cartesian impedance control of a uav with a robotic arm. *IFAC Proceedings Volumes*, 45(22):704–709, 2012. [134](#), [135](#)
 - [156] G. Arleo, F. Caccavale, G. Muscio, and F. Pierri. Control of quadrotor aerial vehicles equipped with a robotic arm. In *21st Mediterranean Conference on Control and Automation*, pages 1174–1180, 2013. Patanias, Greece, June 25-28, 2013. [135](#)