Motion Control of Under-actuated Aerial Robotic Manipulators

MOTION CONTROL OF UNDER-ACTUATED AERIAL ROBOTIC MANIPULATORS

BY

MOHAMMAD JAFARINASAB, M.Sc., B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2017) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Motion Control of Under-actuated Aerial Robotic Manipulators
AUTHOR:	Mohammad Jafarinasab M.Sc., (Mechanical Engineering) B.Sc. (Mechanical Engineering) Isfahan University of Technology, Isfahan, Iran
SUPERVISOR:	Prof. Shahin Sirouspour
NUMBER OF PAGES:	xvi, 183

To my family

Abstract

This thesis presents model-based adaptive motion control algorithms for under-actuated aerial robotic manipulators combining a conventional multi-rotor Unmanned Aerial Vehicle (UAV) and a multi-link serial robotic arm. The resulting control problem is quite challenging due to the complexity of the combined system dynamics, under-actuation, and possible kinematic redundancy. The under-actuation imposes second-order nonholonomic constraints on the system motion and prevents independent control of all system degrees of freedom (DOFs). Desired reference trajectories can only be provided for a selected group of independent DOFs, whereas the references for the remaining DOFs must be determined such that they are consistent with the motion constraints. This restriction prevents the application of common model-based control methods to the problem of this thesis. Using insights from the system under-actuated dynamics, four motion control strategies are proposed which allow for semi-autonomous and fully-autonomous operation. The control algorithm is fully developed and presented for two of these strategies; its development for the other two configurations follows similar steps and hence is omitted from the thesis. The proposed controllers incorporate the combined dynamics of the UAV base and the serial arm, and properly account for the two degrees of under-actuation in the plane of the propellers. The algorithms develop and employ the second-order nonholonomic constraints to numerically determine motion references for the dependent DOFs which are consistent

with the motion constraints. This is a unique feature of the motion control algorithms in this thesis which sets them apart from all other prior work in the literature of UAV-manipulators. The control developments follow the so-called method of virtual decomposition, which by employing a Newtonian formulation of the UAV-Manipulator dynamics, sidesteps the complexities associated with the derivation and parametrization of a lumped Lagrangian dynamics model. The algorithms are guaranteed to produce feasible control commands as the constraints associated with the under-actuation are explicitly considered in the control calculations. A method is proposed to handle possible kinematic redundancy in the presence of second-order motion constraints. The control design is also extended to include the propeller dynamics, for cases that such dynamics may significantly impact the system response. A Lyapunov analysis demonstrates the stability of the overall system and the convergence of the motion tracking errors. Experimental results with an octo-copter integrated with a 3 DOF robotic manipulator show the effectiveness of the proposed control strategies.

Acknowledgements

My thanks and deepest appreciation to my supervisor, Dr. Shahin Sirouspour, for his guidance, support and encouragement throughout my PhD studies. Sincere gratitude to my fellow colleague Eric Dyer for his help and support in my research. My appreciation to my other committee members Dr. Tim Davidson and Dr. Gary Bone for the constructive feedback regarding my research. Finally, very special thanks to my family since they have made it possible for me to excel in my studies with their endless love and support throughout my life.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Mobile manipulation has been an active field of research during the last two decades [1]. Taking advantage of a mobile platform and a dexterous robotic manipulator, mobile manipulators are able to interact with unstructured and dynamic environments and perform complex manipulation tasks. To date, most research on the subject has focused on development of ground mobile manipulators. Stability [2], navigation and trajectory planning [3; 4], motion control [5; 6; 7], hybrid and contact control [8; 9], and coordinated control of multiple mobile manipulators [10; 11; 12], have been widely studied in the literature for ground mobile manipulators. However, this type of mobile manipulation has limitations since the mobile base is restricted to move in a two dimensional space and may encounter obstacles and obstructed places which could limit its workspace and reach.

Equipping an Unmanned Aerial Vehicle (UAV) with a dexterous manipulator can help overcome some of the limitations of ground-based mobile manipulators. This combination would produce a mobile manipulator with expansive three-dimensional workspace, capable

of operating in otherwise unreachable task environments, helping spur new applications of mobile manipulation. Such UAV-Manipulators can be employed in aerial transportation, search and rescue missions, inspection and maintenance in hazardous environments, inspection of civil infrastructure and cooperative aerial assembling [13].

Aerial robot manipulators, although offering great advantages over ground mobile manipulators, would introduce new challenges both in theoretical development and practical implementation of a motion control system meeting the requirements for their safe and effective operation. The most significant of these challenges is to achieve stable high performance motion control in an under-actuated robotic system without a stable base platform, and in the presence of dynamic interactions between the base and the robotic arm, as well as disturbances. In addition to motion control, trajectory planning, contact control, and cooperative control may be of interest in applications of aerial manipulators. Practical limitations in sensing and localization, payload capacity, and power and energy supply can also complicate the control of such systems, although advances in sensing and battery technologies are expected to mitigate many of these issues in the future.

While interest in aerial manipulation has been growing in recent years, the body of research on the subject is still sparse compared to other areas of robotics. More specifically, few advanced model-based motion controllers are available for these systems. This thesis focuses on the motion control of an under-actuated multi-rotor UAV integrated with a serial robotic manipulator and seeks solutions for this problem that yield high-performance and provably stable tracking response.

1.2 Problem Statement

Similar to applications of conventional robots, many tasks involving UAV-Manipulators require some form of motion control. However, while motion control of ground-based robots has been widely covered in the literature, far fewer solutions exist for their aerial counterparts. While this might be partly due to the novelty of UAV-manipulators, the main reason lies with the complexities and challenges associated with the control of such robots.

The first challenge in control design for a UAV-Manipulator system is to ensure system stability. This is difficult not only because of the inherent instability of the flying base but also due to the presence of dynamic coupling between the UAV and manipulator, payload/system parametric uncertainty, interaction with the environment, and possible disturbances. The majority of existing controllers for aerial manipulators are variants of PID controllers, due to their simplicity in design. These methods consider the UAV and the manipulator as two separate systems and treat the interaction forces/torques between them as disturbances. Each system would have its own control unit to achieve its prescribed control objectives, while attempting to reject the disturbance imposed by the other system. While simple to design and implement, such controllers could result in poor control performance and even system instability when the dynamic coupling between the UAV and the manipulator is significant [14]. Typical cases are when the inertia properties of the manipulator/payload are considerable relative to the base platform, the manipulator is interacting with the environment, and fast maneuvers/manipulation tasks are executed. This underscores the need for a model-based control design based on the full nonlinear dynamics of the system which takes into account the dynamics coupling between the two subsystems.

Second, unlike most conventional manipulators, the multi-rotor UAV-Manipulator is

an under-actuated mechanical system. In this system, the manipulator links and the propellers are directly actuated while the UAV base has only four indirect actuations, fewer than the required full six actuations, through the aerodynamic thrust forces produced by its propellers. In a conventional multi-rotor UAV, the propellers' forces are parallel to each other and hence are unable to produce any lateral force in the plane perpendicular to the thrust forces. Only one thrust force and three torques are available to the base. This under-actuation imposes second-order nonholonomic constraints on the system motion and reduces the number of independently controllable degrees of freedom (DOFs) [15]. The constraints preclude direct application of conventional model-based robot motion controllers to a UAV-Manipulator since not all its DOFs can be controlled independently. Only a subset of DOFs consistent with the degrees of actuation can be independently controlled and accept desired reference trajectories. The reference trajectories for the remainder of the system DOFs, required for model-based control development, are unknown and must be determined while satisfying the nonholonomic constraints.

Third, in a task-space control strategy where the manipulator end-effector DOFs are aimed to be controlled directly, the UAV-Manipulator presents a potential kinematic redundancy. This redundancy in motion is desirable since the additional DOFs can be used to perform a secondary task or to maximize a performance criteria. However, the nonholonomic constraints on the multi-body system accelerations prevent the use of commonly known kinematic redundancy resolution methods [16; 17], and [18]. In this case, while resolving the kinematic redundancy, the second-order motion constraints should also be taken into account or otherwise the solution is not dynamically feasible.

Fourth, many applications of such systems involve manipulation of unknown payloads in addition to possibly significant disturbances acting on the vehicle body during flight.

Therefore, the control algorithm must be able to maintain the system stability and a high level of performance in the presence of dynamic uncertainty and disturbances. Stable high-performance motion control requires formulation and parameterization of the full coupled nonlinear dynamics of the system. Using a Lagrangian dynamics formulation, which most model-based controllers rely on, can complicate the control development and real-time implementation, particularly for hyper-DOF systems such as aerial manipulators.

Lastly, in cases where the propellers dynamics significantly affect the entire system dynamics, an example of which is when the number of propellers is odd, it is crucial to take their dynamics into account in the control design. In such cases, when the base experiences a change in orientation, gyroscopic torques affect the vehicle due to the very high speed of the propellers. Moreover, the aerodynamic thrusts, which are used to manipulate the entire system, are dependent on the propeller speeds. As a result, these forces cannot instantaneously change. To include the transient state of thrust generation in the control design, one needs to fully incorporate the propellers dynamics into the system model by considering them as rigid bodies interacting with the base. However, this increases the model DOFs and further complicates the dynamic modeling and control design.

The discussions above highlight the need for model-based control algorithms which not only properly take the system under-actuation into account but also are computationally efficient, capable of maintaining system stability and performance in the presence of uncertainty, and able to resolve possible system kinematic redundancy.

1.3 Thesis Contributions

This thesis tackles some of the key challenges in the motion control of under-actuated UAV-Manipulators. A comprehensive theoretical treatment of the problem, including dynamics

modeling, controller design, and performance and stability analysis is presented. Model-based adaptive control algorithms are developed using the method of Virtual Decomposition Control (VDC) [19; 20]. The model-based motion control algorithms proposed in this thesis divide the system DOFs to independently controlled and dependent DOFs. The control designer/system user would provide reference trajectories for the independent DOFs similar to what is done in conventional robot control. The second-order motion constraints are then employed to derive references consistent with system under-actuation for the remaining dependent DOFs. This is an important unique feature of the controllers in this work, which set them apart from all other relevant prior work on the subject in the literature. Despite the system under-actuation, the proposed model-based controllers achieve the control objectives without using any force/acceleration measurement. As a result, the control system avoids the implementation complexities introduced by such measurements in the feedback loop, such as possible instability due to an algebraic loop and feedback time-delay.

Four control strategies resulting in different system operation modes are proposed and control development is presented fully for two of these strategies; the controllers for the other two strategies can be formulated using similar steps but are not presented to avoid repetition. One strategy allows for a human-in-the-loop mode of system operation whereas the second one is suitable for fully autonomous control. In the former strategy, a human operator can position the UAV by giving the UAV orientation reference commands and manipulate the robotic arm by commanding the joint positions. The other approach allows for full autonomous control of the UAV-Manipulator using an off-line or online trajectory planner that provides references for the UAV position and the manipulator end-effector DOFs. In this case, the UAV-Manipulator presents a possible kinematic redundancy. A

novel approach is proposed to handle this kinematic redundancy in the presence of the under-actuation constraints. This is achieved by reserving some system DOFs to satisfy the second-order motion constraints in the system acceleration space, allowing the rest of system DOFs to satisfy the kinematic constraints in the velocity space.

A desirable feature of the controllers proposed here is its ability to adapt to the system uncertain parameters. These VDC-based algorithms decompose the overall motion control of the multi-body system into local motion control of individual rigid bodies. This decomposition results in a modular model-based control design using the Newton-Euler dynamics formulation. As a result, the control development not only avoids the derivation, parametrization, and implementation complexities of Lagrangian-based control methods, but also allows for decentralized parameter adaptation. Decentralized adaptation allows to easily disable/enable parameter adaption for any rigid body in the system. This feature is especially useful for high performance aerial manipulation since the target object inertia properties are not known a priori.

The control algorithm is developed both with and without considering the propeller dynamics. When these dynamics are considered, the UAV control inputs are the low level propeller motor torques and the propellers are modelled as separate rigid bodies and are fully integrated in the control design. In many scenarios, however, a simpler controller which treats the propellers as part of the UAV rigid body and commands the propeller angular velocities may be sufficient.

The proposed model-based controllers in this thesis are implemented on a prototype of a UAV-Manipulator system. A standard octo-copter configuration UAV is integrated with a 3 DOF serial robotic manipulator for the experiments. The experimental system acquires the required feedback data asynchronously at multiple rates from sensor units and uses them in

the calculations of the control commands. The system integrates and coordinates the operation of different units including a Motion Capture System, IMU, optical encoders, multiple micro-processors, and motor drivers with reliable/robust communication links among these components. To the best of author's knowledge, there is no prior work in the literature to date reporting real-time implementation of a full model-based motion control algorithm for the under-actuated UAV-Manipulator where system under-actuation is taken into account in the control design.

The thesis contributions can be summarized as follows:

- Comprehensive theoretical treatment of the motion control problem for the UAV-Manipulator from dynamics modeling and controller design to performance and closed-loop stability analysis.

- Development of adaptive model-based motion controllers for the under-actuated UAV-Manipulator systems based on the full nonlinear dynamics model of the system in 3D motion. The proposed controllers incorporate a modular parameter adaptation mechanism, making them particularly suited for stable high performance manipulation of unknown objects.

- Guaranteeing the feasibility of the control commands by taking the under-actuation into account in the control algorithm developments; this is a key missing feature of virtually all methods in the literature of the UAV-Manipulator motion control.

- Proposing four motion control strategies which allow for semi-autonomous and fully-autonomous operation of the UAV-Manipulator system. In each case, a method is proposed to derive reference trajectories for the system dependent DOFs which are consistent with the second-order motion constraints imposed by the under-actuation.

- Tackling the problem of model-based task-space motion control for under-actuated aerial manipulators. Proposing an approach for handling kinematic redundancy of the system in the presence of second-order motion constraints imposed on the system generalized accelerations, velocities, and coordinates.
- Extension of the model-based controllers by considering the propeller dynamics in control developments to generalize the control framework for the cases where the propeller dynamics have considerable contribution to the overall system dynamics.
- Providing rigorous analysis for the stability of the overall closed-loop system and demonstrating the convergence of motion tracking errors under the proposed control algorithms.
- Experimental validation of the full model-based motion controllers on an actual UAV-Manipulator for the first and fourth control scenarios.

1.4 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 provides a review of the pertinent literature on aerial manipulation. Chapter 3 studies the kinematics and dynamics of the UAV-Manipulator. The full nonlinear dynamics equations are derived based on the Newton-Euler formulation. Propellers dynamics are fully incorporated in the system dynamics modeling. It is shown that the system is under-actuated in six DOFs if the propellers are considered in the modeling and otherwise in two DOFs. Chapter 4 briefly reviews the method of "Virtual Decomposition Control", based on which the control algorithms in this thesis are developed. Chapter 5 discusses different strategies for the motion control of the UAV-Manipulator system. The effects of under-actuation on the system motion control are

explained in this chapter and four different possible scenarios for the system operation are discussed.

Chapters 6 and 7 develop model-based motion control algorithms for the first and the second control scenarios discussed in Chapter 5. In these chapters, the propeller dynamics are excluded from the model in control design. A method is proposed to handle the non-holonomic constraints on the system motion. Moreover, Chapter 7 tackles a possible kinematic redundancy in the presence of second-order motion constraints. Chapter 8 extends the controllers for the first and the second control scenarios by considering the propeller dynamics. A Lyapunov analysis of the system stability under the proposed control laws is carried out in Chapter 9.

Chapter 10 briefly presents the system implementation for experimental validation of the proposed controllers. Hardware and software developments are explained in this chapter and some of the challenges in controller implementation are discussed. Chapter 11 presents the results of real-time implementation as well as numerical simulations of the proposed controllers. Experimental results for controller implementation in two control scenarios on an octo-copter with a 3 DOF robotic arm are given. Computer simulations are carried out for two control scenarios involving an octo-copter with a 2 DOF robotic manipulator. Simulation and experimental results demonstrate the effectiveness of the proposed model-based controllers. Finally, the thesis is concluded in Chapter 12 where some possible directions for future research are also discussed.

1.5 Related Publications

1.5.1 Journal Articles

- M. Jafarinasab, S. Sirouspour, and Eric Dyer, "Model-based motion control of a robotic manipulator with a flying multi-rotor base", *Submitted to IEEE Transactions on Robotics*, 2017.

- M. Jafarinasab and S. Sirouspour, "Task-space motion control of under-actuated aerial mobile manipulators", *To be submitted to IEEE Transactions on Robotics*, 2017.

- D. Findlay, M. Jafarinasab and S. Sirouspour, "Design optimization and prototyping of a bi-modal aerial/ground mobile manipulator" *Submitted to IEEE/ASME Transactions on Mechatronics*, 2017.

1.5.2 Conference Papers

- M. Jafarinasab and S. Sirouspour, "Adaptive motion control of aerial robotic manipulators based on virtual decomposition," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg,* Sept 2015, pp. 1858-1863. [21]

- D. Findlay, M. Jafarinasab and S. Sirouspour, "Optimization-based design of a novel hybrid aerial/ground mobile manipulator," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg,* Sept 2015, pp. 2467-2472. [22]

1.6 Basic Notation

In this thesis, generalized/concatenated matrices and vectors are denoted by an italic math font, e.g. $\mathcal{A}$, basic matrices are denoted as bold and upper-case, e.g. $\mathbf{A}$, vectors are bold and italic, e.g. $\boldsymbol{A}$, and scalar quantities are non-bold and italic, e.g. a or A.

A vector quantity associated with a coordinate frame such as velocity or force, is also expressed in the same coordinate frame unless otherwise stated. For notational simplicity, throughout the entire thesis, $\boldsymbol{W}_\alpha$ refers to a vector quantity $\boldsymbol{W}$ measured and also expressed in frame α. Furthermore, α belongs to the set of all body-fixed coordinate frames.

Chapter 2

Literature Review

Research in aerial manipulation and transportation can generally be divided into three categories. In a first approach, a UAV equipped with a simple gripper, installed underneath or to one side of the drone, is utilized to grab a target object or perch on an structure. In [23], a helicopter is equipped with a gripper and object grasping in hover is demonstrated. To align the gripper and the object in the presence of large positioning errors, an adaptive compliant gripper is used. Stability of this system in compliant contact is studied under PD-PID control [24].

Several gripper designs have been proposed to allow effective grasp and perch [25; 26]. Construction of a structure composed of small cubic pieces using a team of quad-copters is reported in [27]. In a similar work, quad-copters have been used to pick up, transport, and assemble small pieces of architectural structures [28]. The problem of coordinated and cooperative aerial grasping and transportation of an object by multiple quad-copters with grippers is studied in [29]. Using a team of quad-copters, a relatively large object is transported to a desired position following a three-dimensional trajectory. This type of aerial manipulation has limited applications since the object is rigidly attached to the drone

and has restricted manipulability.

The second approach is the transportation of a cable suspended payload by a single [30; 31] or multiple [32] UAVs. In [31], a motion planning method is proposed to minimize possible oscillations of the cable suspended object due to under-actuation. Coordinated control of a team of quad-copters lifting and carrying a suspended payload is studied in [32]. In this work, attitude and position control of the object is demonstrated in experiments using a team of three quad-copters. The main limitation of this approach in aerial manipulation is the inevitable swings of the object which restrict the trajectory tracking capabilities of the system. The other control challenge here is to guarantee that the cable remains in tension without which instability can occur.

The third approach in aerial manipulation, which is also the subject of this thesis, is to utilize a UAV equipped with a dexterous multi-DOF robotic arm. This approach has a number of advantages over the other two approaches such as active interaction with environment, superior mobility, and precise motion tracking. Performing interactive tasks such as pick-and-place, insertion, valve turning, and opening a drawer using a UAV-Manipulator are demonstrated in experiments [33; 34; 35]. This type of aerial manipulation has been actively studied in recent years. In the remainder of this chapter, the state-of-the-art in the literature of UAV-Manipulator systems is reviewed. First, advances in mechanical design of UAV-Manipulators are presented. Then, the existing research on motion control of such systems is explored. Finally, a few papers on contact control of the UAV-Manipulator are reviewed.

2.1 Mechanical Design

There has been several mechanical designs proposed for the manipulator and its end-effector for aerial manipulation. The main purpose in these designs is to reduce the dynamic coupling between the UAV and the manipulator in order to help with flight stability during arm movements and contact with environment.

A hyper-redundant manipulator is proposed to increase the dexterity and work-space of the UAV-Manipulator in [36]. This redundant manipulator uses a passive compliant 2 DOF gripper with one degree of actuation to grasp a wide variety of objects with minimal grasp planning. In [37], a light weight flexible-joint arm is mounted on a small-sized UAV to achieve intrinsically safe physical interaction with the environment using the joint elasticity. The effectiveness of this system in physical interaction is demonstrated experimentally. A 5 DOF light-weight manipulator is built and integrated with a small quad-copter in [38]. The manipulator is designed such that it can fold on itself during landing. Moreover, the design is conceived to constrain the arm Center of Mass (CM) as close as possible to the base CM, thus reducing the dynamic coupling between the two systems.

In [39], the use of a light-weight arm actuated by Shape Memory Alloy actuators is proposed in order to increase force to mass ratio of the manipulator. A dual-arm aerial manipulator is introduced in [40] where two 5 DOF human-sized arms are integrated with an octo-copter. Emulating the human arm, a compliant manipulator is proposed in [41] to reduce the effects of aerial contact on system stability. In this design, a spring is used to connect the two arm links in order to allow estimation of the payload/contact force through the spring elongation. A 2 DOF manipulator composed of an active rovolute and a passive linear joints is built and tested on a UAV for aerial contact. The impact energy in is absorbed through the passive joint and a directional locking mechanism while the

active joint is used to compensate for the UAV possible pitch movements during contact. A Global Conditioning Index is used as a performance measure to optimize a parallel arm geometric parameters in order to obtain a singularity-free large work-space in [42].

In [43], a novel hybrid aerial/ground mobile manipulator is introduced by our group. A systematic optimization-based approach is employed to minimize the overall mass of the system while manipulator work-space, actuator torque/force limits, and dynamic tip-over stability of the mobile manipulator for ground operations are all considered as design constraints. Some UAV-manipulators are designed for specific applications. Design and implementation of a compliant aerial manipulator with an application for measuring a bridge girder deflection over time is recently reported in [44]. A 4 DOF manipulator is mounted on top of a UAV for this application. The airframe of a quad-copter is adapted to carry a miniature robotic manipulator in [45; 46]. The robotic arm is particularly designed for aerial inspection of industrial plants and to perform non-destructive tests. This manipulator consists of a delta-type 3 DOF parallel arm and a 3 DOF end-effector based on a Cardan gimbal. Finally, a special design is investigated for an aerial manipulator with door-opening capability in [47].

2.2 Motion Control

Motion control methods for the UAV-Manipulator system are categorized into three groups. The first group are decentralized where the UAV and the manipulator controllers are designed independently. The second group consider only part of the dynamics coupling between the two systems in the control design. In the third category are the controllers which are based on the full nonlinear dynamics of the entire system. These motion control methods are reviewed in the remainder of this section.

2.2.1 Decoupled Controllers

The majority of existing controllers for aerial manipulators are PID controllers with different design architectures. This is due to the simplicity of these controllers and to avoid complexities involved in a model-based design for an under-actuated system [48; 49; 50; 51; 52; 53; 54]. In these approaches, the UAV and the manipulator are considered as two separate systems where the force/torque couplings between them are treated as disturbances. Therefore, each system uses its own control unit, while attempting to reject the disturbance imposed by the other system. These methods could perform poorly and even become unstable when the inertia properties of the manipulator/payload are considerable relative to the base platform, the manipulator is interacting with the environment, and fast maneuvers are executed.

In [49; 55], the interaction force/torque exchanged between the UAV and the manipulator during the manipulator movements is studied. To avoid potential crashes, a six DOF gantry system is employed to emulate the dynamics behavior of a UAV during its operation. Having the UAV dynamics model as reference, the gantry motion is controlled by a Model Reference Adaptive Controller (MRAC). Two 4 DOF manipulators attached to the gantry system are used to perform grasping tasks. Computer vision techniques provided the target object and the manipulator position measurements to the control system. The effects of propeller motor dynamics on the stability of the system is studied in [48; 56]. A first-order transfer function is considered for the propeller motor and a simplified linear model of the the UAV attitude is utilized. Assuming the interaction forces between the arm and UAV as disturbances, it is shown that the proportional gain of an attitude PID controller is restricted by an upper bound to maintain system stability.

To control the UAV position, a two-layer algorithm is proposed in [51; 57; 58] where

an outer-loop controls the position by determining the reference roll and pitch orientations of the UAV. An inner-loop controller is tasked with following the orientation reference commands. In this approach, the arm is controlled in joint-space independently. In [51], the UAV position is controlled by a PD controller in an outer-loop; meanwhile an inner-loop high gain PID controller is employed to guarantee reference attitude tracking. Similarly, a PD controller with gravity compensation is proposed for position control as well as a PD attitude controller in [58].

In [59], a model-reference adaptive controller is proposed to adjust the PID gains such that the attitude response follows the reference model and rejects the disturbances from the arm interactions. In a similar approach, the PID gains of the UAV attitude controller are scheduled based on the manipulator joint configurations to improve the attitude trajectory tracking in [60; 61].

2.2.2 Loosely Coupled Controllers

To improve the control performance, some prior work partially considers the dynamics of the manipulator in controlling the base, while the two systems still have their own control units. This is to account for variations of inertial properties and counteract the static torque imposed on the base during the manipulator movements.

A back-stepping attitude controller for the base and a PID controller for the manipulator are suggested in [62]. To improve trajectory tracking precision, the manipulator joint positions are reported to the base controller so to account for variations of inertial properties during arm movements. Simulations show improved tracking precision of the base compared to a case where a PID attitude controller is employed. To counterweight the arm movements, a mechanism is introduced to move the UAV battery such that the entire

system CM remains unchanged as much as possible [63]. In this case, the residual static moment imposed on the base by the manipulator weight, is compensated in the base control algorithm.

In [64], a simplified model of the system dynamics in two-dimensional space is used where the robot arm is modeled as a point mass located at its CM. Using this model, a robust position controller is developed for the UAV. Closed-loop inverse kinematic control is employed to control the position of the end-effector in the plane. A similar approach is reported in [65] where a closed-loop inverse kinematic control algorithm with an integral error feedback are used to independently control the arm. Both these papers, however, treat the end-effector control problem as an inverse kinematic problem in the velocity space, ignoring the motion constraints due to the under-actuation.

2.2.3 Model-based Controllers

There have been some attempts to treat the UAV-Manipulator combination as a unified system in modeling and control in order to account for dynamics coupling of the two systems. In [66], a sliding-mode controller is developed based on the full Lagrangian dynamics of a UAV with a 2-link robotic arm. Later, a passivity-based controller is proposed for the same system in [67]. In these papers, small roll and pitch orientations are assumed to find reference trajectories for these DOFs such that a desired lateral force is created to control the UAV position. In the control design, however, the system is treated like a fully-actuated system. As a result, the control law shows implicit dependency on control inputs as well as their first and second time derivatives. This issue originates from the unknown trajectories for the system dependent DOFs *a priori* due to the under-actuation.

Other attempts use a CM coordinate transformation to transform the system dynamics

into translational dynamics of the CM as well as rotational dynamics of the base orientation and manipulator joints [68; 69; 58]. Based on this transformation, a back-stepping controller is proposed in [68] that requires acceleration measurement of the CM. Moreover, this work assumes system trajectories consistent with the nonholonomic constraints are available for all system DOFs, avoiding to deal with under-actuation. Similarly, a back-stepping control law is developed in [69] to control the 3 DOFs of the end-effector position. However, the system kinematic redundancy is not resolved. This has resulted in a redundant control law that only yields a combination of manipulator joint torques and UAV thrusts; the individual control inputs are not derived in this paper. A similar coordinate transformation is used in [70] to control a quad-copter with a single rigid link carrying a payload. Asymptotic convergence of the quad-copter position and the link orientation are proven.

To simplify the control problem, a two-dimensional model of the UAV-Manipulator is considered in [71; 72; 73] which is restricted to planar motions. In [71], a passivity-based robust controller is developed using the energy-shaping methodology. The controller guarantees boundedness of trajectories under bounded motions of the manipulator. In [72], three cases are studied in order to control the arm end-effector DOFs. The first case assumes a single link attached to the CM of a UAV through a revolute joint. The second one considers an offset between the robotic arm base and the drone CM. The third case studies a redundant arm by adding two prismatic joints to the manipulator where the joint displacements are assumed as control inputs. It is shown that the manipulator redundancy can be employed to improve the tracking performance in certain configurations.

A two-dimensional dynamics model of a quad-copter with a 1 DOF prismatic joint arm is considered in [73]. Using this model, a feed-back linearizing controller combined with a state feed-back control is developed. In [74], an adaptive controller is proposed for

task-space control of a redundant UAV manipulator. However, an assumption is made that the manipulator is much lighter than the UAV in order to simplify the control design by ignoring parts of the dynamics couplings. Moreover, the second-order motion constraints are ignored in resolving the system kinematic redundancy and common closed-loop inverse kinematics algorithms are employed.

The above review of the literature reveals that the under-actuation has significantly restricted the application of available model-based control methods to the UAV-Manipulator system. In all the above referenced papers, the system under-actuation is either ignored, not properly addressed, or has restricted the analysis and design to special cases of a reduced planar system dynamics, all due to the complexities that it introduces in control design.

2.3 Contact Control

Cartesian impedance control for UAV-Manipulator is studied in [75]. Both cases of non-redundant and kinematically redundant systems are considered, and in the case of redundancy, a generic secondary task is achieved using the Jacobian null-space. A hierarchical impedance control algorithm is presented in [76]. The algorithm is composed of an outer loop compliant trajectory generator for the end-effector, a middle-loop inverse kinematics solver, and an inner-loop joint controller for the arm, while the UAV has a separate position controller. These papers, however, disregard the fundamental under-actuation property of the system by not accounting for the second-order motion constraints.

Chapter 3

Kinematics and Dynamics Modeling

This chapter presents kinematics and dynamics analysis of the UAV-Manipulator system. In the first section, the robot structure is described, configuration parameters are defined, and velocity analysis is developed. In the second section of the chapter, dynamics equations of the system motion are derived which lay the foundation for the system control design.

3.1 Robot Kinematics

In this section, the structure of the robot under study is described first and the assumptions made are discussed. Configuration analysis is presented next where the forward/inverse kinematics of the robot are also discussed. The last part of this section is dedicated to the differential kinematics of the robot.

3.1.1 Robot Structure

The mechanical system considered in this thesis is composed of a UAV and a robotic manipulator. The UAV is considered to be a conventional multi-rotor drone with m actuated

propellers in the same direction. As a result, the thrust forces generated by all the propellers are parallel and perpendicular to the propellers plane.

The robotic manipulator is assumed to be a serial robotic arm with an open kinematic chain. The arm has in general n links/joints. The manipulator joints can be either revolute or prismatic. Each joint is assumed to be actuated.

Remark. *In the subsequent sections, all system bodies are treated as rigid bodies and higher order system dynamics caused by any elasticity in the system are ignored.*

Remark. *If the UAV is fully-actuated, the control design problem becomes less complicated and challenging. In this case, the control algorithms proposed in this thesis can be adapted and simplified to be used for the fully-actuated system.*

3.1.2 Configuration Analysis

Fig. 3.1 shows a schematic of a UAV equipped with an n-link serial robotic manipulator holding an arbitrary object. The UAV is considered to have m actuated propellers, all of which are oriented in the same direction and produce a uni-directional thrust force when spinning.

Generalized Coordinates

To describe the system configuration and perform kinematics/dynamics analysis, the following coordinate frames are defined and used in all future developments:

- I : an inertial frame
- B_0: a body-fixed frame attached to the UAV center of mass

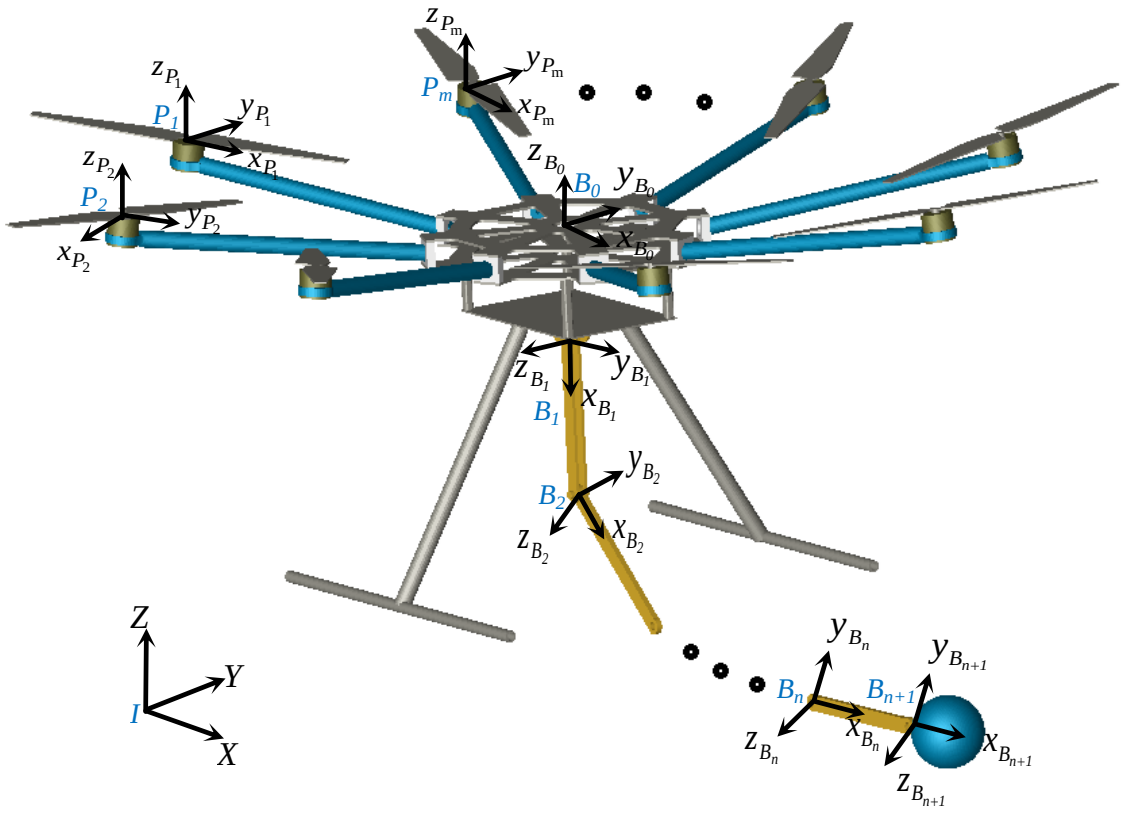

Figure 3.1: Schematic of a UAV with m actuated propellers and an n-link robotic manipulator holding an arbitrary object

- B_i: a body-fixed frame attached to the ith link with $i = 1, \cdots, n$
- B_{n+1}: a body-fixed frame attached to the object
- P_i: a body-fixed frame attached to the ith propeller with $i = 1, \cdots, m$

The above defined frames are all shown in Fig. 3.1. It is worth noting that each frame is attached at a joint connecting two rigid bodies in the system, but not the center of mass of each system body. The z axis of the frame B_0 is aligned with the propeller thrust forces. The z axes of $B_i, i = 1, \cdots, n$ and $P_i, i = 1, \cdots, m$, are aligned with the actuation axis of the corresponding rigid body. Throughout this thesis, we denote a body-fixed frame

by α; this frame belongs to the set of all body-fixed coordinate frames in the system i.e. $\{\boldsymbol{B}_0, \cdots, \boldsymbol{B}_{n+1}, \boldsymbol{P}_1, \cdots, \boldsymbol{P}_m\}$.

This multi-body mechanical system is composed of $N_b = n+m+2$ solid bodies namely n manipulator links, m propellers, the UAV base, and the object. It has $N_d = 6 + n + m$ DOFs. These include six DOFs for the position and orientation of the UAV base, n DOFs for the n-link manipulator, and m DOFs for the UAV propellers.

A minimal representation of the system configuration requires $6 + n + m$ parameters. The following variables are defined as the system generalized coordinates to describe the entire system configuration in the three-dimensional space,

$$\xi \triangleq \left[{}^{I}\boldsymbol{P}_{B_0}^T, \boldsymbol{E}_{I,B_0}^T, \boldsymbol{q}_m^T, \boldsymbol{q}_p^T\right]^T \in \Re^{N_d}, \tag{3.1}$$

The first two components of the generalized coordinates vector namely ${}^{I}\boldsymbol{P}_{B_0}$ and $\boldsymbol{E}_{I,B_0}$ correspond to the 6 DOFs of the UAV base. ${}^{I}\boldsymbol{P}_{B_0} \triangleq [X_{B_0}, Y_{B_0}, Z_{B_0}]^T \in \Re^3$ is the position vector of frame B_0 with respect to frame I and expressed in I. $\boldsymbol{E}_{I,B_0} \triangleq \left[\Phi_{x_{B_0}}, \Theta_{y_{B_0}}, \Psi_{z_{B_0}}\right]^T \in \Re^3$ is the vector of Euler angles (roll, pitch and yaw) representing the rotation of B_0 with respect to I. The manipulator joint positions are described by $\boldsymbol{q}_m \triangleq [q_{m_1}, \ldots, q_{m_n}]^T \in \Re^n$ where q_{m_i} is the rotation/displacement of link i with respect to link $i-1$. The propeller orientations are described by $\boldsymbol{q}_p \triangleq [q_{p_1}, \ldots, q_{p_m}]^T \in \Re^m$ where q_{p_i} is the rotation angle of the ith propeller with respect to the frame B_0.

Remark. *Inertial Navigation Systems use the Euler rotations in the order of yaw, pitch, and roll to align an inertial frame with a body-fixed frame [77]. Here, the same convention is adopted in all developments. The inertial frame is first rotated about the current z axis with yaw. This is followed by a rotation about the current y axis with pitch. The final rotation is about the current x axis with roll which yields the body-fixed frame.*

Forward/Inverse Kinematics

A question of interest in the study of robotic manipulators is the mapping between the so called joint-space parameters and task-space (also known as configuration-space) parameters. This is known as the forward/inverse kinematic problem of a robot manipulator. The joint-space parameters of a conventional robotic arm are the robot joint positions here defined as $\boldsymbol{q}_m$. The task-space parameters, on the other hand, are the position and orientation of the robot end-effector. Here, we extend the definition of these parameters for the UAV-Manipulator system.

Definition. *For the UAV-Manipulator system, the joint-space parameters are defined as the position and orientation of the UAV base w.r.t the inertial frame as well as the manipulator joint positions. Furthermore, the position and orientation of the end-effector w.r.t the inertial frame are referred to as the task-space parameters.*
Joint-space parameters $\triangleq \{{}^I\boldsymbol{P}_{B_0}, \boldsymbol{E}_{I,B_0}, \boldsymbol{q}_m\}$
Task-space parameters $\triangleq \{{}^I\boldsymbol{P}_{B_{n+1}}, \boldsymbol{E}_{I,B_{n+1}}\}$

In the above definition, ${}^I\boldsymbol{P}_{B_{n+1}}$ is the position vector of B_{n+1} (i.e. end-effector frame) w.r.t and expressed in the inertial frame. Furthermore, $\boldsymbol{E}_{I,B_{n+1}}$ is the Euler angle representation of B_{n+1} w.r.t the inertial frame. To derive the mapping between the joint-space and the task-space parameters, one can consider the following vector/matrix equations in the kinematic chain shown in Fig. 3.2,

$$\overrightarrow{\boldsymbol{P}}_{IB_0} + \overrightarrow{\boldsymbol{P}}_{B_0B_1} + \overrightarrow{\boldsymbol{P}}_{B_1B_2} + \cdots + \overrightarrow{\boldsymbol{P}}_{B_nB_{n+1}} - \overrightarrow{\boldsymbol{P}}_{IB_{n+1}} = \mathbf{0} \tag{3.2}$$

$$\mathbf{R}^I_{B_0}\,\mathbf{R}^{B_0}_{B_1}\,\mathbf{R}^{B_1}_{B_2}\,\cdots\,\mathbf{R}^{B_n}_{B_{n+1}} - \mathbf{R}^I_{B_{n+1}} = \mathbf{0} \tag{3.3}$$

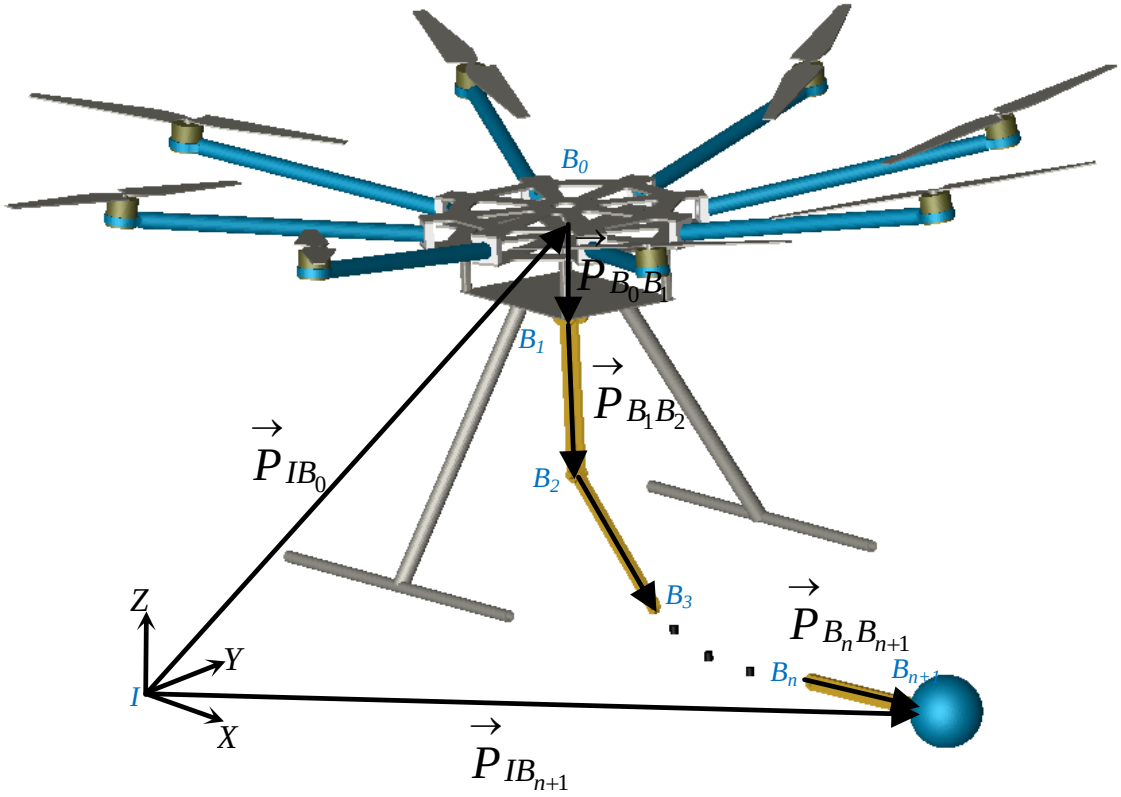

Figure 3.2: Vector loop relating the joint-space and task-space parameters

where in (3.2), $\overrightarrow{\boldsymbol{P}}_{IB_0}$ is the position vector of the origin of B_0 w.r.t I, $\overrightarrow{\boldsymbol{P}}_{B_iB_{i+1}}$ is the position vector of the origin of B_{i+1} w.r.t B_i for $i = 0, \cdots, n$, and $\overrightarrow{\boldsymbol{P}}_{IB_{n+1}}$ is the position vector of the origin of B_{n+1} w.r.t I. In (3.3), $\mathbf{R}^I_{B_0}$ is the rotation matrix describing the orientation of B_0 w.r.t I, $\mathbf{R}^{B_i}_{B_{i+1}}$ is the rotation matrix of B_{i+1} w.r.t B_i for $i = 0, \cdots, n$, and $\mathbf{R}^I_{B_{n+1}}$ is the rotation matrix of the end-effector frame w.r.t the inertial frame I.

When expressed in the inertial coordinate frame, Equation (3.2) represents three algebraic equations between the joint-space and the task-space parameters. Moreover, (3.3) establishes nine algebraic equations among these parameters where three of these equations are independent. As a result, (3.2) and (3.3) yield six independent highly nonlinear algebraic equations between the joint-space and the task-space parameters. These equations can be employed to solve the forward/inverse kinematics of the UAV-Manipulator.

Given the joint-space variables, the forward kinematic deals with the determination of the task-space variables. This is a straightforward problem for the system under study since the robotic arm is a serial manipulator. The inverse kinematic, however, is complicated. On one hand, the equations are highly nonlinear and coupled in terms of the joint-space parameters. On the other hand, the number of joint-space parameters is more than six. This introduces a redundancy in the solution known as kinematic redundancy [78].

To tackle the non-linearity of the kinematic equations, the differential form of these equations is often used. Taking the time derivative of the kinematic equations establishes six linear algebraic constraints between the joint-space and task-space velocities. These equations then can be solved using the well-developed methods in linear algebra. Finally, a numerical integration of the joint-space velocities yields the joint-space parameters. Following this approach, the original nonlinear kinematic equations need to be solved only

for the initial robot configuration. This is required to obtain the initial values for the numerical integration. This method, however, is prone to numerical integration issues which can be solved by including feedback correction terms [79; 80; 81]. The system kinematic redundancy is later discussed in Chapter 7 where control design in task-space is considered.

3.1.3 Velocity Analysis

In this section, the velocity kinematics of the UAV-Manipulator is studied. In order to facilitate the velocity analysis, a generalized transformation matrix is used. A minimal set of velocity variables is defined as the system generalized velocities. Given this set, forward velocity propagation is performed and the velocity vector of each coordinate frame in the system is derived. This is further formulated using a generalized configuration dependent Jacobian matrix. The relationship between the body-expressed angular velocity vector and the Euler angle time derivatives is presented. Finally, forward propagation for the time derivative of the extended velocity vectors is discussed.

In configuration analysis, a coordinate frame was attached to each rigid body of the system. Each frame α has two components of velocity with respect to the inertial frame. The first component is the linear velocity vector of the frame origin $\overrightarrow{\boldsymbol{v}_\alpha}$. The second component is the angular velocity vector of the frame $\overrightarrow{\boldsymbol{\omega}_\alpha}$. Concatenating these components, an extended velocity vector of frame α is defined as $\overrightarrow{\mathbf{V}_\alpha} \triangleq \left[\overrightarrow{\boldsymbol{v}_\alpha}, \overrightarrow{\boldsymbol{\omega}_\alpha}\right]$. This velocity vector can be expressed in any coordinate frame. In this thesis, the velocity vector of a frame is also expressed in that frame unless otherwise stated. As a result, for notational brevity, the extended velocity vector of frame α expressed in α is defined as follows,

$$\mathbf{V}_\alpha \triangleq \left[\boldsymbol{v}_\alpha^T, \boldsymbol{\omega}_\alpha^T\right]^T \in \Re^6. \tag{3.4}$$

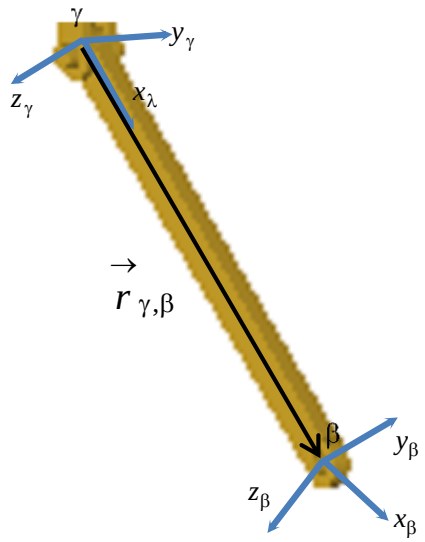

Figure 3.3: A rigid body with two body-fixed coordinate frames

Generalized Transformation

In order to facilitate the velocity analysis, a generalized transformation is described here. Consider two body-attached coordinate frames namely γ and β assigned to a rigid body. This is shown in Fig. (3.3). Given $\mathbf{V}_\gamma = \left[\boldsymbol{v}_\gamma^T, \boldsymbol{\omega}_\gamma^T\right]^T$, and the position vector of β w.r.t γ i.e. $\overrightarrow{\mathbf{r}}_{\gamma,\beta}$, one can derive $\mathbf{V}_\beta$ as follows.

Being attached to the same rigid body, γ and β experience the same angular velocity i.e. $\overrightarrow{\boldsymbol{\omega}_\gamma} = \overrightarrow{\boldsymbol{\omega}_\beta}$. As a result, the following equation holds,

$$\boldsymbol{\omega}_\beta = \mathbf{R}_\gamma^\beta \, \boldsymbol{\omega}_\gamma \tag{3.5}$$

To derive $\boldsymbol{v}_\beta$, the relative velocity equation is employed as follows,

$$\overrightarrow{\boldsymbol{v}_\beta} = \overrightarrow{\boldsymbol{v}_\gamma} + \overrightarrow{\boldsymbol{\omega}_\gamma} \times \overrightarrow{\mathbf{r}}_{\gamma,\beta} \tag{3.6}$$

Equation (3.6) is a vector equation which can be described in any frame. Expressing (3.6) in β yields,

$$\boldsymbol{v}_\beta = \mathbf{R}_\gamma^\beta \left(\boldsymbol{v}_\gamma - \mathbf{S}\left(\mathbf{r}_{\gamma,\beta}^\gamma\right) \boldsymbol{\omega}_\gamma \right) \tag{3.7}$$

in which $\mathbf{R}_\beta^\gamma$ is the rotation matrix describing the orientation of β w.r.t γ, $\mathbf{S}(\cdot)$ is the skew operator and $\boldsymbol{r}_{\gamma,\beta}^\gamma$ is the position vector of β with respect to and expressed in γ. Equations (3.5) and (3.7) can be rewritten in the following compact form,

$$\boldsymbol{V}_\beta = \mathbf{U}_\beta^{\gamma^T} \boldsymbol{V}_\gamma \tag{3.8}$$

where $\mathbf{U}_\beta^\gamma \in \Re^{6\times 6}$ is a generalized transformation matrix between frames γ and β with the following definition,

$$\mathbf{U}_\beta^\gamma \triangleq \begin{bmatrix} \mathbf{R}_\beta^\gamma & \mathbf{0}_{3\times 3} \\ \mathbf{S}\big(\boldsymbol{r}_{\gamma,\beta}^\gamma\big)\, \mathbf{R}_\beta^\gamma & \mathbf{R}_\beta^\gamma \end{bmatrix} \tag{3.9}$$

In the above derivation, the skew symmetric property of $\mathbf{S}$ has been used. This generalized transformation will be widely used in the kinematics/dynamics derivations as well as control design.

Generalized Velocities

Velocity kinematics deals with the derivation of the velocities of all system bodies. The velocity space of a system of N_b rigid bodies is a $6N_b$ dimensional space. However, the system velocities do not cover the entire $6N_b$ space due to the physical constraints between bodies. A minimal set of independent velocity variables which suffices to determine the velocities of all system bodies is called the system generalized velocities. The number of a system generalized velocities is equal to the system DOF. For the UAV-Manipulator, the

generalized velocity vector is defined as,

$$\mathcal{V} \triangleq \left[\boldsymbol{V}_{B_0}^T, \dot{\boldsymbol{q}}_m^T, \dot{\boldsymbol{q}}_p^T\right]^T \in \Re^{N_d}, \tag{3.10}$$

where $\boldsymbol{V}_{B_0} \triangleq \left[\boldsymbol{v}_{B_0}^T, \boldsymbol{\omega}_{B_0}^T\right]^T \in \Re^6$, denotes the linear/angular velocity vector of frame B_0, $\dot{\boldsymbol{q}}_m \triangleq [\dot{q}_{m_1}, \cdots, \dot{q}_{m_n}]^T \in \Re^n$ is the vector of manipulator joint velocities and $\dot{\boldsymbol{q}}_p \triangleq [\dot{q}_{p_1}, \cdots, \dot{q}_{p_m}]^T \in \Re^m$ is the vector of propeller velocities.

From a kinematics perspective, the UAV-Manipulator structure is composed of $m + 1$ open kinematic chains. This includes m similar chains consisting of the base and a propeller as well as a kinematic chain comprising the base and the robot manipulator. The latter starts from the base and ends with the held object. Given the generalized velocity vector in (3.10), the extended velocities of all system frames α can be determined by a forward velocity propagation performed in each of the kinematic chains.

Using the generalized transformation described before, the velocity propagation for a kinematic chain composed of the UAV (B_0) and the ith propeller (P_i) is performed as follows,

$$\boldsymbol{V}_{P_i} = \mathbf{U}_{P_i}^{{B_0}^T} \boldsymbol{V}_{B_0} + \dot{q}_{p_i} \boldsymbol{z}'$$

where $\boldsymbol{z}' \triangleq [0, 0, 0, 0, 0, 1]^T$. The first term in the above equation is due to the transformation of the B_0 velocity vector to P_i and the second term takes into account the extra velocity component of P_i i.e. the propeller speed.

For the kinematic chain composed of B_0, B_1, ..., and B_{n+1}, given $\mathbf{V}_{B_0}$ the velocity is propagated to B_1 first; given $\mathbf{V}_{B_1}$ the extended velocity of B_2 is determined next. Similarly, the velocity is propagated forward until $\mathbf{V}_{B_{n+1}}$ is found. One step velocity propagation in

this chain, from $\mathbf{V}_{B_i}$ to $\mathbf{V}_{B_{i+1}}$, is performed as follows,

$$\boldsymbol{V}_{B_{i+1}} = \mathbf{U}_{B_{i+1}}^{{B_i}^T} \boldsymbol{V}_{B_i} + \dot{q}_{m_{i+1}} \boldsymbol{z},$$

where $\boldsymbol{z} \triangleq [0,0,1,0,0,0]^T$ holds for manipulator prismatic joints, $\boldsymbol{z} \triangleq [0,0,0,0,0,1]^T$ holds for manipulator revolute joints. Finally, the last step propagation to determine the end-effector velocity $\mathbf{V}_{B_{i+1}}$ is performed using the following equation,

$$\boldsymbol{V}_{B_{n+1}} = \mathbf{U}_{B_{n+1}}^{{B_n}^T} \boldsymbol{V}_{B_n},$$

since B_n and B_{n+1} are both attached to the nth link.

The above discussion for velocity forward propagation in the system open kinematic chains can be summarized as follows,

$$\begin{aligned}
&\boldsymbol{V}_{B_{i+1}} = \mathbf{U}_{B_{i+1}}^{{B_i}^T} \boldsymbol{V}_{B_i} + \dot{q}_{m_{i+1}} \boldsymbol{z},\ i = 0, \ldots, n-1, \\
&\boldsymbol{V}_{B_{n+1}} = \mathbf{U}_{B_{n+1}}^{{B_n}^T} \boldsymbol{V}_{B_n}, \\
&\boldsymbol{V}_{P_i} = \mathbf{U}_{P_i}^{{B_0}^T} \boldsymbol{V}_{B_0} + \dot{q}_{p_i} \boldsymbol{z}',\quad i = 1, \ldots, m.
\end{aligned} \tag{3.11}$$

Generalized Jacobian Matrix

The Jacobian matrix of a robot manipulator expresses the mapping between the joint-space velocities and the end-effector velocities. Similarly, a generalized Jacobian matrix can be defined to express the relationship between the system generalized velocities and the extended velocity vectors of all system bodies/frames. To this end, an overall extended velocity vector containing linear/angular velocity vectors of all rigid bodies in the system

is defined as,

$$\mathcal{V}_e \triangleq \left[\boldsymbol{V}_{B_0}^T, \cdots, \boldsymbol{V}_{B_{n+1}}^T, \boldsymbol{V}_{P_1}^T, \cdots, \boldsymbol{V}_{P_m}^T\right]^T \in \Re^{6N_b}. \tag{3.12}$$

The forward velocity propagation equations (3.11) can be used to determine the extended velocity vector of all frames in terms of the system generalized velocity vector $\mathcal{V}$. The results are summarized in the following forward kinematics equation,

$$\mathcal{V}_e = \mathcal{J}_e \mathcal{V}, \tag{3.13}$$

where the generalized Jacobian matrix $\mathcal{J}_e \in \Re^{6N_b \times N_d}$ is given by,

$$\mathcal{J}_e = \begin{bmatrix} \mathcal{J}_{11_{6(n+2)\times 6}} & \mathcal{J}_{12_{6(n+2)\times n}} & \mathbf{0}_{6(n+2)\times m} \\ \mathcal{J}_{21_{6m\times 6}} & \mathbf{0}_{6m\times n} & \mathcal{J}_{23_{6m\times m}} \end{bmatrix} \tag{3.14}$$

where,

$$\mathcal{J}_{11} = \begin{bmatrix} \mathbf{I}_{6\times 6} & \mathbf{U}_{B_1}^{B_0} & \cdots & \mathbf{U}_{B_{n+1}}^{B_0} \end{bmatrix}^T,$$

$$\mathcal{J}_{12} = \begin{bmatrix} 0_{6\times 1} & 0_{6\times 1} & 0_{6\times 1} & \cdots & 0_{6\times 1} \\ \boldsymbol{z} & 0_{6\times 1} & 0_{6\times 1} & \cdots & 0_{6\times 1} \\ {\mathbf{U}_{B_2}^{B_1}}^T \boldsymbol{z} & \boldsymbol{z} & 0_{6\times 1} & \cdots & 0_{6\times 1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ {\mathbf{U}_{B_n}^{B_1}}^T \boldsymbol{z} & {\mathbf{U}_{B_n}^{B_2}}^T \boldsymbol{z} & {\mathbf{U}_{B_n}^{B_3}}^T \boldsymbol{z} & \cdots & \boldsymbol{z} \\ {\mathbf{U}_{B_{n+1}}^{B_1}}^T \boldsymbol{z} & {\mathbf{U}_{B_{n+1}}^{B_2}}^T \boldsymbol{z} & {\mathbf{U}_{B_{n+1}}^{B_3}}^T \boldsymbol{z} & \cdots & {\mathbf{U}_{B_{n+1}}^{B_n}}^T \boldsymbol{z} \end{bmatrix},$$

$$\mathcal{J}_{21} = \begin{bmatrix} \mathbf{U}_{P_1}^{B_0} & \mathbf{U}_{P_2}^{B_0} & \cdots & \mathbf{U}_{P_m}^{B_0} \end{bmatrix}^T,$$

$$\mathcal{J}_{23} = \begin{bmatrix} \boldsymbol{z}' & 0_{6\times 1} & \cdots & 0_{6\times 1} \\ 0_{6\times 1} & \boldsymbol{z}' & \cdots & 0_{6\times 1} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{6\times 1} & 0_{6\times 1} & \cdots & \boldsymbol{z}' \end{bmatrix}.$$

Equations (3.13) and (3.14) are a compact representation of the entire UAV-Manipulator velocity analysis.

Angular Velocity vs Euler Angle Rates

In Section 3.1.2, the Euler angles ($\boldsymbol{E}_{I,B_0}$) were employed as part of the generalized coordinates to represent the UAV orientation w.r.t the inertial frame. However, in this section, the body-expressed angular velocity of the UAV ($\boldsymbol{\omega}_{B_0}$) was introduced as part of the generalized velocity vector. In the subsequent chapters, the relationship between the Euler angle rates and body-expressed angular velocity is needed for control design. The derivation of

this relationship is explained in Appendix A.1 and the result is presented here. The mapping between the UAV body-expressed angular velocity $\boldsymbol{\omega}_{B_0}$ and the UAV Euler angle rates $\dot{\boldsymbol{E}}_{I,B_0}$ is given by,

$$\boldsymbol{\omega}_{B_0} = \mathbf{G}\left(\boldsymbol{E}_{I,B_0}\right) \dot{\boldsymbol{E}}_{I,B_0} \tag{3.15}$$

where,

$$\mathbf{G}\left(\boldsymbol{E}_{I,B_0}\right) = \begin{bmatrix} 1 & 0 & -\sin\left(\Theta_y\right) \\ 0 & \cos\left(\Phi_x\right) & \cos\left(\Theta_y\right)\sin\left(\Phi_x\right) \\ 0 & -\sin\left(\Phi_x\right) & \cos\left(\Phi_x\right)\cos\left(\Theta_y\right) \end{bmatrix} \tag{3.16}$$

Time Derivative of the Extended Velocities

In the next section, where dynamics of the system is studied, the extended velocity vectors of all the system frames as well as the time derivative of these vectors are required. Let $\dot{\mathcal{V}}$ be the time derivative of the generalized velocity vector defined by,

$$\dot{\mathcal{V}} \triangleq \left[\dot{\boldsymbol{V}}_{B_0}^T, \ddot{\boldsymbol{q}}_m^T, \ddot{\boldsymbol{q}}_p^T\right]^T \in \Re^{N_d}, \tag{3.17}$$

where $\dot{\boldsymbol{V}}_{B_0}$ is the time derivative of $\boldsymbol{V}_{B_0}$, $\ddot{\boldsymbol{q}}_m \triangleq \left[\ddot{q}_{m_1}, \cdots, \ddot{q}_{m_n}\right]^T$ is the manipulator joint acceleration vector and $\ddot{\boldsymbol{q}}_p \triangleq \left[\ddot{q}_{p_1}, \cdots, \ddot{q}_{p_m}\right]^T$ is the propeller joint acceleration vector. Given $\dot{\mathcal{V}}$, similar to the forward velocity propagation procedure, a forward propagation can be performed to derive the time derivatives of all the extended velocity vectors $\dot{\boldsymbol{V}}_\alpha$. In another approach, one can take the time derivative of (3.13) as follows,

$$\dot{\mathcal{V}}_e = \dot{\mathcal{J}}_e \mathcal{V} + \mathcal{J}_e \dot{\mathcal{V}}, \tag{3.18}$$

where $\dot{\mathcal{V}}_e \triangleq \left[\dot{\boldsymbol{V}}_{B_0}^T, \cdots, \dot{\boldsymbol{V}}_{B_{n+1}}^T, \dot{\boldsymbol{V}}_{P_1}^T, \cdots, \dot{\boldsymbol{V}}_{P_m}^T\right]^T$ is the time derivative of the overall extended velocity vector and $\dot{\mathcal{J}}_e$ is the time derivative of the generalized jacobian matrix. Equation (3.18), which yields the time derivative of extended velocity vector for all system frames, will be used in future dynamics/control developments.

3.2 Robot Dynamics

In this section, the dynamics of the UAV-Manipulator are presented. An extended force/moment vector is defined and force/moment transformations are explained. The Newton-Euler formulation of rigid body dynamics is employed and the system equations of motion are derived. The inverse/forward dynamics of the system are discussed and a fundamental property of the system dynamics, i.e., under-actuation, is highlighted.

3.2.1 Force/Moment Transformation

Consider a pair of force and moment vectors namely $\overrightarrow{\boldsymbol{f}_\alpha}$ and $\overrightarrow{\boldsymbol{m}_\alpha}$ being applied at the origin of frame α. Concatenating these two, an extended force vector applied at α is defined as $\overrightarrow{\mathbf{F}}_\alpha \triangleq \left[\overrightarrow{\boldsymbol{f}_\alpha}, \overrightarrow{\boldsymbol{m}_\alpha}\right]$. Similar to the velocity vectors, all force/moment vectors in this thesis are expressed in the related body frame. As a result, for notational brevity, the following extended force vector is defined as the force/moment vector measured and expressed in α,

$$\mathbf{F}_\alpha \triangleq \left[\boldsymbol{f}_\alpha^T, \boldsymbol{m}_\alpha^T\right]^T \in \Re^6. \tag{3.19}$$

Consider two body frames γ and β attached to a common rigid body and assume the extended force vector $\mathbf{F}_\gamma$ is applied at the origin of α. It is then straightforward to show that

the static equivalent of this force vector measured and expressed in frame β is given by,

$$\mathbf{F}_\beta = \mathbf{U}_\gamma^\beta \mathbf{F}_\gamma, \tag{3.20}$$

where $\mathbf{U}_\gamma^\beta$ is the generalized transformation defined by (3.9). This further implies the duality of velocity and force transformations.

3.2.2 Rigid Body Dynamics in a Body Frame

The UAV-Manipulator is composed of multiple rigid bodies physically connected together. Each rigid body can be considered as a subsystem with its own dynamics which exchanges interaction force/moments with the neighboring subsystems. Following such a concept, one can start from the dynamics of a single body in the system and recursively derive the entire system dynamics by propagating the interactions between bodies. To this end, the central equations are the equations of motion of a rigid body in free motion which are presented in this section.

The dynamics equations for free rigid body motion in 3-dimensional space are of the following form when expressed in a body frame α,

$$\mathbf{M}_\alpha \dot{\boldsymbol{V}}_\alpha + \mathbf{C}_\alpha (\boldsymbol{\omega}_\alpha) \boldsymbol{V}_\alpha + \boldsymbol{G}_\alpha = \boldsymbol{F}_\alpha^*, \tag{3.21}$$

where $\mathbf{M}_\alpha \in \Re^{6\times 6}$ is a constant inertia matrix, $\mathbf{C}_\alpha \in \Re^{6\times 6}$ is a skew-symmetric matrix being a function of the angular velocity of frame α i.e $\boldsymbol{\omega}_\alpha$, and $\boldsymbol{G}_\alpha \in \Re^6$ is the gravity force/moment vector. Detailed expressions of $\mathbf{M}_\alpha$, $\mathbf{C}_\alpha$ and $\boldsymbol{G}_\alpha$ are found in [20]. Moreover, the force/moment vector $\boldsymbol{F}_\alpha^* \in \Re^6$ is the net force/moment applied to the body, excluding the gravity effects, measured and expressed in frame α.

Remark. *Motion equations here are expressed in a body frame α which is not necessarily attached to the body center of mass. In fact, the frames in this thesis are fixed to one end point of each body where there is a joint. This facilitates the force/moment propagation among the bodies and reduces the number of force transformations.*

3.2.3 Linear-In-Parameter Representation

A desirable property of the motion equations (3.21) is the so called linearity-in-parameter. This property will be used later in the design of parameter adaption laws. The motion equations are written in the linear-in-parameter as follows,

$$\mathbf{M}_\alpha \dot{\boldsymbol{V}}_\alpha^{(r)} + \mathbf{C}_\alpha(\boldsymbol{\omega}_\alpha)\boldsymbol{V}_\alpha^{(r)} + \boldsymbol{G}_\alpha = \mathbf{Y}_\alpha \boldsymbol{\theta}_\alpha, \tag{3.22}$$

where $\mathbf{Y}_\alpha \in \Re^{6\times 13}$ is a regressor matrix and $\boldsymbol{\theta}_\alpha \in \Re^{13}$ is a vector of unknown parameters for the body to which α is assigned. The regressor matrix $\mathbf{Y}_\alpha$ and the parameter vector θ_α can be found in [20].

Remark. *Note that in (3.22), the motion equations are written using $\boldsymbol{V}_\alpha^{(r)}$ instead of $\boldsymbol{V}_\alpha$. $\boldsymbol{V}_\alpha^{(r)}$ is a design vector called "Required Velocity" which will be defined in the next chapter.*

3.2.4 Inverse Dynamics Analysis

This section deals with the derivation of the actuation force/moments given the kinematic analysis. This is referred to as the inverse dynamics problem. Given the motion variables of all bodies obtained from forward velocity propagation, one can begin with the dynamics of the last body in each kinematic chain, i.e., held object/propeller and proceed towards the base to derive the motion equations of all bodies through a backward force recursion.

The dynamics equations for free rigid body motion of the object, manipulator links, propellers and the base platform expressed in their respective body frames are all given by (3.21) where $\alpha \in \{B_0, \cdots, B_{n+1}, P_1, \cdots, P_m\}$. In the remainder of this section, these equations are developed for all system bodies in a recursive approach.

Note that the velocity analysis is performed beforehand and all extended velocity vectors and their time derivatives are determined. In the upcoming free body diagrams, the gravity effects are not included since they have been taken into account in the left-hand side of (3.21).

Dynamics of the Object

Fig. 3.4 depicts the free body diagram of the held object. Its motion is described by,

$$\mathbf{M}_{B_{n+1}} \dot{\boldsymbol{V}}_{B_{n+1}} + \mathbf{C}_{B_{n+1}} \left(\boldsymbol{\omega}_{B_{n+1}}\right) \boldsymbol{V}_{B_{n+1}} + \boldsymbol{G}_{B_{n+1}} = \boldsymbol{F}^*_{B_{n+1}}. \tag{3.23}$$

For a given configuration, all the left-hand side terms of (3.23) are known. As a result, the net force $\boldsymbol{F}^*_{B_{n+1}}$ applied at B_{n+1} can be determined. Moreover, considering $\boldsymbol{F}_{B_{n+1}}$ as the only extended force vector applied to the object, the following equation holds,

$$\boldsymbol{F}_{B_{n+1}} = \boldsymbol{F}^*_{B_{n+1}}. \tag{3.24}$$

Dynamics of the Manipulator Links

Given $\boldsymbol{F}_{B_{n+1}}$, one can proceed backwards and consider the motion equation of the nth link and determine $\boldsymbol{F}_{B_n}$. This recursive approach is followed until the UAV base is reached and all the interaction force/torques among the links are derived. The following equation

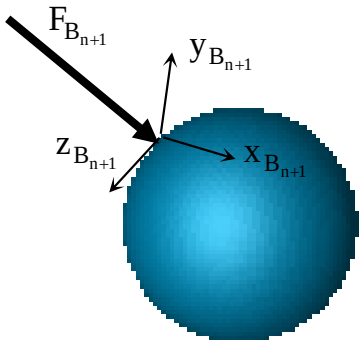

Figure 3.4: Free body diagram of the held object

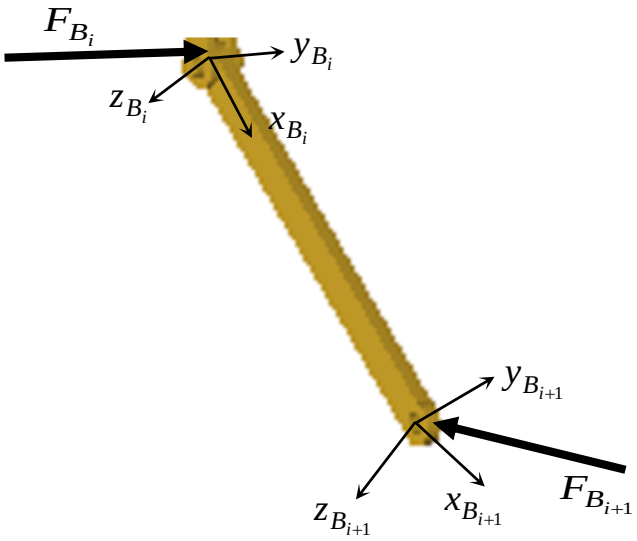

Figure 3.5: Free body diagram of the ith link

describes the motion of the ith manipulator link in this kinematic chain,

$$\mathbf{M}_{B_i}\dot{\boldsymbol{V}}_{B_i} + \mathbf{C}_{B_i}(\boldsymbol{\omega}_{B_i})\boldsymbol{V}_{B_i} + \boldsymbol{G}_{B_i} = \boldsymbol{F}^*_{B_i}, \quad i = n, \cdots, 1. \tag{3.25}$$

Fig. 3.5 shows the free body diagram of the ith link which is subject to two forces at its ends. Given the net force exerted on this link by (3.25), the following equation holds,

$$\boldsymbol{F}_{B_i} = \boldsymbol{F}^*_{B_i} + \mathbf{U}^{B_i}_{B_{i+1}}\boldsymbol{F}_{B_{i+1}}, \quad i = n, \cdots, 1 \tag{3.26}$$

The above equation is used recursively and all the extended interaction forces among the manipulator links are derived. The last step of this recursion yields the extended interaction force between the UAV base and the manipulator.

Dynamics of the Propellers

Similarly, the motion of each propeller can be described by the following equation,

$$\mathbf{M}_{P_i}\dot{\boldsymbol{V}}_{P_i} + \mathbf{C}_{P_i}(\boldsymbol{\omega}_{P_i})\boldsymbol{V}_{P_i} + \boldsymbol{G}_{P_i} = \boldsymbol{F}^*_{P_i}, \quad i = 1, \cdots, m. \tag{3.27}$$

Fig. 3.6 shows the free body diagram of the ith propeller. It is subjected to two extended force vectors at P_i. $\mathbf{F}_{P_i}$ is the interaction force/torque between the propeller and the UAV base whereas $\mathbf{F}^{aero}_{P_i}$ is the extended force vector due to the external aerodynamic force/torque generated by the propeller.

The aerodynamic extended force vector includes a thrust force and a drag torque both of which are perpendicular to the propeller plane. The magnitudes of these two are proportional to the square of the propeller speed [82; 83; 84]. The aerodynamic extended force

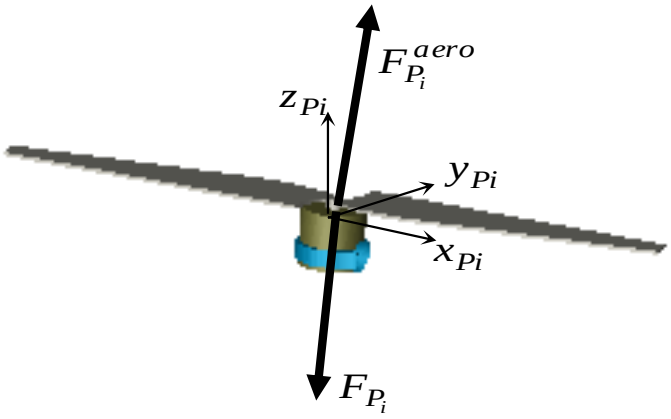

Figure 3.6: Free body diagram of the ith propeller

vector is then given by,

$$\boldsymbol{F}_{P_i}^{aero} = \left[0, 0, k_T \rho D^4, 0, 0, -k_Q \rho D^5 \, \mathrm{sgn}(\dot{q}_{p_i})\right]^T \dot{q}_{p_i}^2. \tag{3.28}$$

Here, ρ is the air density and D is the propeller diameter. Furthermore, k_T and k_Q are the propeller-dependent thrust and torque coefficients respectively. The net extended force vector $\boldsymbol{F}_{P_i}^*$ is determined by (3.27). Moreover, given the propeller speed, the aerodynamic extended force vector is known. As a result, the interaction force/torque between the propeller and the UAV base is given by,

$$\boldsymbol{F}_{P_i} = \boldsymbol{F}_{P_i}^* - \boldsymbol{F}_{P_i}^{aero}, \; i = 1, \cdots, m. \tag{3.29}$$

Dynamics of the UAV Base

The common rigid body between all the $m + 1$ kinematic chains is the UAV base. Having performed the backward force propagation in all the chains, one can proceed to the base

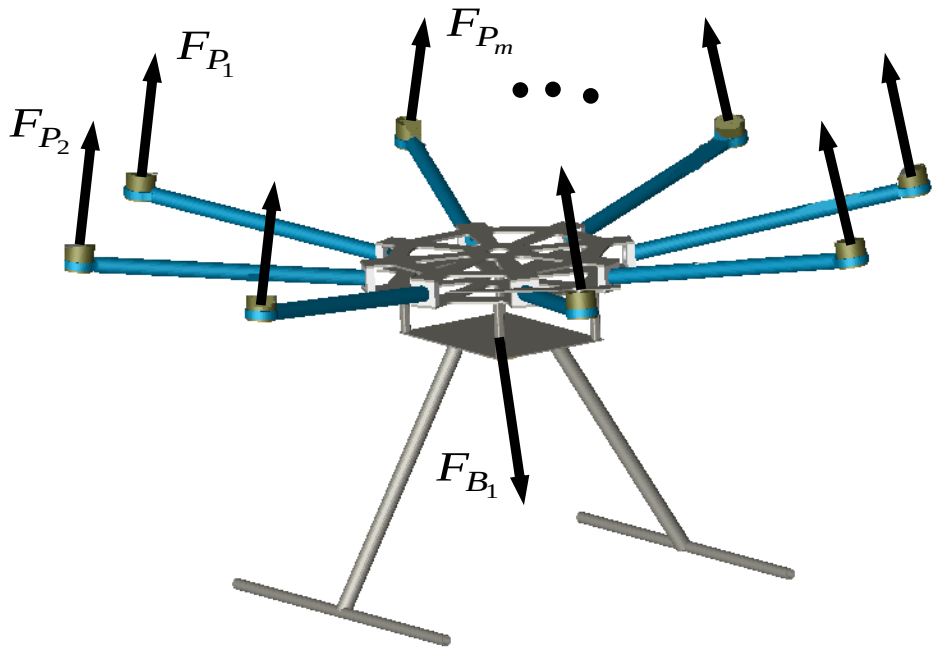

Figure 3.7: Free body diagram of the UAV base

dynamics. The following equation describes the UAV base motion and can be used to determine the net extended force vector exerted on the base at B_0,

$$\mathbf{M}_{B_0}\dot{\boldsymbol{V}}_{B_0} + \mathbf{C}_{B_0}(\boldsymbol{\omega}_{B_0})\boldsymbol{V}_{B_0} + \boldsymbol{G}_{B_0} = \boldsymbol{F}^*_{B_0}. \tag{3.30}$$

Fig. 3.7 depicts the free body diagram of the base. Given the net extended force by (3.30), the base-manipulator extended interaction force by (3.26), and the base-propeller extended interaction forces by (3.29), one can determine the extended force vector at frame B_0 as follows,

$$\boldsymbol{F}_{B_0} = \boldsymbol{F}^*_{B_0} + \mathbf{U}^{B_0}_{B_1}\boldsymbol{F}_{B_1} + \sum_{i=1}^{m}\mathbf{U}^{B_0}_{P_i}\boldsymbol{F}_{P_i} \tag{3.31}$$

Retrieving the Actuation Torques/Forces

Among all system DOFs, the ones introduced by the manipulator joints and the propellers, i.e., $\boldsymbol{q}_m$ and $\boldsymbol{q}_p$ are actuated. The actuation torque/force of each manipulator joint as well as the actuation torque of each propeller is derived by projecting the corresponding interaction force/moment vector to the related joint axis as,

$$\begin{aligned} \tau_{m_i} &= \boldsymbol{z}^T \boldsymbol{F}_{B_i},\ i = 1, \cdots, n, \\ \tau_{p_i} &= \boldsymbol{z'}^T \boldsymbol{F}_{P_i},\ i = 1, \cdots, m, \end{aligned} \tag{3.32}$$

where τ_{m_i} is the ith manipulator joint torque/force and τ_{p_i} is the actuation torque of the ith propeller. The remaining DOFs introduced by position/orientation of the UAV base are not actuated by any external force/torque. As a result, the system is under-actuated in six DOFs. This further implies that the extended force vector $\mathbf{F}_{B_0}$ derived from (3.31) is a zero vector.

Summary of the Inverse Dynamics

The inverse dynamics of the system can be summarized in the following steps. Note that the forward velocity analysis is performed beforehand and the extended velocity vectors of all bodies $\mathbf{V}_\alpha$ and their time derivatives $\dot{\mathbf{V}}_\alpha$ are obtained.

i - Net Force Calculation:

Equations (3.23), (3.25), (3.27), and (3.30) are used to determine the net force exerted on each body at its corresponding frame.

ii - Backward Force Propagation:

Equations (3.24), (3.26), (3.29), and (3.31) are used to find the extended interaction

forces as well as the external extended force at the base.

iii - Actuation Torques/Forces Derivation:

Equation (3.32) is used to find the manipulator actuation torques/forces as well as the propeller actuation torques.

3.2.5 Forward Dynamics Analysis

The forward dynamics deals with determination of the system motion for a given system actuation. Let $\mathcal{F}_e^*$ be an overall net extended force vector defined as,

$$\mathcal{F}_e^* \triangleq \left[\boldsymbol{F}_{B_0}^{*^T}, \cdots, \boldsymbol{F}_{B_{n+1}}^{*^T}, \boldsymbol{F}_{P_1}^{*^T}, \cdots, \boldsymbol{F}_{P_m}^{*^T}\right]^T \in \Re^{6N_b}. \tag{3.33}$$

Employing (3.13) and (3.18), equations (3.23), (3.25), (3.27), and (3.30) may be expressed in the following compact form,

$$\mathcal{F}_e^* = \mathcal{M}_e \mathcal{J}_e \dot{\mathcal{V}} + \mathcal{M}_e \dot{\mathcal{J}}_e \mathcal{V} + \mathcal{C}_e \mathcal{J}_e \mathcal{V} + \mathcal{G}_e, \tag{3.34}$$

in which $\mathcal{M}_e$, $\mathcal{C}_e$, and $\mathcal{G}_e$ are given by,

$$\begin{aligned}
\mathcal{M}_e &= diag\left(\mathbf{M}_{B_0}, \cdots, \mathbf{M}_{B_{n+1}}, \mathbf{M}_{P_1}, \cdots, \mathbf{M}_{P_m}\right) \in \Re^{6N_b \times 6N_b} \\
\mathcal{C}_e &= diag\left(\mathbf{C}_{B_0}, \cdots, \mathbf{C}_{B_{n+1}}, \mathbf{C}_{P_1}, \cdots, \mathbf{C}_{P_m}\right) \in \Re^{6N_b \times 6N_b} \\
\mathcal{G}_e &= \left[\boldsymbol{G}_{B_0}^T, \cdots, \boldsymbol{G}_{B_{n+1}}^T, \boldsymbol{G}_{P_1}^T, \cdots, \boldsymbol{G}_{P_m}^T\right]^T \in \Re^{6N_b}.
\end{aligned} \tag{3.35}$$

Finally, pre-multiplying (3.34) by $\mathcal{J}_e^T$ yields the forward dynamics equations of the entire system as follows,

$$\mathcal{M}\dot{\mathcal{V}} + \mathcal{C}\mathcal{V} + \mathcal{G} = \begin{bmatrix} 0_{6\times 1} \\ \boldsymbol{\tau_m} \\ \boldsymbol{\tau_p} \end{bmatrix}, \tag{3.36}$$

where $\boldsymbol{\tau_m} \triangleq [\tau_{m_1}, \cdots, \tau_{m_n}]^T$ is the manipulator actuator forces/torques, $\boldsymbol{\tau_p} \triangleq [\tau_{p_1}, \cdots, \tau_{p_m}]^T$ is the propeller actuator torques and $\mathcal{M}, \mathcal{C}, \mathcal{G}$ are given by,

$$\begin{aligned} \mathcal{M} &= \mathcal{J}_e^T \, \mathcal{M}_e \, \mathcal{J}_e \in \Re^{N_d \times N_d} \\ \mathcal{C} &= \mathcal{J}_e^T \, \mathcal{M}_e \, \dot{\mathcal{J}}_e + \mathcal{J}_e^T \, \mathcal{C}_e \, \mathcal{J}_e - \mathcal{C}^{aero} \in \Re^{N_d \times N_d} \\ \mathcal{G} &= \mathcal{J}_e^T \, \mathcal{G}_e \in \Re^{N_d}. \end{aligned} \tag{3.37}$$

where,

$$\mathcal{C}^{aero} = \begin{bmatrix} \mathbf{0}_{6\times(6+n)} & \mathbf{U}_{P_1}^{B_0} \boldsymbol{T}_{P_1} \dot{q}_{p_1} & \cdots & \mathbf{U}_{P_m}^{B_0} \boldsymbol{T}_{P_m} \dot{q}_{p_m} \\ \mathbf{0}_{n\times(6+n)} & 0_{n\times 1} & \cdots & 0_{n\times 1} \\ \mathbf{0}_{m\times(6+n)} & z'^T \boldsymbol{T}_{P_1} \dot{q}_{p_1} & \cdots & z'^T \boldsymbol{T}_{P_m} \dot{q}_{p_m} \end{bmatrix},$$

and $\boldsymbol{T}_{P_i} = [0, 0, k_t, 0, 0, -k_d \, \mathrm{sgn}(\dot{q}_{p_i})]^T$ for $i = 1, \cdots, m$.

Equation (3.36) describes the entire system dynamics in a compact form. This equation can be used to solve both the inverse and the forward dynamics problems. For the inverse dynamics, given the system configuration, generalized velocities, and the time derivatives of the generalized velocities, the actuation torques can be determined by a simple evaluation. For the forward dynamics, given the actuation torques and the system initial state, Equation (3.36) can be numerically solved for $\dot{\mathcal{V}}$ at each simulation step time. Numerical integration would then yield the system generalized velocities and coordinates for the next

simulation cycle.

Remark. *The $0_{6\times1}$ in (3.36) is due to the system dynamics being under-actuated in six DOFs since the external force/torque vector applied to the base namely $\boldsymbol{F}_{B_0}$ is zero. This under-actuation is a fundamental property of the system dynamics which makes the system control design significantly more challenging.*

Chapter 4

Introduction to the Virtual Decomposition Control

This thesis uses the Virtual Decomposition Control (VDC) method to develop control algorithms for the UAV-Manipulator system. The purpose of this chapter is to briefly explain the concept of the VDC method. The reader is referred to [20] for the details of this control design methodology.

Inspired by the modular and recursive nature of the Newton-Euler formulation of the multi-body inverse dynamics, the VDC method virtually decomposes the entire multi-body mechanical system to single rigid body units. It then develops local control and adaptation laws for each unit. Finally, a recursive propagation of the local control forces yields the actual system control inputs.

This chapter is organized as follows. First, the terminology of required generalized velocities is described and required velocity propagation is discussed. Local control and adaptation laws in the VDC approach are then described. Finally, the required force propagation and control inputs derivation are presented.

4.1 Required Generalized Velocities

The notion of a "Required Velocity" in VDC approach is different from a common desired velocity or reference trajectory. The required velocity in VDC is a control design variable such that if followed by the corresponding actual velocity, then position control is achieved. Therefore, in a general form, a required velocity consists of both the desired velocity and one or more terms that are related to position errors.

One of the main control objectives in robotics is position control in all system DOFs. Given this and the above description of the required velocity, the first step in VDC approach is to design a set of required velocities for all system DOFs. Here, these are referred to as the required generalized velocities. Once the actual generalized velocities of the system follow the corresponding required ones, the position tracking in all system DOFs is guaranteed.

4.2 Required Velocity Propagation

Having the required velocities of all system DOFs and similar to the velocity analysis in Chapter 3, one can perform a forward velocity propagation through all the system frames. This starts from the UAV base and ends with the manipulator end-effector or the UAV propellers.

The forward propagation of the required velocities is based on the system kinematic constraints. This ensures that the generalized velocities of the system follow the respective required ones as long as each body in the system follows its respective required velocity. This simple result plays a central role in the concept of system virtual decomposition. Decomposition of the system allows for individual control design of each rigid body in the

system such that it follows its respective required velocity. The convergence of velocity errors for each body is guaranteed by ensuring it is "Virtually Stable" with respect to its local control law. Once all bodies are forced to follow their corresponding required velocities, the actual control objectives are achieved successfully. First, the generalized velocity vector is guaranteed to follow the required generalized velocity vector. Next, position tracking in all system DOFs is obtained.

4.3 Virtual Decomposition and Local Control/Adaptation Laws

Performing the required velocity propagation to determine the required velocity of each body sets the stage for the virtual decomposition of the multi-body system. The entire system in this phase is virtually decomposed to individual rigid bodies through the so called "Virtual Cutting Points". These points are the mechanical joints at which two rigid bodies are physically connected.

Having the required velocity of each system body determined, the VDC develops local control and adaptation laws for each body individually. Through the virtual decomposition, each body is isolated from its neighboring bodies. This isolated subsystem is desired to track its required velocity derived in the previous step. In order to achieve this goal, the net extended force vector applied at this body is considered as its local/virtual control input. Model-based control and adaptation laws are designed for this body based on the inverse dynamics equations of a rigid-body in free three-dimensional motion. The remainder of this section describes these local control and adaptation laws.

4.3.1 Local Control Law

For an isolated rigid body with a body-fixed frame α, an extended velocity vector $\mathbf{V}_\alpha$ and a required extended velocity vector $\mathbf{V}_\alpha^{(r)}$, a local control law is designed as follows,

$$\boldsymbol{F}_\alpha^{*(r)} = \mathbf{Y}_\alpha \hat{\boldsymbol{\theta}}_\alpha + \mathbf{K}_\alpha \left(\boldsymbol{V}_\alpha^{(r)} - \boldsymbol{V}_\alpha \right), \tag{4.1}$$

where $\boldsymbol{F}_\alpha^{*(r)} \in \Re^6$ is the required net extended force vector applied at frame α, $\mathbf{K}_\alpha \in \Re^{6\times 6}$ is a constant positive definite design matrix which quantifies the feedback compensation term in the control input, $\mathbf{Y}_\alpha$ is the regressor matrix defined in (3.22), and $\hat{\boldsymbol{\theta}}_\alpha \in \Re^{13}$ is an estimation of the vector of uncertain parameters for the body.

The model-based feedforward compensation term of the control law (4.1) satisfies the following equation where the linear-in-parameter property of the rigid body dynamics is employed,

$$\hat{\mathbf{M}}_\alpha \dot{\boldsymbol{V}}_\alpha^{(r)} + \hat{\mathbf{C}}_\alpha \left(\boldsymbol{\omega}_\alpha \right) \boldsymbol{V}_\alpha^{(r)} + \hat{\boldsymbol{G}}_\alpha = \mathbf{Y}_\alpha \hat{\boldsymbol{\theta}}_\alpha, \tag{4.2}$$

here $\hat{\mathbf{M}}_\alpha \in \Re^{6\times 6}$, $\hat{\mathbf{C}}_\alpha \in \Re^{6\times 6}$, and $\hat{\boldsymbol{G}}_\alpha \in \Re^6$ are the estimations of the matrices defined in (3.21). Note that the feedback compensation term of the control law (4.1) includes both velocity and position feedback due to the definition of the required velocity variables.

4.3.2 Local Adaptation Law

For an isolated body with a body-fixed frame α, the vector of uncertain parameters $\hat{\boldsymbol{\theta}}_\alpha$ is updated according to the following adaptation law,

$$\dot{\hat{\boldsymbol{\theta}}}_\alpha = \boldsymbol{\Gamma}_\alpha^{-1} \mathbf{Y}_\alpha^T \left(\boldsymbol{V}_\alpha^{(r)} - \boldsymbol{V}_\alpha \right) \tag{4.3}$$

where $\boldsymbol{\Gamma}_\alpha \in \Re^{13\times 13}$ is a positive definite design matrix and determines the adaptation rate of $\hat{\boldsymbol{\theta}}_\alpha$. Equations (4.1) and (4.3) describe the local control design for any rigid body in the system. These are used to determine the required net forces applied to all system bodies in order to achieve required velocity tracking and position error convergence.

Remark. *A powerful feature of the VDC approach is decentralization of the parameter adaptation for the system uncertain parameters. Unlike the Lagrangian-based control methods where the entire system dynamics is written in the linear-in-parameter form, VDC parameterizes the dynamics of each body individually and develops adaptation laws for each body independently. This not only helps to avoid the complexity of parameterization of the entire system dynamics, but also allows for enabling/disabling parameter adaptation for individual bodies.*

Remark. *The entries of $\hat{\boldsymbol{\theta}}_\alpha$ should be lower/upper bounded in order to avoid parameter drift which may occur due to unmodeled disturbances. One method is to simply force bounds on the entries of the uncertain parameter vector. In another approach, the so-called projection functions can be employed to disable the integral action in case the integration results in a drift from the acceptable parameter bounds [85].*

4.4 Required Force Propagation and Control Inputs

The last step in VDC is to determine the required interaction force/torque among all bodies in order to derive the actual system control inputs. Given the required net extended force applied at each body and similar to the inverse dynamics analysis presented in Chapter 3, a required force propagation is performed recursively. These recursions begin with the dynamics of the last rigid body in each kinematic chain and end with the UAV base. Following this procedure yields the required interaction extended force vectors among all bodies. Finally, a projection of the required extended force vector onto the respective actuation axis yields the required actuation (control input) at each joint.

In summary, the control design based on the VDC approach involves four major steps. First, the required generalized velocity vector is designed. Second, a forward propagation is performed to obtain the required velocities of all bodies. Third, the entire system is virtually decomposed and model-based control/adaptation laws are designed for all system bodies. Fourth, a recursive propagation is performed for the required forces and the control inputs are derived.

Remark. *Among the steps involved in VDC implementation, the design of local control/adaptation laws for all bodies in the system can be done simultaneously. As a result, the third step of the VDC allows for parallel computing which reduces the computation time of the algorithm. This is especially beneficial for systems with high complexity and DOFs.*

Remark. *Using a Lyapunov analysis, it can be proven that the VDC approach guarantees the stability of the entire closed loop system and the convergence of the trajectory tracking errors [20].*

Chapter 5

Control Design Strategies

This chapter briefly presents different strategies for the motion control of the UAV-Manipulator system. The system operation during the execution of a certain task is discussed first. The effects of under-actuation on the system motion control are explained next. Finally, four different possible scenarios for the system operation are considered and the control variables are defined in each case.

5.1 UAV-Manipulator Operation During a Mission

In general, the UAV-Manipulator system may perform two types of tasks. In some applications, the purpose is to interact with the environment such as exerting a force to press a button. In other applications, the UAV-Manipulator may be used for pick and place tasks. In both cases and from a practical point of view, the system operation involves two phases. First, the UAV-Manipulator is commanded to move towards a target where the target point enters the relative workspace of the manipulator w.r.t the UAV. Second, the manipulator is commanded to perform an assigned task. Fig. 5.1 illustrates the operation concept.

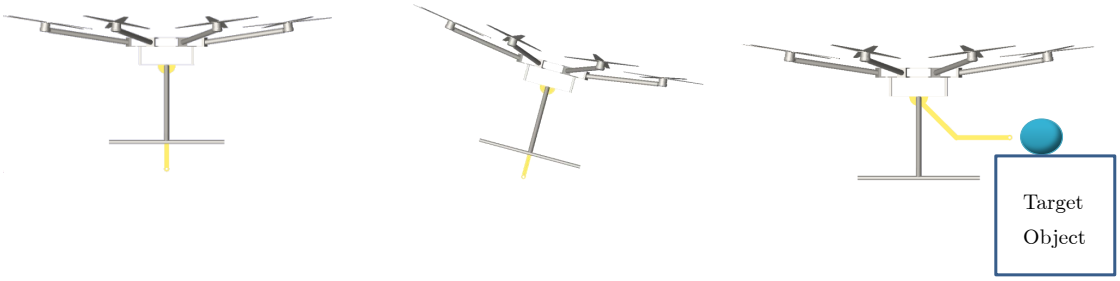

Figure 5.1: Illustration of a UAV-Manipulator operation in approaching/grasping a target object

Note that this does not necessarily mean that the UAV and the manipulator are commanded individually. For agile operation for instance, the object grasping phase may occur while the UAV is also undergoing a motion. This is all addressed in the system trajectory planning which is not within the scope of this research. Here, it is assumed that the desired references for the independent system DOFs are given. These desired motion variables can be determined by a trajectory planning algorithm or a human operator.

Regardless of how the independent desired trajectories are obtained, the choice of variables/DOFs to be controlled affects the design of the control algorithm. In one approach, the manipulator joint positions are among the control variables whereas in another approach the manipulator end-effector position/orientation are selected to be controlled directly. In both cases, however, there are more control variables to take into account. These are related to the UAV DOFs. The combination of the UAV and the Manipulator DOFs to be controlled gives rise to four possible scenarios which will be described later in this chapter.

5.2 System Under-actuation

In Chapter 3, it was shown that the system dynamics is under-actuated. In such a case, not all system DOFs may be controlled independently since there are fewer actuations than the DOFs. As a result, trajectory planning for some DOFs can be performed independently whereas the remaining DOFs trajectories become dependent due to insufficient actuation. While the independent trajectories are given to the control system by the trajectory planner/human operator, the dependent ones are to be determined within the control algorithm, consistent with the motion of the independent degrees. This is one of the major theoretical complications of the model-based motion control for the UAV-Manipulator.

5.2.1 Second-order Motion Constraints

The system under-actuation results in a dependency of the system motion variables and imposes constraints on the system generalized accelerations. Although these constraints involve the generalized accelerations, velocities, and coordinates, they are non-integrable. In other words these would not explicitly establish any constraints on the system velocity/configuration space but on the system acceleration space. These types of constraints are named "Second-order Nonholonomic" constraints in the literature [86; 87; 88].

These motion constraints should be taken into account in the design of a motion control system for the UAV-Manipulator. To this end, the system DOFs are divided into two sets; an independent set and a dependent set. The desired trajectories for the independent set of DOFs are given to the control system. To derive the required motion variables for the dependent set of DOFs, the nonholonomic motion constrains are fully developed and employed. This approach ensures that the motion variables for all system DOFs are consistent with the system motion constraints.

5.2.2 Independent and Dependent DOFs

To accomplish a certain mission, it is desired to fully control the base position/orientation and manipulator joints. Meanwhile, the orientation of each propeller has been considered as a DOF only to fully incorporate the propeller dynamics into the system model and control design; as a result, independent control of the propeller-related DOFs is of no interest. These DOFs can be employed to indirectly control the unactuated base through the external aerodynamic forces/torques applied to the propellers.

Moreover, all the propeller-produced thrust forces and drag torques have the same direction, i.e. z_{B_0}. This provides the base with a maximum of four indirect actuation namely a resultant thrust force in z_{B_0} direction and three moments around the axes of B_0. This actuation can be achieved by a proper arrangement of a minimum of four propellers which is realized in a standard quad-copter configuration. Increasing the number of propellers does not compensate for the lack of external actuation force of the base in x_{B_0} and y_{B_0} directions, but improves the UAV payload capacity and provides actuation redundancy in the actuated DOFs. Henceforth, the number of propellers m is assumed to be at least four to produce as much indirect actuation of the base as possible.

Given the discussion above, the base-manipulator subsystem with $6 + n$ DOFs may be actuated indirectly by four resultant aerodynamic force/torques and directly by n manipulator joint torques. As a result, this subsystem is still under-actuated in two DOFs, i.e. the base translation in x_{B_0} and y_{B_0} directions. This further implies that although desired, independent control of the base position/orientation and manipulator joints is not dynamically feasible. Motion control in two DOFs of the base-manipulator has to be relinquished in order to satisfy the nonholonomic constraints imposed on the system motion.

In summary, among all system DOFs and in all control strategies, $4 + n$ DOFs of the

system are to be controlled independently and the remaining $2 + m$ are employed to tackle the system under-actuation. During the rest of the thesis, the former are referred to as independent DOFs and the latter as dependent DOFs. Different choices of the independent DOFs results in different control strategies which will be discussed shortly in the remaining sections of this chapter.

5.2.3 Under-actuation Effects on VDC Design

As described in Chapter 4, control development based on VDC is carried out in two steps. First, the so-called "required" velocities of all bodies are designed such that if followed by the real velocities, convergence of the tracking errors is guaranteed. For this, the generalized required velocity vector $\mathcal{V}^{(r)}$ is designed and then a forward propagation yields the required velocity of each body. Second, given the required motions, the required interaction forces are designed such that the convergence of real velocities to the required ones is guaranteed.

For fully-actuated systems, the first step can be performed independently of the second step. In other words, any designed $\mathcal{V}^{(r)}$ can be tracked since there is sufficient actuation. For under-actuated systems, however, the two design steps become interdependent. In this case, while designing $\mathcal{V}^{(r)}$, the second-order nonholonomic motion constraints should also be considered. $\mathcal{V}^{(r)}$ should be designed such that its execution requires no forces/torques in the unactuated DOFs in each time instant.

In the subsequent chapters, a method is proposed as part of the control algorithm to handle the dependency of the two design steps in VDC due to the under-actuation.

5.3 Motion Control Strategies

As discussed in the previous section, from the $6 + n + m$ DOFs of the system, $4 + n$ DOFs can be controlled independently. Different choices of these $4 + n$ independent DOFs result in different system operations and control design. This section is concerned with proposing possible scenarios for system operation by discussing different choices of the independent set of DOFs.

As described before, one phase of the UAV-Manipulator operation is to move the system towards a target such that the target point is within the reachable workspace of the manipulator. Given the dynamic of the system in Chapter 3, the net thrust force generated by the propellers can be manipulated to move the entire system in the three dimensional space. This can be achieved by properly orienting and adjusting the magnitude of the net thrust force. This further reveals the dependency of the system motion in the inertial X-Y coordinates to the UAV orientation in roll and pitch. This is evident since the orientation of the net thrust force with respect to the inertial frame is expressed through the roll and pitch components of the UAV Euler angles.

The discussion above gives rise to two control approaches for the coarse movements of the system during its operation. In one approach, the control system may receive the UAV desired roll and pitch trajectories in which case a human operator or a higher-level feedback control loop is required to adjust the desired roll and pitch trajectories in order to move the system towards a target point. In a second approach, the control system may directly receive the UAV desired position trajectory in which case it is responsible to accordingly adjust roll and pitch in order to follow the desired position trajectory. While the former approach is more suitable for a human-in-the-loop mode of system operation, the latter allows for a fully autonomous control mode.

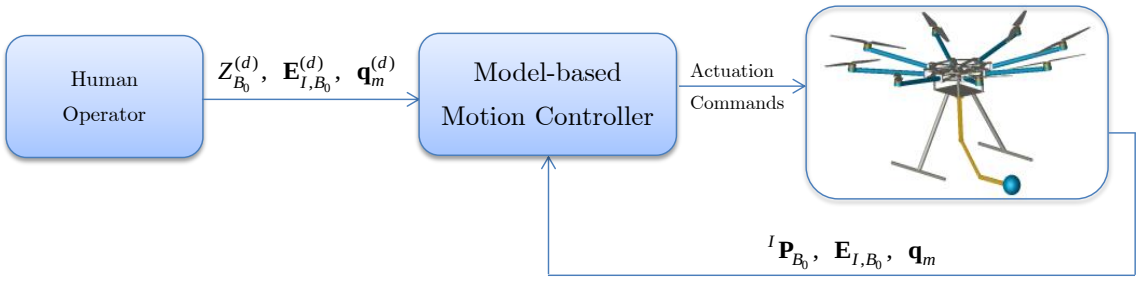

Figure 5.2: Control Scenario I: The UAV altitude and orientation as well as the manipulator joint positions are controlled independently

On the other hand, for the fine system movements where the robotic manipulator is performing a task, there are two possible approaches. In a first approach, the control system may receive the manipulator joint position trajectories where in a second approach, it may receive the desired position/orientation trajectories for the manipulator end-effector. Considering these different approaches for the system coarse and fine movements, four possible control scenarios are proposed and described below.

5.3.1 Control Scenario I

In the first control scenario, the UAV altitude and attitude as well as the manipulator joint positions constitute the independent set of DOFs. As a result, the reference trajectories for these system DOFs are provided to the control system. This control scenario is more suited to a human-in-the-loop mode of system operation. In this case, the human operator can indirectly control the X-Y coordinates of the UAV position in the inertial frame by adjusting the UAV reference roll and pitch angles. Fig. 5.2 shows a simplified block diagram of this control strategy.

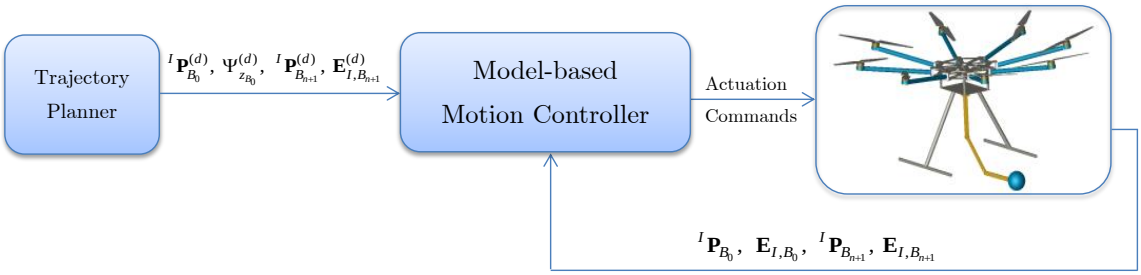

Figure 5.3: Control Scenario II: The UAV position and yaw orientation as well as the end-effector position and orientation are controlled independently

5.3.2 Control Scenario II

In the second control scenario, the control objective is to command the manipulator end-effector directly. In this case, the trajectory planning is performed for the UAV position and yaw orientation as well as the manipulator end-effector position and orientation. Since the UAV position is controlled directly, the motion controller would accordingly determine and execute the proper UAV roll and pitch orientations such that the reference UAV position is followed. This approach not only allows the system to operate in a fully autonomous mode, but also is convenient for executing a manipulation task and object grasping. Exact positioning of the end-effector is achievable in this case even if the UAV base position error does not converge but remains within an acceptable range. Fig. 5.3 illustrates this control strategy.

5.3.3 Control Scenario III

In this control strategy, the three coordinates of the UAV position, the UAV heading/yaw orientation as well as the manipulator joint positions are to be controlled independently. Similar to the second control scenario, the controller is responsible to accordingly adjust

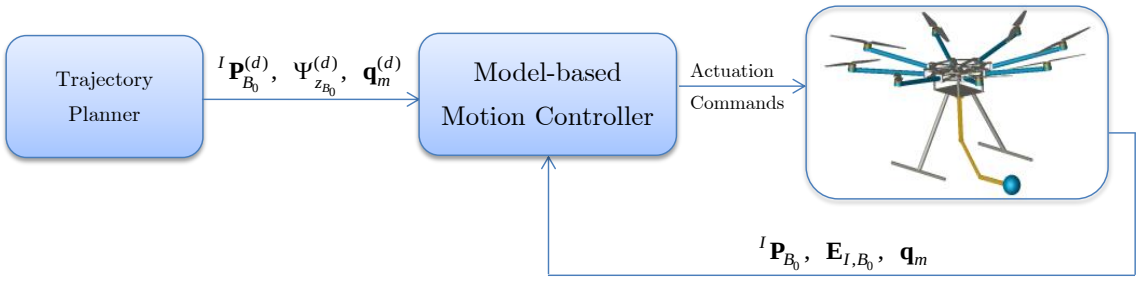

Figure 5.4: Control Scenario III: The UAV position and yaw orientation as well as the manipulator joint positions are controlled independently

the UAV roll and pitch orientations such that the trajectory tracking for the UAV XY coordinates is achieved. This scenario can also be used in a fully autonomous system operation mode since the UAV position trajectories can be designed offline. A simplified block diagram of this control strategy is depicted in Fig. 5.4.

5.3.4 Control Scenario IV

In the last control scenario, the control system receives the desired position/orientation of the end-effector as part of its reference inputs, similar to the second scenario. The UAV altitude and attitude are considered as the rest of system independent DOFs to be controlled directly. In such a case and similar to the first scenario, the roll and pitch reference inputs are used by a human operator to indirectly position the UAV in the horizontal plane. Fig. 5.5 shows a block diagram of this control approach.

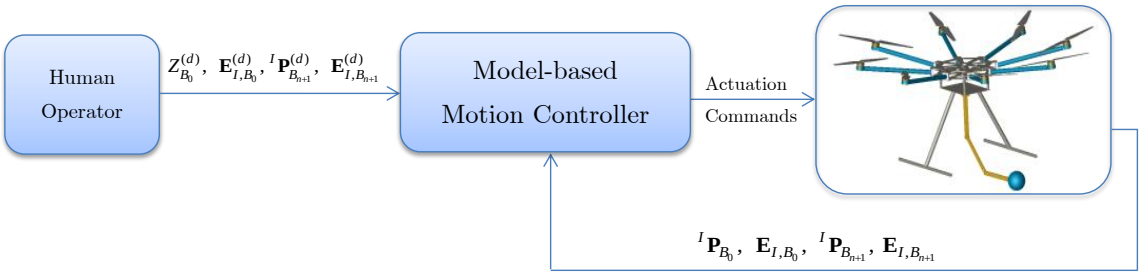

Figure 5.5: Control Scenario IV: The UAV altitude and orientation as well as the end-effector position and orientation are controlled independently

Remark. *Note that in the second and the fourth control strategies proposed here, there are in general* 10 *DOFs to be controlled independently. This includes the six DOFs of the manipulator end-effector and four DOFs of the UAV. Depending on the manipulator configuration design, the number of its joints, the types and the order of the joints, different possible scenarios may occur. For a hyper DOF manipulator, the ten reference inputs to the control algorithm are not sufficient to uniquely determine the* $4 + n$ *DOFs which constitute the independent set of system generalized coordinates. In such a case, the control algorithm needs to also deal with the kinematic redundancy of the system due to having more DOFs than required to achieve the control objectives. In other cases, the manipulator may not provide the system with enough DOFs such that all the ten reference inputs can be controlled. In such a case, the number of the end-effector DOFs to be controlled should be reduced depending on the application. In particular, one may want to only control the end-effector position but not its orientation.*

The three subsequent chapters are concerned with the detailed development of the motion control algorithms discussed here. In each chapter, the control algorithm is developed based on the VDC and an approach is proposed for handling the nonholonomic constraints. Ignoring the dynamics of the propeller, Chapters 6 and 7 develop algorithms for the first

and the second control scenarios respectively. Control design for the other two scenarios can be performed in a similar manner. Chapter 8 presents an extension of these control approaches where the propeller dynamics are fully incorporated in the control design.

Chapter 6

Control Design: Scenario I

In Chapter 5, two different scenarios, namely the first and the third scenarios, were discussed in which the manipulator joint position trajectories were provided for the control system. In this chapter, the first control scenario in is fully developed and discussed. Similar steps may be taken to design a motion control algorithm for the third scenario.

In the first control strategy, the control system may receive and respond to orientation and elevation control commands from a human operator/trajectory planner. The operator/planner can then manipulate the orientation reference commands to indirectly position the UAV in the horizontal plane, as needed. Considering such a strategy, the control system provides for independent control of the orientation and altitude of the base platform as well as all the joint positions.

In this chapter, the propeller dynamics are ignored in the design of the motion controller. In the first section of this chapter, the system dynamics are reviewed to describe the implications of neglecting the propellers dynamics on the system modeling. Next, the VDC is employed to develop a motion control algorithm for the UAV-Manipulator system. This

is followed by proposing an approach to handle the nonholonomic motion constraints. Finally, the chapter is concluded by summarizing the steps involved in the control algorithm implementation.

Before proceeding, it should be noted that for brevity, explicit dependencies of all desired and required variables on time are omitted in the notation.

6.1 System Dynamics Revision

The kinematics and dynamics formulation of the UAV-Manipulator are briefly reformulated here ignoring the propeller dynamics.

6.1.1 Kinematics

The generalized velocity vector is reduced to,

$$\mathcal{V} \triangleq \left[\boldsymbol{V}_{B_0}^T, \dot{\boldsymbol{q}}_m^T\right]^T \in \Re^{n+6}. \tag{6.1}$$

Similarly, the overall extended velocity vector is reduced to,

$$\mathcal{V}_e \triangleq \left[\boldsymbol{V}_{B_0}^T, \cdots, \boldsymbol{V}_{B_{n+1}}^T\right]^T \in \Re^{6(n+2)}. \tag{6.2}$$

In this case, there is only a single kinematic chain composed of the UAV base, the n manipulator links, and the held object. The forward velocity propagation in this kinematic chain

is performed by $\mathcal{V}_e = \mathcal{J}_e \mathcal{V}$ where the generalized Jacobian matrix is given by,

$$\mathcal{J}_e = \begin{bmatrix} \mathbf{I}_{6\times6} & 0_{6\times1} & 0_{6\times1} & 0_{6\times1} & \cdots & 0_{6\times1} \\ \mathbf{U}_{B_1}^{B_0\,T} & \boldsymbol{z} & 0_{6\times1} & 0_{6\times1} & \cdots & 0_{6\times1} \\ \mathbf{U}_{B_2}^{B_0\,T} & \mathbf{U}_{B_2}^{B_1\,T}\boldsymbol{z} & \boldsymbol{z} & 0_{6\times1} & \cdots & 0_{6\times1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{U}_{B_n}^{B_0\,T} & \mathbf{U}_{B_n}^{B_1\,T}\boldsymbol{z} & \mathbf{U}_{B_n}^{B_2\,T}\boldsymbol{z} & \mathbf{U}_{B_n}^{B_3\,T}\boldsymbol{z} & \cdots & \boldsymbol{z} \\ \mathbf{U}_{B_{n+1}}^{B_0\;\;T} & \mathbf{U}_{B_{n+1}}^{B_1\;\;T}\boldsymbol{z} & \mathbf{U}_{B_{n+1}}^{B_2\;\;T}\boldsymbol{z} & \mathbf{U}_{B_{n+1}}^{B_3\;\;T}\boldsymbol{z} & \cdots & \mathbf{U}_{B_{n+1}}^{B_n\;\;T}\boldsymbol{z} \end{bmatrix}. \tag{6.3}$$

6.1.2 Dynamics

The equations of motion of the entire system are reduced to,

$$\mathcal{M}\dot{\mathcal{V}} + \mathcal{C}\mathcal{V} + \mathcal{G} = \begin{bmatrix} \boldsymbol{F}_{B_0} \\ \boldsymbol{\tau_m} \end{bmatrix}, \tag{6.4}$$

where $\mathcal{M}, \mathcal{C}, \mathcal{G}$ are given by,

$$\begin{aligned} \mathcal{M} &= \mathcal{J}_e^T\, \mathcal{M}_e\, \mathcal{J}_e \in \Re^{(n+6)\times(n+6)} \\ \mathcal{C} &= \mathcal{J}_e^T\, \mathcal{M}_e\, \dot{\mathcal{J}}_e + \mathcal{J}_e^T\, \mathcal{C}_e\, \mathcal{J}_e \in \Re^{(n+6)\times(n+6)} \\ \mathcal{G} &= \mathcal{J}_e^T\, \mathcal{G}_e \in \Re^{n+6}, \end{aligned} \tag{6.5}$$

in which $\mathcal{M}_e$, $\mathcal{C}_e$, and $\mathcal{G}_e$ are given by,

$$\mathcal{M}_e = diag\left(\mathbf{M}_{B_0}, \cdots, \mathbf{M}_{B_{n+1}}\right) \in \Re^{6(n+2)\times 6(n+2)}$$
$$\mathcal{C}_e = diag\left(\mathbf{C}_{B_0}, \cdots, \mathbf{C}_{B_{n+1}}\right) \in \Re^{6(n+2)\times 6(n+2)}$$
$$\mathcal{G}_e = \left[\boldsymbol{G}_{B_0}^T, \cdots, \boldsymbol{G}_{B_{n+1}}^T\right]^T \in \Re^{6(n+2)\times 6(n+2)} \tag{6.6}$$

In (6.4), $\boldsymbol{F}_{B_0}$ is the extended force applied at the UAV center of mass. In Chapter 3, where the propellers were included in the dynamics derivation, there was no direct actuation for the UAV base. As a result, the base was under-actuated in all six DOFs and $\boldsymbol{F}_{B_0}$ was a zero vector. When the propellers are not considered as separate rigid bodies in the system dynamics, the propeller produced thrust forces/drag torques are applied directly to the UAV base. This is evident since the whole UAV is considered as a single rigid body in this case. It is easy to show that the resultant force/torque vector of all the propeller thrust forces, when measured in B_0 is $\boldsymbol{F}_{B_0} = [0, 0, F_{z_{B_0}}, M_{x_{B_0}}, M_{y_{B_0}}, M_{z_{B_0}}]^T$. This implies that the entire system remains under-actuated in two DOFs. In this case, the degree of under-actuation is reduced compared to the case where the propellers are fully included in the dynamics derivation and control design.

Note that in the current version of the system dynamics, the control inputs are considered to be the resultant extended force vector at the base $\boldsymbol{F}_{B_0}$ as well as the manipulator joint torque vector $\boldsymbol{\tau}_m$.

6.2 Motion Control Design

In view of (6.1), a generalized required velocity vector is defined as,

$$\mathcal{V}^{(r)} \triangleq \left[\boldsymbol{V}_{B_0}^{(r)^T}, \dot{\boldsymbol{q}}_m^{(r)^T} \right]^T \in \Re^{n+6}, \tag{6.7}$$

where $\boldsymbol{V}_{B_0}^{(r)} \triangleq \left[\boldsymbol{v}_{B_0}^{(r)^T}, \boldsymbol{\omega}_{B_0}^{(r)^T} \right]^T \in \Re^6$ denotes the extended required velocity vector of the base, and $\dot{\boldsymbol{q}}_m^{(r)} \in \Re^n$ is the required joint velocity vector. Note that the propeller velocity vector is excluded from the generalized required velocity vector since the propeller dynamics are ignored in the model-based control design in this chapter.

In this control scenario, the desired trajectories for the manipulator joint positions $\boldsymbol{q}_m^{(d)}$, base orientation $\boldsymbol{E}_{I,B_0}^{(d)}$, and base altitude $Z_{B_0}^{(d)}$ are provided to the control system. As a result, the corresponding required velocities for these DOFs are designed independently as follows. Let $\boldsymbol{q}_m^{(d)} \triangleq \left[q_{m_1}^{(d)}, \cdots, q_{m_n}^{(d)} \right]^T \in \Re^n$ be the desired joint trajectories. Then, $\dot{\boldsymbol{q}}_m^{(r)}$ is designed as,

$$\dot{\boldsymbol{q}}_m^{(r)} = \dot{\boldsymbol{q}}_m^{(d)} + \boldsymbol{\Lambda}_{q_m} \left(\boldsymbol{q}_m^{(d)} - \boldsymbol{q}_m \right), \tag{6.8}$$

where $\boldsymbol{\Lambda}_{q_m} \in \Re^{n \times n}$ is a constant positive definite diagonal matrix. The elements of $\boldsymbol{\Lambda}_{q_m}$ are design parameters that determine the convergence rate of joint position tracking errors. Also, let $\boldsymbol{E}_{I,B_0}^{(d)}$ be the desired Euler angles of the base; then $\boldsymbol{\omega}_{B_0}^{(r)}$ is designed as,

$$\boldsymbol{\omega}_{B_0}^{(r)} = \mathbf{G} \left(\boldsymbol{E}_{I,B_0} \right) \left[\dot{\boldsymbol{E}}_{I,B_0}^{(d)} + \boldsymbol{\Lambda}_{E_{B_0}} \left(\boldsymbol{E}_{I,B_0}^{(d)} - \boldsymbol{E}_{I,B_0} \right) \right], \tag{6.9}$$

where $\boldsymbol{\Lambda}_{E_{B_0}} \in \Re^{3 \times 3}$ is a constant positive diagonal matrix whose elements are design parameters and $\mathbf{G}$ is given by (3.16).

The UAV required linear velocity $\boldsymbol{v}_{B_0}^{(r)}$ has to be determined such that on one hand the

motion constraints are satisfied and on the other hand the altitude tracking error converges. In fact, although the under-actuation of the base imposes non-integrable motion constraints on the system acceleration space, the velocity trajectories become dependent over time. Consequently, the two components of $\boldsymbol{\nu}_{B_0}^{(r)}$ in the unactuated directions namely $v_{x_{B_0}}^{(r)}$ and $v_{y_{B_0}}^{(r)}$ are dependent on the required velocities in the actuated DOFs, in a window of time. In the next subsection, a method is proposed to formulate this dependency and determine $v_{x_{B_0}}^{(r)}$ and $v_{y_{B_0}}^{(r)}$. Also, given the desired base altitude $Z_{B_0}^{(d)}$, a reference vertical velocity in the inertial frame is designed as,

$$\dot{Z}_{B_0}^{(r)} = \dot{Z}_{B_0}^{(d)} + \lambda_z \left(Z_{B_0}^{(d)} - Z_{B_0} \right), \tag{6.10}$$

where λ_z is a positive design parameter which defines the convergence rate of altitude tracking error. Now given $v_{x_{B_0}}^{(r)}$ and $v_{y_{B_0}}^{(r)}$ obtained through satisfaction of motion constraints over time, the remaining component $v_{z_{B_0}}^{(r)}$ is designed such that the projection of the UAV required linear velocity $\boldsymbol{\nu}_{B_0}^{(r)}$ onto the vertical inertial axis *Z* produces the reference velocity $\dot{Z}_{B_0}^{(r)}$. Hence, $v_{z_{B_0}}^{(r)}$ is derived from the following equation,

$$\dot{Z}_{B_0}^{(r)} = [0, 0, 1]\, \mathbf{R}_{B_0}^{I}\, \boldsymbol{\nu}_{B_0}^{(r)}. \tag{6.11}$$

where $\mathbf{R}_{B_0}^{I}$ is the rotation matrix of B_0 w.r.t I.

Given the dependent components $v_{x_{B_0}}^{(r)}$, $v_{y_{B_0}}^{(r)}$, the generalized required velocity vector $\mathcal{V}^{(r)}$ would be completely determined. As a result, the required velocities of all bodies are

obtained by a forward propagation in the system kinematic chain as follows,

$$\begin{aligned} \boldsymbol{V}_{B_{i+1}}^{(r)} &= \mathbf{U}_{B_{i+1}}^{B_i{}^T} \boldsymbol{V}_{B_i}^{(r)} + \dot{q}_{m_{i+1}}^{(r)} \boldsymbol{z},\ i = 0, \cdots, n-1 \\ \boldsymbol{V}_{B_{n+1}}^{(r)} &= \mathbf{U}_{B_{n+1}}^{B_n{}^T} \boldsymbol{V}_{B_n}^{(r)}. \end{aligned} \tag{6.12}$$

An overall extended required velocity vector is defined by,

$$\mathcal{V}_e^{(r)} \triangleq \left[\boldsymbol{V}_{B_0}^{(r)^T}, \cdots, \boldsymbol{V}_{B_{n+1}}^{(r)^T} \right]^T \in \Re^{6(n+2)}, \tag{6.13}$$

and the required velocity propagation in (6.12) is written in the following compact form,

$$\mathcal{V}_e^{(r)} = \mathcal{J}_e \mathcal{V}^{(r)}. \tag{6.14}$$

where $\mathcal{J}_e$ is given by (6.3).

Having the required motion of each body in the system, we may now develop the control equations. According to (4.2), the motion equations for the required velocities are written in the linear-in-parameter form,

$$\hat{\mathbf{M}}_{B_i} \dot{\boldsymbol{V}}_{B_i}^{(r)} + \hat{\mathbf{C}}_{B_i}(\boldsymbol{\omega}_{B_i}) \boldsymbol{V}_{B_i}^{(r)} + \hat{\boldsymbol{G}}_{B_i} = \mathbf{Y}_{B_i} \hat{\boldsymbol{\theta}}_{B_i}, \quad i = 0, \cdots, n+1, \tag{6.15}$$

where $\mathbf{Y}_{B_i}$ and $\hat{\boldsymbol{\theta}}_{B_i}$ have been defined in (4.2). Treating the net extended force vector of each body as its local virtual control input, the control laws for all system bodies are designed as,

$$\boldsymbol{F}_{B_i}^{*(r)} = \mathbf{Y}_{B_i} \hat{\boldsymbol{\theta}}_{B_i} + \mathbf{K}_{B_i} \left(\boldsymbol{V}_{B_i}^{(r)} - \boldsymbol{V}_{B_i} \right), \quad i = 0, \cdots, n+1, \tag{6.16}$$

where $\mathbf{K}_{B_i}$ was defined in (4.1) and $\hat{\boldsymbol{\theta}}_{B_i}$ is updated by the following adaptation law,

$$\dot{\hat{\boldsymbol{\theta}}}_{B_i} = \boldsymbol{\Gamma}_{B_i}^{-1} \mathbf{Y}_{B_i}^T \left(\boldsymbol{V}_{B_i}^{(r)} - \boldsymbol{V}_{B_i} \right), \;\; i = 0, \cdots, n+1, \tag{6.17}$$

here $\boldsymbol{\Gamma}_{B_i} > 0$ is a design matrix and determines the adaptation rate of $\hat{\boldsymbol{\theta}}_{B_i}$. Once the net extended required force vectors are designed, the extended required interaction force vectors are obtained by a backward force recursion from the held object to the UAV base, i.e.,

$$\begin{aligned} &\boldsymbol{F}_{B_{n+1}}^{(r)} = \boldsymbol{F}_{B_{n+1}}^{*(r)}, \\ &\boldsymbol{F}_{B_i}^{(r)} = \boldsymbol{F}_{B_i}^{*(r)} + \mathbf{U}_{B_{i+1}}^{B_i} \boldsymbol{F}_{B_{i+1}}^{(r)}, \; i = n, \cdots, 0. \end{aligned} \tag{6.18}$$

Finally, the actual control inputs are derived from the required interaction forces/torques as,

$$\begin{aligned} &\boldsymbol{F}_{B_0} = \boldsymbol{F}_{B_0}^{(r)}, \\ &\tau_{m_i} = \boldsymbol{z}^T \boldsymbol{F}_{B_i}^{(r)}, \; i = 1, \cdots, n. \end{aligned} \tag{6.19}$$

6.3 Second-Order Nonholonomic Constraints

This part is concerned with determining the required motion variables for the dependent DOFs by taking the under-actuation into account. Note that nonholonomic constraints are imposed on the system motion since $F_{x_{B_0}} = F_{y_{B_0}} = 0$. This establishes two non-integrable constraints between the linear accelerations of all bodies once projected onto the x_{B_0}-y_{B_0} plane. As will be proven in stability analysis, the control laws (6.16) along with the adaptation laws (6.17) guarantee the convergence of $\mathcal{V}$ to $\mathcal{V}^{(r)}$ which in turn ensures

asymptotic convergence of the tracking errors to zero in all DOFs. This is provided the control system does not request any force/torque in the unactuated DOFs. To ensure this, we enforce the required force/torque at the base to be zero in the unactuated directions. This is achieved by imposing the following constraints,

$$F_{x_{B_0}}^{(r)} = F_{y_{B_0}}^{(r)} = 0. \tag{6.20}$$

In the remainder of this section, (6.20) is fully developed and the constraints imposed on required accelerations are explored. Since the required accelerations for independent DOFs are determined by the time derivatives of (6.8) and (6.9), the nonholonomic constraints will be used to derive the required accelerations for dependent DOFs. Numerical integration then yields the dependent required velocities. Following this approach guarantees that the required motions of all system bodies are dynamically feasible.

Similar to (6.2), let $\mathcal{F}_e^{*(r)} \in \Re^{6(n+2)}$ be the extended required net force/moment vector of all bodies and $\mathcal{U}_{B_0}$ be an overall extended transformation matrix defined as,

$$\mathcal{U}_{B_0} \triangleq \left[\mathbf{I}_{6\times 6}, \mathbf{U}_{B_1}^{B_0}, \cdots, \mathbf{U}_{B_{n+1}}^{B_0}\right] \in \Re^{6\times 6(n+2)}. \tag{6.21}$$

Then a backward force propagation through (6.18) yields,

$$\boldsymbol{F}_{B_0}^{(r)} = \mathcal{U}_{B_0}\mathcal{F}_e^{*(r)} \tag{6.22}$$

Moreover using (6.15) and (6.16), it follows that,

$$\mathcal{F}_e^{*(r)} = \hat{\mathcal{M}}_e\mathcal{J}_e\dot{\mathcal{V}}^{(r)} + \hat{\mathcal{M}}_e\dot{\mathcal{J}}_e\mathcal{V}^{(r)} + \hat{\mathcal{C}}_e\mathcal{J}_e\mathcal{V}^{(r)} + \hat{\mathcal{G}}_e + \mathcal{K}_e\mathcal{J}_e\left(\mathcal{V}^{(r)} - \mathcal{V}\right), \tag{6.23}$$

where $\hat{\mathcal{M}}_e$, $\hat{\mathcal{C}}_e$ and $\hat{\mathcal{G}}_e$ are estimates of the variables defined in (6.6), and $\mathcal{K}_e = diag\left(\mathbf{K}_{B_0}, \cdots, \mathbf{K}_{B_{n+1}}\right)$.

Substituting (6.23) in (6.22), multiplying it by a constant matrix $\mathcal{L}$ to extract the first two rows of the equation, and applying (6.20) yields,

$$\mathbf{0}_{2\times 1} = \mathcal{L}\mathcal{U}_{B_0}\hat{\mathcal{M}}_e\mathcal{J}_e\dot{\mathcal{V}}^{(r)} + \mathcal{L}\mathcal{U}_{B_0}\left[\hat{\mathcal{M}}_e\dot{\mathcal{J}}_e\mathcal{V}^{(r)} + \hat{\mathcal{C}}_e\mathcal{J}_e\mathcal{V}^{(r)} + \hat{\mathcal{G}}_e + \mathcal{K}_e\mathcal{J}_e\left(\mathcal{V}^{(r)} - \mathcal{V}\right)\right] \tag{6.24}$$

Equation (6.24) characterizes the nonholonomic constraints and completely describes the dependency of the required accelerations. In this equation, the freely designed required accelerations namely $\frac{d}{dt}(\dot{\boldsymbol{q}}_m^{(r)})$ and $\frac{d}{dt}(\boldsymbol{\omega}_{B_0}^{(r)})$ are given by the time derivatives of (6.8) and (6.9), respectively. Also, the numerical values of all other variables in this equation are available in each instant of time in the control algorithm implementation. Hence, (6.24) essentially represents two explicit constraints on $\frac{d}{dt}(\boldsymbol{\nu}_{B_0}^{(r)})$. On the other hand, the constraint (6.11) on $\boldsymbol{\nu}_{B_0}^{(r)}$ should be always satisfied over time. Hence, its time derivative reveals another constraint on $\frac{d}{dt}(\boldsymbol{\nu}_{B_0}^{(r)})$. This, along with (6.24), constitute a set of three linear algebraic equations to be solved for $\frac{d}{dt}(\boldsymbol{\nu}_{B_0}^{(r)})$. Finally, the dependent required velocities $v_{x_{B_0}}^{(r)}$ and $v_{y_{B_0}}^{(r)}$ are obtained through a numerical integration of the corresponding acceleration variables.

For future reference, (6.24) is written in the following compact form,

$$\mathcal{A}\dot{\mathcal{V}}^{(r)} = \mathcal{B}, \tag{6.25}$$

where,

$$\begin{aligned}
&\mathcal{A} = \mathcal{L}\mathcal{U}_{B_0}\hat{\mathcal{M}}_e\mathcal{J}_e \in \Re^{2\times(n+6)},\\
&\mathcal{B} = -\mathcal{L}\mathcal{U}_{B_0}\left[\hat{\mathcal{M}}_e\dot{\mathcal{J}}_e\mathcal{V}^{(r)} + \hat{\mathcal{C}}_e\mathcal{J}_e\mathcal{V}^{(r)} + \hat{\mathcal{G}}_e + \mathcal{K}_e\mathcal{J}_e\left(\mathcal{V}^{(r)} - \mathcal{V}\right)\right] \in \Re^2.
\end{aligned} \tag{6.26}$$

Next section summarizes the steps involved in the implementation of the control algorithm and clarifies the derivation of the dependent motion variables within the controller implementation.

6.4 Control Algorithm Implementation

Fig. 6.1 depicts a block diagram of the model-based motion control algorithm. There are five major consecutive sections involved in this algorithm which are summarized below.

i - Calculation of independent required motion variables

- Given the desired trajectories of the manipulator joints and the base orientation, (6.8), (6.9) and the time derivative of these equations are employed to design $\dot{\boldsymbol{q}}_m^{(r)}$, $\boldsymbol{\omega}_{B_0}^{(r)}$, $\frac{d}{dt}(\dot{\boldsymbol{q}}_m^{(r)})$ and $\frac{d}{dt}(\boldsymbol{\omega}_{B_0}^{(r)})$.

ii - Derivation of dependent required motion variables

- The integrator of dependent required accelerations yields $v_{x_{B_0}}^{(r)}$, and $v_{y_{B_0}}^{(r)}$. This integrator is initialized at the real linear velocity of the base obtained from measurements.
- Given the desired trajectory of the base altitude, (6.11) is used to design $\dot{Z}_{B_0}^{(r)}$.
- Given $v_{x_{B_0}}^{(r)}$, $v_{y_{B_0}}^{(r)}$ and $\dot{Z}_{B_0}^{(r)}$, (6.11) is used to find $v_{z_{B_0}}^{(r)}$. At this point, all the components of $\mathcal{V}^{(r)}$ are determined.

- Equation (6.24) and the time derivative of (6.11) are solved simultaneously to obtain $\frac{d}{dt}(\boldsymbol{v}_{B_0}^{(r)})$. This completes the derivation of $\dot{\mathcal{V}}^{(r)}$.

iii - Forward propagation of required motion variables

- Given $\mathcal{V}^{(r)}$ and $\dot{\mathcal{V}}^{(r)}$, the required velocity/acceleration for all system bodies are obtained using (6.14) and its time derivative.

iv - Required local control input allocation

- Given $\boldsymbol{V}_{B_i}^{(r)}$ and $\dot{\boldsymbol{V}}_{B_i}^{(r)}$, the required net force $\boldsymbol{F}_{B_i}^{*(r)}$ for each body is designed using (6.16) and (6.17).

v - Backward recursion of required forces/moments

- Given $\boldsymbol{F}_{B_i}^{*(r)}$ for all bodies, the required interaction forces/moments $\boldsymbol{F}_{B_0}^{(r)}$ are derived from (6.18).
- Finally, the control inputs $\boldsymbol{F}_{B_0}$, and τ_{m_i} are obtained from (6.19).

Remark. *It should be mentioned that this implementation requires measurements of the base altitude, attitude and linear/angular velocity as well as the manipulator joint positions/velocities.*

Remark. *The method proposed in the control algorithm to generate system dependent trajectories does not use any force/acceleration measurements. As a result, the control system avoids the implementation complexities introduced by such measurements in the feedback loop. The most significant of these is the possible instability due to algebraic loop and feedback time-delay.*

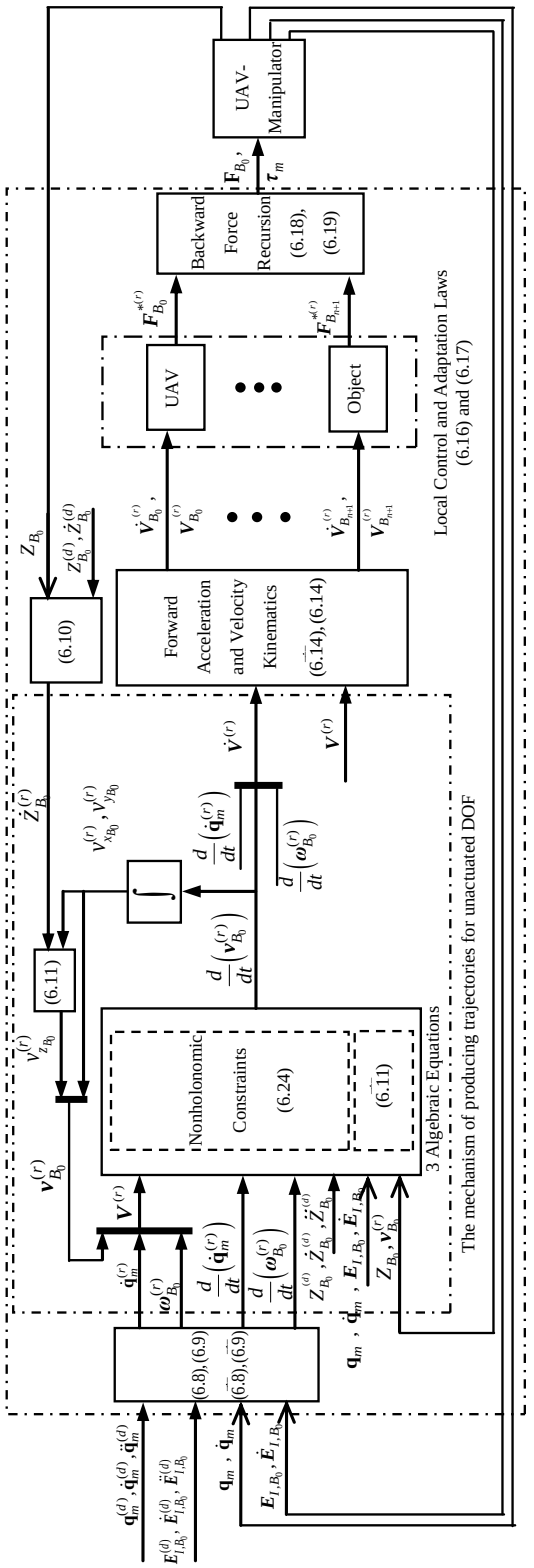

Figure 6.1: Block diagram of the proposed VDC algorithm for Scenario I. $v_{x_{B_0}}^{(r)}$, $v_{y_{B_0}}^{(r)}$ and $v_{z_{B_0}}^{(r)}$ are the components of $\boldsymbol{\nu}_{B_0}^{(r)}$. The $\dot{(i)}$ indicates time derivative of the ith equation. Also, the solid arrows indicate the directions of the signals flows while the other ones show feed-back signals.

Control algorithm development and implementation for Scenario III follow similar steps to what described in this chapter for Scenario I. In Scenario III, the UAV position and yaw orientation as well as the manipulator joint positions are controlled independently. Therefore, the required velocities for these DOFs are designed independently. In this case, the required joint velocities are still defined according to (6.8). The base required linear velocity and a yaw reference velocity are designed similar to (6.10). The dependent required motions for the UAV roll and pitch orientations are numerically derived by developing (6.20) which yields a similar equation as (6.24). Given the generalized velocities and accelerations, the rest of the algorithm remains the same as Scenario I.

Chapter 7

Control Design: Scenario II

In Chapter 5, two different scenarios were proposed in which the manipulator end-effector DOFs were controlled directly. In this chapter, the control design for the second scenario is fully developed. Similar steps may be taken to design an algorithm for the fourth control scenario.

In the second control strategy, the trajectory planner/human operator directly provides reference commands for the base platform position and yaw orientation as well as position and orientation reference commands for the arm end-effector. In particular, the UAV may be first flown to a position where the target is within the reach of the manipulator end-effector. It may then be commanded to hover in place while the manipulator end-effector executes a task. The reduced system kinematics and dynamics presented in Chapter 6 will be used for the controller development here and the propeller dynamics will be ignored.

7.1 Motion Control Design

Let ${}^{I}\boldsymbol{P}_{B_{n+1}} \triangleq \left[X_{B_{n+1}}, Y_{B_{n+1}}, Z_{B_{n+1}}\right]^T \in \Re^3$ be the position vector of frame B_{n+1} (end-effector) with respect to frame I and expressed in I, and $\boldsymbol{E}_{I,B_{n+1}} \triangleq [\Phi_{x_{B_{n+1}}}, \Theta_{y_{B_{n+1}}}, \Psi_{z_{B_{n+1}}}]^T \in \Re^3$ be the vector of Euler angles representing the rotation of B_{n+1} with respect to I. Also, let ${}^{I}\boldsymbol{P}^{(d)}_{B_{n+1}}$ be the desired position vector and $\boldsymbol{E}^{(d)}_{I,B_{n+1}}$ be the vector of desired Euler angles of B_{n+1} with respect to I. Then, the body-expressed required linear and angular velocities of B_{n+1} are respectively designed as,

$$\boldsymbol{v}^{(r)}_{B_{n+1}} = \mathbf{R}^{B_{n+1}}_{I}\left[{}^{I}\dot{\boldsymbol{P}}^{(d)}_{B_{n+1}} + \boldsymbol{\Lambda}_{P_{B_{n+1}}}\left({}^{I}\boldsymbol{P}^{(d)}_{B_{n+1}} - {}^{I}\boldsymbol{P}_{B_{n+1}}\right)\right], \tag{7.1}$$

$$\boldsymbol{\omega}^{(r)}_{B_{n+1}} = \mathbf{G}\big(\boldsymbol{E}_{I,B_{n+1}}\big)\left[\dot{\boldsymbol{E}}^{(d)}_{I,B_{n+1}} + \boldsymbol{\Lambda}_{E_{B_{n+1}}}\Big(\boldsymbol{E}^{(d)}_{I,B_{n+1}} - \boldsymbol{E}_{I,B_{n+1}}\Big)\right] \tag{7.2}$$

where $\boldsymbol{\Lambda}_{P_{B_{n+1}}} \in \Re^{3\times3}$ and $\boldsymbol{\Lambda}_{E_{B_{n+1}}} \in \Re^{3\times3}$ are constant positive definite diagonal design matrices and $\mathbf{G}$ is defined similar to (3.16). The kinematic mapping between the end-effector extended required velocity $\boldsymbol{V}^{(r)}_{B_{n+1}}$ given by (7.1) and (7.2) and the generalized required velocity vector $\mathcal{V}^{(r)}$ defined in (6.7) is represented by,

$$\boldsymbol{V}^{(r)}_{B_{n+1}} = \mathcal{J}\mathcal{V}^{(r)}, \tag{7.3}$$

where the Jacobian matrix $\mathcal{J} \in \Re^{6\times(n+6)}$ is extracted from (6.3) as,

$$\mathcal{J} = \left[\mathbf{U}^{B_0{}^T}_{B_{n+1}}, \mathbf{U}^{B_1{}^T}_{B_{n+1}}\boldsymbol{z}, \cdots, \mathbf{U}^{B_n{}^T}_{B_{n+1}}\boldsymbol{z}\right]. \tag{7.4}$$

The potential redundancy for the required velocities, introduced in the inverse kinematics of the hyper-DOF UAV-Manipulator system and described by (7.3), cannot be treated like the kinematic redundancy of common fully-actuated robotic systems. Solutions to this kinematic redundancy must be dynamically feasible. In other words, these solutions have to be consistent with the nonholonomic constraints imposed on the system accelerations.

To tackle this problem, in an approach similar to the first control scenario, components of the generalized required velocity vector $\mathcal{V}^{(r)}$ are separated into two dependent components and $n+4$ independent ones. The former are reserved to meet the motion constraints while the latter are designated to satisfy (7.3). To this end, the dependent required accelerations are determined by solving the second-order motion constraints and the corresponding required velocities are obtained through numerical integration. Given these dependent components of $\mathcal{V}^{(r)}$, the independent components are then derived such that (7.3) holds and the end-effector undergoes its designed required velocity. This guarantees the consistency of the kinematic redundancy resolution with the motion constraints.

In this control strategy, two components of the required angular velocity of the base platform namely $\omega_{x_{B_0}}^{(r)}$ and $\omega_{y_{B_0}}^{(r)}$ are the dependent variables. This is due to one of the control objectives that requests for independent control of the base position. These angular velocities can be used to manipulate the aerodynamic thrust and indirectly control the base position. Following the aforementioned approach, the kinematic redundancy problem is reformulated as follows.

Let $\mathbf{T} \triangleq [\mathbf{T}_1\ \mathbf{T}_2] \in \Re^{(n+6)\times(n+6)}$ be a constant reordering matrix such that,

$$\mathcal{V}^{(r)} = [\mathbf{T}_1\ \mathbf{T}_2] \begin{bmatrix} \mathcal{V}_v^{(r)} \\ \mathcal{V}_u^{(r)} \end{bmatrix}, \tag{7.5}$$

where, by definition, $\mathcal{V}_v^{(r)} \triangleq \left[\boldsymbol{v}_{B_0}^{(r)^T}, \omega_{z_{B_0}}^{(r)}, \dot{\boldsymbol{q}}_m^{(r)^T}\right]^T \in \Re^{n+4}$ is the independent generalized required velocity vector and $\mathcal{V}_u^{(r)} \triangleq \left[\omega_{x_{B_0}}^{(r)}, \omega_{y_{B_0}}^{(r)}\right]^T \in \Re^2$ is the dependent one. Substituting (7.5) in (7.3) results in,

$$\boldsymbol{V}_{B_{n+1}}^{(r)} = \mathcal{J}_v \mathcal{V}_v^{(r)} + \mathcal{J}_u \mathcal{V}_u^{(r)}, \tag{7.6}$$

where $\mathcal{J}_v = \mathcal{J}\,\mathbf{T}_1 \in \Re^{6\times(n+4)}$ and $\mathcal{J}_u = \mathcal{J}\,\mathbf{T}_2 \in \Re^{6\times 2}$.

In addition, to control the position of the base independently, and given the desired position vector of frame B_0 (base platform) ${}^I\boldsymbol{P}_{B_0}^{(d)}$, the required linear velocity of this frame expressed in the inertial frame I is designed as,

$${}^I\dot{\boldsymbol{P}}_{B_0}^{(r)} = \left[{}^I\dot{\boldsymbol{P}}_{B_0}^{(d)} + \boldsymbol{\Lambda}_{P_{B_0}}\left({}^I\boldsymbol{P}_{B_0}^{(d)} - {}^I\boldsymbol{P}_{B_0}\right)\right], \tag{7.7}$$

with $\boldsymbol{\Lambda}_{P_{B_0}} \in \Re^{3\times 3}$ being a constant design matrix and positive definite. Similarly, given the desired trajectory of the base yaw orientation $\Psi_{z_{B_0}}^{(d)}$, the following reference velocity is defined,

$$\dot{\Psi}_{z_{B_0}}^{(r)} = \dot{\Psi}_{z_{B_0}}^{(d)} + \lambda_{\Psi_{z_{B_0}}}\left(\Psi_{z_{B_0}}^{(d)} - \Psi_{z_{B_0}}\right), \tag{7.8}$$

with $\lambda_{\Psi_{z_{B_0}}}$ being a positive design constant. Equations (7.7), (7.8) impose four constraints on the generalized required velocity vector as follows,

$$\begin{bmatrix} {}^I\dot{\boldsymbol{P}}_{B_0}^{(r)} \\ \dot{\Psi}_{z_{B_0}}^{(r)} \end{bmatrix} = \mathcal{J}_v' \mathcal{V}_v^{(r)} + \mathcal{J}_u' \mathcal{V}_u^{(r)}, \tag{7.9}$$

where $\mathcal{J}_v' = \mathcal{Q}\mathbf{T}_1' \in \Re^{4\times(n+4)}$ and $\mathcal{J}_u' = \mathcal{Q}\mathbf{T}_2' \in \Re^{4\times 2}$ while $\mathbf{T}_1' \in \Re^{6\times(n+4)}$ and $\mathbf{T}_2' \in \Re^{6\times 2}$

are constant reordering matrices such that $\boldsymbol{V}_{B_0}^{(r)} = [\mathbf{T}_1'\mathbf{T}_2'][{\mathcal{V}_v^{(r)}}^T {\mathcal{V}_u^{(r)}}^T]^T$ and $\mathcal{Q}$ is given by,

$$\mathcal{Q} = \begin{bmatrix} \mathbf{R}_{B_0}^I & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{1\times 3} & [0,0,1]\mathbf{G}^{-1}\left(\boldsymbol{E}_{I,B_0}\right) \end{bmatrix} \in \Re^{4\times 6}. \tag{7.10}$$

The kinematic constraints described in (7.9) have to be taken into consideration while resolving the kinematic redundancy in (7.6). Equations (7.6) and (7.9) may be represented together as follows,

$$\mathcal{L}_v \mathcal{V}_v^{(r)} = \begin{bmatrix} \boldsymbol{V}_{B_{n+1}}^{(r)} \\ {}^I\dot{\boldsymbol{P}}_{B_0}^{(r)} \\ \dot{\Psi}_{z_{B_0}}^{(r)} \end{bmatrix} - \mathcal{L}_u \mathcal{V}_u^{(r)}, \tag{7.11}$$

in which $\mathcal{L}_v = [\mathcal{J}_v^T \mathcal{J}_v'^T]^T \in \Re^{10\times(n+4)}$ and $\mathcal{L}_u = [\mathcal{J}_u^T \mathcal{J}_u'^T]^T \in \Re^{10\times 2}$. Equation (7.11) characterizes the kinematic redundancy along with the additional constraints. The first term of the right-hand side of this equation is given by (7.1), (7.2), (7.7), and (7.8). Also, $\mathcal{V}_u^{(r)}$ is determined from the integration of its derivative, which is obtained from the nonholonomic constraints as will be discussed shortly. At this point, any kinematic redundancy resolution algorithm can be employed to solve (7.11) since the second-order motion constrains are already considered. This includes kinematic control methods which can simultaneously achieve secondary objectives such as maximizing manipulability or avoiding joint limits. Without loss of generality, the minimum norm solution [89] is utilized here for redundancy resolution as follows,

$$\mathcal{V}_v^{(r)} = \mathcal{L}_v^{\dagger} \left(\begin{bmatrix} \boldsymbol{V}_{B_{n+1}}^{(r)} \\ {}^I\dot{\boldsymbol{P}}_{B_0}^{(r)} \\ \dot{\Psi}_{z_{B_0}}^{(r)} \end{bmatrix} - \mathcal{L}_u \mathcal{V}_u^{(r)} \right), \tag{7.12}$$

where $\mathcal{L}_v^{\dagger}$ is the right pseudo-inverse of $\mathcal{L}_v$.

Once the generalized required velocity vector $\mathcal{V}^{(r)}$ is completely obtained, the remainder of the control algorithm follows similar steps to in Scenario I. Forward propagation for required velocities/accelerations of all frames is carried out according to (6.14) and its time derivative. The local control and adaptation laws are designed based on (6.16) and (6.17), respectively. Finally, upon backward recursion of required forces by (6.18), the control inputs are calculated from (6.19).

7.2 Second-order Nonholonomic Constraints

To obtain the dependent part of the generalized required velocity vector i.e., $\mathcal{V}_u^{(r)}$, the nonholonomic constraints described by (6.24) are utilized. Replacing $\mathcal{V}^{(r)}$, $\mathcal{V}_v^{(r)}$, and $\dot{\mathcal{V}}_v^{(r)}$ respectively from (7.5), (7.12), and the time-derivative of (7.12), in (6.24) and rearranging the equation yields,

$$\mathcal{A}_u \, \dot{\mathcal{V}}_u^{(r)} = \mathcal{B}_u, \tag{7.13}$$

where the expressions of $\mathcal{A}_u \in \Re^{2\times 2}$ and $\mathcal{B}_u \in \Re^2$ are given by,

$$\begin{aligned}
\mathcal{A}_u &= \mathcal{A}\left(\mathbf{T}_1 \, \mathcal{L}_v^{\dagger} \, \mathcal{L}_u - \mathbf{T}_2\right) \\
\mathcal{B}_u &= -\mathcal{B} + \mathcal{A}\mathbf{T}_1 \left[\frac{d(\mathcal{L}_v^{\dagger})}{dt} \left(\begin{bmatrix} \boldsymbol{V}_{B_{n+1}}^{(r)} \\ {}^{I}\dot{\boldsymbol{P}}_{B_0}^{(r)} \\ \dot{\Psi}_{z_{B_0}}^{(r)} \end{bmatrix} - \mathcal{L}_u \mathcal{V}_u^{(r)} \right) + \mathcal{L}_v^{\dagger} \left(\frac{d}{dt} \begin{bmatrix} \boldsymbol{V}_{B_{n+1}}^{(r)} \\ {}^{I}\dot{\boldsymbol{P}}_{B_0}^{(r)} \\ \dot{\Psi}_{z_{B_0}}^{(r)} \end{bmatrix} - \frac{d(\mathcal{L}_u)}{dt} \mathcal{V}_u^{(r)} \right) \right]
\end{aligned} \tag{7.14}$$

and $\mathcal{A}$ and $\mathcal{B}$ were defined in (6.26).

Given the numerical values of $\mathcal{A}_u$ and $\mathcal{B}_u$ at each time step, (7.13) represents two linear

algebraic equations to be solved for $\dot{\mathcal{V}}_u^{(r)}$ and obtain $\frac{d}{dt}(\omega_{x_{B_0}}^{(r)})$ and $\frac{d}{dt}(\omega_{y_{B_0}}^{(r)})$. Finally, $\mathcal{V}_u^{(r)}$ is obtained through numerical integration of $\dot{\mathcal{V}}_v^{(r)}$.

7.3 Control Algorithm Implementation

Fig. 7.1 depicts a block diagram of the model-based motion control algorithm. The steps involved in the implementation of this control algorithm are outlined below in chronological order.

i- Calculation of the end-effector and the base required velocities

- To fulfill the manipulation task, the required velocities/accelerations of the end-effector and the base are designed by (7.1), (7.2), (7.7), (7.8), and the time-derivatives of these equations.

ii- Resolution of kinematic redundancy while taking the under-actuation into account

- The integrator of dependent required accelerations, initialized at the measured angular velocities of the base, yields $\mathcal{V}_u^{(r)}$.
- Given $\boldsymbol{V}_{B_{n+1}}^{(r)}$, ${}^I\dot{\boldsymbol{P}}_{B_0}^{(r)}$, $\dot{\Psi}_{z_{B_0}}^{(r)}$, and $\mathcal{V}_u^{(r)}$, the independent required velocities are obtained from (7.12). This completely determines $\mathcal{V}^{(r)}$.
- Given all these variables, the two linear algebraic equations described by (7.13) are solved for $\frac{d}{dt}(\omega_{x_{B_0}}^{(r)})$ and $\frac{d}{dt}(\omega_{y_{B_0}}^{(r)})$.
- Given $\dot{\mathcal{V}}_u^{(r)}$, the time-derivative of (7.12) is used to determine $\dot{\mathcal{V}}_v^{(r)}$. This completes the derivation of $\dot{\mathcal{V}}^{(r)}$.

The remainder of the algorithm identically follows Sections (iii), (iv), and (v) of the first control scenario implementation.

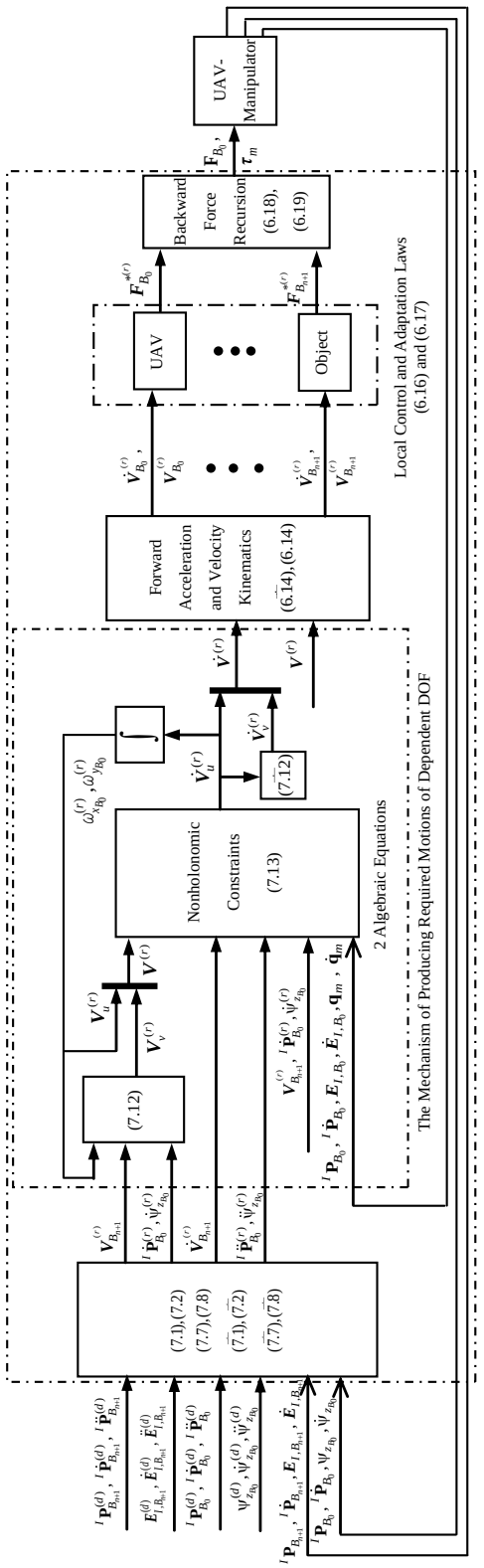

Figure 7.1: Block diagram of the proposed VDC algorithm for Scenario II. $\omega_{xB_0}^{(r)}$ and $\omega_{yB_0}^{(r)}$ are the components of $\boldsymbol{\omega}_{B_0}^{(r)}$. The $\dot{(i)}$ indicates time derivative of the ith equation. Also, the solid arrows indicate the directions of the signals flows while the other ones show feed-back signals.

Remark. *It should be mentioned that this implementation requires measurements of the base position, attitude and linear/angular velocity as well as the manipulator joint positions/velocities and end-effector position/orientation.*

Remark. *The algorithm proposed here provides for independent control of the end-effector position/orientation as well as the base position and yaw orientation i.e.,* 10 *DOFs. Also, recall that two DOFs of the base are released to meet the motion constraints. Consequently, a minimum number of six manipulator DOFs are needed to achieve the control objectives considered here. Otherwise, the number of independently controllable DOFs is reduced. In such a case, not all 6 DOFs of the the end-effector can be controlled. For example, to control the end-effector position, the manipulator needs to have at least three DOFs.*

Control algorithm development and implementation for Scenario IV follow similar steps to what described in this chapter for Scenario II. In Scenario IV, the UAV altitude and orientation as well as the end-effector position and orientation are controlled independently. Therefore, the required end-effector velocities are designed according to (7.1) and (7.2). Moreover, the base required angular velocity and an altitude reference velocity are designed according to (6.9) and (6.10), respectively. The dependent required linear velocities/accelerations of the UAV are numerically derived from a similar equation to (7.13) which is obtained by developing the under-actuation constraints. Given these dependent components, the possible kinematic redundancy is resolved with a similar equation to (7.12) and the rest of the generalized velocities/accelerations are determined. The remainder of the algorithm remains the same as Scenario II.

Chapter 8

Control Design Considering Propeller Dynamics

This chapter extends the control algorithms developed in Chapter 6 and 7 for the UAV-Manipulator by taking into account the dynamics of the UAV propellers in model-based control design. Although the propellers mass/inertia are small compared to the UAV base, their angular speed is significant. As a result, in scenarios where the angular momentum of these propellers are not well-balanced, such as when there is an odd number of propellers, the dynamics introduced by the propellers cannot be ignored. Moreover, the aerodynamic thrusts, which are used to manipulate the entire system and commonly considered as UAV control inputs, are proportional to the propeller speeds. Therefore, these forces cannot instantly change and there is a transient state in their generation. Ignoring this transient phase in actuation may undermine the overall system performance if fast motions are planned. In this chapter, to fully incorporate the propeller dynamics into the system model, each propeller is considered as a rigid body and the control/adaptation laws are designed for them as well. In this case, the UAV control inputs are the low-level propeller motor torques.

In the previous developments, the resultant force/torque of the propeller produced thrusts at the center of the UAV base, was considered as a direct control actuation for the UAV base. This resultant extended force vector was composed of a total thrust force and three actuation torques. The actuation forces in the body-fixed $x_{B_0} - y_{B_0}$ directions were zeros. As a result, the entire system was under-actuated in two DOFs.

In this chapter, however, the control inputs are narrowed down to the low level propeller motor torques. Since each propeller is considered as a rigid body, the individual thrust forces are applied to each propeller but not the UAV base. Consequently, there is no direct actuation for the base; the extended force vector at the base is entirely zero in this case and the system is under-actuated in six DOFs. However, as previously discussed, the independent control of the propeller DOFs is of no interest. The propeller motion variables can be used to indirectly actuate the UAV base which lacks any force/torque actuation. Therefore, in this chapter, the propeller motion variables are part of the dependent set of the generalized coordinates.

Note that the set of m propeller can only provide the base with four out of six possible actuation forces/torques. This implies that two of the remaining generalized coordinates should still be included in the dependent set. In summary, in both control strategies in this chapter, $4 + n$ coordinates will be controlled independently and $2 + m$ coordinates will form the dependent set to address the system under-actuation in six DOFs.

In the first section of this chapter, the control design is reformulated for the first control scenario. The second section discusses the extension of the second control scenario. For brevity and to avoid repetition of the formulations, we refer to the equations of the previous chapters when possible.

8.1 Control Scenario I Revisited

8.1.1 Motion Control Design

In view of (3.10), the generalized required velocity vector is defined as,

$$\mathcal{V}^{(r)} \triangleq \left[\boldsymbol{V}_{B_0}^{(r)^T}, \dot{\boldsymbol{q}}_m^{(r)^T}, \dot{\boldsymbol{q}}_p^{(r)^T} \right]^T \in \Re^{N_d}, \tag{8.1}$$

where $\boldsymbol{V}_{B_0}^{(r)} \triangleq \left[\boldsymbol{v}_{B_0}^{(r)^T}, \boldsymbol{\omega}_{B_0}^{(r)^T} \right]^T \in \Re^6$ denotes the base extended required velocity, $\dot{\boldsymbol{q}}_m^{(r)} \in \Re^n$ is the required joint velocity vector, and $\dot{\boldsymbol{q}}_p^{(r)} \in \Re^m$ is the required propeller velocity vector. Here, the independent generalized coordinate set includes the UAV altitude and its Euler orientation as well as the manipulator joint positions. Therefore, $v_{z_{B_0}}^{(r)}$, $\boldsymbol{\omega}_{B_0}^{(r)}$, and $\dot{\boldsymbol{q}}_m^{(r)}$ form the independent part of the generalized required velocity vector whereas $v_{x_{B_0}}^{(r)}$, $v_{y_{B_0}}^{(r)}$, and $\dot{\boldsymbol{q}}_p^{(r)}$ form the dependent part.

Given the desired trajectories for the manipulator joint positions ($\boldsymbol{q}_m^{(d)}$) and the base orientation ($\boldsymbol{E}_{I,B_0}^{(d)}$), the corresponding required velocities for these DOFs are designed according to (6.8) and (6.9), respectively. Moreover, given the base desired altitude ($Z_{B_0}^{(d)}$), a reference velocity along the inertial Z axis is designed for the base according to (6.10). Note that the UAV's required linear velocity $\boldsymbol{v}_{B_0}^{(r)}$ has to be obtained such that not only the motion constraints are satisfied but also this vertical reference velocity is achieved. To this end, once the dependent components of $\boldsymbol{v}_{B_0}^{(r)}$ are found through satisfaction of the motion constraint, the $v_{z_{B_0}}^{(r)}$ is obtained from (6.11).

Similar to the two dependent components of $\boldsymbol{v}_{B_0}^{(r)}$ discussed above, the propeller required velocities $\dot{\boldsymbol{q}}_p^{(r)}$ are also dependent since they are employed to actuate the base in four DOFs indirectly. These dependent components of $\mathcal{V}^{(r)}$ will also be derived in the next subsection through imposing the motion constraints. Given all the dependent components, i.e., $v_{x_{B_0}}^{(r)}$,

$v_{y_{B_0}}^{(r)}$ and $\dot{\boldsymbol{q}}_p^{(r)}$, the generalized required velocity vector $\mathcal{V}^{(r)}$ is completely determined, and the required velocities of all bodies are obtained by a forward propagation in each kinematic chain as follows,

$$\begin{aligned}
&\boldsymbol{V}_{B_{i+1}}^{(r)} = \mathbf{U}_{B_{i+1}}^{{B_i}^T} \boldsymbol{V}_{B_i}^{(r)} + \dot{q}_{m_{i+1}}^{(r)} \boldsymbol{z},\ i = 0, \cdots, n-1 \\
&\boldsymbol{V}_{B_{n+1}}^{(r)} = \mathbf{U}_{B_{n+1}}^{{B_n}^T} \boldsymbol{V}_{B_n}^{(r)}, \\
&\boldsymbol{V}_{P_i}^{(r)} = \mathbf{U}_{P_i}^{{B_0}^T} \boldsymbol{V}_{B_0}^{(r)} + \dot{q}_{p_i}^{(r)} \boldsymbol{z}',\quad i = 1, \ldots, m.
\end{aligned} \tag{8.2}$$

By defining the overall extended required velocity vector similar to (3.12), one can rewrite (8.2) in the following form,

$$\mathcal{V}_e^{(r)} = \mathcal{J}_e \mathcal{V}^{(r)}. \tag{8.3}$$

where $\mathcal{J}_e$ is the generalized Jacobian matrix defined in (3.14). The local control and adaptation laws can be designed in a similar approach to Chapter 6. Having the required motion of each body in the system, Equations (4.1), (4.2), and (4.3) are employed to design the net extended required force to be applied at each body where $\alpha \in \{B_0, \cdots, B_{n+1}, P_1, \cdots, P_m\}$.

Once the net extended required forces are designed, the required interaction force/moment vectors are obtained by a backward recursion in all kinematic chains, i.e.,

$$\begin{aligned}
&\boldsymbol{F}_{B_{n+1}}^{(r)} = \boldsymbol{F}_{B_{n+1}}^{*(r)}, \\
&\boldsymbol{F}_{B_i}^{(r)} = \boldsymbol{F}_{B_i}^{*(r)} + \mathbf{U}_{B_{i+1}}^{B_i} \boldsymbol{F}_{B_{i+1}}^{(r)},\ i = n, \cdots, 1, \\
&\boldsymbol{F}_{P_i}^{(r)} = \boldsymbol{F}_{P_i}^{*(r)} - \boldsymbol{F}_{P_i}^{aero},\ i = 1, \cdots, m \\
&\boldsymbol{F}_{B_0}^{(r)} = \boldsymbol{F}_{B_0}^{*(r)} + \mathbf{U}_{B_1}^{B_0} \boldsymbol{F}_{B_1}^{(r)} + \sum_{i=1}^{m} \mathbf{U}_{P_i}^{B_0} \boldsymbol{F}_{P_i}^{(r)}.
\end{aligned} \tag{8.4}$$

Finally, the actual control inputs are derived from the required interaction forces/torques as,

$$\begin{aligned} \tau_{m_i} &= \boldsymbol{z}^T \boldsymbol{F}_{B_i}^{(r)},\ i = 1, \cdots, n, \\ \tau_{p_i} &= \boldsymbol{z'}^T \boldsymbol{F}_{P_i}^{(r)},\ i = 1, \cdots, m. \end{aligned} \tag{8.5}$$

8.1.2 Second-Order Nonholonomic Constraints

Similar to Chapter 6, the control algorithm guarantees that no actuation is requested in the unactuated DOFs; this is achieved by imposing the following constraints,

$$\mathbf{F}_{B_0}^{(r)} = 0. \tag{8.6}$$

Equation (8.6) ensures the consistency of the designed generalized required velocity vector with the system motion constraints. In the remainder of this section, this equation is further developed and the dependent motion variables are derived. Similar to (3.33), let $\mathcal{F}_e^{*(r)} \in \Re^{6N_b}$ be the extended required net force/moment vector of all bodies and $\mathcal{U}_{B_0} \triangleq \left[\mathcal{U}_B^{B_0}, \mathcal{U}_P^{B_0}\right] \in \Re^{6\times 6N_b}$ be an overall extended transformation matrix where,

$$\begin{aligned} \mathcal{U}_B^{B_0} &\triangleq \left[\mathbf{I}_{6\times 6}, \mathbf{U}_{B_1}^{B_0}, \cdots, \mathbf{U}_{B_{n+1}}^{B_0}\right] \in \Re^{6\times 6(n+2)}, \\ \mathcal{U}_P^{B_0} &\triangleq \left[\mathbf{U}_{P_1}^{B_0}, \cdots, \mathbf{U}_{P_m}^{B_0}\right] \in \Re^{6\times 6m}. \end{aligned} \tag{8.7}$$

Then a backward force propagation through (8.4) yields,

$$\boldsymbol{F}_{B_0}^{(r)} = \mathcal{U}_{B_0} \mathcal{F}_e^{*(r)} - \mathcal{U}_P^{B_0} \mathcal{F}_p^{aero} \tag{8.8}$$

where $\mathcal{F}_p^{aero} \triangleq \left[\boldsymbol{F}_{P_1}^{aero^T}, \cdots, \boldsymbol{F}_{P_m}^{aero^T}\right]^T \in \Re^{6m}$. Moreover using (4.1) and (4.2), it follows that,

$$\mathcal{F}_e^{*^{(r)}} = \hat{\mathcal{M}}_e \mathcal{J}_e \dot{\mathcal{V}}^{(r)} + \hat{\mathcal{M}}_e \dot{\mathcal{J}}_e \mathcal{V}^{(r)} + \hat{\mathcal{C}}_e \mathcal{J}_e \mathcal{V}^{(r)} + \hat{\mathcal{G}}_e + \mathcal{K}_e \mathcal{J}_e \left(\mathcal{V}^{(r)} - \mathcal{V}\right), \tag{8.9}$$

where $\hat{\mathcal{M}}_e$, $\hat{\mathcal{C}}_e$ and $\hat{\mathcal{G}}_e$ are estimates of the variables defined in (3.35), and $\mathcal{K}_e = diag\left(\mathbf{K}_{B_0}, \cdots, \mathbf{K}_{B_{n+1}}, \mathbf{K}_{P_1}, \cdots, \mathbf{K}_{P_m}\right)$. Finally, substituting (8.9) in (8.8) and applying (8.6) yields,

$$\mathcal{A}\dot{\mathcal{V}}^{(r)} = \mathcal{B}, \tag{8.10}$$

where,

$$\begin{aligned} &\mathcal{A} = \mathcal{U}_{B_0} \hat{\mathcal{M}}_e \mathcal{J}_e \in \Re^{6 \times N_d}, \\ &\mathcal{B} = \mathcal{U}_P^{B_0} \mathcal{F}_p^{aero} - \mathcal{U}_{B_0} \left[\hat{\mathcal{M}}_e \dot{\mathcal{J}}_e \mathcal{V}^{(r)} + \hat{\mathcal{C}}_e \mathcal{J}_e \mathcal{V}^{(r)} + \hat{\mathcal{G}}_e + \mathcal{K}_e \mathcal{J}_e \left(\mathcal{V}^{(r)} - \mathcal{V}\right)\right] \in \Re^6. \end{aligned} \tag{8.11}$$

Equation (8.10) characterizes the nonholonomic constraints and describes the dependency of the motion variables. In this equation, $\mathcal{A}$ and $\mathcal{B}$ are given numerically in each instant of time in the control algorithm implementation. Moreover, as previously mentioned, the freely designed required acceleration terms, namely $\frac{d}{dt}(\dot{\boldsymbol{q}}_m^{(r)})$ and $\frac{d}{dt}(\boldsymbol{\omega}_{B_0}^{(r)})$, are given by the time derivatives of 6.8 and 6.9, respectively. Therefore, (8.10) represents six explicit constraints on $\frac{d}{dt}(\boldsymbol{v}_{B_0}^{(r)})$ and $\frac{d}{dt}(\dot{\boldsymbol{q}}_p^{(r)})$, which through a rearrangement of this equation are given by,

$$\mathcal{A}_w \begin{bmatrix} \frac{d}{dt}(\boldsymbol{v}_{B_0}^{(r)}) \\ \frac{d}{dt}(\dot{\boldsymbol{q}}_p^{(r)}) \end{bmatrix} = \mathcal{B}_w, \tag{8.12}$$

where $\mathcal{A}_w \in \Re^{6 \times (3+m)}$ and $\mathcal{B}_w \in \Re^6$ are still numerically known in each iteration of the

control implementation. Equation (8.12) represents six linear algebraic equations for the unknown dependent required accelerations.

To solve these equations, an inspection reveals that in the first two rows of $\mathcal{A}_w$ the entries corresponding to $\frac{d}{dt}(\dot{\boldsymbol{q}}_p^{(r)})$ are zero. Therefore, the only unknown variables in the first two equations of (8.12) are $\frac{d}{dt}(\boldsymbol{v}_{B_0}^{(r)})$. On the other hand, the constraint (6.11) on $\boldsymbol{v}_{B_0}^{(r)}$ should always be satisfied over time. Hence, the time derivative of (6.11) reveals another constraint on $\frac{d}{dt}(\boldsymbol{v}_{B_0}^{(r)})$. This along with the first two constraints in (8.12) constitute a set of three linear algebraic equations to be solved for $\frac{d}{dt}(\boldsymbol{v}_{B_0}^{(r)})$.

Once $\frac{d}{dt}(\boldsymbol{v}_{B_0}^{(r)})$ is determined, the last four equations in (8.12) can be solved for the m unknown variables $\frac{d}{dt}(\dot{\boldsymbol{q}}_p^{(r)})$. These equations have a unique solution when $m = 4$. In the case of redundant propellers, i.e. $m > 4$, a secondary objective can also be achieved while satisfying the motion constraints. Minimum-energy consumption in actuation is an example of such an objective for which optimization algorithms are needed within the controller implementation. Minimum actuation-norm is another example with a closed-form solution, which is desirable for real-time implementation. The propeller redundancy can also be employed to robustify the control system against possible system faults.

Finally, the dependent required velocities i.e $v_{x_{B_0}}^{(r)}$, $v_{y_{B_0}}^{(r)}$ and $\dot{\boldsymbol{q}}_p^{(r)}$ are obtained through numerical integration of the related required accelerations. At this point, the required motion variables, i.e., velocities and accelerations, for all system DOFs are determined. Accordingly, forward required velocity/acceleration propagation is performed in all kinematic chains followed by backward required force/moment recursion and the control inputs are obtained. The following section further clarifies the steps involved in the implementation of the proposed control algorithm.

8.1.3 Control Algorithm Implementation

Fig. 8.1 depicts a block diagram of the motion control algorithm in the first scenario. There are five major consecutive sections involved in this algorithm which are summarized below respectively.

i - Calculation of independent required motion variables

- Given the desired trajectories of manipulator joints and base orientation, (6.8), (6.9) and the time derivative of these equations are employed to design $\dot{\boldsymbol{q}}_m^{(r)}$, $\boldsymbol{\omega}_{B_0}^{(r)}$, $\frac{d}{dt}(\dot{\boldsymbol{q}}_m^{(r)})$ and $\frac{d}{dt}(\boldsymbol{\omega}_{B_0}^{(r)})$.

ii - Derivation of dependent required motion variables

- The integration of dependent required accelerations yields $v_{x_{B_0}}^{(r)}$, $v_{y_{B_0}}^{(r)}$ and $\dot{\boldsymbol{q}}_p^{(r)}$. This integration is initialized at the real linear velocity of the base and the propeller velocities obtained from measurements.
- Given the desired trajectory of the base altitude, (6.10) is used to design $\dot{Z}_{B_0}^{(r)}$.
- Given $v_{x_{B_0}}^{(r)}$, $v_{y_{B_0}}^{(r)}$ and $\dot{Z}_{B_0}^{(r)}$, (6.11) is used to find $v_{z_{B_0}}^{(r)}$. At this point, all the components of $\mathcal{V}^{(r)}$ are determined.
- The first two equations of (8.12) and the time derivative of (6.11) are solved simultaneously to obtain $\frac{d}{dt}(\boldsymbol{v}_{B_0}^{(r)})$.
- Given $\frac{d}{dt}(\boldsymbol{v}_{B_0}^{(r)})$, the last four equations of (8.12) are solved for $\frac{d}{dt}(\dot{\boldsymbol{q}}_p^{(r)})$. This completes the derivation of $\dot{\mathcal{V}}^{(r)}$.

iii - Forward propagation of required motion variables

- Given $\mathcal{V}^{(r)}$ and $\dot{\mathcal{V}}^{(r)}$, the required velocity/acceleration for all system bodies are obtained using (8.3) and its time derivative.

iv - Required local control input allocation

- Given $\boldsymbol{V}_{\alpha}^{(r)}$ and $\frac{d}{dt}(\boldsymbol{V}_{\alpha}^{(r)})$, the required net force $\boldsymbol{F}_{\alpha}^{*^{(r)}}$ for each body is designed using (4.1) and (4.3).

v - Backward recursion of required forces/moments

- Given $\boldsymbol{F}_{\alpha}^{*^{(r)}}$ for all bodies, the required interaction forces/moments $\boldsymbol{F}_{\alpha}^{(r)}$ are derived from (8.4).
- Finally, the control inputs τ_{m_i} and τ_{p_i} are obtained from (8.5).

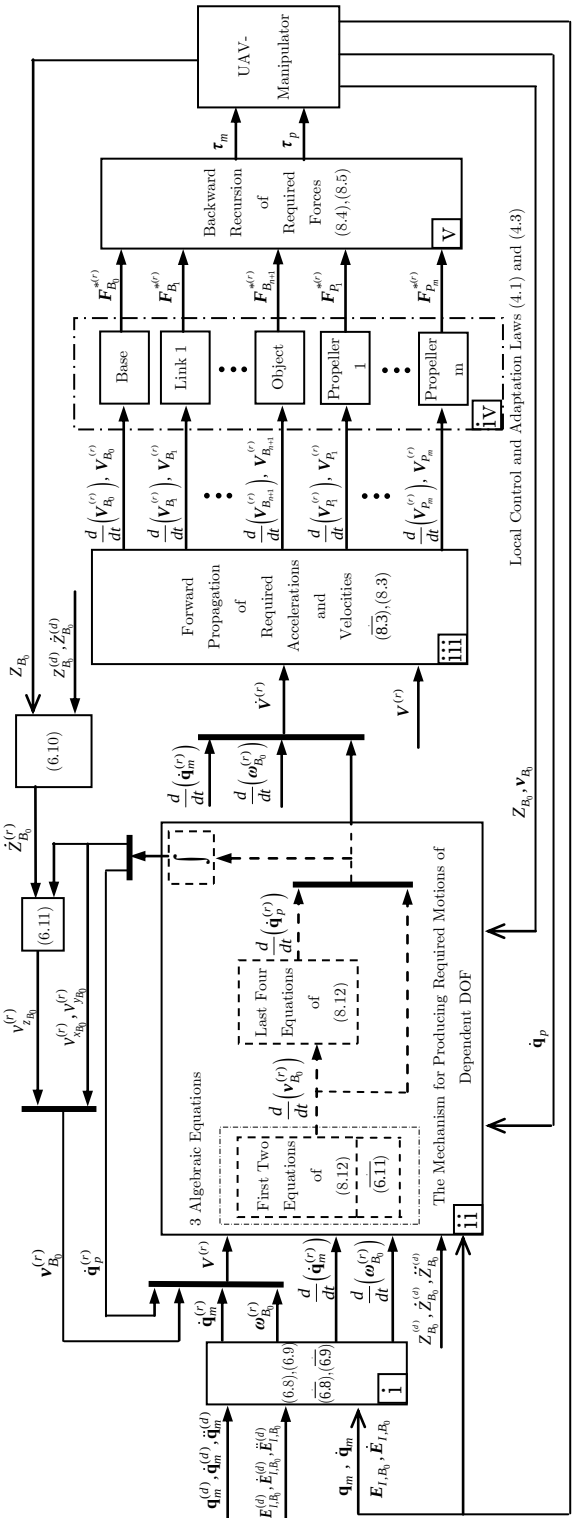

Figure 8.1: Block diagram illustrating the implementation of the proposed VDC algorithm in Scenario I including the propeller dynamics. $v_{xB_0}^{(r)}$, $v_{yB_0}^{(r)}$ and $v_{zB_0}^{(r)}$ are the components of $\boldsymbol{\nu}_{B_0}^{(r)}$. The $\dot{(i)}$ indicates time derivative of the ith equation. Also, the solid arrows indicate the directions of the signals flows while the other ones show feed-back signals.

Remark. *: It should be mentioned that this implementation requires measurements of the base altitude, attitude and linear/angular velocity as well as the manipulator joint positions/velocities and propeller speeds.*

8.2 Control Scenario II Revisited

8.2.1 Motion Control Design

Similar to what was done in Chapter 7, given the desired position vector ${}^{I}\boldsymbol{P}_{B_{n+1}}^{(d)}$ and the vector of desired Euler angles $\boldsymbol{E}_{I,B_{n+1}}^{(d)}$ of the end-effector, the body-expressed required linear and angular velocities of B_{n+1} are designed according to (7.1) and (7.2), respectively.

The kinematic mapping between the end-effector required linear/angular velocities $\boldsymbol{V}_{B_{n+1}}^{(r)}$ and the generalized required velocity vector $\mathcal{V}^{(r)}$ defined in (8.1) is represented by,

$$\boldsymbol{V}_{B_{n+1}}^{(r)} = \mathcal{J}\mathcal{V}^{(r)}, \tag{8.13}$$

where the Jacobian matrix $\mathcal{J} \in \Re^{6\times N_d}$ is extracted from (3.14) as,

$$\mathcal{J} = \left[\mathbf{U}_{B_{n+1}}^{{B_0}^T}, \mathbf{U}_{B_{n+1}}^{{B_1}^T}\boldsymbol{z}, \cdots, \mathbf{U}_{B_{n+1}}^{{B_n}^T}\boldsymbol{z}, \mathbf{0}_{6\times m}\right]. \tag{8.14}$$

Another control objective in the second scenario was to control the UAV base position and yaw orientation. The reference velocities for these DOFs are designed according to (7.7) and (7.8), respectively. This establishes four more constraints on the generalized required velocity vector.

Similar to Chapter 7, in order to resolve the potential kinematic redundancy in (8.13) while considering the system under-actuation, the generalized required velocity vector

is decomposed into two sets. The independent set includes $\boldsymbol{v}_{B_0}^{(r)}$, $\omega_{z_{B_0}}^{(r)}$, and $\dot{\boldsymbol{q}}_m^{(r)}$ while $\omega_{x_{B_0}}^{(r)}$,$\omega_{y_{B_0}}^{(r)}$, and $\dot{\boldsymbol{q}}_p^{(r)}$ form the dependent set. Note that $\boldsymbol{v}_{B_0}^{(r)}$ is included in the independent set since the base position here is controlled independently.

Let $\mathbf{T} \triangleq [\mathbf{T}_1\, \mathbf{T}_2] \in \Re^{N_d \times N_d}$ be a constant reordering matrix such that,

$$\mathcal{V}^{(r)} = [\mathbf{T}_1\, \mathbf{T}_2] \begin{bmatrix} \mathcal{V}_v^{(r)} \\ \mathcal{V}_u^{(r)} \end{bmatrix}, \tag{8.15}$$

where, by definition, $\mathcal{V}_v^{(r)} \triangleq \left[\boldsymbol{v}_{B_0}^{(r)^T}, \omega_{z_{B_0}}^{(r)}, \dot{\boldsymbol{q}}_m^{(r)^T}\right]^T \in \Re^{4+n}$ is the independent generalized required velocity vector and $\mathcal{V}_u^{(r)} \triangleq \left[\omega_{x_{B_0}}^{(r)}, \omega_{y_{B_0}}^{(r)}, \dot{\boldsymbol{q}}_p^{(r)^T}\right]^T \in \Re^{2+m}$ is the dependent one. Substituting (8.15) in (8.13) results in,

$$\boldsymbol{V}_{B_{n+1}}^{(r)} = \mathcal{J}_v \mathcal{V}_v^{(r)} + \mathcal{J}_u \mathcal{V}_u^{(r)}, \tag{8.16}$$

where $\mathcal{J}_v = \mathcal{J}\,\mathbf{T}_1 \in \Re^{6\times(n+4)}$ and $\mathcal{J}_u = \mathcal{J}\,\mathbf{T}_2 \in \Re^{6\times(2+m)}$. Moreover, the four extra constraints on the generalized required velocity vector imposed by (7.7) and (7.8), can be written as,

$$\begin{bmatrix} {}^I\dot{\boldsymbol{P}}_{B_0}^{(r)} \\ \dot{\Psi}_{z_{B_0}}^{(r)} \end{bmatrix} = \mathcal{J}_v' \mathcal{V}_v^{(r)} + \mathcal{J}_u' \mathcal{V}_u^{(r)}, \tag{8.17}$$

where $\mathcal{J}_v' = \mathcal{Q}\mathbf{T}_1' \in \Re^{4\times(n+4)}$ and $\mathcal{J}_u' = \mathcal{Q}\mathbf{T}_2' \in \Re^{4\times(2+m)}$ while $\mathbf{T}_1' \in \Re^{6\times(n+4)}$ and $\mathbf{T}_2' \in \Re^{6\times(2+m)}$ are constant reordering matrices such that $\boldsymbol{V}_{B_0}^{(r)} = [\mathbf{T}_1'\mathbf{T}_2'][\mathcal{V}_v^{(r)^T}\mathcal{V}_u^{(r)^T}]^T$ and $\mathcal{Q}$ is given by (7.10). The kinematic constraints (8.16) and (8.17) on the generalized required velocity vector should be taken into account simultaneously in order to determine

$\mathcal{V}^{(r)}$. These two sets of kinematic constraints are represented together as,

$$\mathcal{L}_v \mathcal{V}_v^{(r)} = \begin{bmatrix} \boldsymbol{V}_{B_{n+1}}^{(r)} \\ {}^I\dot{\boldsymbol{P}}_{B_0}^{(r)} \\ \dot{\Psi}_{z_{B_0}}^{(r)} \end{bmatrix} - \mathcal{L}_u \mathcal{V}_u^{(r)}, \tag{8.18}$$

in which $\mathcal{L}_v = [\mathcal{J}_v^T \mathcal{J}_v'^T]^T \in \Re^{10\times(n+4)}$ and $\mathcal{L}_u = [\mathcal{J}_u^T \mathcal{J}_u'^T]^T \in \Re^{10\times(2+m)}$.

Equation (8.18) can be used to determine the independent set of the required velocities. In this equation, $\boldsymbol{V}_{B_{n+1}}^{(r)}$, ${}^I\dot{\boldsymbol{P}}_{B_0}^{(r)}$, and $\dot{\Psi}_{z_{B_0}}^{(r)}$ are given by (7.1), (7.2), (7.7), and (7.8). Also, $\mathcal{V}_u^{(r)}$ is derived by exploiting the nonholonomic constraints. Without loss of generality, among all possible solutions for (8.18), the minimum norm solution is utilized here,

$$\mathcal{V}_v^{(r)} = \mathcal{L}_v^{\dagger} \left(\begin{bmatrix} \boldsymbol{V}_{B_{n+1}}^{(r)} \\ {}^I\dot{\boldsymbol{P}}_{B_0}^{(r)} \\ \dot{\Psi}_{z_{B_0}}^{(r)} \end{bmatrix} - \mathcal{L}_u \mathcal{V}_u^{(r)} \right), \tag{8.19}$$

where $\mathcal{L}_v^{\dagger}$ is the right pseudo-inverse of $\mathcal{L}_v$.

Once the generalized required velocity vector $\mathcal{V}^{(r)}$ is completely obtained, the remainder of the control algorithm follows the same procedure as the control development in Chapter 6. Forward propagation for required velocities/accelerations of all frames is carried out according to (8.3) and its time derivative. The local control and adaptation laws are designed based on (4.1) and (4.3), respectively. Finally, upon backward recursion of required forces by (8.4), the control inputs are calculated from (8.5).

8.2.2 Second-Order Nonholonomic Constraints

To obtain the dependent part of the generalized required velocity vector, $\mathcal{V}_u^{(r)}$, the nonholonomic constraints described by (8.10) are utilized. Replacing $\mathcal{V}^{(r)}$, $\mathcal{V}_v^{(r)}$, and $\dot{\mathcal{V}}_v^{(r)}$ respectively from (8.15), (8.19), and the time-derivative of (8.19), in (8.10) and rearranging the equation yields,

$$\mathcal{A}_u \, \dot{\mathcal{V}}_u^{(r)} = \mathcal{B}_u, \tag{8.20}$$

where the expressions of $\mathcal{A}_u \in \Re^{6\times(2+m)}$ and $\mathcal{B}_u \in \Re^6$ are given by,

$$\begin{aligned}
\mathcal{A}_u &= \mathcal{A}\left(\mathbf{T}_1 \, \mathcal{L}_v^{\dagger} \, \mathcal{L}_u - \mathbf{T}_2\right) \\
\mathcal{B}_u &= -\mathcal{B} + \mathcal{A}\mathbf{T}_1 \left[\frac{d(\mathcal{L}_v^{\dagger})}{dt} \left(\begin{bmatrix} \boldsymbol{V}_{B_{n+1}}^{(r)} \\ {}^{I}\dot{\boldsymbol{P}}_{B_0}^{(r)} \\ \dot{\Psi}_{z_{B_0}}^{(r)} \end{bmatrix} - \mathcal{L}_u \mathcal{V}_u^{(r)} \right) + \mathcal{L}_v^{\dagger} \left(\frac{d}{dt} \begin{bmatrix} \boldsymbol{V}_{B_{n+1}}^{(r)} \\ {}^{I}\dot{\boldsymbol{P}}_{B_0}^{(r)} \\ \dot{\Psi}_{z_{B_0}}^{(r)} \end{bmatrix} - \frac{d(\mathcal{L}_u)}{dt} \mathcal{V}_u^{(r)} \right) \right]
\end{aligned} \tag{8.21}$$

and $\mathcal{A}$ and $\mathcal{B}$ are defined in (8.11).

Given the numerical values of $\mathcal{A}_u$ and $\mathcal{B}_u$ at each time step, (8.20) represents six linear algebraic equations to be solved for $m+2$ variables of $\dot{\mathcal{V}}_u^{(r)}$. An inspection of $\mathcal{A}_u$ shows that $\frac{d}{dt}(\omega_{x_{B_0}}^{(r)})$ and $\frac{d}{dt}(\omega_{y_{B_0}}^{(r)})$ are the only unknown variables in the fourth and the fifth equations of (8.20). Consequently, these two equations are solved first to obtain $\frac{d}{dt}(\omega_{x_{B_0}}^{(r)})$ and $\frac{d}{dt}(\omega_{y_{B_0}}^{(r)})$. Given these variables, the remaining four equations are solved for the m propeller required angular accelerations $\frac{d}{dt}(\dot{\boldsymbol{q}}_p^{(r)})$. The previous discussion about the unique or redundant solutions of these equations holds true here as well. Finally, $\mathcal{V}_u^{(r)}$ is obtained through numerical integration of $\dot{\mathcal{V}}_v^{(r)}$.

8.2.3 Control Algorithm Implementation

The steps involved in the implementation of the control algorithm in the second scenario are outlined below in chronological order.

i- Calculation of the end-effector and the base required velocities

- To fulfill the manipulation task, the required velocities/accelerations of the end-effector and the base are designed by (7.1), (7.2), (7.7), (7.8), and the time-derivatives of these equations.

ii- Resolution of kinematic redundancy while taking the under-actuation into account

- The integration of dependent required accelerations, initialized at the measured angular velocities of the base and propellers, yields $\mathcal{V}_u^{(r)}$.
- Given $\boldsymbol{V}_{B_{n+1}}^{(r)}$, ${}^I\dot{\boldsymbol{P}}_{B_0}^{(r)}$,$\dot{\Psi}_{z_{B_0}}^{(r)}$, and $\mathcal{V}_u^{(r)}$, the independent required velocities are obtained from (8.19). This completely determines $\mathcal{V}^{(r)}$.
- Given all these variables, the fourth and the fifth equations of the six linear algebraic equations described by (8.20) are solved for $\frac{d}{dt}(\omega_{x_{B_0}}^{(r)})$ and $\frac{d}{dt}(\omega_{y_{B_0}}^{(r)})$.
- Given $\frac{d}{dt}(\omega_{x_{B_0}}^{(r)})$ and $\frac{d}{dt}(\omega_{y_{B_0}}^{(r)})$, the remaining four equations of (8.20) are solved to obtain $\frac{d}{dt}(\dot{\boldsymbol{q}}_p^{(r)})$.
- Given $\dot{\mathcal{V}}_u^{(r)}$, the time-derivative of (8.19) is used to determine $\dot{\mathcal{V}}_v^{(r)}$. This completes the derivation of $\dot{\mathcal{V}}^{(r)}$.

The remainder of the algorithm identically follows Sections (iii), (iv), and (v) of the control implementation in Chapter 6.

Chapter 9

Stability Analysis

This chapter studies the stability of the closed-loop system with the proposed control approaches. The stability analysis are performed for the general case where the propeller dynamics are also included in the system control design. The results directly apply to the control design in Chapter 6 and 7 since they can be considered as special cases of the developments in Chapter 8.

The stability analysis of the closed-loop system is carried out in two steps. First, convergence of the system generalized velocity vector $\mathcal{V}$ to the required one $\mathcal{V}^{(r)}$ is proven and then asymptotic convergence of the tracking errors to zero is shown. The following lemmas are instrumental in performing the first step. Similar results can be found in [20]. Also, for notational simplicity, the following vectors are defined,

$$\tilde{\boldsymbol{V}}_\alpha \triangleq \boldsymbol{V}_\alpha^{(r)} - \boldsymbol{V}_\alpha,\ \tilde{\boldsymbol{\theta}}_\alpha \triangleq \boldsymbol{\theta}_\alpha - \hat{\boldsymbol{\theta}}_\alpha,\ \tilde{\boldsymbol{F}}_\alpha^* \triangleq \boldsymbol{F}_\alpha^{*^{(r)}} - \boldsymbol{F}_\alpha^*, \tag{9.1}$$

where $\alpha \in \{B_0, \cdots, B_{n+1}, P_1, \cdots, P_m\}$.

Lemma 1:

For the ith body of the system whose motion is described by (3.21) with its respective control and adaptation laws (4.1) and (4.3), if the following positive definite scalar function is considered,

$$\mathfrak{V}_\alpha \left(\tilde{\boldsymbol{V}}_\alpha, \tilde{\boldsymbol{\theta}}_\alpha \right) = \frac{1}{2} \left(\tilde{\boldsymbol{V}}_\alpha^T \mathbf{M}_\alpha \tilde{\boldsymbol{V}}_\alpha + \tilde{\boldsymbol{\theta}}_\alpha^T \boldsymbol{\Gamma}_\alpha \tilde{\boldsymbol{\theta}}_\alpha \right) \tag{9.2}$$

then,

$$\dot{\mathfrak{V}}_\alpha = -\tilde{\boldsymbol{V}}_\alpha^T \mathbf{K}_\alpha \tilde{\boldsymbol{V}}_\alpha + \tilde{\boldsymbol{V}}_\alpha^T \tilde{\boldsymbol{F}}_\alpha^*, \tag{9.3}$$

holds.

Proof. Taking the time derivative of (9.2) yields,

$$\dot{\mathfrak{V}}_\alpha = \tilde{\boldsymbol{V}}_\alpha^T \mathbf{M}_\alpha \left(\dot{\boldsymbol{V}}_\alpha^{(r)} - \dot{\boldsymbol{V}}_\alpha \right) - \dot{\tilde{\boldsymbol{\theta}}}_\alpha^T \boldsymbol{\Gamma}_\alpha \tilde{\boldsymbol{\theta}}_\alpha,$$

with $\boldsymbol{\theta}_\alpha$ being considered constant. Adding and subtracting the terms $\tilde{\boldsymbol{V}}_\alpha^T \left(\mathbf{C}_\alpha \tilde{\boldsymbol{V}}_\alpha + \boldsymbol{G}_\alpha \right)$ and exploiting (3.21), (4.2) and (4.3) result in,

$$\dot{\mathfrak{V}}_\alpha = \tilde{\boldsymbol{V}}_\alpha^T \left(\mathbf{Y}_\alpha \boldsymbol{\theta}_\alpha - \boldsymbol{F}_\alpha^* \right) - \tilde{\boldsymbol{V}}_\alpha^T \mathbf{C}_\alpha \tilde{\boldsymbol{V}}_\alpha - \tilde{\boldsymbol{V}}_\alpha^T \mathbf{Y}_\alpha \tilde{\boldsymbol{\theta}}_\alpha.$$

Finally, substituting from (9.1), employing (4.1) and in view of the skew-symmetric property of matrix $\mathbf{C}_\alpha$,

$$\dot{\mathfrak{V}}_\alpha = -\tilde{\boldsymbol{V}}_\alpha^T \mathbf{K}_\alpha \tilde{\boldsymbol{V}}_\alpha + \tilde{\boldsymbol{V}}_\alpha^T \tilde{\boldsymbol{F}}_\alpha^*$$

which completes the proof.

□

Lemma 2:

The sum of $\tilde{\boldsymbol{V}}_\alpha^T \tilde{\boldsymbol{F}}_\alpha^*$ for all bodies is null, i.e.

$$\sum_\alpha \tilde{\boldsymbol{V}}_\alpha^T \tilde{\boldsymbol{F}}_\alpha^* = 0. \tag{9.4}$$

Proof. Defining $\tilde{\boldsymbol{F}}_\alpha \triangleq \boldsymbol{F}_\alpha^{(r)} - \boldsymbol{F}_\alpha$ and utilizing force propagation equations (3.24), (3.26), (3.29), (3.31), and (8.4), the following relations hold,

$$\begin{aligned}
&\tilde{\boldsymbol{F}}^*_{B_{n+1}} = \tilde{\boldsymbol{F}}_{B_{n+1}},\\
&\tilde{\boldsymbol{F}}^*_{B_i} = \tilde{\boldsymbol{F}}_{B_i} - \mathbf{U}^{B_i}_{B_{i+1}} \tilde{\boldsymbol{F}}_{B_{i+1}},\ i = n, \cdots, 1,\\
&\tilde{\boldsymbol{F}}^*_{P_i} = \tilde{\boldsymbol{F}}_{P_i},\ i = 1, \cdots, m,\\
&\tilde{\boldsymbol{F}}^*_{B_0} = \tilde{\boldsymbol{F}}_{B_0} - \mathbf{U}^{B_0}_{B_1} \tilde{\boldsymbol{F}}_{B_1} - \sum_{i=1}^{m} \mathbf{U}^{B_0}_{P_i} \tilde{\boldsymbol{F}}_{P_i}.
\end{aligned} \tag{9.5}$$

Similarly, defining $\dot{\tilde{q}}_{m_i} \triangleq \dot{q}^{(r)}_{m_i} - \dot{q}_{m_i}$ and $\dot{\tilde{q}}_{p_i} \triangleq \dot{q}^{(r)}_{p_i} - \dot{q}_{p_i}$, the velocity propagation equations (3.11) and (8.2) yield,

$$\begin{aligned}
&\tilde{\boldsymbol{V}}_{B_{i+1}} = \mathbf{U}^{{B_i}^T}_{B_{i+1}} \tilde{\boldsymbol{V}}_{B_i} + \dot{\tilde{q}}_{m_{i+1}} \boldsymbol{z},\ i = 0, \cdots, n-1,\\
&\tilde{\boldsymbol{V}}_{B_{n+1}} = \mathbf{U}^{{B_n}^T}_{B_{n+1}} \tilde{\boldsymbol{V}}_{B_n},\\
&\tilde{\boldsymbol{V}}_{P_i} = \mathbf{U}^{{B_0}^T}_{P_i} \tilde{\boldsymbol{V}}_{B_0} + \dot{\tilde{q}}_{p_i} \boldsymbol{z}',\ i = 0, \cdots, m.
\end{aligned} \tag{9.6}$$

Now, in view of (9.5) and (9.6), it follows that,

$$\begin{aligned}
\sum_{\alpha} \tilde{\boldsymbol{V}}_{\alpha}^{T} \tilde{\boldsymbol{F}}_{\alpha}^{*} &= \tilde{\boldsymbol{V}}_{B_0}^{T} \left(\tilde{\boldsymbol{F}}_{B_0} - \mathbf{U}_{B_1}^{B_0} \tilde{\boldsymbol{F}}_{B_1} - \sum_{i=1}^{m} \mathbf{U}_{P_i}^{B_0} \tilde{\boldsymbol{F}}_{P_i} \right) + \\
&\quad \tilde{\boldsymbol{V}}_{B_{n+1}}^{T} \tilde{\boldsymbol{F}}_{B_{n+1}} + \sum_{i=1}^{m} \tilde{\boldsymbol{V}}_{P_i}^{T} \tilde{\boldsymbol{F}}_{P_i} + \\
&\quad \sum_{i=1}^{n} \left(\tilde{\boldsymbol{V}}_{B_i}^{T} \tilde{\boldsymbol{F}}_{B_i} - \tilde{\boldsymbol{V}}_{B_i}^{T} \mathbf{U}_{B_{i+1}}^{B_i} \tilde{\boldsymbol{F}}_{B_{i+1}} \right) \\
&= \tilde{\boldsymbol{V}}_{B_0}^{T} \tilde{\boldsymbol{F}}_{B_0} - \tilde{\boldsymbol{V}}_{B_0}^{T} \mathbf{U}_{B_1}^{B_0} \tilde{\boldsymbol{F}}_{B_1} + \\
&\quad \tilde{\boldsymbol{V}}_{B_{n+1}}^{T} \tilde{\boldsymbol{F}}_{B_{n+1}} + \sum_{i=1}^{m} \dot{\tilde{q}}_{p_{i+1}} \boldsymbol{z}'^{T} \tilde{\boldsymbol{F}}_{P_{i+1}} + \\
&\quad \sum_{i=1}^{n} \left(\tilde{\boldsymbol{V}}_{B_i}^{T} \tilde{\boldsymbol{F}}_{B_i} - \tilde{\boldsymbol{V}}_{B_{i+1}}^{T} \tilde{\boldsymbol{F}}_{B_{i+1}} \right) + \sum_{i=1}^{n-1} \dot{\tilde{q}}_{m_{i+1}} \boldsymbol{z}^{T} \tilde{\boldsymbol{F}}_{B_{i+1}} \\
&= \tilde{\boldsymbol{V}}_{B_0}^{T} \boldsymbol{F}_{B_0}^{(r)} + \sum_{i=1}^{m} \dot{\tilde{q}}_{p_{i+1}} \left(\boldsymbol{z}'^{T} \boldsymbol{F}_{P_i}^{(r)} - \tau_{p_i} \right) + \\
&\quad \sum_{i=1}^{n} \dot{\tilde{q}}_{m_{i+1}} \left(\boldsymbol{z}^{T} \boldsymbol{F}_{B_i}^{(r)} - \tau_{m_i} \right) = 0,
\end{aligned}$$

in which the last equality is ensured by the control law (8.5) and the imposed constraint (8.6).

□

The Lyapunov function candidate for the entire system is proposed as the summation of all $\mathfrak{V}_{\alpha}$'s,

$$\mathfrak{V} = \sum_{\alpha} \mathfrak{V}_{\alpha} \left(\tilde{\boldsymbol{V}}_{\alpha}, \tilde{\boldsymbol{\theta}}_{\alpha} \right), \tag{9.7}$$

Differentiating (9.7) with respect to time and considering (9.3) and (9.4) yields,

$$\dot{\mathfrak{V}} = \sum_{\alpha} -\tilde{\boldsymbol{V}}_{\alpha}^{T} \mathbf{K}_{\alpha} \tilde{\boldsymbol{V}}_{\alpha}. \tag{9.8}$$

Since $\mathfrak{V} > 0$ and $\dot{\mathfrak{V}} \leq 0$, it can be concluded that $\tilde{\boldsymbol{V}}_\alpha \in L_2 \cap L_\infty$ and $\tilde{\boldsymbol{\theta}}_\alpha \in L_\infty$ for all bodies. This ensures the boundedness of $\hat{\boldsymbol{\theta}}_\alpha$. In addition, considering (3.11) and (8.2), $\dot{\tilde{q}}_{m_i} = \dot{q}_{m_i}^{(r)} - \dot{q}_{m_i} \in L_2 \cap L_\infty$ and $\dot{\tilde{q}}_{p_i} = \dot{q}_{p_i}^{(r)} - \dot{q}_{p_i} \in L_2 \cap L_\infty$ also holds. As a result,

$$\mathcal{V}^{(r)} - \mathcal{V} \in L_2 \cap L_\infty. \tag{9.9}$$

Using these results and (6.8), (6.9), (6.10), (6.11), (7.1), (7.2), and (7.7), it follows that,

$$\begin{aligned}
&\dot{\boldsymbol{q}}_m^{(d)} - \dot{\boldsymbol{q}}_m,\, \boldsymbol{q}_m^{(d)} - \boldsymbol{q}_m,\\
&\dot{Z}_{B_0}^{(d)} - \dot{Z}_{B_0},\, Z_{B_0}^{(d)} - Z_{B_0},\\
&\dot{\boldsymbol{E}}_{I,B_0}^{(d)} - \dot{\boldsymbol{E}}_{I,B_0},\, \boldsymbol{E}_{I,B_0}^{(d)} - \boldsymbol{E}_{I,B_0},\\
&{}^I\dot{\boldsymbol{P}}_{B_{n+1}}^{(d)} - {}^I\dot{\boldsymbol{P}}_{B_{n+1}},\, {}^I\boldsymbol{P}_{B_{n+1}}^{(d)} - {}^I\boldsymbol{P}_{B_{n+1}},\\
&\dot{\boldsymbol{E}}_{I,B_{n+1}}^{(d)} - \dot{\boldsymbol{E}}_{I,B_{n+1}},\, \boldsymbol{E}_{I,B_{n+1}}^{(d)} - \boldsymbol{E}_{I,B_{n+1}},\\
&{}^I\dot{\boldsymbol{P}}_{B_0}^{(d)} - {}^I\dot{\boldsymbol{P}}_{B_0},\, {}^I\boldsymbol{P}_{B_0}^{(d)} - {}^I\boldsymbol{P}_{B_0}
\end{aligned} \in L_2 \cap L_\infty. \tag{9.10}$$

Moreover, when the desired trajectories and their time derivatives are bounded, (9.10) results in boundedness of the independent required velocity/acceleration vectors, i.e., $\mathcal{V}_v^{(r)}$ and $\dot{\mathcal{V}}_v^{(r)}$. Given this, it can be shown that the dependent required acceleration vector $\dot{\mathcal{V}}_u^{(r)}$ derived from the nonholonomic constraints is also bounded. As a result the generalized required acceleration vector $\dot{\mathcal{V}}^{(r)}$ is bounded. This further leads to boundedness of $\boldsymbol{F}_\alpha^{*(r)}$ since $\hat{\boldsymbol{\theta}}_\alpha$ is also bounded. Consequently, the boundedness of $\boldsymbol{F}_\alpha^{(r)}$ is guaranteed which further ensures the control inputs $\boldsymbol{\tau}_m$ and $\boldsymbol{\tau}_p$ are bounded. Referring to the dynamics equations (3.36), it follows that $\dot{\mathcal{V}}$ is bounded which in turn yields,

$$\dot{\mathcal{V}}^{(r)} - \dot{\mathcal{V}} \in L_\infty. \tag{9.11}$$

Equations (9.9) and (9.11) guarantee the convergence of $\mathcal{V}$ to $\mathcal{V}^{(r)}$ [89].

Once $\mathcal{V}$ converges to $\mathcal{V}^{(r)}$, in the first control scenario, the equations (6.8), (6.9), and (6.10) become strictly stable first-order linear equations for the tracking errors of joint positions, UAV Euler angles, and UAV altitude. Thus, asymptotic convergence of these errors to zero is guaranteed. The same argument is valid for the second control scenario where (7.1), (7.2), (7.7), and (7.8) guarantee the convergence of the tracking errors for the end-effector position/orientation as well as the UAV position and yaw orientation.

Remark. *Note that for the dependent DOFs, only the convergence of the required velocity vector to the actual one is guaranteed. However, the boundedness of the dependent required velocity vector $\mathcal{V}_u^{(r)}$ is not guaranteed since the actual one may grow unbounded. An example of this happens when a constant nonzero desired roll/pitch is commanded in Scenario I. In this case, the dependent required/actual velocities grow unbounded. In reality, however, the aerodynamic drag forces would bring the system acceleration to zero avoiding the dependent velocity terms to grow unbounded.*

Chapter 10

System Implementation

To validate the control design, the proposed control algorithms for the first and fourth control scenarios are implemented on a prototype of the UAV-Manipulator system. A 3 DOFs robotic manipulator with revolute joints is designed, built, and integrated with the commercial octo-copter DJI S1000. Fig. 10.1 shows this system. The UAV mass including a six-cell high capacity battery is $6kg$ whereas the robotic arm mass is $2.75kg$, making the interaction forces/moments between the two system components considerable. In the remainder of this chapter, the communication link between the sensor/actuator units and the controller module is first described. Then, the system hardware and software designs are presented. The last section of this chapter discusses the system fail-safe and precautionary measures taken during the experiments. [1]

[1] It should be mentioned that the system has been implemented with the help of Eric Dyer who is a graduate student in the Department of Electrical and Computer Engineering at McMaster University.

Figure 10.1: Aerial Manipulator: DJI S1000 octo-copter integrated with a 3 DOF robotic manipulator during a test flight

10.1 Communication

A testbed was created to run the experiments. One of the main challenges which emerged during the system development was establishing a reliable real-time communication link among the sensor units, controller modules, and actuators. The system integration involved linking 14 microprocessors using C and Python code across different platforms.

Hardware and software design decisions were predominantly driven by the need for robust real-time communication at sufficiently high update rates commensurate with the closed-loop control response requirements. In real-time communication of audio and video signals data packets are large but delays, inconsistency and occasional intermittent transmission can be tolerated. However real-time control of a safety critical system such as a UAV requires reliable high-rate low-latency communication of small data packets. Tolerance for delay, jitter, and data dropouts is fairly low in such systems. Wireless links are able to achieve high data bandwidths, for non-safety critical systems, but are notorious for their unreliability due to unpredictable latency spikes when it comes to real-time control applications.

Numerous implementations of wireless communication failed to achieve the required performance and was eventually abandoned. To minimize the latency and maximize the communication packet transfer rate, the UAV-Manipulator was directly interfaced to a ground PC via a USB serial connection.

10.2 Hardware Development

The entire system hardware comprises three main subsystems. This includes a motion capture system, an octo-copter, and a serial robotic arm; these systems are briefly explained next.

10.2.1 Motion Capture System

Implementation of the control algorithm requires measurements of the UAV position and velocity. In indoors applications, these measurements can be obtained by a motion capture system. An OptiTrack motion capture system was used which implements an Optical-Passive motion capture approach [90]. This system employs eight Flex 13 infrared cameras and retroreflective markers to track targeted objects. The retroreflective markers, mounted on the UAV, are tracked by the infrared cameras. Measurement data from these cameras is analyzed by the Motive application on a ground PC to estimate the UAV position and orientation. These estimates with an accuracy of less than a millimeter in position and half a degree in orientation are available in real-time at a rate of 120 Hz.

In addition to the position measurements, the yaw orientation of the octo-copter is also streamed from the Motive application and used by the control algorithm. This is because the on-board magnetometer of an Inertial Navigation Unit (IMU) cannot be used to accurately measure yaw orientation in an indoor environment due to magnetic field interference.

10.2.2 DJI S1000 Octo-copter

The commercial octo-copter DJI S1000 is used in the experiments [91]. The main electric hardware components of this octo-copter are an on-board STM32 F3 microprocessor with

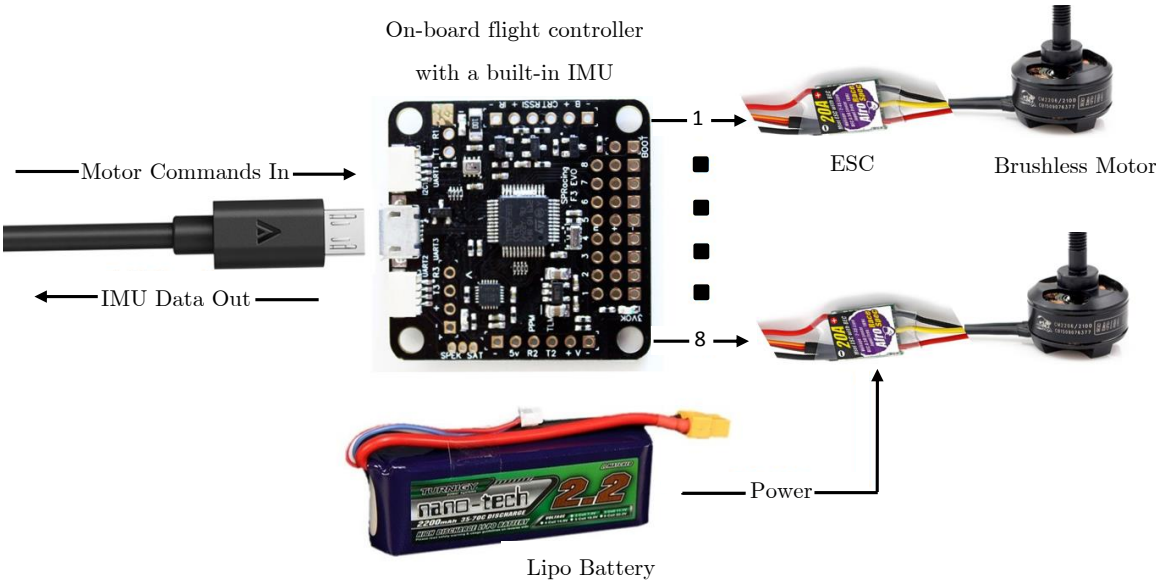

Figure 10.2: Hardware architecture of the UAV actuation and sensing units

a built-in IMU, Electronic Speed Controllers (ESCs), brushless DC electric motors, and a lithium polymer (LiPo) battery. Fig. 10.2 shows the hardware architecture of the octocopter actuation and sensing units.

The on-baord flight controller is used to receive and process data from the IMU to determine the attitude and angular velocity of the drone. It also sends out Pulse Width Modulation (PWM) commands to eight ESCs. The ESCs are powered directly from a six-cell (6S) 22.2V LiPo battery and interface with each brushless motor using three wires. Each ESC has its own onboard microcontroller to receive the motor commands from the central microprocessor and sequentially pulse the three motor wires to obtain a desired motor speed. The on-board flight controller communicates with a ground PC controller over a USB cable sending the IMU measurements and receiving the control PWM commands for all the eight ESCs.

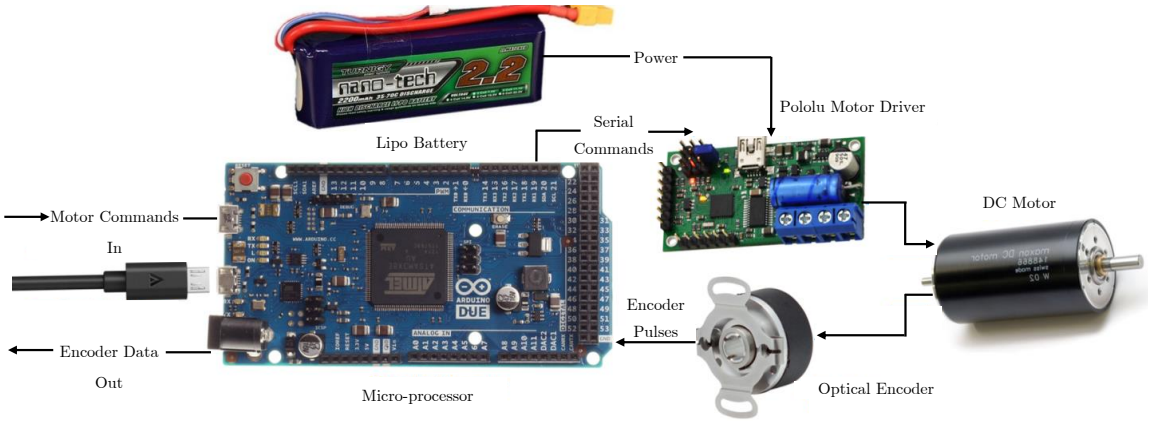

Figure 10.3: Hardware architecture of the manipulator actuation and sensing units

10.2.3 Robotic Manipulator

The robotic manipulator hardware architecture works in a similar way to the drone architecture and is shown in Fig. 10.3. This hardware comprises a SAM3X8E microcontroller board, three Pololu motor drivers, three Maxson DC motors, three optical incremental encoders, and a LiPo battery.

The microprocessor communicates with the ground station over a serial USB cable and with the Pololu motor drivers over separate serial RX/TX ports. It receives the joint actuation commands from the ground PC and transmits them over to the Pololu motor drivers. These commands are then interpreted by the motor drivers on-board logic before being converted into a power PWM signal sent to the Maxon DC motors.

Each motor is equipped with a high resolution optical encoder that provide 5000 pulses per revolution to precisely measure the motor shaft position. Each encoder has two logic A and B channels that are connected into the digital inputs of the microprocessor board. An interrupt routine is implemented to count the pulses and determine the direction of motion, and relay this information back to the ground PC.

10.3 Software Development

Each software module is coded, tested separately, and then integrated to the main application. The focus here is to describe how the required data for the control implementation is obtained and used in each control cycle. The main control algorithm is developed using Python and executed off-board on the ground PC. At the low-level, however, each of the microprocessors described in the previous section has its own firmware. The octo-copter flight controller, for example, runs a C application using a real-time scheduler in order to perform multiple tasks including processing IMU/accelerometer measurements, sensor fusion, and issuing PWM commands to the ESCs.

The measurement data needed for implementation of the model-based control algorithm is provided by a variety of sensing units each of which sampling the respective quantities at a different rate. An asynchronous multi-threaded programming architecture was adopted for the implementation such that the control algorithm can run at the highest sampling rate available while using the latest available measurements from the sensors. An overview of the dataflow and feed-back loop rates in real-time implementation of the control algorithm is captured in Fig. 10.4. The three functional blocks of sensor data acquisition, control algorithm, and trajectory planner constitute the Python application. From an implementation point of view, the application is running three asynchronous threads at different loop rates which are explained below.

10.3.1 Motion Capture Thread

The motion capture thread is dedicated to receiving the octo-copter position and yaw orientation from the motion capture system at a rate of 120 Hz. It invokes a call-back function as soon as new measurement is available. The call-back function numerically differentiates

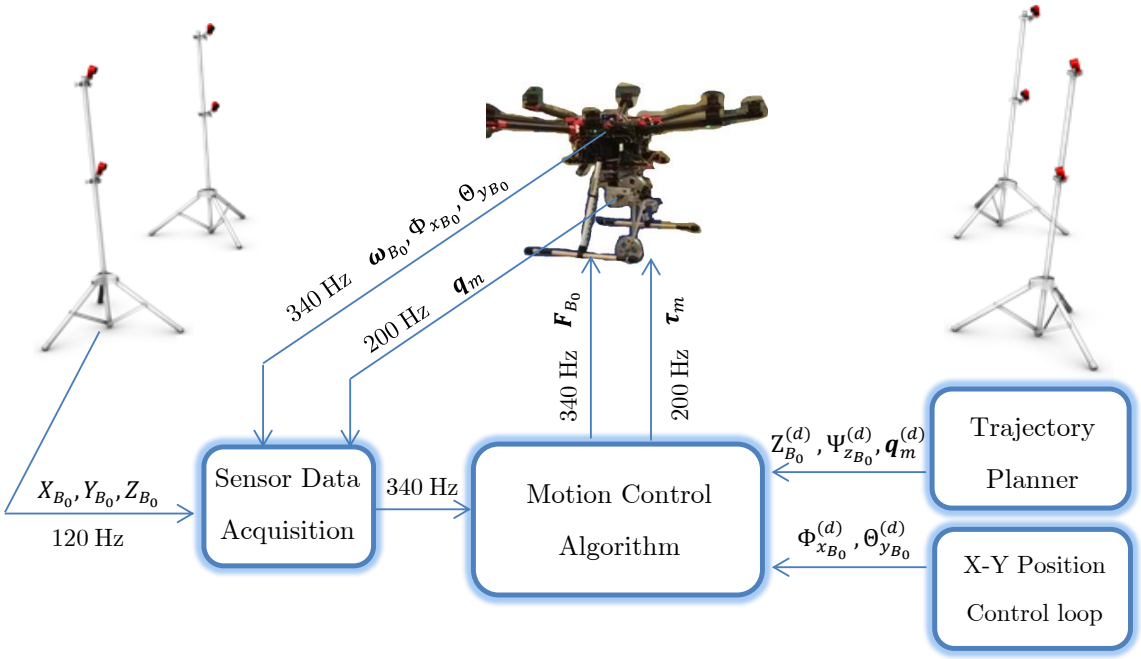

Figure 10.4: An overview of the dataflow and feedback loop rates in real-time implementation of the control algorithm

and low-pass filters the data to compute the octo-copter linear velocity. It then updates the position, velocity, and yaw variables in the application making them available to the other threads. This thread uses socket programming to receive position and attitude data over a software link from the Motive application.

10.3.2 Robotic Arm Thread

The second thread is dedicated to communicating with the manipulator microprocessor in order to send the manipulator actuation commands and receive the encoder pulse counts. This thread runs at a rate of 200 Hz. Similar to the first thread, the encoder data is numerically differentiated and low-pass filtered to calculate the manipulator joint velocities. The joint position and velocity variables are then updated and made available to the other threads. This thread transmits the latest computed joint actuation commands to the manipulator on-board microprocessor using its own custom protocol.

10.3.3 Main Thread

The main thread communicates with the octo-copter flight controller sending out the PWM command values and receiving the IMU attitude and angular velocity measurements. This is the fastest thread running at a fundamental loop rate of 340 Hz, which is the update rate of IMU measurements. Trajectory planning and control algorithm computations are implemented within this thread.

At each cycle of the computations, the control algorithm receives the IMU measurements and the desired trajectories. It then reads the latest available octo-copter position, linear velocity, and yaw orientation provided by the motion capture thread as well as the

manipulator joint positions and velocities provided by the robotic arm thread. Upon performing the computations, the eight required PWM values of the ESCs as well as the three required joint commands are updated in the application. The main thread then sends the PWM values to the octo-copter microprocessor at a rate of 340 Hz. The manipulator joint commands are sent to the respective microprocessor by the robotic arm thread at a rate of 200 Hz. It is paramount that this entire process happens with minimal latency.

A software-level communication protocol is implemented to receive data packets from the octo-copter microprocessor. The Multiwii Serial Protocol (MSP) is used due to its minimalistic packet structure [92]. The protocol comes native with the microprocessor firmware with the ability to send and receive up to 256 unique commands containing up to 256 bytes of data. The MSP protocol also contains a single 1-byte XOR checksum at the end of the packet for simple yet effective error checking. A library is written in Python to interface the microprocessor with the ground PC using this protocol.

10.4 Fail-safe and Precautionary Measures

An aerial robotics platform comes with inherent risk during experimentation. If any of the real-time systems fail, the UAV could behave unpredictably and cause damage to itself and/or those working with the system. To mitigate these risks, a number of fail-safes are built into the system.

At the lowest level, the on-board octo-copter microcontroller constantly monitors the motor commands it receives. If for some reason, the communication link between the on-board controller and the ground PC were lost, the on-board controller would time out after 100 ms and cut the power to the drone. The system would immediately stop operating and fall out of the air. This crude fail-safe may lead to equipment damage but its simplicity

ensures that it would act quickly to prevent personnel injury due to a UAV runaway. The floor space of the flight zone is covered with half-inch thick soft exercise mats to absorb some of the shock from a crashing UAV-Manipulator. An emergency stop (E-stop) button is also mounted beside the ground PC. The E-stop button must be pulled out to allow the system to arm and fly. At any point in the experiment, one can press down onto the locking button to immediately deactivate the system.

Other safety measures exist in the Python application itself. If the system deviates too far from the desired trajectories, the experiment will terminate. Actuator commands also pass through saturation functions to ensure that unreasonably large commands are never sent to the system. Over the hundreds of tests that have been performed, this system of fail-safes has proven as an effective means to keep the equipment and people involved safe.

Remark. *The modular design of the hardware/software described in this chapter makes it adaptable to a variety of different experiments in the future as different sensor combinations could be added and control architectures could be swapped without significant changes to the skeleton of the system.*

Chapter 11

Simulation and Implementation Results

This chapter presents the results of numerical simulations as well as experimental validation of the proposed control algorithms. The first section of this chapter is concerned with simulation results where the most general control framework developed in Chapter 8 is used in the simulations. Propeller dynamics are considered in the model as well as in the control development and the effectiveness of both model-based control scenarios are investigated. Moreover, the control performance of these model-based controllers are compared with a conventional PID controller. The second part of this chapter is concerned with implementation results for two of the control strategies. First, the results for Scenario I, developed in Chapter 6, are presented where the UAV altitude and orientation as well as the arm joint positions are controlled independently. Then, the implementation results of Scenario IV are presented where the UAV altitude and orientation as well as the arm end-effector positions are controlled independently.

Table 11.1: Inertia properties of the system rigid bodies

	Mass (kg)	Inertia ($kg\,m^2$)
UAV	2.930	$\begin{bmatrix} 0.2114 & 0.0001 & 0.0000 \\ 0.0001 & 0.2024 & -0.0001 \\ 0.0000 & -0.0001 & 0.3515 \end{bmatrix}$
Link 1	0.590	$\begin{bmatrix} 0.00003 & 0 & 0 \\ 0 & 0.00205 & 0 \\ 0 & 0 & 0.00205 \end{bmatrix}$
Link 2	0.550	$\begin{bmatrix} 0.00002 & 0 & 0 \\ 0 & 0.00190 & 0 \\ 0 & 0 & 0.00191 \end{bmatrix}$
Propellers	0.031	$\begin{bmatrix} 0.00001 & 0 & 0 \\ 0 & 0.00024 & 0 \\ 0 & 0 & 0.00024 \end{bmatrix}$

11.1 Numerical Simulations

Computer simulations are carried out for an octo-copter equipped with a 2-link robotic arm. A precise dynamics model of the UAV-Manipulator is created in the Matlab/SimMechanics environment on which the control algorithms are implemented. The inertia properties of the UAV base, propellers and the manipulator links used in these simulations are given in Table 11.1. It is worth noting that the manipulator mass ($1.14\,kg$) is significant relative to the UAV mass ($2.93\,kg$) resulting in considerable interactions between the two.

The control design parameters for both control algorithms are given in Table 11.2. The aerodynamic coefficients $k_T\rho D^4$ and $k_Q\rho D^5$ are 2.9×10^{-5} $(N.sec^2)$ and 1.1×10^{-6} $(N.m.sec^2)$ respectively. These values were obtained through real thrust/torque experiments. In the remainder of this section, the simulation results for the first control scenario are presented. Later, the effectiveness of the first and the second control strategies as well as a conventional PID controller in the object grabbing phase of the system mission are compared.

Table 11.2: Control algorithms parameters used in simulations

First control scenario	Second control scenario
$\mathbf{\Lambda}_{q_m} = \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}$	$\mathbf{\Lambda}_{P_{B_3}} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 10 \end{bmatrix}$
$\mathbf{\Lambda}_{E_{B_0}} = 2\,\mathbf{I}_{3\times 3}$	$\mathbf{\Lambda}_{P_{B_0}} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 8 \end{bmatrix}$
$\lambda_z = 10$	
Common parameters	
$\mathbf{\Gamma}_{B_0} = 1\,\mathbf{I}_{13\times 13}$	$\mathbf{K}_{B_0} = \begin{bmatrix} 10\,\mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & 2\,\mathbf{I}_{3\times 3} \end{bmatrix}$
$\mathbf{\Gamma}_{B_1} = 0.2\,\mathbf{I}_{13\times 13}$	$\mathbf{K}_{B_1} = 2\,\mathbf{I}_{6\times 6}$
$\mathbf{\Gamma}_{B_2} = 0.2\,\mathbf{I}_{13\times 13}$	$\mathbf{K}_{B_2} = 2\,\mathbf{I}_{6\times 6}$
$\mathbf{\Gamma}_{B_3} = 0.2\,\mathbf{I}_{13\times 13}$	$\mathbf{K}_{B_3} = 3\,\mathbf{I}_{6\times 6}$
$\mathbf{\Gamma}_{P_i} = 0\,\mathbf{I}_{13\times 13}, \quad i = 1, \cdots, 8$	$\mathbf{K}_{P_i} = 0.1\,\mathbf{I}_{6\times 6}, \quad i = 1, \cdots, 8$

11.1.1 Simulation Results: Control Scenario I

Task Description

To examine the control performance in all DOFs, a desired multi-stage maneuver is planned. Fig. 11.1 shows the actual trajectories for the manipulator joints as well as the octo-copter position and attitude. In the first stage (0-$4sec$), the octo-copter experiences a yaw rotation of $\pi/2rad$ while elevating for $1.5m$ to obtain suitable orientation/position. During this stage, the robotic arm is fully stretched pointing vertically downwards. In the second stage (4-$6sec$), the octo-copter hovers at a fixed position in the three dimensional space while the robotic arm starts moving towards a target object (a $0.5kg$ ball). Upon approaching the target, the end-effector grabs the ball at time instant $6sec$. It is worth noting that exact positioning of the end-effector is required for grasping the ball at this time. In the third stage (6-$8sec$), the UAV-Manipulator keeps the ball fixed in its place. In the last stage ($8-10sec$), the octo-copter undergoes $\pi/8rad$ of roll and pitch change in order to manipulate the ball

towards another target position; meanwhile, the two links are commanded to return to their home positions. Fig. 11.2 illustrates the system configuration at different time instances during its mission.

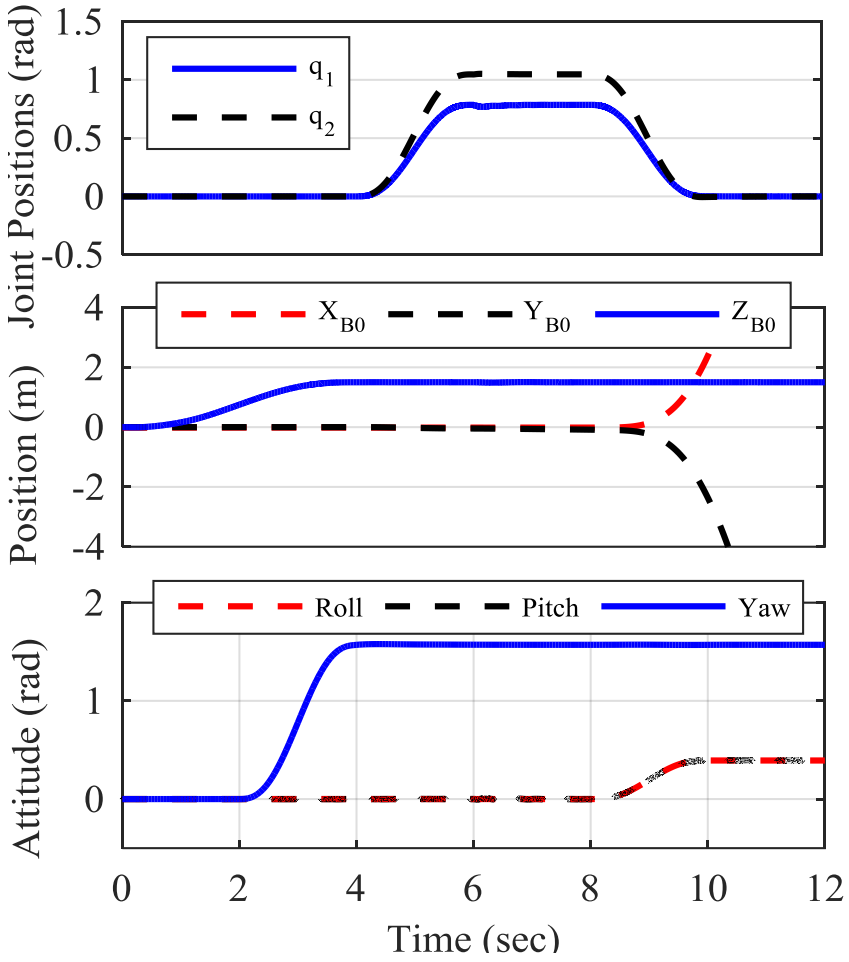

Figure 11.1: Actual trajectories of the UAV-Manipulator for VDC implementation in Scenario I

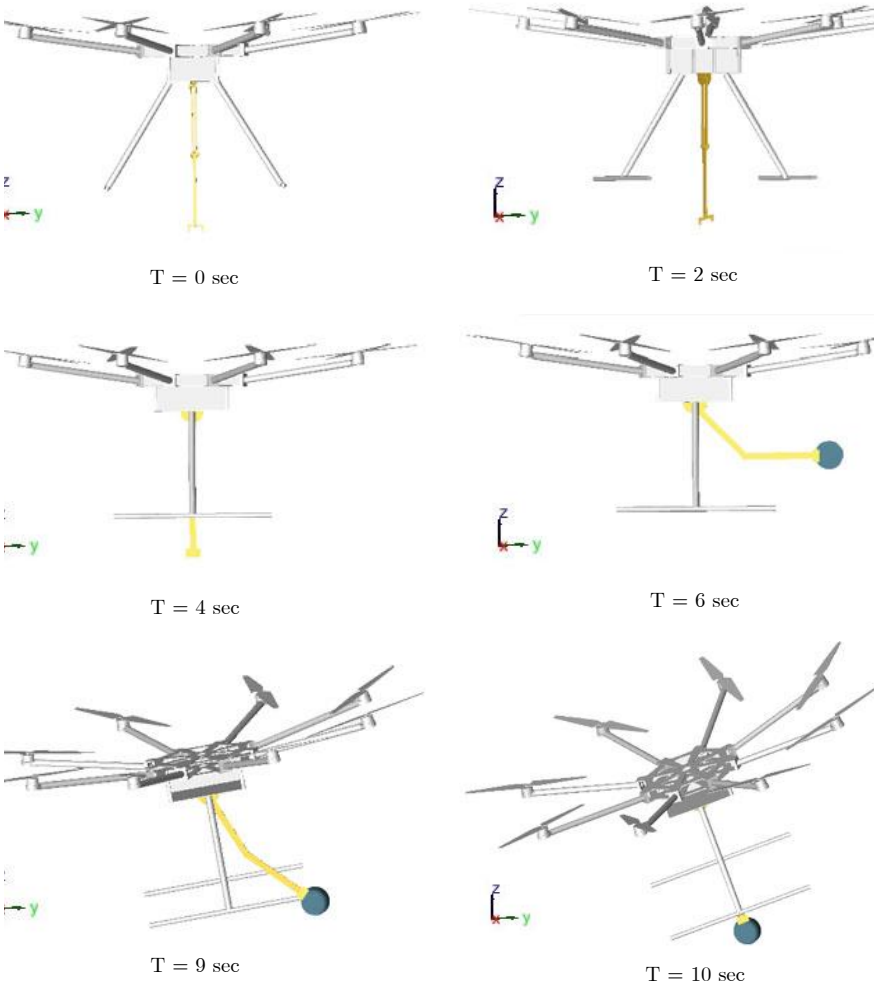

Figure 11.2: System configuration at different time instances during the simulated mission

Results and Discussion

Fig. 11.3 shows the trajectory tracking errors of joint positions, altitude and attitude. As it is seen, the error values remain less than $1.5cm$ for the altitude, $0.015rad$ for the attitude and $0.02rad$ for the joint positions. Moreover, the transient errors at the grasping stage are well compensated, which demonstrates the robustness and error compensation ability of

the control algorithm during the manipulation of an uncertain object. Figs. 11.4 and 11.5 show the actuation torques of the propellers/manipulator joints and the propeller speeds, respectively.

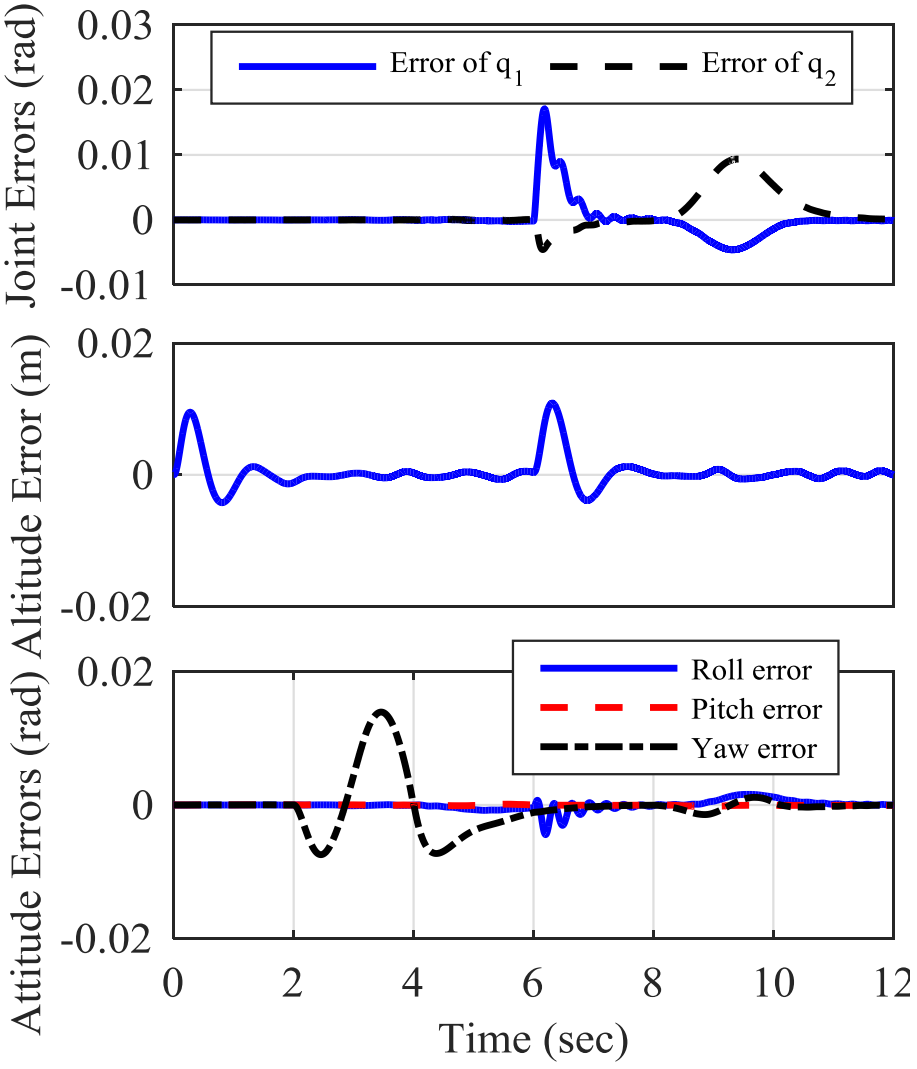

Figure 11.3: Trajectory tracking errors of joint positions, altitude and attitude for VDC implementation in Scenario I

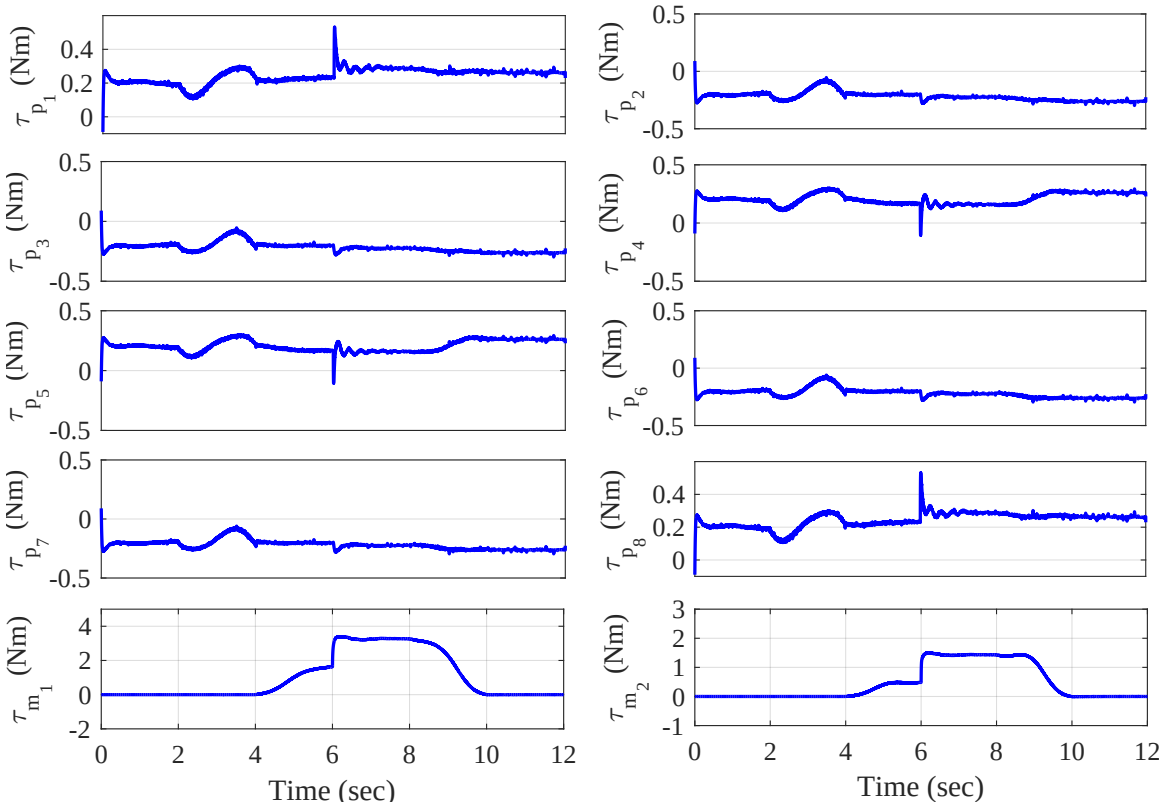

Figure 11.4: Actuation torques of the eight UAV propellers as well as the two manipulator joints in Scenario I

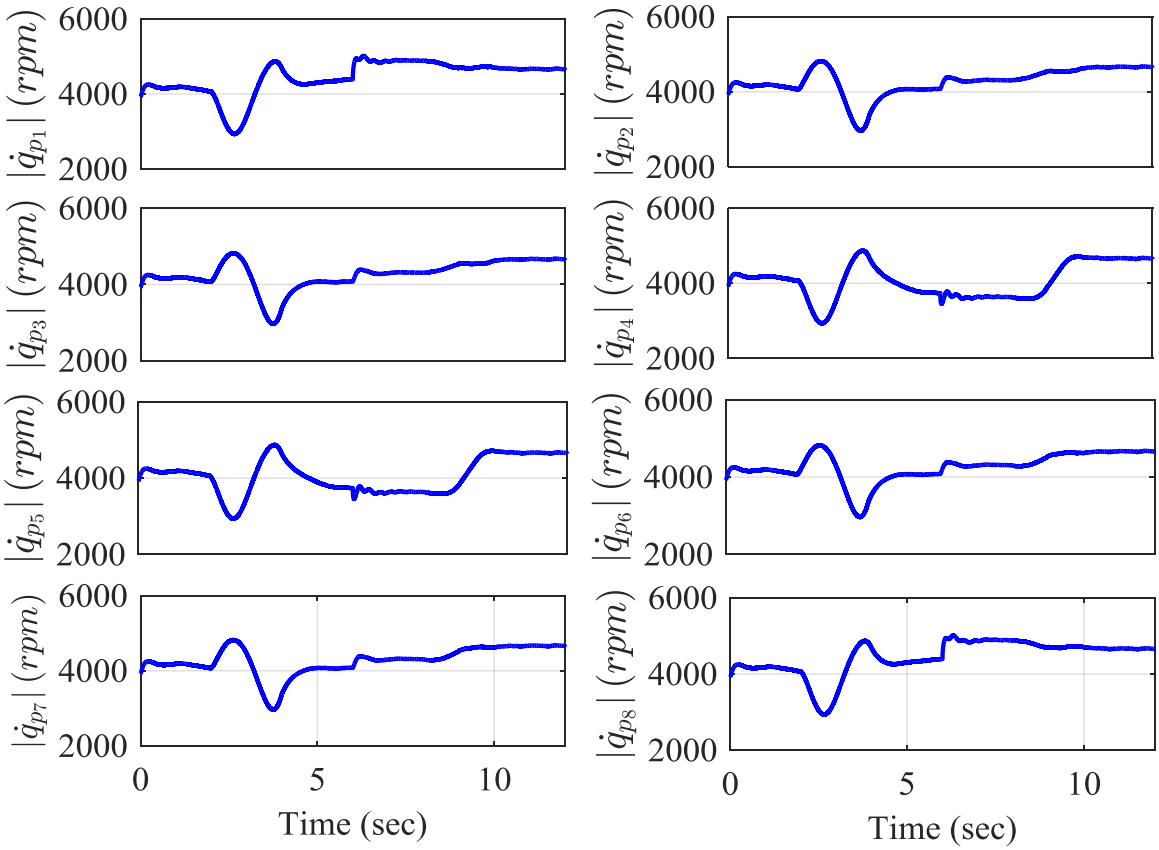

Figure 11.5: Propellers spinning speeds in Scenario I

11.1.2 Simulation Results: A Comparison of Scenarios

To better illustrate the performance and effectiveness of the proposed control approaches in positioning the end-effector, simulations have been also carried out with the VDC for the second control scenario as well as a conventional PID controller used in most of the commercially available UAV flight controllers. Similar to the first control scenario, the PID algorithm receives reference trajectories for the UAV orientation and altitude as well as joint positions. In this case, a PID controller is designed for each of these DOFs independently. In the second control scenario, two coordinates of the end-effector position are controlled since the robotic manipulator has only two DOFs.

The results are compared in Figs. 11.6 and 11.7. Fig. 11.6 shows the position of the UAV between 4 and 8 seconds, where the UAV should remain fixed in its place, hovering without moving. This is crucial for accurate end-effector positioning while the manipulator extends to grab the object. As it is seen in Fig. 11.6, in the period of $4-6$ seconds where the arm is extending towards the target object, the UAVs with VDC algorithm in the first scenario and PID controllers move in the Y-direction for $4cm$ as opposed to the one with VDC controller in the second scenario which perfectly remains in its place. In tele-operation scenarios, this undesirable drift of the base would cause inconvenience for the operator as he/she has to continuously adjust the reference joint commands to guide the end-effector towards the target. This issue, originating from the under-actuation of UAV-Manipulator, further illustrates the importance of the second control approach for such systems.

Fig. 11.7 depicts the end-effector position error in the period of $6-8$ seconds where the target ball should be kept in a fixed position. As it is shown, both VDC control algorithms

have kept the end-effector fixed in place whereas the PID controller shows poor performance and significant position errors of more than $10cm$ in Y and $3cm$ in Z directions. This happens due to the drift of the base, a behaviour that further underscores the need for full model-based controllers for UAV-Manipulators due to the under-actuation.

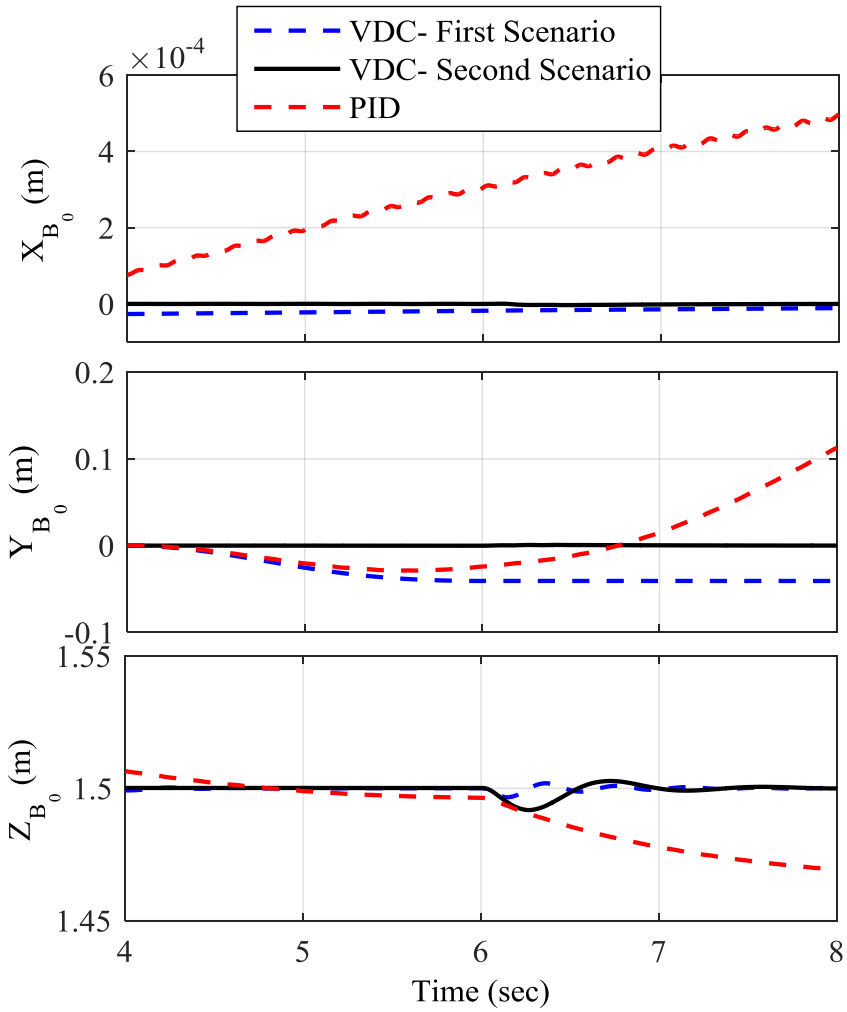

Figure 11.6: Position of the UAV in the hover stage (4-8 seconds) for the VDC and PID controllers

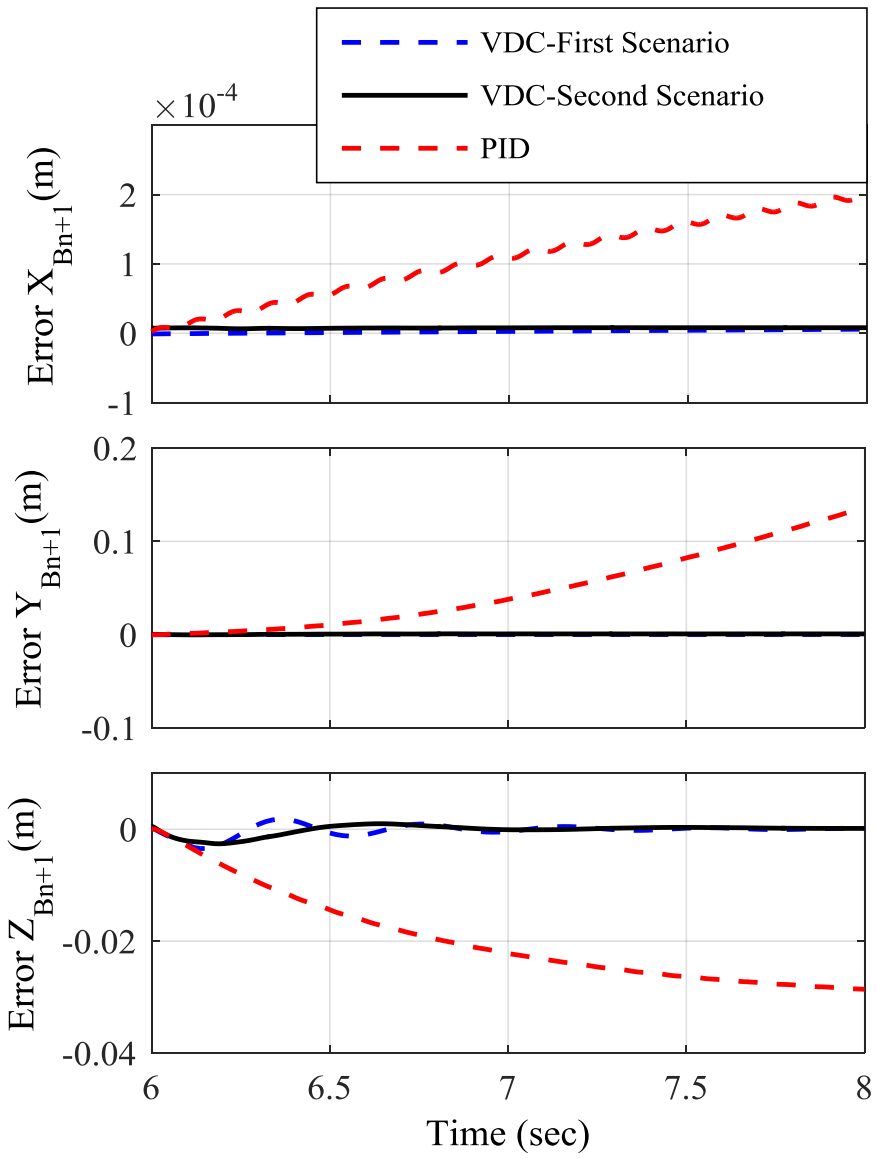

Figure 11.7: End-effector position error within 2 seconds of the hover stage for the VDC and PID controllers

11.2 Real-time Implementation

This section presents the implementation results for control Scenarios I and IV. Before proceeding with presenting the experimental results, a few points should be mentioned.

First, in Chapter 6, the resultant thrust/moment at the UAV center of mass produced by the propeller forces was considered as the UAV control input. This control input can be mapped to the individual propeller forces through a transformation which depends on the UAV geometry and the number of propellers. Using an standard octo-copter introduces actuation redundancy in the actuated DOFs while the system remains under-actuated in two DOFs. The propeller redundancy can be employed to robustify the control system against possible system faults. For the experiments in this thesis, however, a pair of propellers are coupled together treating the UAV as an standard quad-copter.

Second, the low-level commands sent to the UAV actuators are PWM signals. These are individually sent to each ESC which in turn drives a propeller motor. A mapping is needed to convert the required propeller forces to the actual PWM commands for the ESCs. This mapping is presented in this section.

Third, as described in Chapter 5, control Scenarios I and IV are suitable for a human-in-the-loop operation where a human operator would command roll and pitch angles to laterally position the UAV in the XY horizontal plane. Consider a case where the UAV needs to hover in place. In this case, even with a good tracking performance for the UAV orientation, the system is prone to drift in the horizontal plane due to the system under-actuation. This is where the human operator could actively adjust the reference orientation commands to avoid the drift. In the experiments of the thesis, an outer-loop proportional-derivative controller is implemented to play this role and to stabilize the UAV lateral position. The outputs of this controller are the reference commands for the roll and pitch angles in the

model-based controllers. This strategy was adopted to ensure safe operation in the confined space of the indoor flight zone, and also to promote consistency in the experiments.

11.2.1 Thrust-PWM Mapping

The aerodynamic force/torque relationships with the propeller speed and the describing equations of the dc motor electric circuit actuating each propeller are used to derive a mapping between the steady-state propeller thrust force and the commanded PWM to the ESC. The derivation of the mapping is presented in Appendix (A.2) and the result is given in the following equation,

$$D_{pwm} = \gamma_1 F_{thrust} + \gamma_2 \sqrt{F_{thrust}} + \gamma_3, \tag{11.1}$$

where D_{pwm} is the scaled duty cycle of the commanded PWM signal and F_{thrust} is the propeller thrust force. Moreover, γ_1, γ_2, and γ_3 are constant coefficients which depend on a number of factors including the propeller-dependent thrust and torque coefficients, motor torque-current coefficient, motor back EMF coefficient, and motor resistance.

The coefficients in (11.1) were determined through an identification experiment. For this purpose, a test setup was developed. The thrust force/drag torque were measured using a six-axis ATI Mini40 force/torque sensor and a 16-bit DAQ. Measurements were captured at 1 kHz sampling rate and a resolution of 0.01 N. A 2-ms PWM pulse was sent to the built-in ESC of the DJI 4114 PRO 400KV motor.

The experiment was carried out for 50 different duty cycle values and a least-squares problem was solved to find a set of γ_1, γ_2, and γ_3 which fits best to the experiment samples. The results are $\gamma_1 = -7.3$, $\gamma_2 = 178.5$, and $\gamma_3 = 1007.9$. Fig. 11.8 compares a verification experiment sample points with the curve obtained from Equation (11.1). The mapping

proposed by in (11.1) fits well to the experiment results and can be used to map a required thrust force to the PWM command.

Remark. *Note that the coefficients γ_1 and γ_2 may be used as a criteria for the actuation efficiency. The higher the value of γ_2 compared to γ_1, the more efficient the propeller is since for a given PWM duty cycle the generated thrust force is higher.*

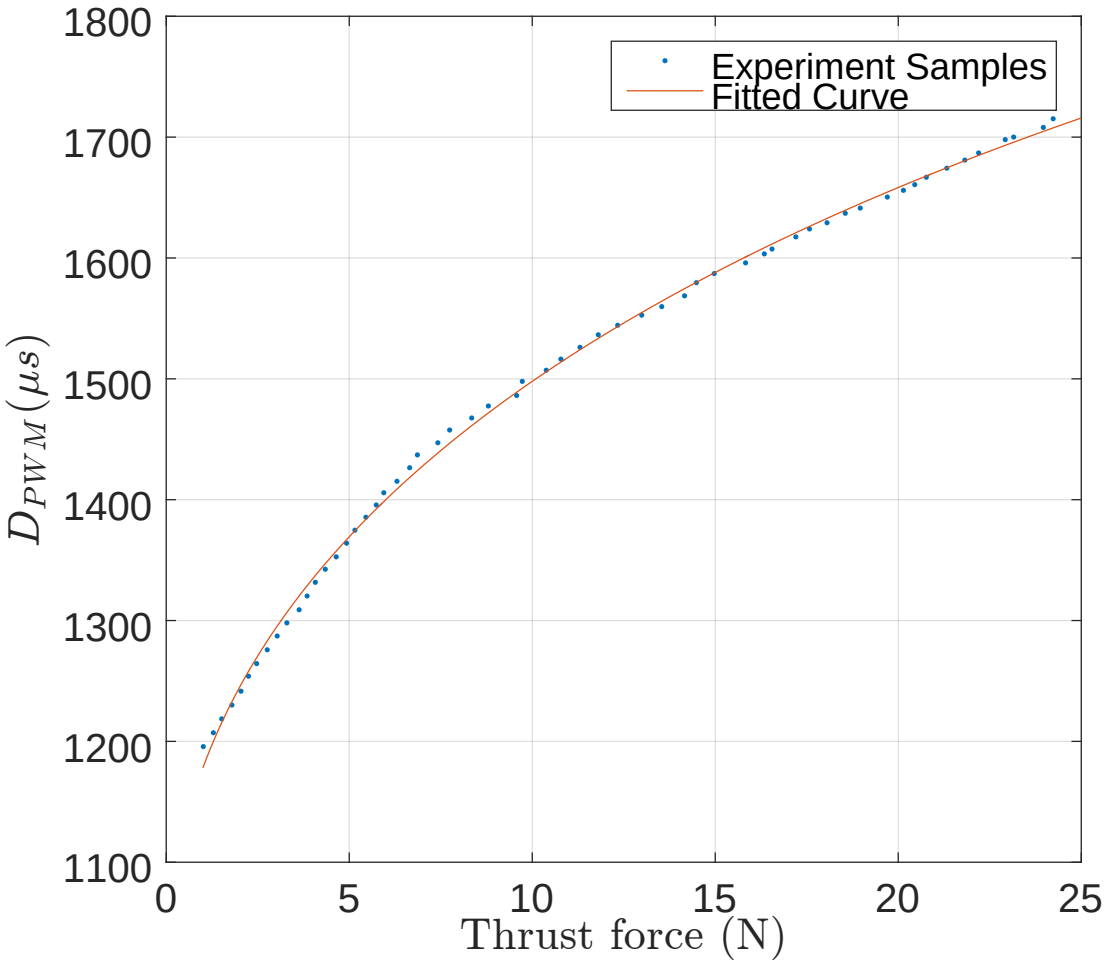

Figure 11.8: Commanded PWM vs measured thrust force

Table 11.3: Control algorithm parameters used in Scenario I

$\mathbf{\Lambda}_{q_m} = diag\,(15, 10, 10)$	$\mathbf{K}_{B_0} = diag\,(40, 40, 300, 5.6, 5.1, 0.3)$
$\mathbf{\Lambda}_{E_{B_0}} = diag\,(13.4, 12.7, 10)$	$\mathbf{K}_{B_1} = diag\,(3, 3, 3, 1.2, 1.2, 3.7)$
$\lambda_z = 2.7$	$\mathbf{K}_{B_2} = diag\,(2, 2, 2, 0.9, 0.9, 2.1)$
$\mathbf{\Gamma}_{B_0}(1, 1) = 0.2$	$\mathbf{K}_{B_3} = diag\,(2, 2, 2, 0.8, 0.8, 1.1)$

11.2.2 Experimental Results: Scenario I

A desired multi-stage maneuver is planned to examine the control performance in all system DOFs. In the first stage (0-$5sec$), the octo-copter undergoes a yaw rotation of $\pi/4rad$ while elevating for $1m$ to obtain suitable orientation/position. During this stage, the robotic arm remains in its home configuration. In the second stage (5-$10sec$), the octo-copter hovers at a fixed position while the robotic arm moves towards a target point by changing all its DOFs. In the third stage (10-$15sec$), while the octo-copter is still hovering, the robotic arm retracts towards the home position. In the last stage ($15 - 20sec$), the octo-copter rotates back towards its original orientation while descending and finally lands in the original position. All the designed trajectories are smooth polynomials. Fig. 11.9 shows the UAV-Manipulator in different time instances during the execution of the maneuver. A video demonstrating this experiment is available at https://youtu.be/MqZYHxAHICM. The control design parameters used during the experiment are given in Table 11.3.

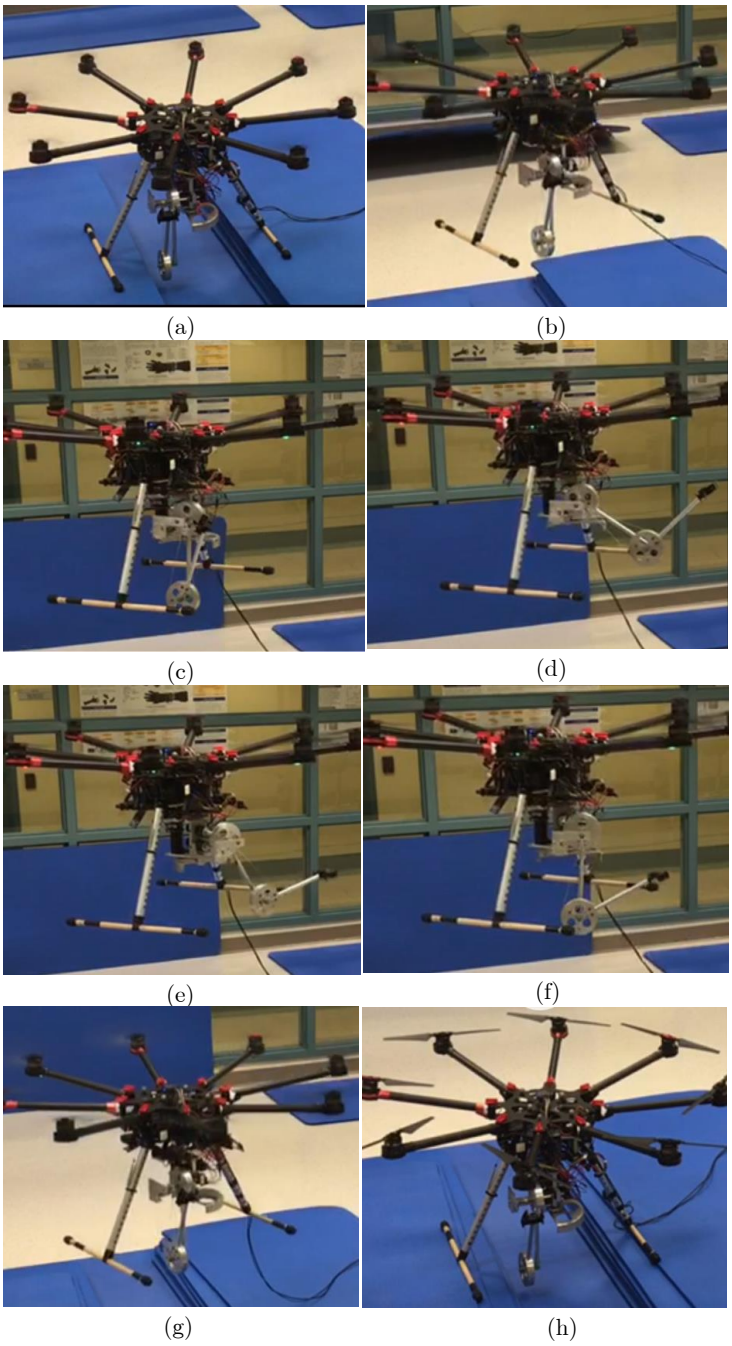

Figure 11.9: UAV-Manipulator during execution of the Scenario I maneuver, (a): UAV starting to elevate, (b): UAV elevating and changing its orientation, (c): UAV in hover, manipulator starting to move towards a target, (d): manipulator second and third links expanding, (e): manipulator first joint orienting towards the target point, (f): manipulator retracting towards its home position, (g): UAV descending and rotating back towards its original orientation, (h): UAV landing.

Figs. 11.10 to 11.16 show the control performance for the copter altitude, yaw, roll, and pitch as well as the manipulator joint positions, respectively. These figures compare the desired and actual trajectories associated with the corresponding DOF. The UAV altitude follows the desired altitude with an error of less than $3.5cm$. The UAV yaw, roll, and pitch tracking errors are less than $0.03rad$, $0.02rad$, and $0.02rad$, respectively. The manipulator joint position errors remain less than $0.015rad$, $0.015rad$, and $0.04rad$. The results demonstrate high performance and stable tracking behaviour in all system DOFs. The manipulator joint position tracking error for the third joint is relatively higher than the first and the second joints. This is due to a considerable Coulomb friction observed in this joint.

As it is observed in Fig. 11.10, the altitude tracking error remains less than $3.5cm$ while it does not converge. The main reason for this observation is an uncertainty in the UAV actuation due to the UAV battery voltage variations. From the derivation of Equation (11.1) given in Appendix (A.2), it is clearly seen that the coefficients γ_1, γ_2, and γ_3 are dependent on the battery voltage. The output voltage of the battery itself is dependent on the motor current and hence the commanded PWM value. As the PWM value increases the voltage decreases due to the internal resistance of the battery. In the identification tests which were performed to find the coefficients of (11.1), the battery voltage was monitored and recorded for any commanded PWM. The voltage variation was then taken into account in the identification of these coefficients. As a result, these coefficients were considered as constants and a least-squares problem was solved to determine their values. As it was shown before in Fig. 11.8, the derived model was an effective mapping between the propeller thrust force and the commanded PWM. However, in the identification tests, the battery was supplying voltage to only one motor and a propeller whereas in the real experiments 8 of these

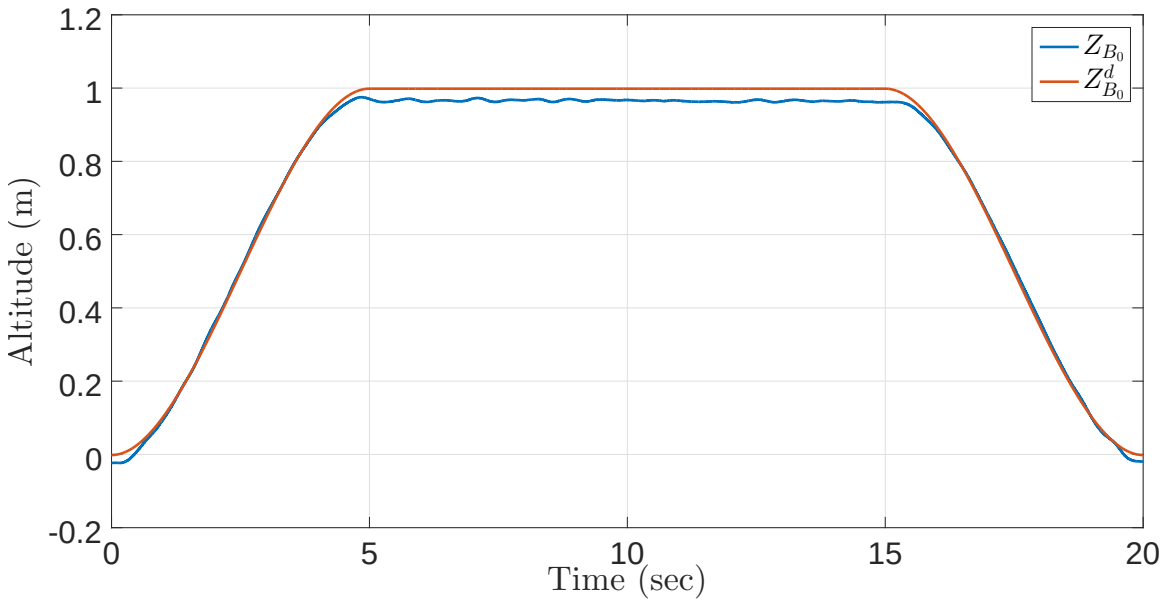

Figure 11.10: Scenario I experiment results: UAV altitude tracking performance

were involved. This battery voltage reduction introduced some inaccuracy in the actuation model presented in (11.1). While the altitude tracking error is quite satisfactory, i.e. less than 3.5 percent of the reference altitude, further improvement can be achieved by using battery voltage measurements in real-time and properly accounting for its variations.

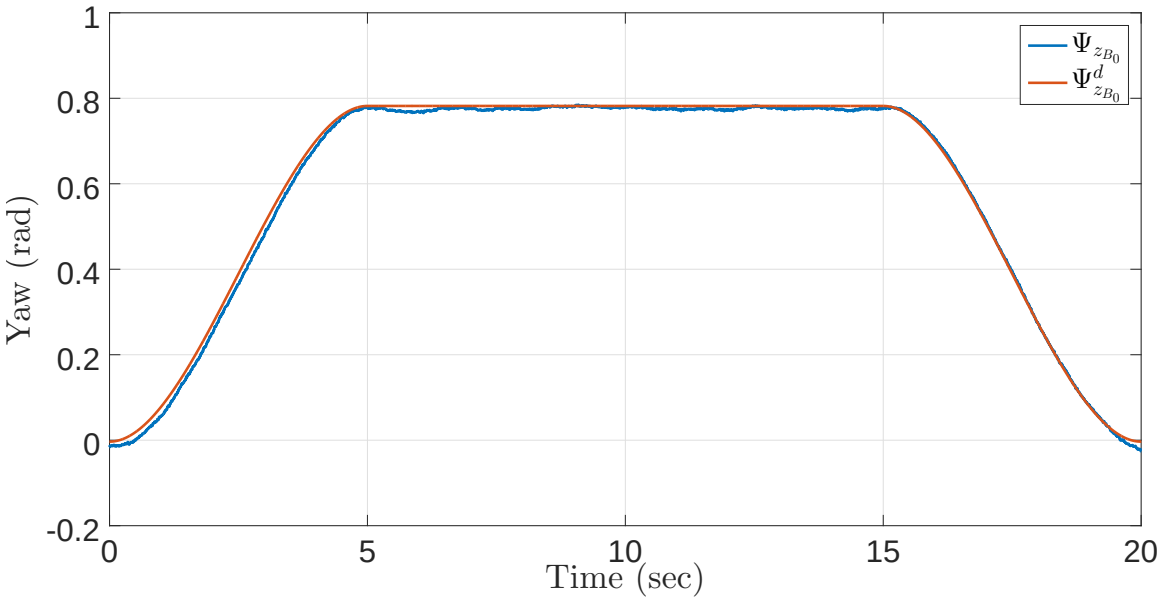

Figure 11.11: Scenario I experiment results: UAV yaw tracking performance

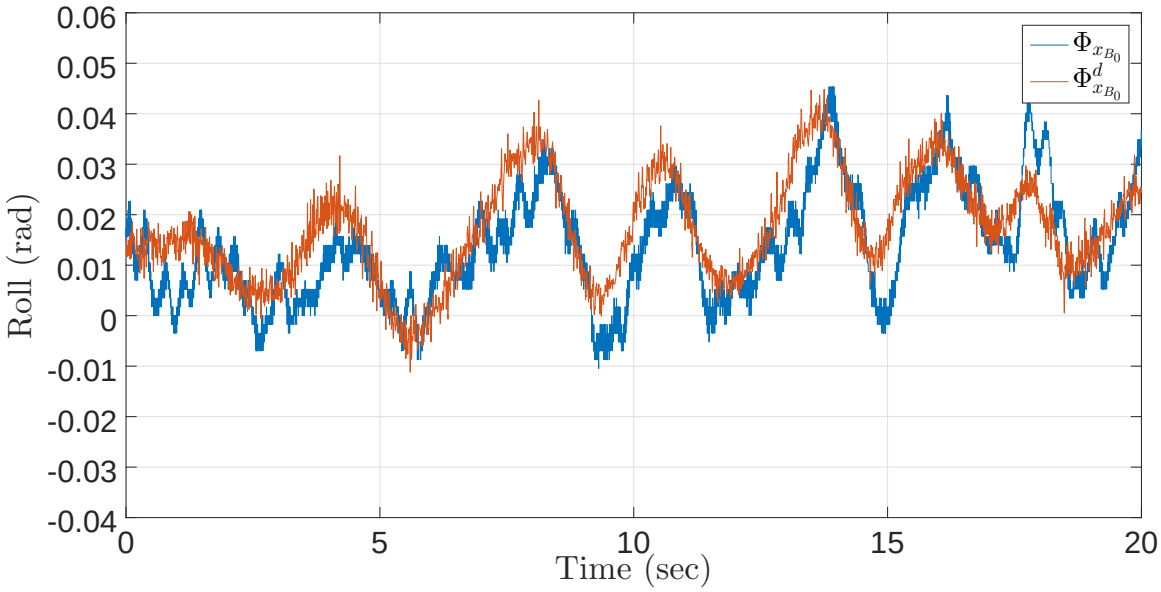

Figure 11.12: Scenario I experiment results: UAV roll tracking performance

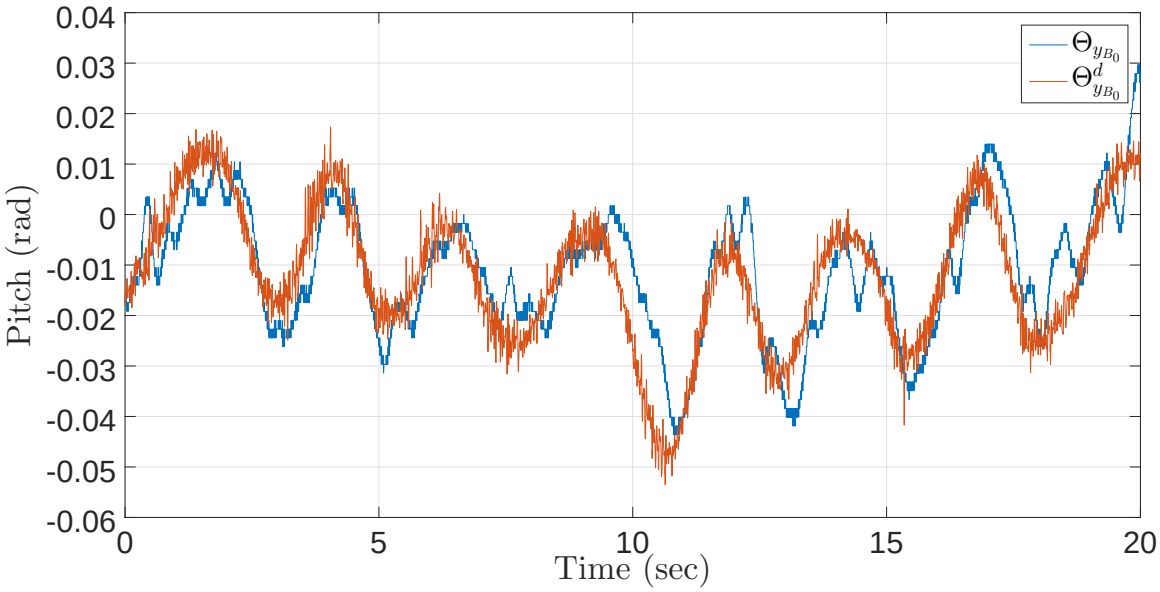

Figure 11.13: Scenario I experiment results: UAV pitch tracking performance

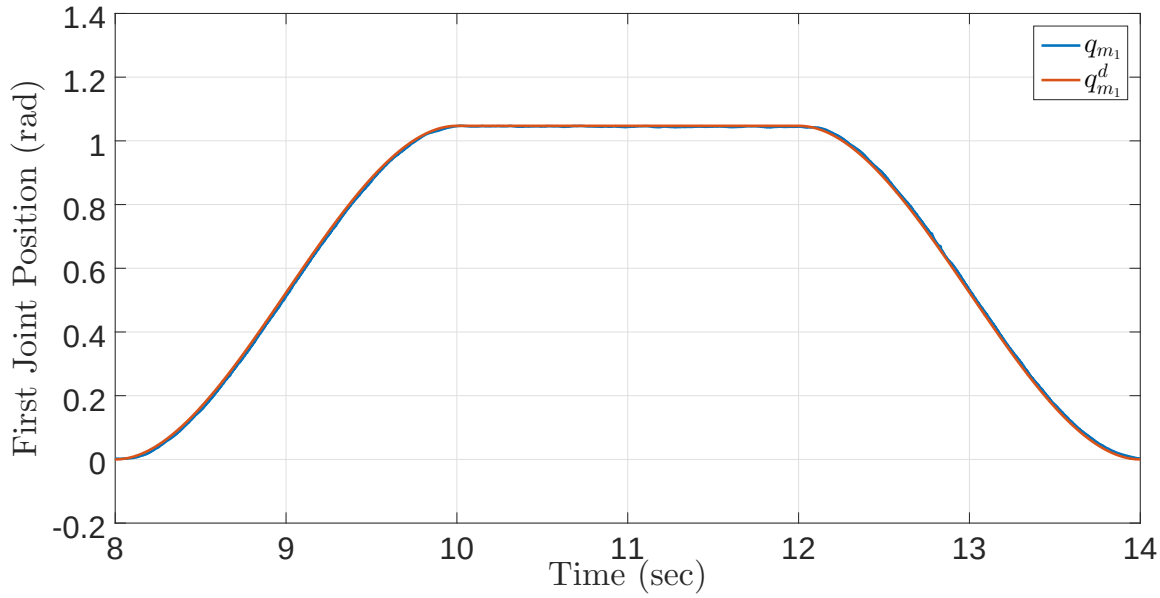

Figure 11.14: Scenario I experiment results: Manipulator first joint tracking performance

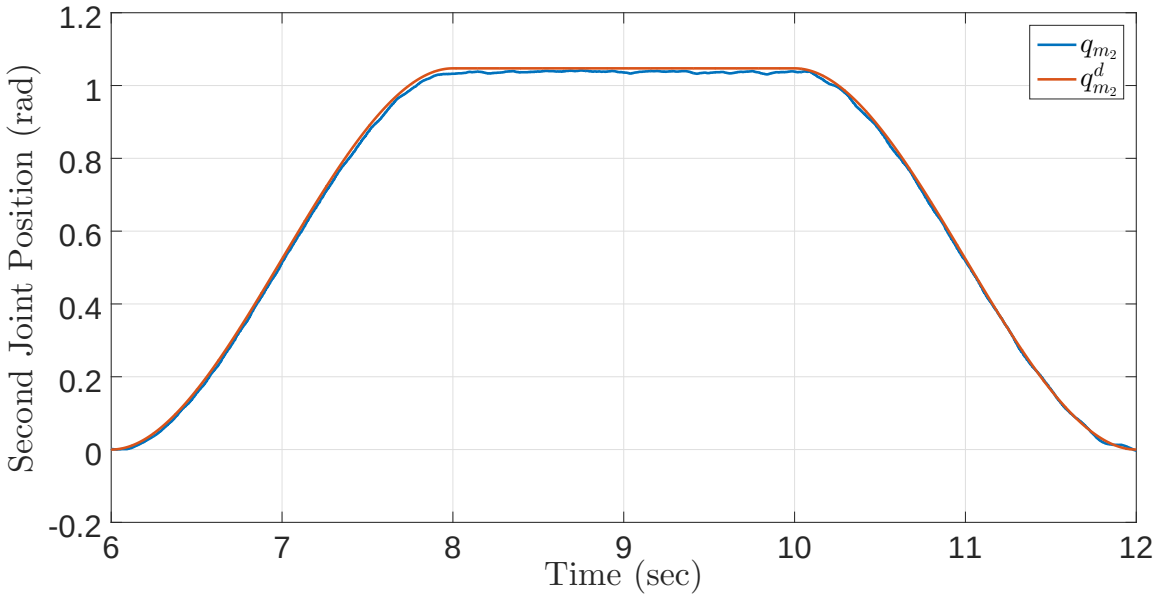

Figure 11.15: Scenario I experiment results: Manipulator second joint tracking performance

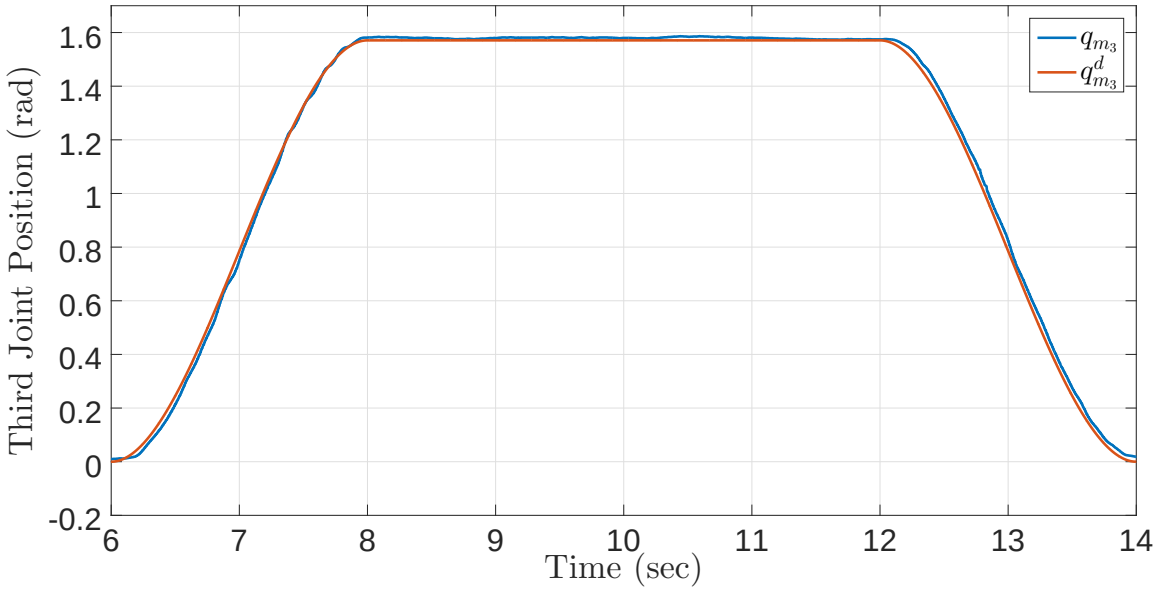

Figure 11.16: Scenario I experiment results: Manipulator third joint tracking performance

Figs. 11.17 to 11.19 show the system control inputs namely the UAV thrust and moment actuations as well as the manipulator joint torques. The present noise in these control commands is due to the noise in feed-back variables. During the system operation, the high-speed propellers create high frequency/small amplitude vibrations. These vibrations introduce noise with small magnitude in the octo-copter position measurement captured by the infrared cameras, its orientation captured by the IMU, and the manipulator joint positions captured by the high-resolution optical encoders. These vibrations further make the measured angular velocity significantly noisy.

This noise is amplified in the octo-copter linear velocity and the manipulator joint velocities through numerical differentiation of the position variables. Although, these variables are low-passed filtered after derivation, there is still a considerable amount of noise left in these signals. There is a trade-off between the reduction of the cut-off frequency of these filters and the latency introduced in the velocity feed-back variables through filtering. Numerous experiments were performed to smooth out these velocity estimates while not undermining the system overall performance due to the extra lag added by filtering.

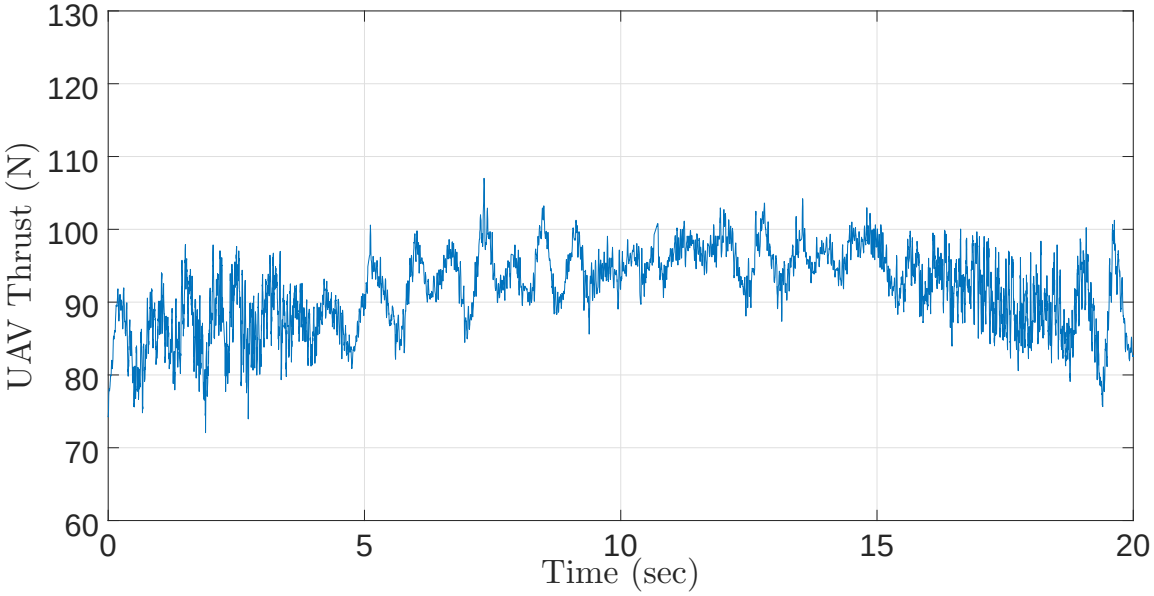

Figure 11.17: Scenario I experiment results: UAV thrust actuation

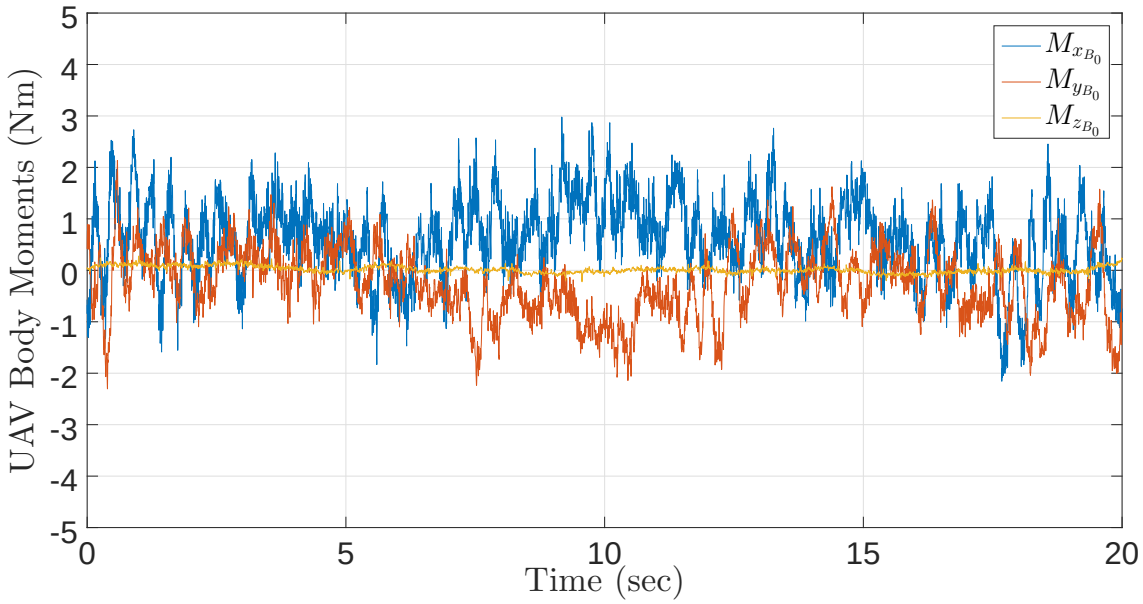

Figure 11.18: Scenario I experiment results: UAV moment actuations

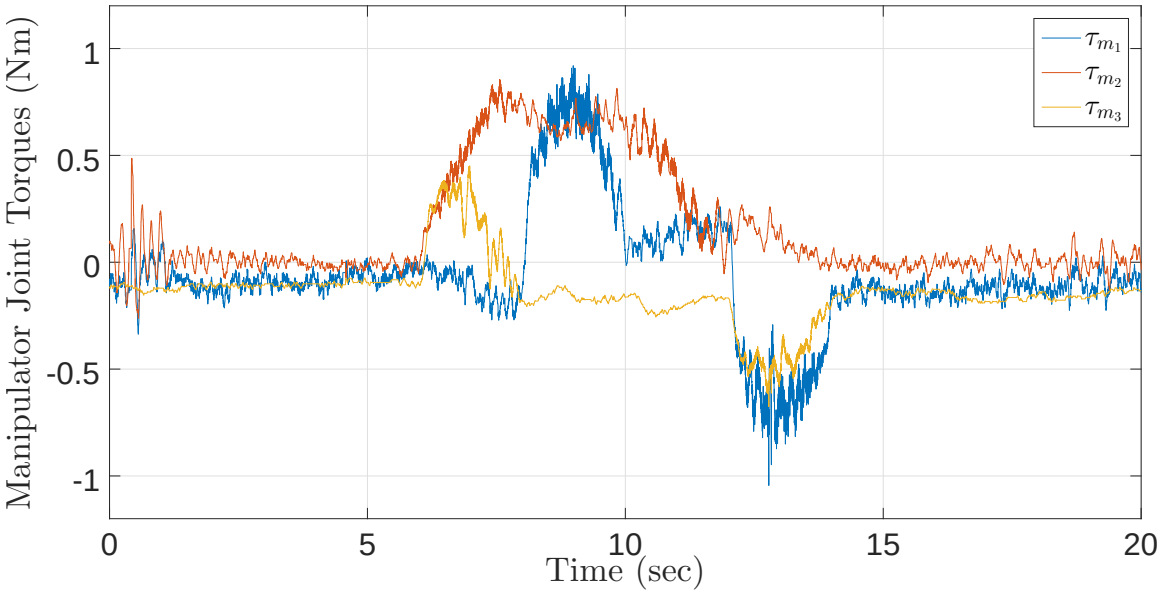

Figure 11.19: Scenario I experiment results: Manipulator joint torques

11.2.3 Experimental Results: Scenario IV

Scenario IV selects the UAV altitude and orientation as well as the end-effector position and orientation as the independently controlled DOFs. In the implementation of this control scenario, the first joint was locked treating the arm as a 2 DOF manipulator. Therefore, two out of six configuration variables of the end-effector could be controlled. In this case, the end-effector X-Z coordinates in the inertial frame are controlled in addition to the UAV altitude and orientation.

A desired multi-stage maneuver is planned to examine the control performance in all independently controlled system DOFs. In the first stage (0-5sec), the octo-copter undergoes a yaw rotation of $\pi/4rad$ while elevating for $1m$. During this stage, the end-effector regulates its X coordinate at zero while its Z coordinate follows a desired trajectory. In the second stage (5-15sec), the octo-copter hovers at a fixed position and the end-effector holds

Table 11.4: Control algorithm parameters used in Scenario IV

$\mathbf{\Lambda}_{P_{B_3}} = diag\,(10, 10)$	$\mathbf{K}_{B_0} = diag\,(40, 40, 300, 5.6, 5.1, 0.3)$
$\mathbf{\Lambda}_{E_{B_0}} = diag\,(13.4, 12.7, 10)$	$\mathbf{K}_{B_1} = diag\,(2, 2, 2, 0.9, 0.9, 2.1)$
$\lambda_z = 2.7$	$\mathbf{K}_{B_2} = diag\,(2, 2, 2, 0.8, 0.8, 1.1)$
$\mathbf{\Gamma}_{B_0}(1, 1) = 0.2$	

its position in the X-Y plane. In the last stage ($15 - 20sec$), the octo-copter rotates back towards its original orientation while descending. During this stage, the end-effector tracks a desired trajectory in Z direction while still regulating its X coordinate at zero. All designed trajectories are smooth polynomials. Fig. 11.20 shows the UAV-Manipulator in different time instances during execution of the maneuver. A video demonstrating this experiment is available at https://youtu.be/8eJYPr_-LDU. The control design parameters used during this experiment are given in Table 11.4.

Figs. 11.21 to 11.24 show the control performance for the copter altitude, yaw, roll, and pitch. These figures compare the desired and actual trajectories associated with the corresponding DOFs. The UAV altitude follows the desired altitude with an error of less than $4cm$. The UAV yaw, roll, and pitch tracking errors are less than $0.03rad$, $0.02rad$, and $0.02rad$, respectively.

Figs. 11.25 and 11.26 show the end-effector position tracking performance in the X and Z directions, respectively. Moreover, Fig. 11.27 depicts the end-effector position tracking errors which remain less than $1cm$ in both X and Z coordinates. The effectiveness of this control strategy in precise positioning of the end-effector is evident when the UAV position errors are compared to those of the end-effector. Fig. 11.28 shows the X-Y coordinates of the UAV. The UAV position error in the X direction is less than $7cm$. Therefore, while the UAV shows $4cm$ positioning error in the Z direction and $7cm$ in the X direction, the end-effector position errors remain less than $1cm$ in these coordinates. This demonstrates

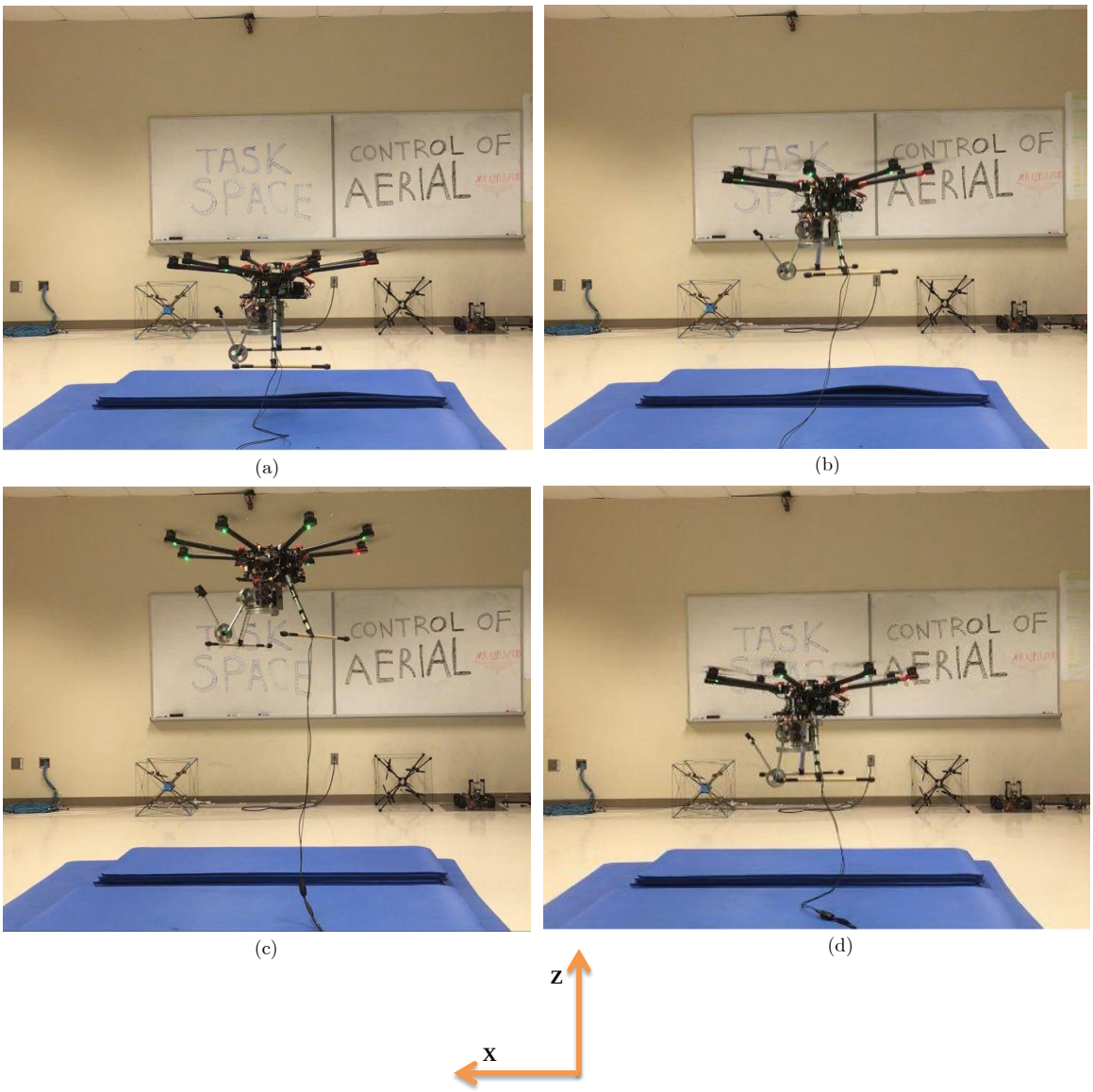

Figure 11.20: Scenario IV first experiment, (a): system in home configuration, (b): UAV elevating and changing its orientation, end-effector tracking its desired trajectory in Z direction and regulating its position in X direction, (c): UAV in hover, end-effector holding its position, (d): UAV descending and rotating back towards its original orientation, end-effector following its desired trajectory in Z direction and regulating its position in X direction.

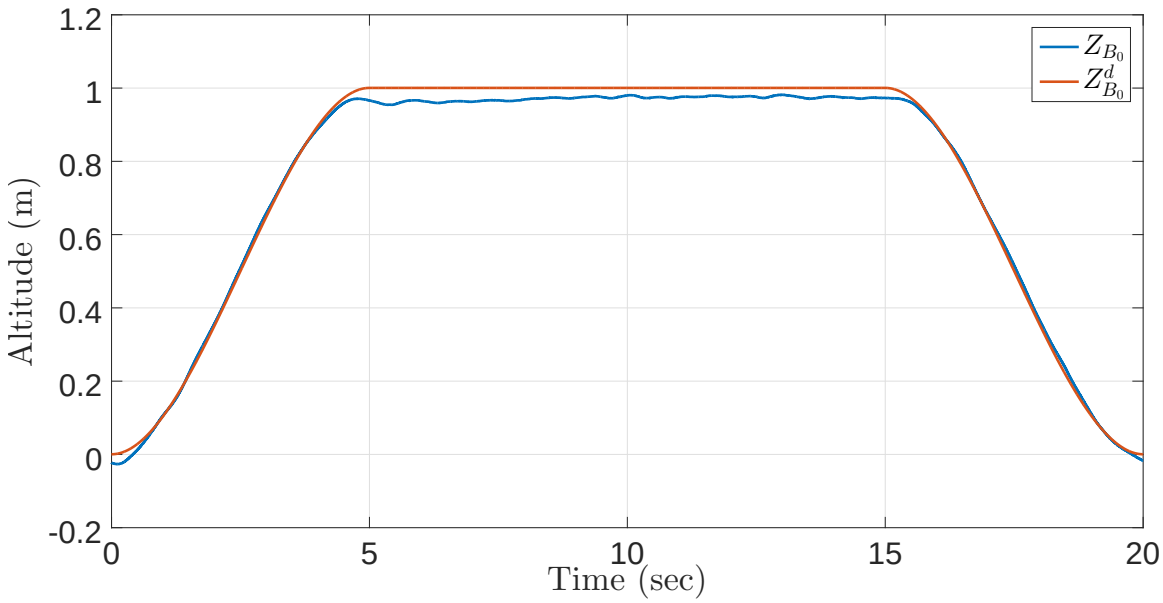

Figure 11.21: Scenario IV experiment results: UAV altitude tracking performance

that the control strategy in Scenario IV is more suitable than the one in Scenario I whenever precise task-space control is required as the manipulator actively adjusts its configuration to compensate for the base positioning errors. Figs. 11.29 to 11.31 show the system control inputs namely the UAV thrust and moment actuations as well as the manipulator joint torques.

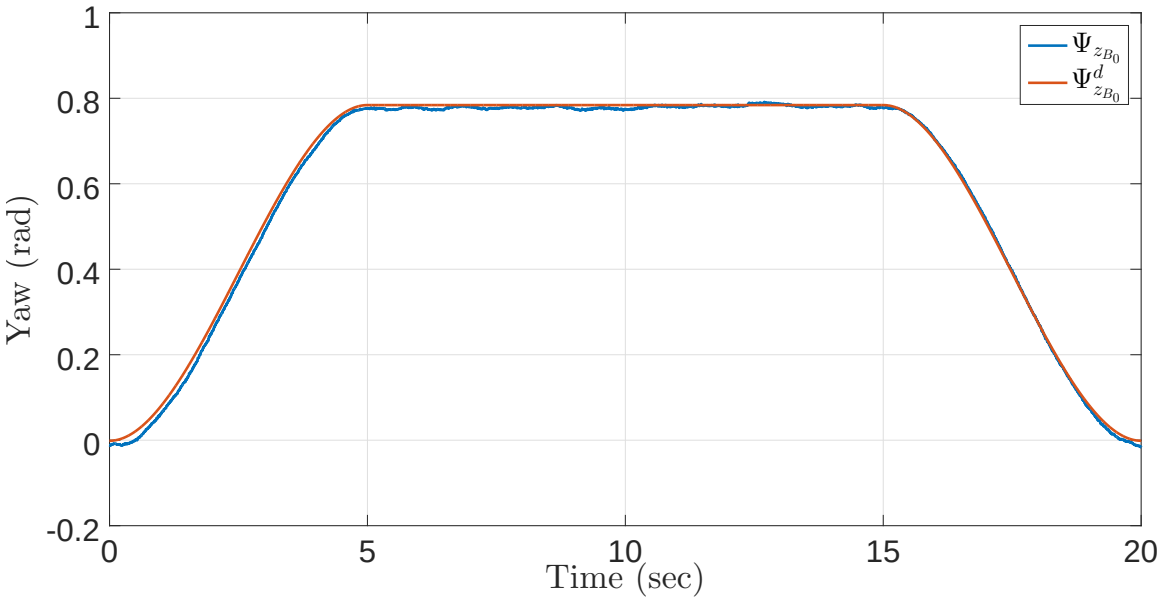

Figure 11.22: Scenario IV experiment results: UAV yaw tracking performance

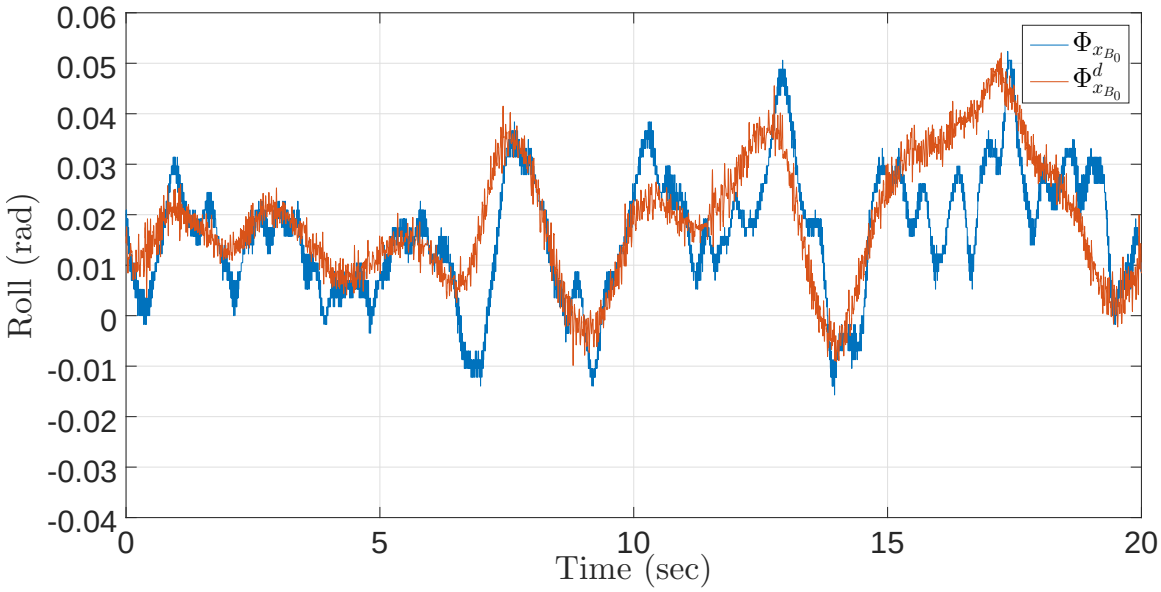

Figure 11.23: Scenario IV experiment results: UAV roll tracking performance

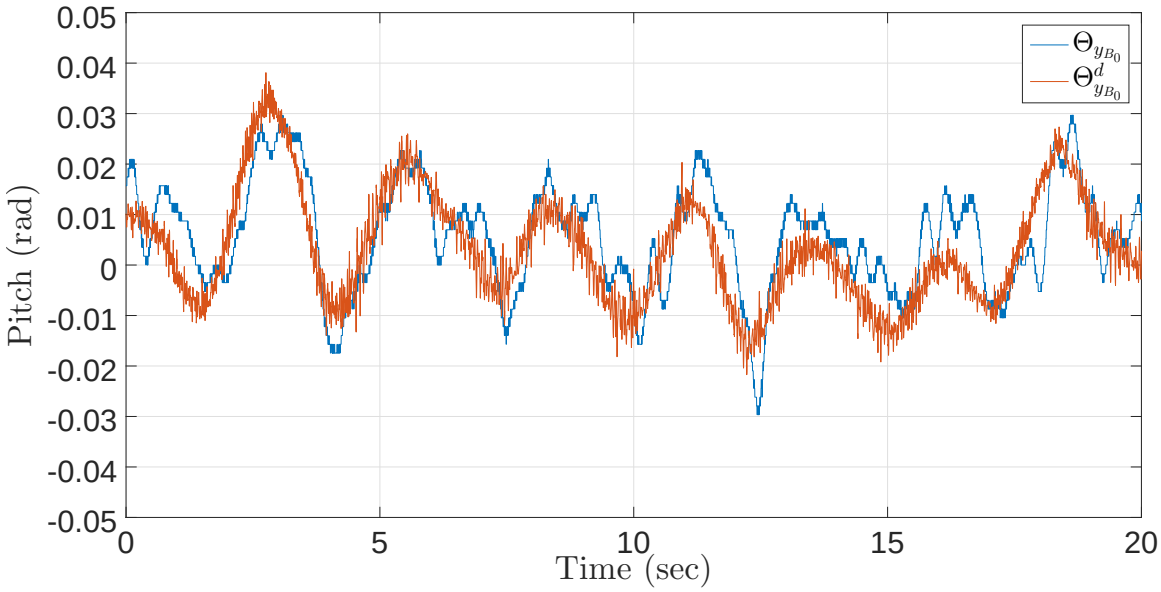

Figure 11.24: Scenario IV experiment results: UAV pitch tracking performance

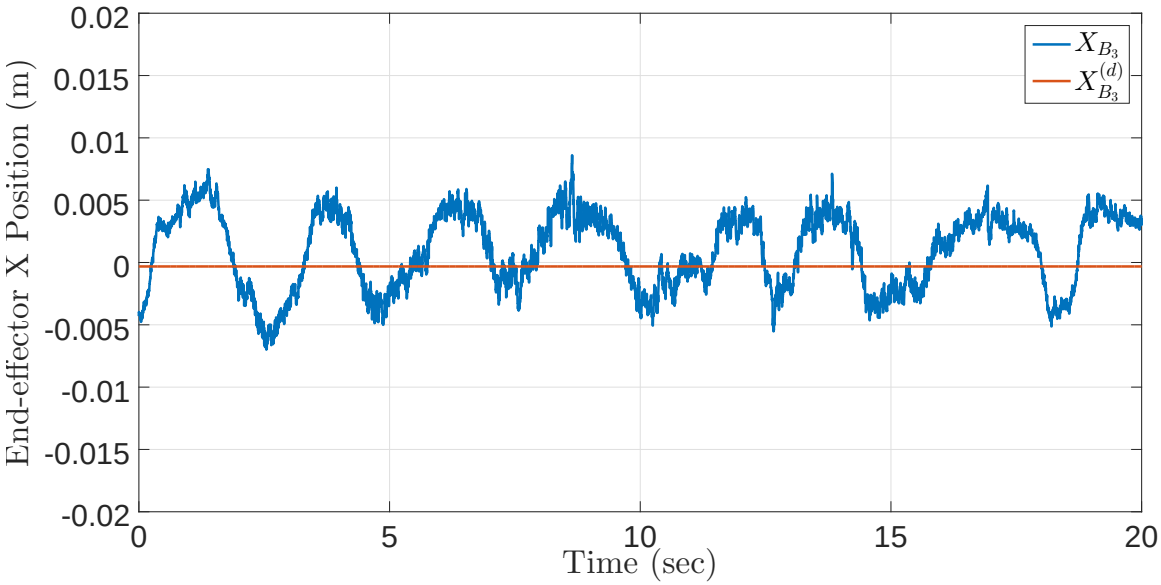

Figure 11.25: Scenario IV experiment results: End-effector position tracking performance in X direction

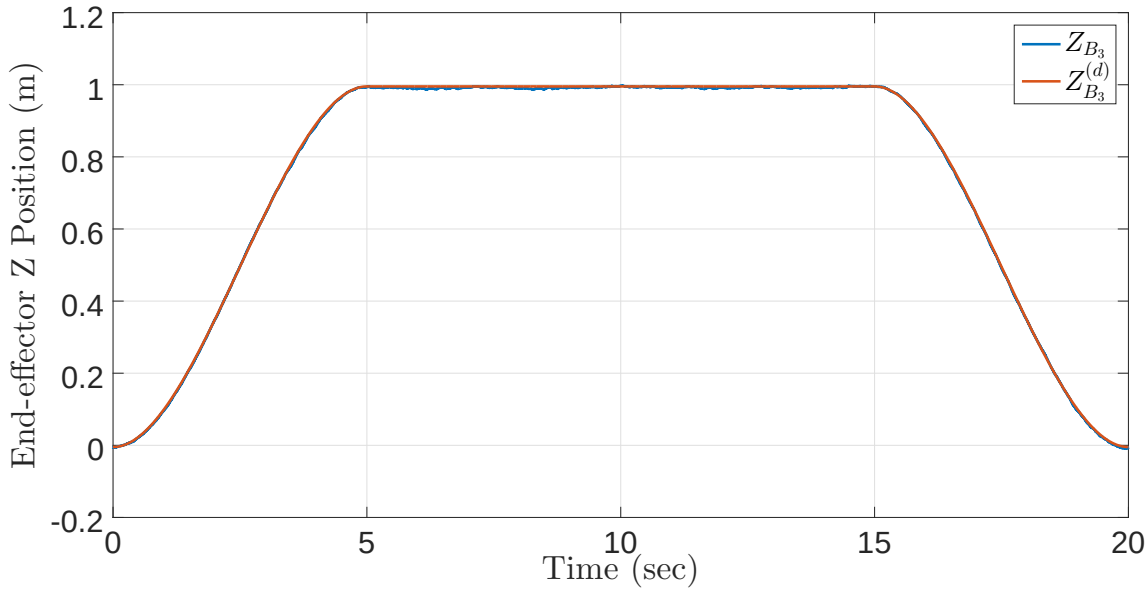

Figure 11.26: Scenario IV experiment results: End-effector position tracking performance in Z direction

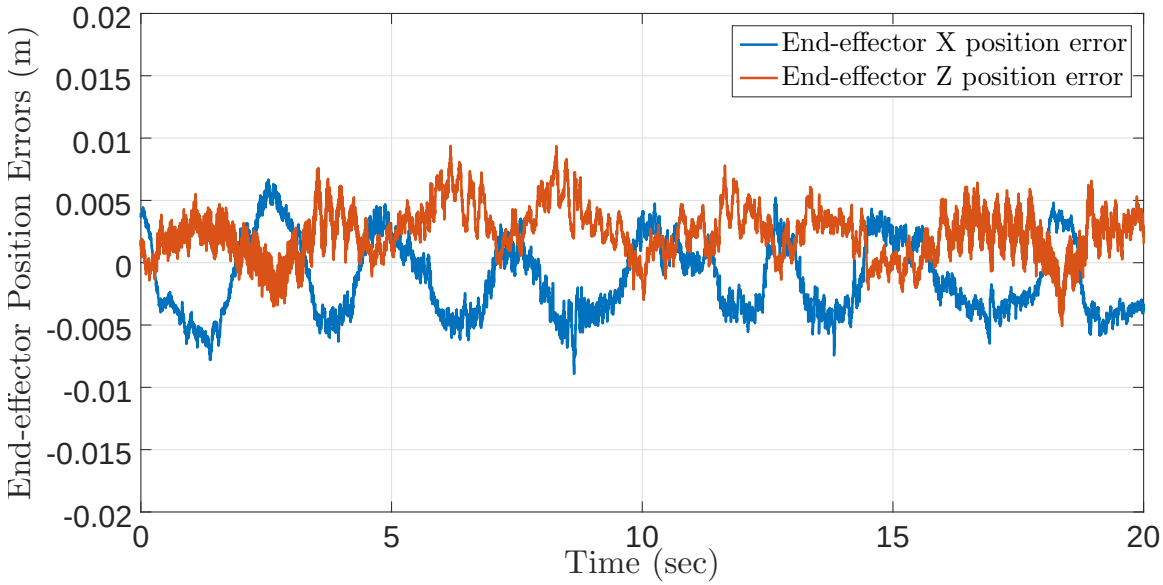

Figure 11.27: Scenario IV experiment results: End-effector position tracking errors

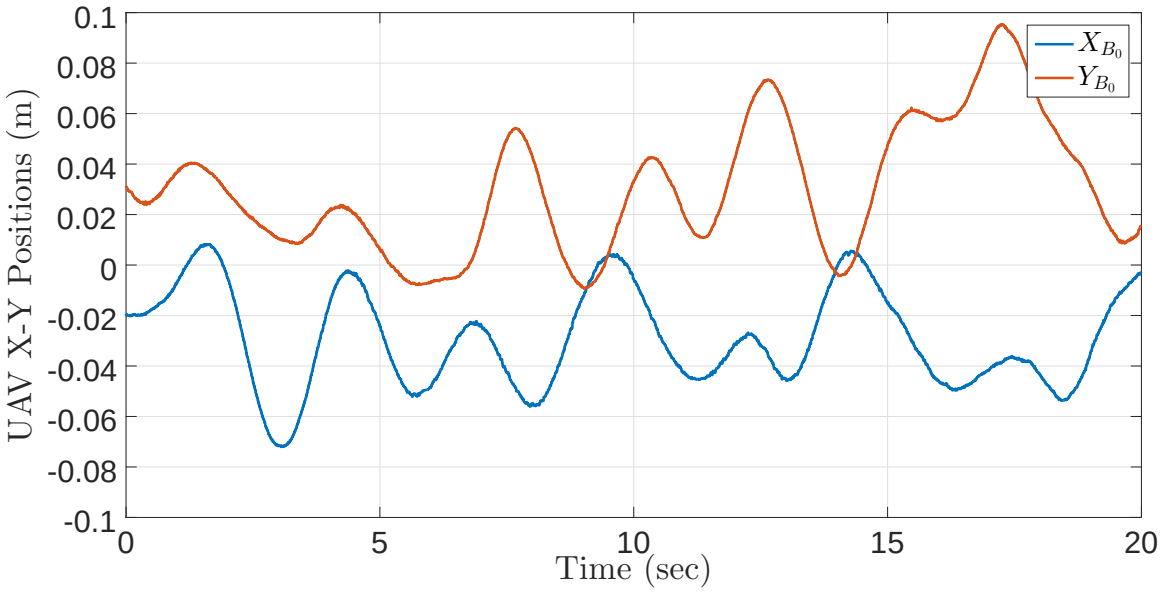

Figure 11.28: Scenario IV experiment results: UAV X-Y position coordinates

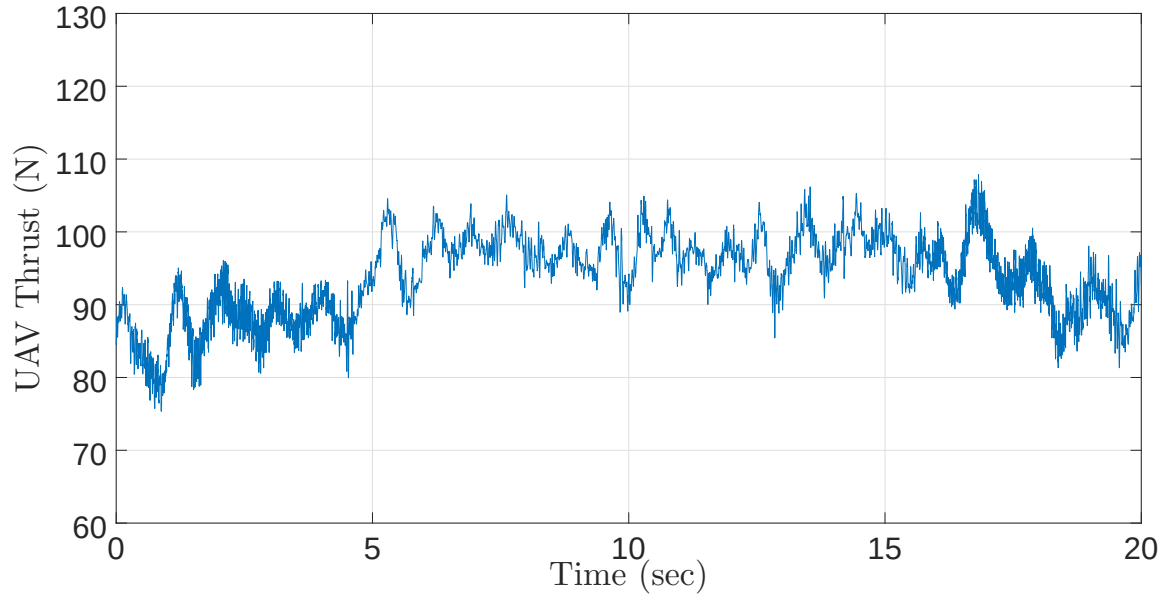

Figure 11.29: Scenario IV experiment results: UAV thrust actuation

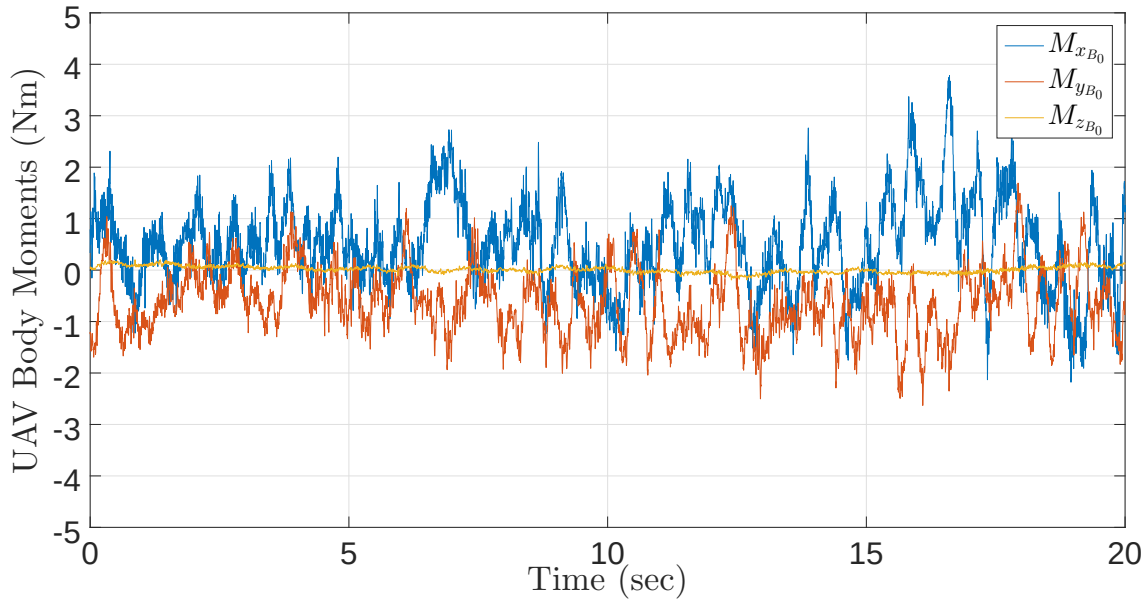

Figure 11.30: Scenario IV experiment results: UAV moment actuations

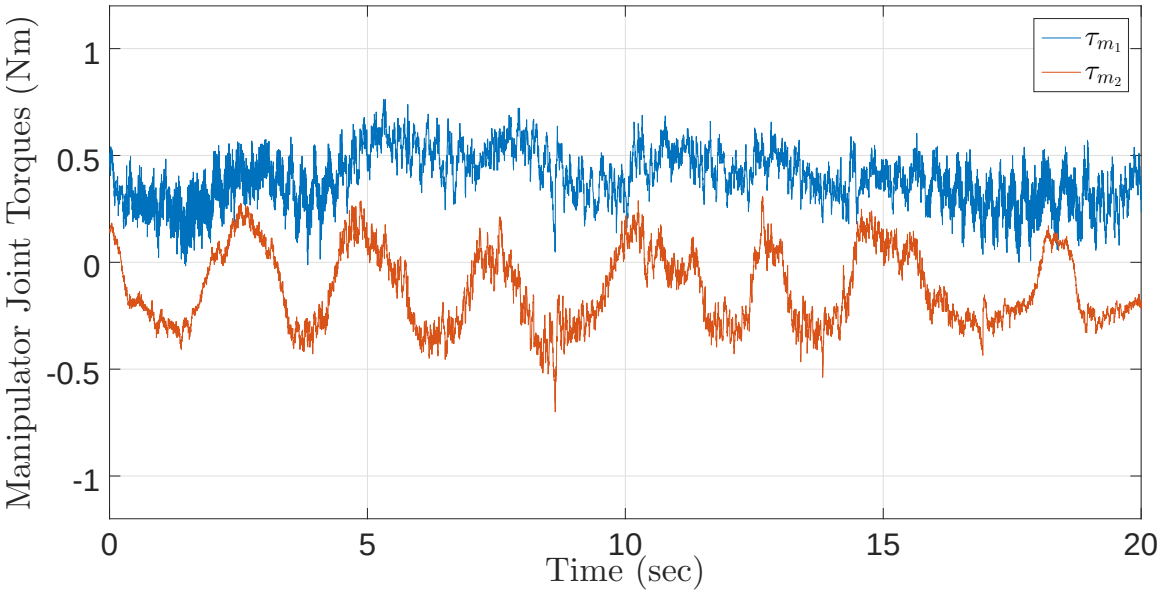

Figure 11.31: Scenario IV experiment results: Manipulator joint torques

The effectiveness of the control Scenario IV was further investigated in two other experiments. Fig. 11.32 shows the system configuration at different time instances during the second experiment. The purpose of this test is to illustrate accurate positioning of the end-effector despite base movements. Here, the end-effector first moves in the X direction to approach a target vertical line. Then it moves along the target line in the Z direction while the base also starts elevating. Upon reaching the end point of the target line, the end-effector holds its position at this point while the base continues to elevate. Finally, the end-effector sweeps the target line in the negative Z direction while the base also descends. A video demonstrating this experiment is available at https://youtu.be/9qpxdH5y_wc.

A third experiment is performed where the end-effector is commanded to regulate its position in the inertial X direction while the UAV undergoes a $45deg$ yaw orientation. In such a case, the end-effector is constrained to move along a line in the inertial Y direction since its desired X coordinate is constant. Fig. 11.33 shows four snapshots of the system during this experiment. First, the end-effector moves in the X direction towards a desired X coordinate. Then, the UAV starts to smoothly change its yaw orientation from 0 to $45deg$. The figure shows that the end-effector remains on a target line in the Y direction. As the yaw increases, the manipulator extends to hold the end-effector at the desired X coordinate. A video demonstrating this experiment is available at https://youtu.be/d9YMUf0nmXo.

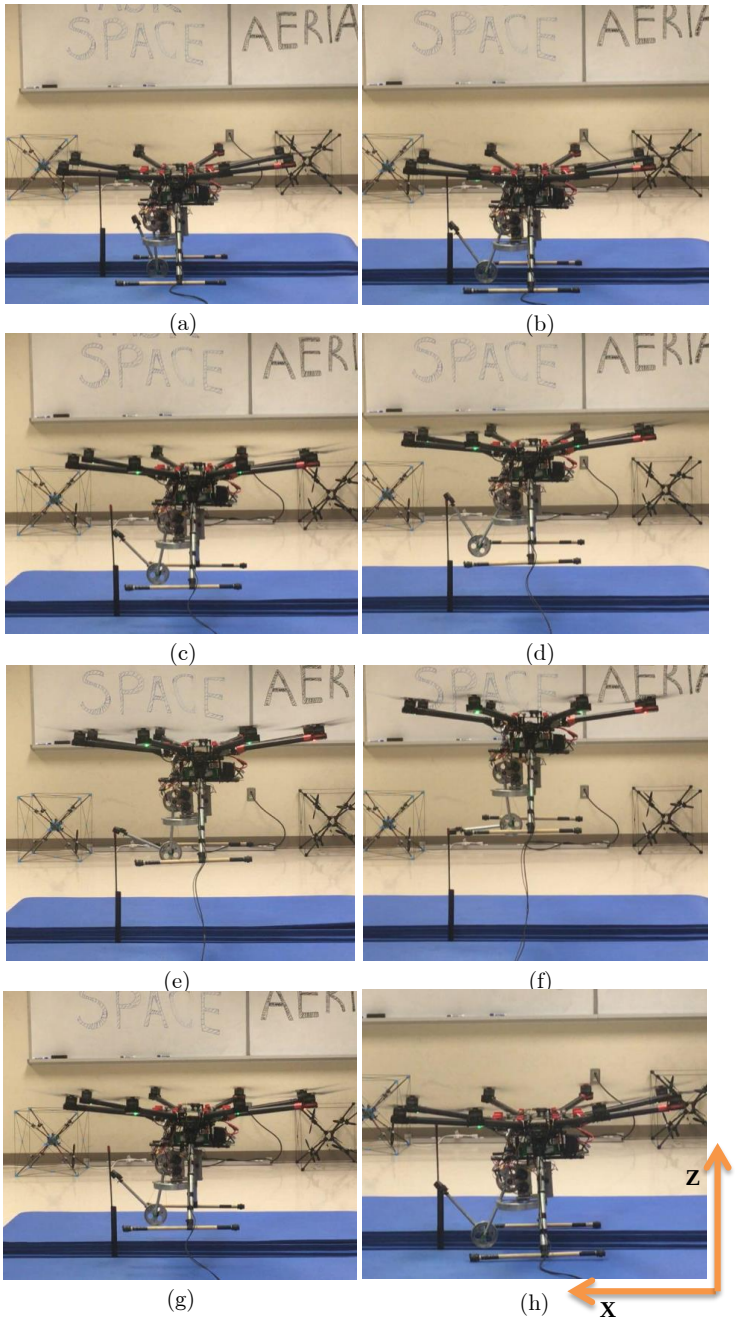

Figure 11.32: Scenario IV second experiment, (a): system in home configuration, (b): end-effector approaching the target line, (c): UAV elevating, end-effector moving upwards along the target line, (d): end-effector reaching the target point, (e): end-effector remains at the target point while the UAV continues to elevate, (f): UAV in hover, end-effector remains at the target point, (g): UAV descending, end-effector moving downwards along the target line, (h): system back to the home configuration.

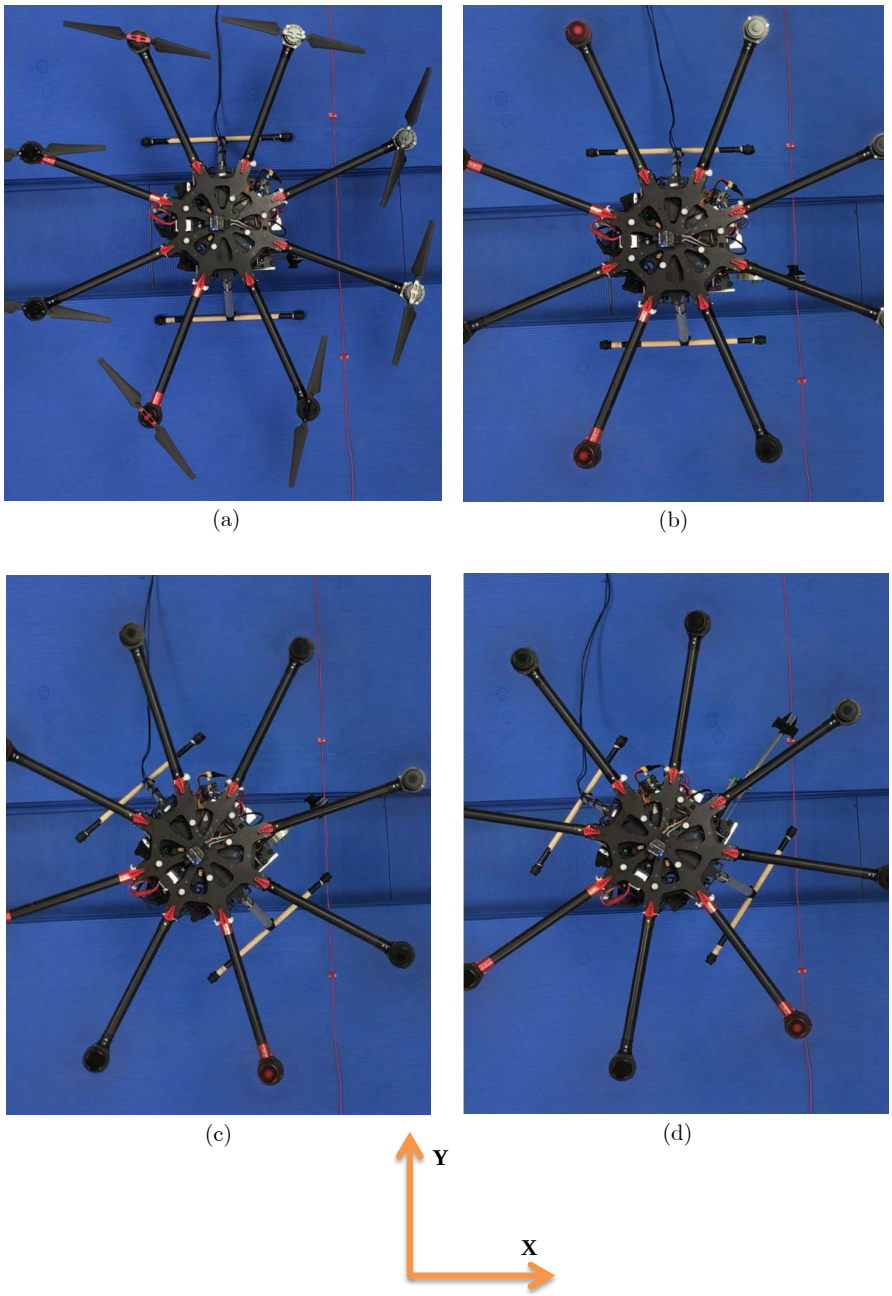

Figure 11.33: Scenario IV third experiment, (a): system in home configuration, (b): end-effector moving in X direction and approaching the red target line (c): end-effector remains on the target line while the UAV has a $30deg$ yaw orientation, (d): end-effector remains on the target line while the UAV has a $45deg$ yaw orientation.

Chapter 12

Conclusions and Future Work

12.1 Conclusions

This thesis was concerned with model-based motion control of aerial mobile manipulators composed of an under-actuated UAV and a fully actuated robotic manipulator. A complete theoretical treatment of the problem, including modeling, controller design, and performance and stability analysis was presented. High-performance stable control of such hyper-DOF robotic systems requires a model-based approach that takes into account the full coupled dynamics of the multi-body system, as well as under-actuation, kinematic redundancy, and uncertainty. Under-actuation significantly complicates the motion control problem for UAV-Manipulators in different aspects including motion planning, model-based design, and kinematic redundancy resolution. If not properly addressed, under-actuation can lead to drift and even instability as the UAV-Manipulator may reach uncontrollable/singular configurations.

Four control strategies were discussed and control development was carried out in details for two of the four scenarios. The controllers for the other two strategies can be

formulated using similar steps but were not presented to avoid repetition. The first control scenario is suitable for a semi-autonomous operation mode where a human operator could provide the UAV orientation reference commands to position the UAV in the horizontal plane as needed. The second control scenario is geared for fully autonomous operation where the trajectory planning could be performed off-line for the UAV position. In both cases, model-based adaptive control algorithms were proposed based on the method of virtual decomposition. The controllers addressed the under-actuation issue for both control strategies. These control algorithms use the second-order motion constraints to derive the required motion variables for the system dependent DOFs that are needed in model-based controller. In the second control scenario, where the manipulator task-space variables were independently controlled, a novel method was presented for resolving a possible kinematic redundancy within an under-actuated system. The proposed controllers can adapt to the system uncertain parameters. The decentralized parameter adaptation can be easily enabled/disabled for any rigid body in the system. This feature is especially useful for high performance aerial manipulation since the target object inertia properties are not known a priori.

Control algorithm developments, for both control scenarios, were extended to the case where propeller dynamics have significant contribution to overall system dynamics. In this case, the propeller motor torques as well as the manipulator joint torques are the control inputs. When the propeller dynamics are included, the system is under-actuated in six DOFs, which was taken into account in the control design.

Numerical simulations were carried out to examine the performance of the controller with the propeller dynamics using a realistic model of an octo-copter with a two-link serial manipulator. The results demonstrated high performance and stable tracking behaviour

whereas a conventional PID control exhibited poor response because of the base under-actuation as well as dynamic interaction between the flying base and the robotic arm.

An actual aerial manipulator system was developed to experimentally validate the proposed control approaches. The system combined a commercial conventional octo-coptor with two degrees of under-actuation with a serial 3R robotic arm. The model-based control algorithms in the first and fourth control scenarios were implemented using asynchronous multi-rate measurements from different sensing units. A motion capture system estimated the position and yaw orientation of the UAV. An on-boad IMU measured the UAV angular velocity as well as its roll and pitch orientation. High-resolution optical encoders were used to measure the manipulator joint positions. Numerical differentiation was carried out to estimate the UAV linear velocity as well as the manipulator joint velocities from the corresponding position variables. The experimental results demonstrated high performance and a stable tracking behaviour in all system DOFs despite limitations in sensing and actuation.

12.2 Future Work

Aerial manipulation is a recent field of research which has gained more attention lately from academia and industry. This area of research involves very challenging problems from trajectory planning and motion control to navigation, perception, and interaction with the environment. While this thesis laid a theoretical foundation for model-based motion control of under-actuated UAV-Manipulator systems, it leaves some interesting open problems for further investigation. Some possible avenues for future research are discussed below.

- The intended applications of this thesis require high-performance trajectory tracking for aerial manipulators which in turn needs accurate measurements of the system states. In particular, measuring the UAV position can be challenging. For indoor applications accurate motion tracking systems can be used to measure the UAV 3D position. For outdoor applications, however, there are limited options. Commercial GPS data is not sufficiently accurate or reliable. High precision Inertial Navigation Systems (INS) are available which combine IMU measurements with Real-Time-Kinematic Differential GPS (RTK-DGPS)[93]. Although, these measuring units can provide high accuracy position measurements ($2cm$ estimation error) with a high feedback rate ($200Hz$), they are relatively expensive [94]. The use of an on-board 3D vision in conjunction with IMU and GPS, for estimating the relative position/orientation of the UVA with respect to its intended target could be an approach to study for accurate motion control in the manipulation phase of the system operation in outdoor applications. This in turn would introduce new challenges in control problem among which are higher computational complexity of the algorithms, lower relative position/orientation update rates of the targeted object, and increased feedback time delays between perception and actuation. Some studies on vision-guided aerial manipulation are reported in [95; 96; 97; 98].

- Contact control of the UAV-Manipulator while interacting with environment is another interesting open problem. Development of control strategies which guarantee system stability as well as effective interaction with environment requires further investigation in the field of aerial manipulators. This control problem is particularly challenging due to the extra geometric constraints added through contact. These new constraints could be holonomic or even first-order nonholonomic. Handling the

under-actuation motion constraints in the presence of these new contact constraints introduces new challenges in the algorithm development for an effective model-based control.

- While motion planning for UAVs has been widely studied in the literature [99; 100; 101], there is far fewer results for the UAV-Manipulators [102; 103]. Similar to every other fully-autonomous system, on-line motion planning algorithms are required for applications of the UAV-Manipulator in unstructured environments. Localization, perception, obstacle detection and avoidance along with the presence of extra constraints such as less agility due to the added robotic arm, are among challenges involved in optimal path and trajectory planning of such systems, yielding plenty of room for future research.

- There are a few studies with preliminary results on cooperative control of multiple UAV-Manipulators [104; 105; 106; 107; 108; 109; 110]. Employing a team of such systems could expand their applications, an example of which is cooperative aerial assembly [111; 112]. However, this control problem gives rise to a number of challenges such as coordinated trajectory planning, optimal cooperative force distribution at contact, compliant interaction, and contact maintenance. Designing a high-level motion planning algorithm for coordinated motion and contact control of a team of UAV-Manipulators is a possible research direction where the model-based controllers developed in this thesis could be employed as the low-level motion control units of each UAV-Manipulator.

Appendix A

Appendix

A.1 Derivation of the G Matrix

The orientation of the body frame B_0 w.r.t the inertial frame I was represented by a set of Euler angle rotations. The order of these rotations is Yaw, Pitch, and Roll, respectively. More precisely, in order to align I with B_0, one first rotates I around the current z axis with Ψ_z; this is followed by a rotation about the current y axis with Θ_y; finally, a rotation about the current x axis with Φ_x yields B_0. Given this, the angular velocity vector can be written as,

$$\overrightarrow{\omega}_{B_0} = \dot{\Psi}_z \hat{\mathbf{n}}_z + \dot{\Theta}_y \hat{\mathbf{n}}_y + \dot{\Phi}_x \hat{\mathbf{n}}_x \tag{A.1}$$

where $\overrightarrow{\omega}_{B_0}$ is the vector of angular velocity of frame B_0, and $\hat{\mathbf{n}}_z$, $\hat{\mathbf{n}}_y$, and $\hat{\mathbf{n}}_x$ are the unit vectors of the successive rotations described before. The vector equation (A.1) can be described in any coordinate frame. Here, this equation needs to be expressed in the body-frame B_0 since the relationship between $\boldsymbol{\omega}_{B_0}$ (the angular velocity vector of B_0 expressed

in B_0) and the Euler angle rates ($\dot{\Psi}, \dot{\Theta}, \dot{\Phi}$) is investigated.

In order to find the expressions of $\hat{\mathbf{n}}_z$, $\hat{\mathbf{n}}_y$, and $\hat{\mathbf{n}}_x$ in B_0, one needs to find the rotation matrices of the intermediate coordinate frames w.r.t B_0. This is because these unit vectors are current axes of rotation and the principal axes of the intermediate frames. To this end, consider the following intermediate frames introduced in successive rotations from frame I to frame B_0,

$$I \xrightarrow{\text{Yaw}} B' \xrightarrow{\text{Pitch}} B'' \xrightarrow{\text{Roll}} B_0$$

The rotation matrix from B' to B'' is a basic rotation around $y_{B'}$ given by,

$$\mathbf{R}_{B''}^{B'}(\Theta_y) = \begin{bmatrix} \cos(\Theta_y) & 0 & \sin(\Theta_y) \\ 0 & 1 & 0 \\ -\sin(\Theta_y) & 0 & \cos(\Theta_y) \end{bmatrix}$$

Similarly, the rotation matrix from B'' to B_0 is a basic rotation around $x_{B''}$ given by,

$$\mathbf{R}_{B_0}^{B''}(\Phi_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi_x) & -\sin(\Phi_x) \\ 0 & \sin(\Phi_x) & \cos(\Phi_x) \end{bmatrix}$$

These further lead to the rotation matrix from B' to B_0 as follows,

$$\mathbf{R}_{B_0}^{B'}(\Theta_y, \Phi_x) = \mathbf{R}_{B''}^{B'}(\Theta_y)\, \mathbf{R}_{B_0}^{B''}(\Phi_x)$$

Given the above rotation matrices, one can express the unit vectors of rotation in frame B_0. The first vector of rotation $\hat{\mathbf{n}}_z$ is expressed in B' as $[0,\ 0,\ 1]^T$, consequently its expression in B_0 is given by,

$$ {}^{B_0}\mathbf{n}_z = \left(\mathbf{R}_{B_0}^{B'}\right)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\sin(\Theta_y) \\ \cos(\Theta_y)\sin(\Phi_x) \\ \cos(\Phi_x)\cos(\Theta_y) \end{bmatrix} \tag{A.2} $$

The second vector of rotation $\hat{\mathbf{n}}_y$ is expressed in B'' as $[0,\ 1,\ 0]^T$; its expression in B_0 is then given by,

$$ {}^{B_0}\mathbf{n}_y = \left(\mathbf{R}_{B_0}^{B''}\right)^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \cos(\Phi_x) \\ -\sin(\Phi_x) \end{bmatrix} \tag{A.3} $$

The third vector of rotation, $\hat{\mathbf{n}}_x$ is already expressed in B_0 as,

$$ {}^{B_0}\mathbf{n}_x = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{A.4} $$

Finally, replacing (A.2), (A.3), and (A.4) in (A.1) yields,

$$ \boldsymbol{\omega}_{B_0} = \begin{bmatrix} 1 & 0 & -\sin(\Theta_y) \\ 0 & \cos(\Phi_x) & \cos(\Theta_y)\sin(\Phi_x) \\ 0 & -\sin(\Phi_x) & \cos(\Phi_x)\cos(\Theta_y) \end{bmatrix} \begin{bmatrix} \dot{\Phi}_x \\ \dot{\Theta}_y \\ \dot{\Psi}_z \end{bmatrix} $$

A.2 Derivation of the Thrust-PWM Mapping

Two sets of equations must be considered for developing the mapping between the steady-state propeller thrust force and the commanded PWM to the ESCs. The first set is the equations of the aerodynamic force/torque and the second one is the equations describing

the motor electric circuit which actuates the propeller.

Referring to (3.28), the propeller thrust force and the drag torque are given as follows [82; 83; 84],

$$F_{thrust} = k_T \rho D^4 \dot{q}_p^2 \tag{A.5}$$

$$\tau_{drag} = k_Q \rho D^5 \dot{q}_p^2 \tag{A.6}$$

where $\dot{q}_p$ is the propeller speed and all the other constants have been defined before. This further yields,

$$\tau_{drag} = \frac{D k_Q}{k_T} F_{thrust} \tag{A.7}$$

For an electric dc motor, the source voltage across the motor V_m and the current through the motor I_m have the following relationship,

$$V_m = k_{bemf} \dot{q}_p + R_m I_m \tag{A.8}$$

where k_{bemf} is the back EMF voltage coefficient and R_m is the motor circuit resistance. Here, the motor inductance effects are ignored. Furthermore, the following relationship holds for the motor current I_m and the motor actuation torque τ_p,

$$\tau_p = k_m I_m \tag{A.9}$$

where k_m is the motor torque constant. Substituting (A.9) in (A.8) yields,

$$V_m = k_{bemf} \dot{q}_p + \frac{1}{k_m} \tau_p \tag{A.10}$$

Since the propeller inertia is considerably small, the propeller generated drag torque and the motor actuation torque are assumed to be equal, as a result,

$$\tau_p = \tau_{drag} \tag{A.11}$$

Back substitution of (A.11), (A.7), and (A.5) in (A.10) yields,

$$V_m = \frac{Dk_Q}{k_m k_T} F_{thrust} + \frac{k_{bemf}}{D^2\sqrt{\rho k_T}} \sqrt{F_{thrust}} \tag{A.12}$$

The effective voltage across the motor V_m is equal to $D_{pwm} V_{battery}$. Moreover, ESCs are usually calibrated such that they do not react to duty cycles below a certain value. Taking these two into account, Equation (A.12) is modified as follows,

$$D_{pwm} = \gamma_1 F_{thrust} + \gamma_2 \sqrt{F_{thrust}} + \gamma_3 \tag{A.13}$$

where $\gamma_1 = \frac{Dk_Q}{k_m k_T V_{battery}}$, $\gamma_2 = \frac{k_{bemf}}{D^2\sqrt{\rho k_T} V_{battery}}$, and γ_3 is the duty cycle threshold below which the ESCs are not activated.

Bibliography

[1] O. Khatib, "Mobile manipulation: The robotic assistant," *Robotics and Autonomous Systems*, vol. 26, no. 2, pp. 175 – 183, 1999.

[2] S. A. A. Moosavian and K. Alipour, "On the dynamic tip-over stability of wheeled mobile manipulators," *International Journal of Robotics & Automation*, vol. 22, no. 4, p. 322, 2007.

[3] S. Dubowsky and E. E. Vance, "Planning mobile manipulator motions considering vehicle dynamic stability constraints," in *Proceedings of the International Conference on Robotics and Automation*, vol. 3, May 1989, pp. 1271–1276.

[4] J. P. Desai and V. Kumar, "Nonholonomic motion planning for multiple mobile manipulators," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, Apr 1997, pp. 3409–3414.

[5] H. Seraji, "A unified approach to motion control of mobile manipulators," *The International Journal of Robotics Research*, vol. 17, no. 2, pp. 107–118, 1998.

[6] N. A. M. Hootsmans and S. Dubowsky, "Large motion control of mobile manipulators including vehicle suspension characteristics," in *Proceedings of the IEEE*

International Conference on Robotics and Automation (ICRA), vol. 3, Apr 1991, pp. 2336–2341.

[7] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 53–64, Feb 2003.

[8] M. Souzanchi-K, A. Arab, M. R. Akbarzadeh-T., and M. M. Fateh, "Robust impedance control of uncertain mobile manipulators using time-delay compensation," *To appear in the IEEE Transactions on Control Systems Technology*, 2017.

[9] Z. Li, S. S. Ge, and A. Ming, "Adaptive robust motion/force control of holonomic-constrained nonholonomic mobile manipulators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 3, pp. 607–616, June 2007.

[10] H. Osumi, M. Terasawa, and H. Nojiri, "Cooperative control of multiple mobile manipulators on uneven ground," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, May 1998, pp. 3198–3203.

[11] T. G. Sugar and V. Kumar, "Control of cooperating mobile manipulators," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 94–103, Feb 2002.

[12] Z. Li, J. Li, and Y. Kang, "Adaptive robust coordinated control of multiple mobile manipulators interacting with rigid environments," *Automatica*, vol. 46, no. 12, pp. 2028 – 2034, 2010.

[13] A. Ollero, "Aerial Robotics Cooperative Assembly System (ARCAS): First results," in *Aerial Physically Acting Robots (AIRPHARO) workshop, IROS, Vilamoura, Portugal*, 2012.

[14] K. Kondak, K. Krieger, A. Albu-Schaeffer, M. Schwarzbach, M. Laiacker, I. Maza, A. Rodriguez-Castano, and A. Ollero, "Closed-loop behavior of an autonomous helicopter equipped with a robotic arm for aerial manipulation tasks," *International Journal of Advanced Robotic Systems*, vol. 10, no. 2, p. 145, 2013.

[15] M. W. Spong, "Underactuated mechanical systems," in *Control problems in robotics and automation*, Springer, 1998, pp. 135–150.

[16] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, Jun 1997.

[17] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE Journal on Robotics and Automation*, vol. 4, no. 4, pp. 403–410, Aug 1988.

[18] R. G. Roberts and A. A. Maciejewski, "Repeatable generalized inverse control strategies for kinematically redundant manipulators," *IEEE Transactions on Automatic Control*, vol. 38, no. 5, pp. 689–699, May 1993.

[19] W. Zhu, B. Z., and J. De Schutter, "Adaptive motion/force control of multiple manipulators with joint flexibility based on virtual decomposition," *IEEE Transactions on Automatic Control*, vol. 43, no. 1, pp. 46–60, Jan 1998.

[20] W. Zhu, *Virtual Decomposition Control: Toward Hyper Degrees of Freedom Robots*. Springer, 2010, vol. 60.

[21] M. Jafarinasab and S. Sirouspour, "Adaptive motion control of aerial robotic manipulators based on virtual decomposition," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 1858–1863.

[22] D. Findlay, M. Jafarinasab, and S. Sirouspour, "Optimization-based design of a novel hybrid aerial/ground mobile manipulator," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 2467–2472.

[23] P. E. Pounds, D. R. Bersak, and A. M. Dollar, "Grasping from the air: Hovering capture and load stability," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 2491–2498.

[24] P. E. Pounds and A. M. Dollar, "Stability of helicopters in compliant contact under pd-pid control," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1472–1486, 2014.

[25] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 2668–2673.

[26] V. Ghadiok, J. Goldin, and W. Ren, "Autonomous indoor aerial gripping using a quadrotor," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 4645–4651.

[27] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction with quadrotor teams," *Autonomous Robots*, vol. 33, no. 3, pp. 323–336, 2012.

[28] J. Willmann, F. Augugliaro, T. Cadalbert, R. D'Andrea, F. Gramazio, and M. Kohler, "Aerial robotic construction towards a new field of architectural research," *International journal of architectural computing*, vol. 10, no. 3, pp. 439–459, 2012.

[29] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed autonomous robotic systems*. Springer, 2013, pp. 545–558.

[30] I. Palunko, R. Fierro, and P. Cruz, "Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 2691–2697.

[31] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning swing-free trajectories for uavs with a suspended load," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 4902–4909.

[32] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[33] M. Orsag, C. Korpela, S. Bogdan, and P. Oh, "Dexterous aerial robots ;mobile manipulation using unmanned aerial systems," *To appear in the IEEE Transactions on Robotics*, 2017.

[34] C. Korpela, M. Orsag, and P. Oh, "Towards valve turning using a dual-arm aerial manipulator," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, 2014, pp. 3411–3416.

[35] S. Kim, H. Seo, and H. J. Kim, "Operating an unknown drawer using an aerial manipulator," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5503–5508.

[36] T. W. Danko and P. Y. Oh, "Design and control of a hyper-redundant manipulator for mobile manipulating unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, p. 709, 2014.

[37] B. Yüksel, S. Mahboubi, C. Secchi, H. H. Bülthoff, and A. Franchi, "Design, identification and experimental testing of a light-weight flexible-joint arm for aerial physical interaction," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 870–876.

[38] 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 *Proceedings of the 23th Mediterranean Conference on Control and Automation (MED)*, 2015, pp. 853–858.

[39] S. Quintanar-Guzmán, S. Kannan, M. A. Olivares-Mendez, and H. Voos, "Lightweight robotic arm actuated by shape memory alloy (sma) wires," in *Proceedings of the 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2016, pp. 1–6.

[40] A. Suarez, A. Jimenez-Cano, V. Vega, G. Heredia, A. Rodríguez-Castaño, and

A. Ollero, "Lightweight and human-size dual arm aerial manipulator," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 1778–1784.

[41] A. Suarez, G. Heredia, and A. Ollero, "Lightweight compliant arm for aerial manipulation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1627–1632.

[42] M. Kamel, K. Alexis, and R. Siegwart, "Design and modeling of dexterous aerial manipulator," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4870–4876.

[43] D. Findlay, M. Jafarinasab, and S. Sirouspour, "Optimization-based design of a novel hybrid aerial/ground mobile manipulator," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 2467–2472.

[44] A. E. Jimenez-Cano, G. Heredia, and A. Ollero, "Aerial manipulator with a compliant arm for bridge inspection," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2017, pp. 1217–1222.

[45] A. Macchelli, F. Forte, A. Keemink, S. Stramigioli, R. Carloni, M. Fumagalli, R. Naldi, and L. Marconi, "Developing an aerial manipulator prototype," *IEEE Robot Autom Mag*, vol. 21, no. 3, pp. 41–55, 2014.

[46] A. Q. Keemink, M. Fumagalli, S. Stramigioli, and R. Carloni, "Mechanical design of a manipulation system for unmanned aerial vehicles," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 3147–3152.

[47] H. Tsukagoshi, M. Watanabe, T. Hamada, D. Ashlih, and R. Iizuka, "Aerial manipulator with perching and door-opening capability," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4663–4668.

[48] M. Orsag, C. Korpela, M. Pekala, and P. Oh, "Stability control in aerial manipulation," in *Proceedings of the American Control Conference (ACC)*, June 2013, pp. 5581–5586.

[49] M. Orsag, C. Korpela, and P. Oh, "Modeling and Control of MM-UAV: Mobile Manipulating Unmanned Aerial Vehicle," *Journal of Intelligent and Robotic Systems*, vol. 69, no. 1-4, pp. 227–240, 2013.

[50] A. Torre, D. Mengoli, R. Naldi, F. Forte, A. Macchelli, and L. Marconi, "A prototype of aerial manipulator," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2012, pp. 2653–2654.

[51] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, "Modeling and control of a flying robot for contact inspection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2012, pp. 3532–3537.

[52] M. Orsag, C. Korpela, and P. Oh, "Modeling and control of mm-uav: Mobile manipulating unmanned aerial vehicle," *Journal of Intelligent & Robotic Systems*, pp. 1–14, 2013.

[53] G. Chmaj, T. Buratowski, T. Uhl, K. Seweryn, and M. Banaszkiewicz, "The dynamics influence of the attached manipulator on unmanned aerial vehicle," in *Aerospace Robotics*, Springer, 2013, pp. 109–119.

[54] 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 *Proceedings of the 24th Mediterranean Conference on Control and Automation (MED)*, 2016, pp. 665–670.

[55] C. M. Korpela, T. W. Danko, and P. Y. Oh, "Designing a system for mobile manipulation from an unmanned aerial vehicle," in *Proceedings of the IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, 2011, pp. 109–114.

[56] C. Korpela, M. Orsag, M. Pekala, and P. Oh, "Dynamic stability of a mobile manipulating unmanned aerial vehicle," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 4922–4927.

[57] J. R. Kutia, K. A. Stol, and W. Xu, "Initial flight experiments of a canopy sampling aerial manipulator," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 1359–1365.

[58] S. Kannan, M. Alma, M. A. Olivares-Mendez, and H. Voos, "Adaptive control of aerial manipulation vehicle," in *Proceedings of the IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2014, pp. 273–278.

[59] M. Orsag, C. Korpela, S. Bogdan, and P. Oh, "Lyapunov based model reference adaptive control for aerial manipulation," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, 2013, pp. 966–973.

[60] M. Orsag, C. M. Korpela, S. Bogdan, and P. Y. Oh, "Hybrid adaptive control for aerial manipulation," *Journal of intelligent & robotic systems*, vol. 73, no. 1-4, p. 693, 2014.

[61] X. Meng, Y. He, F. Gu, Q. Li, and J. Han, "Dynamics modeling and simulation analysis for rotorcraft aerial manipulator system," in *Proceedings of the 36th Chinese Control Conference (CCC)*, 2017, pp. 1156–1161.

[62] A. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano, "Control of an aerial robot with multi-link arm for assembly tasks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 4916–4921.

[63] F. Ruggiero, M. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Peréz, V. Lippiello, A. Ollero, and B. Siciliano, "A multilayer control for multirotor uavs equipped with a servo robot arm," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4014–4020.

[64] J. Acosta, C. de Cos, and A. Ollero, "A robust decentralised strategy for multi-task control of unmanned aerial systems. application on underactuated aerial manipulator," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 1075–1084.

[65] M. Sánchez, J. Acosta, and A. Ollero, "Integral action in first-order closed-loop inverse kinematics. application to aerial manipulators," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5297–5302.

[66] S. Kim, S. Choi, and H. Kim, "Aerial manipulation using a quadrotor with a two DOF robotic arm," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2013, pp. 4990–4995.

[67] H. Lee, S. Kim, and H. J. Kim, "Control of an aerial manipulator using on-line

parameter estimator for an unknown payload," in *Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE)*, 2015, pp. 316–321.

[68] M. Kobilarov, "Nonlinear Trajectory Control of Multi-body Aerial Manipulators," *Journal of Intelligent and Robotic Systems*, vol. 73, no. 1-4, pp. 679–692, 2014.

[69] H. Yang and D. Lee, "Dynamics and control of quadrotor with robotic manipulator," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 5544–5549.

[70] P. O. Pereira, R. Zanella, and D. V. Dimarogonas, "Decoupled design of controllers for aerial manipulation with quadrotors," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4849–4855.

[71] J. Acosta, M. Sanchez, and A. Ollero, "Robust control of underactuated aerial manipulators via ida-pbc," in *Proceedings of the 53rd IEEE Conference on Decision and Control*, 2014, pp. 673–678.

[72] R. Naldi, P. Pounds, S. D. Marco, and L. Marconi, "Output tracking for quadrotor-based aerial manipulators," in *Proceedings of the American Control Conference (ACC)*, July 2015, pp. 1855–1860.

[73] A. Y. Mersha, S. Stramigioli, and R. Carloni, "Exploiting the dynamics of a robotic manipulator for control of uavs," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1741–1746.

[74] F. Caccavale, G. Giglio, G. Muscio, and F. Pierri, "Adaptive control for uavs

equipped with a robotic arm," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11 049–11 054, 2014.

[75] V. Lippiello and F. Ruggiero, "Exploiting redundancy in Cartesian impedance control of UAVs equipped with a robotic arm," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2012, pp. 3768–3773.

[76] E. Cataldi, G. Muscio, M. A. Trujillo, Y. Rodríguez, F. Pierri, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Impedance control of an aerial-manipulator: Preliminary results," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 3848–3853.

[77] D. Titterton and J. L. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.

[78] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley New York, 2006, vol. 3.

[79] B. Siciliano, "A closed-loop inverse kinematic scheme for on-line joint-based robot control," *Robotica*, vol. 8, no. 3, pp. 231–243, 1990.

[80] F. Caccavale, S. Chiaverini, and B. Siciliano, "Second-order kinematic control of robot manipulators with jacobian damped least-squares inverse: Theory and experiments," *IEEE/ASME Transactions on Mechatronics*, vol. 2, no. 3, pp. 188–194, 1997.

[81] P. Falco and C. Natale, "On the stability of closed-loop inverse kinematics algorithms

for redundant robots," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 780–784, 2011.

[82] M. S. Selig, "Modeling propeller aerodynamics and slipstream effects on small uavs in realtime," *AIAA Paper*, vol. 7938, 2010.

[83] J. B. Brandt, M. S. Selig *et al.*, "Propeller performance data at low reynolds numbers," in *49th AIAA aerospace sciences meeting*, 2011, pp. 2011–1255.

[84] F. J. Ruiz-Sanchez, "Model of thrust/load-torque in terms of geometric parameters of the blade for design and simulation of small scale propellers used in miniature uavs," in *Proceedings of the IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, Nov 2016, pp. 1–6.

[85] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and adaptive control design*. Wiley, 1995.

[86] G. Oriolo and Y. Nakamura, "Control of mechanical systems with second-order nonholonomic constraints: underactuated manipulators," in *Proceedings of the 30th IEEE Conference on Decision and Control*, vol. 3, Dec 1991, pp. 2398–2403.

[87] H. Arai, K. Tanie, and N. Shiroma, "Nonholonomic control of a three-dof planar underactuated manipulator," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 681–695, Oct 1998.

[88] G. Oriolo and Y. Nakamura, "Free-joint manipulators: motion control under second-order nonholonomic constraints," in *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems*, vol. 3, Nov 1991, pp. 1248–1253.

[89] G. Tao, "A simple alternative to the Barbalat lemma," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 698–, May 1997.

[90] "Motion capture systems," http://www.optitrack.com.

[91] "Dji-s1000," https://www.dji.com.

[92] "Multiwii serial protocol," http://www.multiwii.com/wiki.

[93] G. Heredia, A. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J. Acosta, and A. Ollero, "Control of a multirotor outdoor aerial manipulator," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 3417–3422.

[94] "High precision inertial navigation systems," https://www.sbg-systems.com.

[95] V. Lippiello, J. Cacace, A. Santamaria-Navarro, J. Andrade-Cetto, M. A. Trujillo, Y. R. Esteves, and A. Viguria, "Hybrid visual servoing with hierarchical task composition for aerial manipulation," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 259–266, 2016.

[96] A. Santamaria-Navarro, P. Grosch, V. Lippiello, J. Sola, and J. Andrade-Cetto, "Uncalibrated visual servo for unmanned aerial manipulation," *To appear in IEEE/ASME Transactions on Mechatronics*, 2017.

[97] M. Laiacker, F. Huber, and K. Kondak, "High accuracy visual servoing for aerial manipulation using a 7 degrees of freedom industrial manipulator," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1631–1636.

[98] S. Kim, H. Seo, S. Choi, and H. J. Kim, "Vision-guided aerial manipulation using a multirotor with a robotic arm," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 4, pp. 1912–1923, 2016.

[99] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous uav guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, p. 65, 2010.

[100] S. A. Bortoff, "Path planning for uavs," in *Proceedings of the American Control Conference (ACC)*, vol. 1, no. 6, 2000, pp. 364–368.

[101] T. W. McLain and R. W. Beard, "Trajectory planning for coordinated rendezvous of unmanned air vehicles," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, vol. 4369, 2000, pp. 1–8.

[102] D. Lunni, A. Santamaria-Navarro, R. Rossi, P. Rocco, L. Bascetta, and J. Andrade-Cetto, "Nonlinear model predictive control for aerial manipulation," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 87–93.

[103] G. Garimella and M. Kobilarov, "Towards model-predictive control for aerial pick-and-place," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4692–4697.

[104] H. Yang and D. Lee, "Hierarchical cooperative control framework of multiple quadrotor-manipulator systems," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4656–4662.

[105] H. Lee and H. J. Kim, "Constraint-based cooperative control of multiple aerial manipulators for handling an unknown payload," *IEEE Transactions on Industrial Informatics*, 2017.

[106] J. Kim, H.-N. Nguyen, and D. Lee, "Preliminary control design on spherically-connected multiple-quadrotor manipulator system," in *Proceedings of the 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2015, pp. 206–207.

[107] M. Mohammadi, A. Franchi, D. Barcelli, and D. Prattichizzo, "Cooperative aerial tele-manipulation with haptic feedback," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 5092–5098.

[108] H. Lee, H. Kim, and H. J. Kim, "Planning and control for collision-free cooperative aerial transportation," *IEEE Transactions on Automation Science and Engineering*, 2016.

[109] B. Arbanas, A. Ivanovic, M. Car, T. Haus, M. Orsag, T. Petrovic, and S. Bogdan, "Aerial-ground robotic system for autonomous delivery tasks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5463–5468.

[110] H. Lee, H. Kim, and H. J. Kim, "Path planning and control of multiple aerial manipulators for a cooperative transportation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 2386–2391.

[111] G. Muscio, F. Pierri, M. A. Trujillo, E. Cataldi, G. Giglio, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, and A. Ollero, "Experiments on coordinated motion of

aerial robotic manipulators," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1224–1229.

[112] F. Caccavale, G. Giglio, G. Muscio, and F. Pierri, "Cooperative impedance control for multiple uavs with a robotic arm," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 2366–2371.