Adaptive Nonlinear Control for Unmanned Aerial Vehicles: Visual Servoing and Aerial Manipulation

by

Muhammad Awais Rafique

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Control Systems

Department of Electrical and Computer Engineering

University of Alberta

Abstract

With the improvements in the autonomy and capability of Unmanned Aerial Vehicles (UAVs), there is an increased interest in their applications in infrastructure inspection and maintenance. The focus of this thesis is to study new methods for improved UAV autonomy. In this regard, the motion control of two UAV configurations is studied. First, a novel visual servoing control is presented, intended for inspection of linear structures e.g., power transmission lines and pipelines. Second, the motion control of an Unmanned Aerial Manipulator (UAM) that attaches a robotic arm to the UAV is proposed. This configuration makes a UAV capable of tasks involving interaction with the environment e.g., maintenance or disaster recovery.

The inspection of linear structures such as power transmission lines depends on high-quality video used for managing maintenance and repair. A UAV with an onboard camera is ideally suited to safely and efficiently collect this inspection data. Accurate and robust motion control is key to obtaining quality line video. Traditionally, UAV position and linear velocity estimates are obtained using a Global Positioning System (GPS). However, GPS lacks the accuracy needed for close inspection and its dependence on an external signal limits the vehicle's autonomy. Also, the infrastructure spans a vast area and is often inaccurately mapped. Image-Based Visual Servoing (IBVS) is an appropriate framework for accurately controlling the relative position between the UAV and a linear target. IBVS detects the line in the image and describes its relative position and yaw using image features. The coordinates of these features are used directly in the state feedback control for UAVs. We present a new IBVS method with a number of features, including an output feedback design that removes the need for linear velocity measurements. The control

adapts to change in sensor bias, vehicle thrust constant and external disturbances. Also, the proposed approach is robust against variations in vehicle mass and camera focal length. An inner-outer loop control structure is used. The feature and linear velocity estimate errors are shown to be exponentially convergent using Lyapunov stability analysis. The effectiveness of the approach is evaluated in simulation and experiment.

The second part of the thesis considers the motion control of a UAM, which is a UAV combined with a multiple degrees of freedom (DOF) robot arm. Unlike traditional UAVs, a UAM is designed to interact with the environment and used in maintenance and similar applications. Motion control of a UAM is a challenging problem, given the system's coupled dynamics. The effect of arm motion on the UAV results in a highly nonlinear and complex system model. We present the dynamic model of the UAM and simplify it to obtain UAV dynamics which includes uncertainty, to model the effect of the arm. We propose two motion controllers using the adaptive backstepping approach based on this model. An inner-outer loop control structure, which treats translational and rotational dynamics of UAV subsystem separately and designs respective outer and inner loop controllers, is employed. The outer loop provides thrust input and roll and pitch references that serve as reference trajectories for the inner loop control, providing torque inputs. Since the inner-outer loop control approach lacks the stability analysis for the entire closed-loop, another adaptive backstepping method that uses the entire UAV dynamics is presented. The proposed approaches are rigorously tested in a multi-body simulation environment.

Preface

- An earlier version of Sections 3.2 and 3.3 have been published as [M. A. Rafique and A. F. Lynch, "Vision-based motion control of UAVs for GPS-denied environments," in Proceedings of the 2018 Unmanned Systems Canada Conference, Vancouver, BC, Oct. 2018, pp. 1–16]. I was responsible for theoretical derivation, simulation and manuscript composition. Dr. Lynch was the supervisory author and was involved with the concept formation and manuscript composition.

- Sections 3.2, 3.3 and 3.4 in the current form have been published as [M. A. Rafique and A. F. Lynch, "Output-Feedback Image-Based Visual Servoing for Multirotor Unmanned Aerial Vehicle Line Following," IEEE Transactions on Aerospace and Electronic Systems, vol. 56, no. 4, pp. 3182–3196, 2020]. I was responsible for theoretical derivation, experimental implementation and manuscript composition. Dr. Lynch was the supervisory author and was involved with the concept formation and manuscript composition.

- Section 2.2.4 and Chapter 4 has been submitted as [M. A. Rafique and A. F. Lynch, "Modelling and Decentralized Control of UAV equipped with a robot arm," Robotica, Submitted: Dec. 2021.]. I was responsible for theoretical derivation, simulation and manuscript composition. Dr. Lynch was the supervisory author and was involved with the concept formation and manuscript composition.

- Sections 5.3, 5.5 and 5.6 have been submitted as [M. A. Rafique and A. F. Lynch, "Adaptive Backstepping control of Unmanned Aerial Vehicles with unknown parameters and disturbances," Journal of Intelligent & Robotic Systems, Submitted: Dec. 2021]. I was responsible for theoretical derivation, simulation and manuscript composition. Dr. Lynch was the supervisory author and was involved with the concept formation and manuscript composition.

To my wife, parents, and cute little daughter Zoya whose innocent smiles fuel my life.

Acknowledgements

All glory to the creator of this universe, who blessed me with potential thoughts, talented teachers, helping friends, loving wife, parents, daughter, sisters and brother, and an opportunity to make this humble contribution.

I express my most sincere appreciation and thanks to my supervisor Dr. Alan Lynch for his outstanding leadership, encouragement and guidance. His keen interest, personal involvement and criticism for the betterment were all sources of courage, inspiration and strength during the completion of this thesis. I want to thank my committee members, Dr. Qing Zhao and Dr. Hui Xie, for providing valuable and constructive feedback on my research. I also want to extend my gratefulness to the external examiner, Dr. Farrokh Janabi-Sharifi, for providing a detailed review of my thesis that led to significant improvements.

I am thankful to my friends and colleagues, including Dr. Geoff Fink, Dr. Amir Moeini, Zifei Jiang, Arash M. Hasani, Mohamed Al Lawati and other folks at the Applied Nonlinear Control Laboratory and friends in Canada and abroad, for their continuous support. Finally, I would like to express my deepest gratitude to my wife, parents, brother, sisters, and other friends and relatives for their emotional and moral support throughout my academic career and their love, patience, encouragement, and prayers.

Table of Contents

List of Tables

List of Figures

List of Symbols

a	Inverse of UAM mass m
α	Joint coordinates of arm, except Sec. 2.2.2: angle of a line
α_m	Mean of line angle parameter α for multiple lines
$\mathcal{B}$	Body fixed frame attached to the GC of a UAV
$C(q, \dot{q})$	Coriolis-Centrifugal matrix
c_{kj}	Christoffel symbols of the first kind
c_ξ	Cosine of ξ, i.e., $\cos \xi$
d_f	Sum of arm and external force disturbances on a UAM-V
d_τ	Sum of arm and external torque disturbances a UAM-V
D_h	Transformed disturbance in height subsystem
D_l	Transformed disturbance in lateral subsystem
δ	Disturbance term in $\dot{v}^v$ dynamics
e_h	Height moment feature error
e_l	Lateral moment feature error
e_ψ	Yaw moment feature error
η	UAV attitude vector $[\phi, \theta, \psi]^T$
f_T	UAV normalized thrust input
F	UAV force input vector
g	Gravitational acceleration constant
$G(q)$	Gravitational vector
$J, J_{\eta,\eta}$	Inertia of UAM-V including effects due to arm
J_q	Inertial of a UAV
J_{ω,L_i}	Jacobian for angular velocity of the ith link in $\mathcal{B}$
J_{v,L_i}	Jacobian for linear velocity of the ith link in $\mathcal{B}$
K	Kinetic energy of UAM
L	Lagrangian $= K - P$
m	Mass of UAM
m_q	UAV mass
$M(q)$	Inertia Tensor for UAM
μ	Variance of ρ for multiple lines, except Sec. 4.2: regressor function

$\mathcal{N}$	Navigation or Inertial frame
O_c	Optical center of a camera in 3D
O_I	Image center in 2D image plane
ω	UAV angular velocity in $\mathcal{B}$
ω_{L_i}	Angular velocity of the ith link in $\mathcal{B}$
p	Only for Sec. 2.2.2: an arbitrary point on 2D image plane
p, p^n	Position of COM of UAV in $\mathcal{N}$
p^b	Position of COM of UAV in $\mathcal{B}$
$p^b_{L_i}$	Position of COM of ith link of arm in $\mathcal{B}$
$p^n_{L_i}$	Position of COM of ith link of arm in $\mathcal{N}$
P	A 3D point in $\mathcal{B}$, except Sec. 2.2.4: Potential energy of UAM
ϕ	UAV roll angle or angular displacement around b_1
ψ	UAV yaw angle or angular displacement around b_3
q	Vector of generalized coordinates of UAM
R^k_j	Rotation matrix between arbitrary frames j and k
R^n_b, R	Rotation matrix from $\mathcal{B}$ to $\mathcal{N}$
ρ	Perpendicular distance from line to origin in a 2D image
ρ_m	Mean of line parameter ρ for multiple lines
s_h	Height moment feature in $\mathcal{V}$
s_l	Lateral moment feature in $\mathcal{V}$
s_ψ	Yaw moment feature $\mathcal{V}$
s_ξ	Sine of ξ, i.e., $\sin\xi$
θ	UAV pitch angle or angular displacement around b_2
τ_q	UAV torque input vector
$\tau_{\text{ext},\eta}$	External torque disturbance on a UAV
$\tau_{\text{ext},p}$	External force disturbance on a UAV
S(.)	Skew matrix
SO(3)	Special Orthogonal Group containing 3D rotation matrices
$W(\eta)$	Interaction matrix between angular velocity ω and angular rates $\dot{\eta}$
u	UAV thrust input
$v,\ v^n$	UAV translational velocity in $\mathcal{N}$
v^b	UAV translational velocity in $\mathcal{B}$
v^v	UAV translational velocity in $\mathcal{V}$
$\mathcal{V}$	Virtual Camera frame
$\xi_d,\ \xi^*$	Desired or reference value or trajectory of an arbitrary quantity ξ
$\hat{\xi}$	Estimate of an arbitrary quantity ξ
$\frac{\partial}{\partial z}(.)$	Gradient if argument is scalar and Jacobian if vector

List of Acronyms

ABSC	Adaptive Backstepping Control of Chapter 5
AKBSC	Backstepping Control with all parameters and disturbances known
ANCL	Applied Nonlinear Control Lab
AHRS	Attitude and Heading Reference System
BIBO	Bounded Input Bounded Output
BIBS	Bounded Input Bounded State
CAD	Computer-Aided Design
COM	Center of Mass
CVS	Computer Vision System
DOBSC	Disturbance Observer based Backstepping Control of [1]
DOF	Degrees of Freedom
DT	Desired Trajectory
ESC	Electronic Speed Controller
FoV	Field of View
GAS	Globally Asymptotically Stable
GC	Geometric Center
GES	Globally Exponentially Stable
GPS	Global Positioning Systems
GUI	Graphical User Interface
IBVS	Image-based Visual Servoing
IMU	Inertial Measurement Units
IOABSC	Inner-outer loop Adaptive Backstepping Control of Chapter 4
EKF	Extended Kalman Filter
FMU	Flight Management Unit
LAS	Locally Asymptotically Stable
LES	Local Exponentially Stable
LiDAR	Light Detection and Ranging
LQR	Linear Quadratic Regulator
MCS	Motion Capture System
OS	Operating System

OpenCV	Open Source Computer Vision Library
PBVS	Position-based Visual Servoing
PD	Proportional-Derivative
PE	Persistence of Excitation
PID	Proportional-Integral-Derivative
PWM	Pulse Width Modulation
QGC	The ground station software Q-Ground Control
RPM	Revolutions Per Minute
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
UART	Universal Asynchronous Receiver/Transmitter
UAM	Unmanned Aerial Manipulator
UAM-A	Arm subsystem in an Unmanned Aerial Manipulator
UAM-V	UAV subsystem in an Unmanned Aerial Manipulator
UAV	Unmanned Aerial Vehicle
UDBSC	Unknown Disturbance Backstepping Control
VTOL	Vertical Take-off and Landing

Chapter 1

Introduction

1.1 Background

Multirotor Unmanned Aerial Vehicles (UAV) have several advantages over fixed-wing UAVs. They have the benefits of vertical take-off and landing (VTOL) capability: no need for a runway, hover, and low-speed flight without the risk of stalling. Their high thrust-to-weight ratio makes them highly manoeuvrable, and simple electro-mechanical design makes them robust to failure and inexpensive. Due to these characteristics, the UAVs are commonly employed in applications like infrastructure inspection, rescue, surveillance, and load transportation [2, 3]. The work presented in this thesis is geared towards infrastructure inspection and maintenance. In this regard, we deal with the motion control aspects of UAVs to improve their autonomy and capability.

Among essential infrastructure requiring inspection or maintenance are Power Transmission lines and Oil and Gas pipelines. These linear structures spread over thousands of kilometres, require periodic inspections to determine maintenance needs and ensure continuity of supply. Inspection and maintenance of these structures by humans from the ground or the air with manned helicopters is hazardous and expensive. Therefore, UAVs have been employed recently to perform these tasks [4–7]. A Camera-equipped UAV is flown along the length of the linear structure to obtain an inspection video, which can be processed online or offline for abnormalities. The accuracy of UAV pose relative to the infrastructure determines the quality of the inspection data and the safety of the mission. Here *pose* refers to the position and heading or yaw angle of the UAV relative to the infrastructure.

Motion control of UAVs requires accurate knowledge of the full UAV state, including position, linear velocity, attitude, and angular velocity. The UAV position and linear velocity are usually estimated using Global Positioning System (GPS) [8] along with other onboard sensors such as ultrasonic rangefinders and barometers. In contrast, UAV attitude and angular velocity are usually estimated by Attitude

(a) A Visual Servoing UAV (b) A Flying hand [16] (c) A UAM [17]

Figure 1.1: Examples of UAVs.

and Heading Reference System (AHRS) using only the onboard Inertial Measurement Unit (IMU) measurements. The position estimate from an average low-cost GPS module has an accuracy of a few meters, which is not suitable for inspection. Moreover, the target GPS coordinates of the linear structures may not be accurately known. Also, a GPS position estimate, depending on an *external* signal from satellites, limits the autonomy of a UAV since a GPS signal may be weak or unavailable due to obstructions [9], such as near tall mountains or inside tunnels. Extending the UAV's ability to inspect GPS-denied environments is a difficult and practically relevant challenge.

A camera image is a rich source of information about the environment. Modern computer vision has developed high-performance algorithms for extracting features such as lines or points from images. Hence, cameras are a good substitute for GPS for UAVs. Other sensors can provide similar information to cameras but have increased cost or weight, e.g., Light Detection and Ranging (LiDAR). A camera is compact, light-weight, low-cost and often already found on a UAV for collecting data or providing a first-person view [10]. Indeed, there exists significant research on using computer vision for visual simultaneous localization and mapping (SLAM) or visual odometry. The UAV pose estimate can be used for motion control [11, 12]. The use of visual information from a camera for the motion control of a mechanical system is called *visual servoing* [13]. Visual servoing based kinematic estimation for general robotic systems is studied in [14]. A recent survey on the use of vision for sensing, flight control, navigation and guidance of UAVs is in [15].

We propose an Image-Based Visual Serving (IBVS) approach for motion control of a camera-equipped quadrotor UAV that can be used for inspecting linear structures. Since the linear structures are detected as lines in a camera image, we use line features and their dynamics coupled with the UAV dynamics to design control inputs to the UAV. Line features in the case of IBVS are mathematical quantities based on the parameters of a line measured from the projection of the target in the camera image. The features provide meaningful information about the relative position of the target to the camera. Controlling these features allows us to control the motion of the UAV directly with respect to the linear target due to the dependence of line features on linear and angular velocities. Our design incorporates output

feedback to estimate linear velocity. The effect of unknown external disturbances such as wind and variations in UAV and camera parameters, including mass, thrust constant, target distance and camera focal length, are also addressed. The proposed method is robust to uncertainty in system parameters and adapts to the unknown external disturbances.

Usually, when damage is found through an inspection, corrective maintenance action is required. Providing the UAV with the capability to interact with the environment would enable it to repair the infrastructure or perform an action to mitigate damage. There are many useful scenarios where this capability to intervene could be used. For example, if through inspection, it is determined that a system needs to be shut down, a UAV can press a button or pull a lever while keeping humans at a safe distance. Another example of environment-interaction is load transportation. UAVs capable of interacting with the environment are known as Unmanned Aerial Manipulators (UAMs). Several UAMs designs exist, with the most common configuration being a multi-rotor UAV with some form of a robotic arm. [18–21]

As in the case of the inspection example, traditionally, UAVs can only observe or 'see' the environment. Their contact with external objects is avoided as it often leads to a crash or damage due to the high-speed rotors extending at their periphery. However, a UAM, thanks to its robot arm, can "touch" and physically interact with the objects in its environment. Also, its arm can extend to reach confined spaces where a UAV may not fit or fly due to its larger body and rotors.

The importance of aerial manipulation can be seen in several government-funded projects around the world. The remote inspection of industrial structures by contact is studied in the European project AIRobots (Innovative aerial service robots for remote inspections by contact) [22]. More detail on the project is available in [23–25]. The project's focus was to develop aerial robots that can interact with the environment in a safe manner. Therefore, ducted-fan UAVs were developed, which allow the end-effector to be mounted on either side of the propeller. This configuration ensured the UAV interacted with its environment safely. The ARCAS (Aerial Robotics Cooperative Assembly System) project [26] dealt with cooperative manipulation of aerial robots for construction and assembly applications. Another project that specifically deals with the inspection and maintenance application of UAMs is AEROARMS [27, 28]. It mainly focuses on the outdoor operation of UAMs for taking measurements, sensor installation and robotic crawler (a robot that moves within pipes) deployment in the oil and gas industry. Further, the infrastructure inspection and maintenance using multiple cooperative UAMs is studied in AEROWORKS [27]. These large-scale, international projects demonstrate governments, academia and industry's serious interest in aerial manipulation.

The complexity and capability of a UAM are determined by its arm configuration, which involves its degrees of freedom (DOF), types of joints, link parameters and constraints. Also, the ability to perform specific tasks is determined by the last link of the robotic arm, known as an end-effector. An end-effector is the part of UAM that comes in contact with the environment and could be a gripper, a particular tool or a sensor. UAMs can have a fixed end-effector directly attached to the UAV without any joints or links. However, it is the addition of a multi-DOF arm to a UAV that significantly adds to the capability of UAM, allowing it to interact with the environment. It also makes areas accessible that were otherwise inaccessible, e.g., the arm can extend to reach confined spaces.

The most common tasks requiring physical interaction involve payload pick up, transport and placement, e.g., in construction, and application of a force/torque, e.g., to push an object. In these applications, a UAM is required to approach its operational area, extend its robot arm to align its end-effector for the task at hand, perform an interaction while maintaining a stable flight and safely disengage to return. This process involves precise motion control of the UAM in order to perform the task successfully. However, the motion control of a UAM is a challenging problem.

A UAV is already a highly nonlinear and underactuated system with bounded inputs, which may also be affected by external disturbances and measurement noise [29–32]. The addition of a robot arm significantly complicates its dynamics and makes it difficult to control. The coupling between UAV and arm dynamics affects UAV control. Our approach focuses on the motion control of the UAV subsystem, considering the model parameters and states of the arm as unknown. We model the effect of the arm as a disturbance and parameter uncertainty that appears in a traditional UAV model. Adaptive backstepping is then used to compensate for the model uncertainty.

Throughout this document, we refer to the UAV subsystem in a UAM as UAM-vehicle (UAM-V) and the arm subsystem as UAM-Arm (UAM-A). For greater clarity, UAM-V refers to a UAV affected by the coupling forces and toques due to arm. Also, UAM-V mass includes the arm's mass and any object held by the arm. The UAM-V inertia includes the imbalances due to arm and usually a non-diagonal matrix.

1.2 Literature Survey

1.2.1 Line following: Image-Based Visual Servoing

As discussed in the previous section, UAVs are well-suited for efficient inspection of electric transmission lines and pipelines. Inspection of such infrastructure is an

important and challenging problem and key to ensuring a high standard of service. For example, regular detailed inspection reduces the potentially disastrous effects of line failures. Since lines are spread over vast areas and inhospitable terrains, manned vehicle inspections are not efficient. Hence, UAVs are an emerging solution, e.g., [33–37].

Existing work has considered various aspects of using UAVs equipped with a video camera for inspecting linear structures. Some of the work focuses on the automatic extraction of useful information from the video. For example, the authors of [38] present a computer vision technique for estimating the 3D position of power transmission lines from video obtained using a quadrotor UAV. The method monitors transmission line sag to ensure safe ground clearance. In [4] UAV video is used to extract the position of power lines and map vegetation along the transmission line corridor. In [39] a multi-UAV system is proposed to improve the efficiency of collecting transmission line inspection data. In [40] various remote sensing techniques and data sources for monitoring transmission lines are compared, and the authors eventually recommend a UAV-based approach. The problems of pipeline and transmission line inspection are similar. Work on pipeline inspection using UAVs includes [5, 34, 41].

The main requirement for any UAV-based infrastructure inspection is to automatically control the relative pose of the onboard sensor to what is being inspected. The automatic control of the relative pose ensures that consistent and accurate measurements are obtained to perform the inspection while allowing the UAV operator to focus on higher-level objectives. The camera image of a 3D linear structure, e.g., pipeline, power transmission line, duct, corridor or road etc., has two or more lines at the edges of the structure. Therefore, we can assume that our target consists of two or more lines constrained to a horizontal plane. This assumption makes it possible to control the relative pose (position and yaw angle) of the UAV to the target using feedback from an onboard monocular computer vision system.

Visual servoing methods are typically categorized into two main techniques: Position-Based Visual Servoing (PBVS) and Image-Based Visual Servoing (IBVS) [13, 42]. PBVS is a more traditional approach that involves estimating or reconstructing the robot's 3D pose using the appearance and pixel dimensions of the target object in the image, together with the known Euclidean dimensions and a 3D model of the object. The reconstructed pose is then used in a motion control algorithm [43, 44]. Reconstruction depends on an accurate 3D model of the target and camera calibration parameters. On the other hand, IBVS associates certain features to the target in the image plane called image features. Image features provide a measure of the error in UAV pose relative to the target [45]. These features are directly controlled in the image plane and driven to the desired value, which in

turn controls the relative pose between UAV and target. This approach is known to be insensitive to camera calibration error and does not require a 3D target model [46, 47]. An IBVS approach with an uncalibrated camera for fixed-base robots is studied in [48, 49]. A comparison of IBVS and PBVS approaches is given in [50].

The IBVS approach has its challenges and limitations. The interaction matrix that relates the translational and angular velocity of the camera to points' motion in the image depends upon the distance of the target from the camera along the principal axis, known as depth. Since the depth information cannot be obtained directly from the image measurements, it must be estimated or approximated [13]. Also, the interaction matrix depends upon the intrinsic camera parameters e.g., focal length. The classical IBVS that employs a set of points as features; requires a minimum of three points for 6 degrees of freedom (DOF) control. Selecting minimum points may lead to a singular Interaction matrix in some configurations [51] e.g., three collinear points. Also, there exist four global minima for feature error i.e., four distinct camera poses could lead to feature error being zero [52]. Therefore, more than three points are usually needed.

Further, the estimate of the interaction matrix affects the transient performance of the feature error e.g., for a pose with pure rotation around the optical axis, the straight line convergence of feature points to the desired location involves rotation and a retreating transnational motion the optical axis. The IBVS also requires an image of the target at the desired pose. Furthermore, the classical IBVS suffers from a local minima problem i.e., the feature errors converge to a non-zero value that is locally minimum but could be far from the desired configuration. Due to this reason, the traditional IBVS can only provide local asymptotic stability as larger displacement could reach a local minimum or may cross a singularity in the interaction matrix. Advanced IBVS approaches use rich and intuitive geometric primitives instead of image point coordinates as features. This resolves some of the issues described above. An example of such primitive is image moments [53, 54].

The IBVS approach also has some control design challenges associated with it. Due to the perspective projection of the camera, the resulting image kinematics are nonlinear and destroy a triangular structure in the dynamics due to angular velocity dependence. Various methods have been employed in the literature to deal with the perspective projection effects. The work which focuses on the closed-loop stability of IBVS can be divided into four major categories based on the selection and processing of image features used in the feedback law.

The first IBVS method projects image features onto a virtual spherical image plane [55]. The spherical projection removes angular velocity dependence from the image feature kinematics. The resulting kinematics have a triangular structure that allows the backstepping design. This approach is further studied in [56–59]. Work

[60] points out the difficulty in defining a feature for yaw control using spherical projection. A second IBVS technique employs the homography matrix to define feature error between current and desired target view. A homography matrix between two images provides the relative orientation and translation of the camera [61]. The homography matrix can be used to control the pose of the UAV. Homography based techniques are studied in [62–65]. This method assumes a planar target and a small error range, particularly for yaw.

The third IBVS method uses a virtual spring approach for points features [66, 67]. Here, the translational velocity component of image feature kinematics takes a simple form. The work assumes that the desired height is known, and the image plane is parallel to the planar target. This assumption is restrictive for traditional multi-rotor UAVs, as horizontal motion requires a non-zero roll or pitch. The fourth IBVS approach is the virtual camera method, which uses attitude estimates from an IMU to define a virtual image plane that remains parallel to the planar horizontal target [45, 60, 68–74].

Image-based visual servoing mainly uses two feature definitions, depending upon the application. These types are point features and line features. Point features are based on a certain number of detectable points in the target and are usually suitable for controlling position relative to a discrete target. On the other hand, line features depend on detecting lines in the image plane and are suitable for linear targets that extend to larger distances in the longitudinal sense. For example, line features are used in [75] for autonomous landing of a fixed-wing UAV on a runway or landing strip.

Most of the work mentioned above, discussed in the contexts of IBVS, focuses on the point feature case, and less work has considered the line feature case. In [76], line following is considered using Euclidean Plücker coordinates. In [77], the work in [76] is extended using bi-normalized Euclidean Plücker coordinates while considering uncertainty in the depth of the measured image feature. It employs the point feature IBVS result in [55] and inherits its lack of sensitivity to UAV height. Other aspects of IBVS have attracted the attention of researchers, such as improved open-loop reference trajectory generation [78]. In our proposed method, the focus is on the design of feedback control and a constant reference feature is used.

Since the UAV flights for the inspection of linear structures are mostly outdoors, the UAV is often affected by wind. Strong wind gusts can displace the UAV away from the target; therefore, it is crucial to consider the effects of wind disturbance on system dynamics. When modelling an inspection UAV, it is also important to consider the effects of mass variations. The UAV mass could change due to adding a payload of unknown mass, e.g., an instrument or a sensor for inspection. Moreover, the UAV thrust is proportional to the square of a motor speed, which is proportional

to the square of to the normalized PWM signals provided at the input of Electronic Speed controllers (ESCs) driving the motors. The square of the ratio of motor speed to the normalized PWM signal defines a constant of proportionality called the thrust constant. Contrary to what its name suggests, the thrust constant is a slowly varying quantity and depends upon the battery voltage. During the flight, as the battery discharges, the motor speed and hence the thrust produced by each motor slowly reduces for the same input PWM. The thrust constant variation and its effects are discussed in [45, 79]. It is important to consider the thrust constant variation to achieve a stable flight. Although an IBVS approach is relatively insensitive to camera calibration parameters, the camera focal length shows up in the interaction matrix and could affect control performance. Determining camera focal length is not complicated; however, modern cameras come with auto-focus and magnification options, and a user may unknowingly alter focal length. Therefore, it is important to consider variations in camera focal length. Most of the work mentioned above does not consider these important parameter variations.

Among the literature on IBVS line following, the article [60] proposes a virtual camera approach with a simple PID controller. A backstepping virtual camera-based IBVS for line features was presented in [72]. However, these two articles assume the UAV's linear velocity measurements are available, which in turn depends on GPS or a motion capture system external to the UAV for velocity estimation. These articles also assume that UAV mass and thrust constant are known constant quantities. Further, they do not include disturbance forces arising due to wind. We propose an output feedback approach to avoid measuring linear velocity or using an optical flow sensor. A few output feedback IBVS methods for point features have been proposed in the past [80], [74]. However, the linear feature case has not been considered.

Some work on the use of machine learning-based IBVS found in the literature is described below. In [81] an IBVS based on the Gaussian mixture model is developed to improve the speed of image processing. In [82], a reinforcement learning-based IBVS approach is employed that uses Q-learning and SARSA algorithms coupled with neural networks to adapt to the change in conditions in the environment. The method provides robustness to uncertainties due to the field of view constraints, camera calibration, system model, and image noise. In [83] an IBVS method using an Extreme Learning Machine combined with Q-learning is developed to provide robustness against uncertainties in the interaction matrix, camera calibration and image noise. The approach provides self-tuning of gains to improve convergence. Work in [84] proposes an IBVS method using gradient descent and Levenberg–Marquardt algorithms along with neural networks. The approaches mentioned above are applied to fixed-base robot arms. Little work has been performed on the visual servoing of UAVs using machine learning, with a few exceptions. In [85] employs deep neural

networks using stereo triangulation to detect wind turbines and calculate relative distance. Images are obtained from a stereo camera onboard the UAV. The proposed framework is intended to be used with visual servoing. Work in [86] proposes a visual servoing method based on deep learning for detecting and following of a small UAV.

1.2.2 Unmanned Aerial Manipulators

A UAM is a special kind of UAV capable of interacting with its environment. The interaction involves several different forms. It could involve close observation, such as taking a sensor measurement that requires contact or extension of a sensor using some mechanism attached to the UAV (e.g., sophisticated high DOF manipulator arm or simple gripper) to a point otherwise unreachable. Pickup and placement of a tool or component or application of force or toque are all forms of interactions with the environment. This section discusses various UAM designs, applications, and control challenges from literature.

A simple configuration that enables UAV to interact with the environment is the attachment of a robotic tool rigidly or through a simple mechanism [87–90]. These types of UAMs are sometimes referred to as *flying hands* in literature. A UAM dedicated to driving a screw is considered in [91] where the UAV motion provides the force required to operate the tool. A UAM for cleaning high-rise buildings is considered in [87]. A UAM to apply force to push an object is considered in [92]. The quadrotor designs for a door opening UAV and surface scratching/aerial writing are presented in [93], and [94], respectively. Other examples of flying hands involve load transporting aerial vehicles divided into two categories: slung load through a non-rigid cable and using a fixed gripper. Fixed grippers on helicopters are studied in [95, 96]. The work in [18] and [97] considers different types of grippers on a quadrotor UAV for load transport.

Flying hands apply a force through a rigid tool or surface or carry an object through a cable or a gripper whose position cannot be controlled independently of the UAV due to the motion constraints resulting from a passive attachment of the end-effector. Also, the end-effector access may be limited by the dimensions of the UAV frame and propellers. This configuration could be useful in some applications when the payload must be precisely controlled [91] and the addition of a manipulator arm is not feasible. However, the manipulation capability of such a UAM is limited without additional DOF provided by movable links in an arm.

UAMs which integrate multi-DOF arms have increased capability for interaction with the environment. They provide a broad range of motion configurations for the task at hand and, in general, can be applied to perform manipulation tasks. Also, they can compensate for the motion constraints of a UAV itself. For example, an

object directly attached to a quadrotor UAV can not be positioned and oriented arbitrarily because a quadrotor cannot hover and have a non-zero attitude at the same time. However, if the object is held through a multi-DOF arm attached to the quadrotor, it could orient an object attached to its end-effector in any position and orientation. Given the UAV's underactuated dynamics, this pose control would not be possible without the additional DOFs provided by the arm.

Common design attributes of a UAM system involve the UAV platform type, the number of arms, their type (i.e.,, series or parallel), and their number of DOF. A quadrotor, although traditionally underactuated, is the most commonly used UAV platform. Most of the UAMs in literature are single series type robotic arms, with a few exceptions [98–101]. The number of UAM DOF is an important design specification. More DOFs improve the arm's dexterity and increase its range of motion and reach. However, each DOF increases the weight, power requirements, and complexity of the system. The types of joints and their axis of motion also contribute to the reachability of the arms. Most commonly used arms are mounted under the UAV, use revolute joints, and operate in a hemispherical region. The UAM design also has some technical challenges. The addition of an arm adds a significant mass and inertia to the UAV [16, 18]. The range of operation of UAV actuators is reduced since more thrust is required to compensate for UAV weight and less actuation is available to generate torques for fast maneuvers. Therefore, a UAM design requires a higher payload capacity UAV that requires higher ratings for actuators and batteries.

A UAM system with a multi-DOF robotic arm has complicated dynamics due to the UAV and arm interaction. It is necessary to account for the effects of both subsystems on each other when performing position control or manipulation. For example, the COM and the inertia matrix of the UAM vary with the arm motion. Picking up unknown payloads leads to unknown COM, mass, and inertia. When the UAM is in contact with the environment during an interaction, external forces act on it. For example, when a UAV applies a torque to drive a screw in a fixture attached to the navigation frame, the duration when the UAV is in contact with the fixtures, the dynamics of UAV are coupled with the fixture and the side in contact may shift its weight on the fixture, altering system parameters. Therefore, it is crucial to consider the variation of unknown system parameters during control design.

Three approaches are mainly used for modelling mechanical systems in the literature. These include the Euler-Lagrange formulation, Newton-Euler formulation and Kane's method. The Euler-Lagrange and Newton-Euler formulations are based on the generalized coordinates that describe bodies' configuration in a multi-body system. The Euler-Lagrange formulation treats all links or bodies in a multi-body

system by finding the Lagrangian, the difference between kinetic and potential energy. The derivatives of Lagrangian with respect to generalized coordinates and their velocities are calculated for use in the Euler-Lagrange equation. This leads to lengthy expressions in the process, starting with kinetic and potential energies to the final equations of motion. In the Newton-Euler formulation, the equations for linear and angular motion of each link are derived while considering coupling forces and torques from only the neighbouring components. The resulting equations are solved recursively to find the complete dynamics of the system. This method is less complicated at the start, but complexity increases as the equations are solved to find the system's dynamics in compact form. Kane's method is less used and based on generalized speeds instead of generalized coordinates. It is more procedural and non-intuitive. Although all three methods result in the same system dynamics, the Euler-Lagrange method is the most elegant as it can be applied even when bodies are not rigid. More detail on modelling approaches is included in Section 2.2.4. Work in [102] uses an Euler-Lagrange modelling approach from [103] to derive the system dynamics for UAM. A Newton-Euler formulation for the system modelling is employed in [104, 105].

Most of the exiting models in literature do not provide expressions for all terms in the equations of motion. Expressly, the matrix containing Coriolis and Centrifugal terms is not provided even in the most recent models presented in the literature [106–108]. Instead, an equation to obtain the entries of this matrix known as Christoffel symbols is provided. This equation is based on the partial derivatives of the scalar entries of the inertia matrix. The inertia matrix can be written as three-dimensional sub-matrices specifying the inertial contributions of individual bodies of the UAM. However, obtaining the Coriolis-Centrifugal matrix using Christoffel symbols result in scalar expressions for its entries, which lacks the intuitive understanding of the coupling between the arm and UAV. We revisit the UAM modelling using the Euler-Lagrange equation and in contrast to the existing literature, provide the complete model of the system with the Coriolis-Centrifugal matrix written in the form of sub-matrices. This provides an intuitive understanding of the coupling between the UAV and the arm dynamics.

Different control techniques to deal with various UAM challenges have been studied in the literature. The UAM motion control schemes can be categorized into two methods based on their structure. One of them is a decentralized approach, where the control is separately designed for the UAM-V and the UAM-A. The coupling forces or torques due to UAM-A can be treated as a disturbance input on the UAM-V [17, 104, 109–112]. Such a decoupled design for each subsystem is simpler given the reduced model complexity of each subsystem. Also, the designer can benefit from the rich literature on robust control of UAVs and arms. The other approach

involves considering the entire UAM as a single system. This approach is challenging theoretically due to the high-order nonlinear model where the order and complexity increases with the number of arm DOF [102, 103, 113–115]. Using a decentralized approach, the controllers for UAM-V and UAM-A can be made independent by using estimation techniques to capture the coupling effects. However, in a centralized approach, the dynamic model, parameters and states of both the UAM-V and the UAM-A are usually assumed to be known [116].

Among control schemes, a Lyapunov-based model reference adaptive control is presented in [117] to control a quadrotor with a multi DOF arm. This work is further extended in [118] with gain scheduling. A linearized system model is employed in [119] to design a linear quadratic regulator (LQR) control for a UAM. The forces during end-effector interaction for aerial manipulation are explored in [120], and an impedance control scheme is proposed using a three-loop structure. A Cartesian impedance control for a force control is presented in [103] and [102]. An inner-outer loop control structure with two loops is employed in [121] where the outer-loop obtains the reference trajectories for actuated variables through inverse kinematics. At the same time, the inner-loop ensures the desired motion references generated in the outer-loop are achieved. State and output feedback linearization for UAM control is studied in [104] and [122], respectively.

The decentralized approach is taken in [17, 109, 110]. A hexrotor with a 4 DOF freedom robotic arm is considered in [17] to study the effects of arm motion on UAV trajectory. A high gain observer is designed to estimate the uncertain disturbances acting on the UAV body due to arm movement. It employs an inner-outer loop control approach, where the inner-loop is a disturbance observer design based on [123]. It does not have an outer-loop design; instead, it presents a control augmentation scheme for attitude references. The augmentation scheme can be applied to any outer-loop control design which is locally asymptotically stable. The outer-loop controller used to test the augmentation scheme is taken from [124].

A bio-inspired UAM equipped with a single DOF arm is employed in [125] for high-speed aerial grasping by trajectory planning using the differential flatness property of the system. An end-effector trajectory control algorithm similar to backstepping is proposed in [114] for a UAM system with a 2-DOF robotic arm. The scheme employs a decoupled approach, where the kinematics of the UAV's COM and rotational dynamics together with the arm dynamics are treated as separate subsystems. A moving battery counterweight system that compensates for shift in COM due to the motion of the arm is in [126].

Depending upon its DOF, a UAM can be overactuated. The relative positioning of the end-effector can be achieved by the motion of both UAV and the arm; therefore, a motion selection algorithm is needed to control the relative positioning.

The paper [127] proposes a velocity weighting matrix as a function of feature error to avoid arm joint limits. A similar idea for assigning weights to the UAV and the arm velocities is used in [128]. However, the criterion for the selection of weight matrix is described heuristically. No analytical expressions for weights as a function of feature error are given.

Other UAM work studies the control of applied force without using a sensor. Human-robot interaction has been studied in [129] where an admittance controller defines the quadrotor trajectory using estimated force. A Kalman filter is used to obtain a force estimate. A hybrid pose and wrench control is studied in [89] where a quadrotor wrench is estimated using the measured pose and the inputs, avoiding the use of a wrench sensor. Visual servoing of a UAM is studied in [98, 127, 128, 130]. The stability of switching controllers to deal with the model discontinuity arising from environment interaction is discussed in [131]. Surveys on recent developments in aerial manipulation are given in [132–138].

As discussed earlier, motion control of a UAM using a decentralized approach leads to a simple design. It allows modifications in the arm configuration, e.g., using different end-effectors for different applications, without the need to remodel the complete UAM dynamics. This broad usage is possible because the arm is treated as generic uncertainty acting on the UAV subsystem. Due to its benefits, we use a decentralized approach in this thesis. We model the complete UAM system using the Euler-Lagrange model and extract the dynamics for the UAV subsystem. We show how the arm affects the UAV dynamics and approximately model its effect as a disturbance and parametric uncertainty. Using this model, we design a robust adaptive controller for output tracking.

UAV motion control normally uses two control structures. The first is an inner-outer loop control structure where the closed-loop is divided into translational and rotational subsystems [139, 140]. The second uses the entire UAV dynamics [141–143]. The inner-outer loop control structure is usually simpler and easier to implement. A user can either specify an attitude trajectory for low-level control or a position trajectory for high-level control. Also, the controller can be designed without dynamic extension [144]. However, this structure often lacks a closed-loop stability analysis, which is challenging. It is more straightforward to prove closed-loop stability for a design based on the entire UAV dynamics. However, it is harder to implement and usually requires dynamic state feedback. We consider both control structures in this thesis.

Since the UAV subsystem dynamics in the navigation frame can have a triangular structure, we use the backstepping control design method. Backstepping recursively solves the stability of the control error, cancelling indefinite terms in the derivative of a Lyapunov function with the input [145]. In the perspective of the closed-loop

UAV control design, backstepping is beneficial because it only requires one dynamic state extension, i.e., representing thrust as a state [146–148], in contrast to the state feedback linearization, which requires dynamic extension by two states, i.e., thrust and its derivative [149]. Also, backstepping approaches show promising results in experimental validation [150–152].

Since our control objective is time-varying trajectory tracking of the UAV subsystem while the parameters and states of the arm subsystem are unknown, we must consider the UAV subsystem parameters and the force and torque disturbances exerted by the arm on a UAV as unknowns. Therefore, we use an adaptive control that only assumes the UAV state is known while all other parameters and disturbances are unknown. In [153] an adaptive controller only for the outer-loop of UAM is considered where only the mass is considered unknown. In [153] a model reference adaptive control for a UAM is proposed. This approach only considers the uncertainty in mass and inertia, while the effect of coupling forces and torques due to the UAM-A on UAM-V is not considered. In [154] a Luenberger observer is used to estimate torque disturbance due to arm on the UAV. The estimate is used in control based on the inner-outer loop structure for position stabilization. In [155], the torque estimation using Leunberger observer has been compared to Kalman filter-based estimation. These articles assume the effects of arm motion are lumped in the torque coupling, and variation in inertia, mass and force coupling due to arm is not considered. An adaptive scheme considering all parameters and disturbances as unknowns is stated in [156]. However, the adaptive controller is only designed for the outer-loop, and simple proportional derivative (PD) control for attitude control is used. Similar to the above-mentioned approaches, we propose an inner-outer loop control design where uncertainty in mass and inertia is considered. Also, the effect of coupling forces and torques due to UAM-A is considered in the form of unknown parameters. Asymptotic convergence of the tracking error for outer and inner-loops is individually shown in the case of the inner-outer loop control approach, and all the parameter estimates are shown to be bounded.

Existing work considering the entire UAV dynamics for control is discussed below. An adaptive controller with thrust saturation is proposed in [32, 157] which considers only a force disturbance and no yaw control. Unknown force and torque disturbances are considered in [158]. An adaptive controller considering unknown mass, inertia and force disturbances is considered in the [142]. This approach assumes a diagonal inertia matrix, which is not suitable for a UAM since its arm creates an asymmetric mass distribution. Also, the effect of torque disturbance is not included in the model that is the main factor acting on the UAV due to the arm even when the arm and the UAV are stationary. A brief survey on the control of UAVs with disturbance compensation can be found in [159]. The design using

entire UAM-V dynamics in a single loop presented in this thesis is based on the backstepping approach used in [1] and [160]. These contributions only consider the force and torque disturbances, while all the parameters are assumed known. Similarly, in the case of the complete closed-loop control design, asymptotic convergence of the tracking errors is shown while all the parameter estimates are bounded.

1.3 Outline

This document is organized as follows:

Chapter 2 starts with the presentation of the hardware and software components of the quadrotor UAV platforms developed, upgraded and maintained by the author for experimental research. It briefly explains the specifications and purpose of each component and highlights the important interconnections between them. The chapter includes modelling for a quadrotor UAV, pinhole camera, line feature kinematics and image moments. The modelling of a UAM is presented using the Euler-Lagrange method. We reformulate this model as a simpler UAV dynamics where the disturbance and parametric uncertainty are added to approximately account for the arm subsystem.

Chapter 3 presents the image-based visual servoing of a quadrotor UAV for line-following applications. Based on the inner-outer loop control structure, an outer-loop controller for thrust input is presented along with a velocity observer and an estimator for unknown parameters and disturbance. An inner-loop control based on a PID design is provided. The exponential stability of the outer and both inner-outer loop is proven. The experimental setup is explained, and both simulation and experimental results are provided. Analysis of the results is provided that discusses the effectiveness of the proposed approach.

Chapter 4 presents an inner-outer loop control approach for motion control of UAM-V. It describes the simplifying assumptions used for the UAM model. A backstepping-based inner-outer loop control is described. Simulation results are presented for three scenarios: a UAM tracking a figure-8 trajectory with a fixed arm, a moving arm, and a UAM performing a pick and place operation.

Chapter 5 presents a closed-loop design approach using backstepping where the UAV thrust and torques inputs are directly designed to control the position trajectory of the UAV. The closed-loop asymptotic stability is proven. Simulation results for figure-8 trajectory tracking and pick and place application are provided. The results in the case of tracking with moving arms are compared with other designs.

The document concludes with chapter 6 where it summarizes the whole thesis and provides a conclusion. Also, the limitations of the work presented and future directions for research are outlined.

1.4 Contributions

The contributions made in this thesis are listed below:

- A dynamic IBVS method to control the relative pose of a quadrotor UAV to a target consisting of two or more parallel lines is proposed in [161]. The design is based on a single monocular camera that is downward facing. Line moment features defined in the virtual camera frame preserve passivity property. The primary contribution in contrast to existing literature [60, 72, 76, 77] is the estimation of lateral and vertical velocities of the UAV relative to the line target using the line features measured in the image plane. Further, the closed-loop system is shown to be exponentially stable in contrast to [72], which shows only asymptotic stability.

- In [162], a dynamic IBVS approach for lines target is presented similar to [161]. In addition to the velocity estimation using output feedback, the control design is robust against variation in UAV mass and thrust constant. Although the IBVS approach has built-in robustness to camera calibration parameters, the robustness is further strengthened by treating camera focal length in the interaction matrix as an unknown parameter in control design. Also, the effect of external disturbance such as wind is considered on the system. A robust adaptive controller that controls the image features to the desired reference and estimates velocity and disturbance is proposed. A stability proof for the outer-loop, i.e., feature kinematics and linear velocity dynamics, and the inner-loop dynamics, i.e., attitude and angular velocity dynamics, is given. The Lyapunov theorem-based stability analysis shows the exponential convergence of both inner and outer loops. Further, in contrast to [72, 76, 77], the approach is experimentally tested using a quadrotor platform to track a target using identifiable colour patches in a row, similar to lanes markings on the road.

- An inner-outer loop control design for the UAM-V is proposed. The effect of the arm is considered in the form of unknown parameters and disturbances. In contrast to existing literature [153, 155, 156], all parameters of the UAM-V are considered unknown. In addition, the effects of the arm on the UAV subsystem are considered in the form of the unknown force and torque disturbances acting on translational and angular velocity dynamics, respectively. Unlike many existing approaches, we do not assume the inertia matrix to be diagonal. Lyapunov-based proof for asymptotic stability of inner and outer loop dynamics is provided. The approach is tested for trajectory tracking in two cases: when the arm attached to the UAV is fixed and when it is moving. Moreover, the proposed approach is applied to a practical case of pick and place task.

- A design based on entire UAM-V dynamics that eliminates the drawbacks of inner-outer loop design is presented in Chapter 5. This approach directly designs the UAV control inputs, i.e., thrust and torque, to control position. A proof that shows the asymptotic stability of complete dynamics of the UAV subsystem is given. This approach also considers the unknown mass, non-diagonal unknown inertia, and forces and torques due to the arm acting on the system. This should be compared with the works in [1, 157, 158] where a similar control structure is used; however, all parameters and disturbances are not considered unknown. The approach developed avoids the need to compute numerical derivatives. The simulation results are provided and show the effectiveness of the proposed approach for tracking a figure-8 trajectory for a fixed and moving arm. Also, the control technique is tested for a pick and place operation. A comparison of the proposed approach with other methods is provided in the case of trajectory tracking with a moving arm.

- The quadrotor UAV platforms presented in Chapter 2 have been developed, upgraded and maintained by the author. These have been employed in the research by other group members in [151, 152].

Chapter 2

Modelling and Experimental Platforms

This chapter introduces the experimental platforms developed, upgraded and maintained by the author during this Ph.D. Also, the mathematical models of a quadrotor UAV, the line features, and the UAM are presented. The mathematical notations defined in this chapter are used throughout the thesis.

This chapter is organized as follows. Section 2.1 starts with the overview of experimental setup. The key software and hardware components of a Quadrotor UAV are outlined in Sections 2.1.1 and 2.1.2, respectively. Section 2.2 presents the kinematic and dynamic models. Dynamic model of a Quadrotor UAV is given in Section 2.2.1. Section 2.2.2 describes the model of a Pinhole camera, the camera projection of a line and the kinematics of a line in image plane. Section 2.2.3 defines line moment features and presents their kinematics. Section 2.2.4 derives the complete model of a UAM system. This chapter concludes by providing a summary and conclusion in Section 2.3.

2.1 Experimental Platforms

Experimental validation is an essential step of the applied research on UAVs. It evaluates the theoretical design and gauges the control performance and robustness to unmodeled dynamic effects of a physical system. The indoor setup at the Applied Nonlinear Control Laboratory (ANCL) consists of a motion capture system (MCS), three battery-powered Quadrotor platforms, and multiple computers that serve as ground stations during flight operations. The three UAV platforms namely ANCLQ 1.0, ANCLQ 2.0 and ANCLQ 3.0 are shown in Figures 2.1, 2.2 and 2.3, respectively.

Every Quadrotor UAV has a set of hardware components to perform various functions. Some of the hardware components have a microprocessor on them that

Figure 2.1: Quadrotor UAV ANCLQ 1.0

Figure 2.2: Quadrotor UAV ANCLQ 2.0

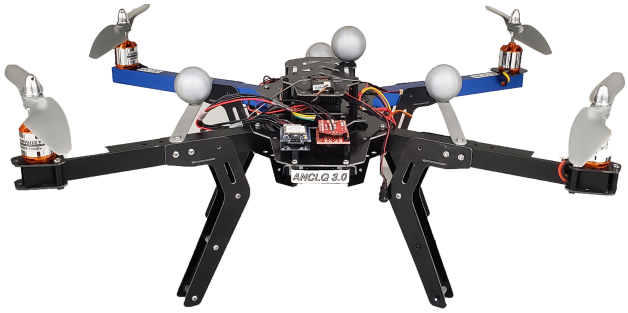


Figure 2.3: Quadrotor UAV ANCLQ 3.0

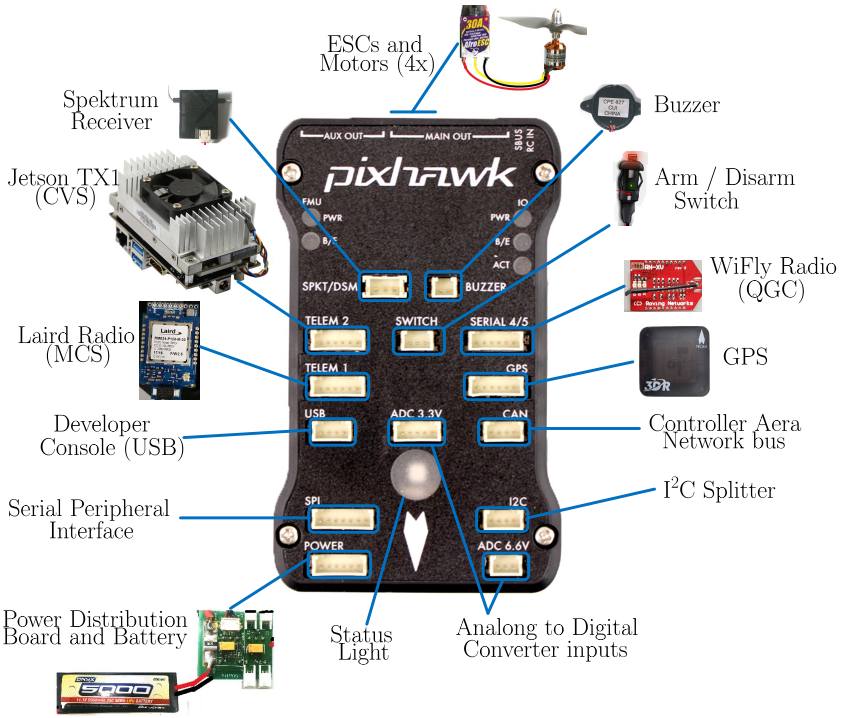


Figure 2.4: 3DR Pixhawk and its peripherals on ANCLQ 2.0.

runs a software. The essential hardware and software components of a Quadrotor UAV are described in the following sections.

2.1.1 Hardware Components

The common components of a Quadrotor UAV are shown in Figure 2.4 and their interconnection is shown in Figure 2.5. The arrangement of hardware components onboard a UAV is shown in Figure 2.6. The essential components of a quadrotor UAV and the specifications of those used in the ANCL platforms are described below.

Flight Controller

A flight controller is a mini-computer onboard a UAV with built-in sensors and communication ports. The onboard CPU on a flight controller runs the autopilot software that contains the estimation, control and safety-related decision-making algorithms to fly the UAV. Built-in sensors provide various measurements that are used by the estimation algorithms to determine system states. The communication ports allow connection with external sensors, actuators, ground station computers, and remote controllers. A list of flight controllers commonly used for small UAVs is given in [163] and segregated into open-source and closed-source in [164]. ANCLQ 2.0 and ANCLQ 3.0 use the 3DR Pixhawk, while ANCLQ 1.0 uses the HKPilot32

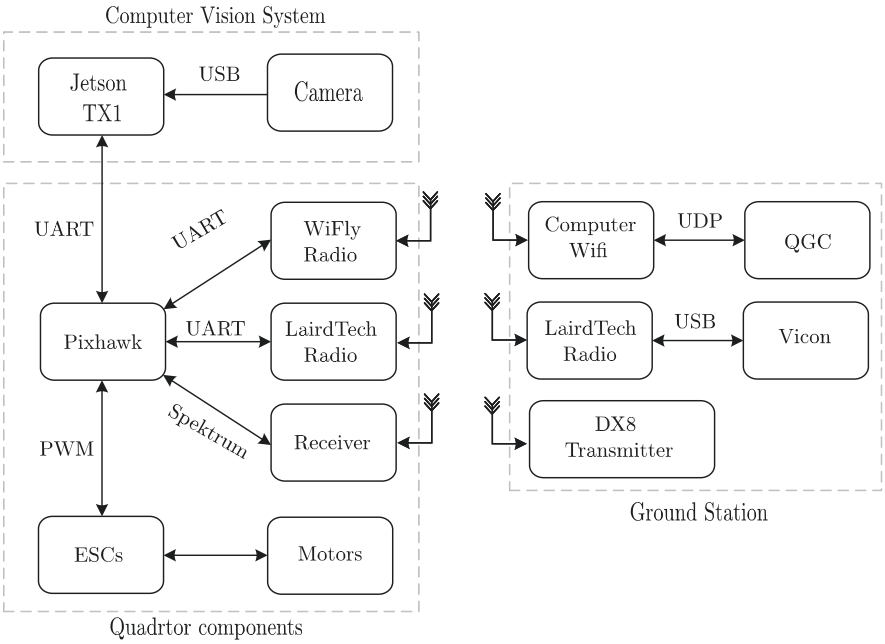


Figure 2.5: Hardware interconnections of ANCL quadrotor platforms.

flight controller. Both 3DR and HKpilot32 configurations are based on Pixhawk 1 Flight Management Unit version 2 (FMUv2) design [165]. Pixhawk 1 flight controller has 2 CPUs; one serves as the main CPU while the other is a backup to ensure safety. The main CPU is a 180 MHz ARM Cortex M4 CPU with 256 kB SRAM, while the fail-safe CPU is a 24 MHz ARM Cortex M3 with 8 kB SRAM. Pixhawk 1's Inertial Measurement Unit (IMU) consists of multiple built-in sensors that include ST Micro L3GD20H gyroscope, ST Micro LSM303D accelerometer/magnetometer, and Invensense MPU 6000 3-axis gyroscope/accelerometer. It has a set of two gyros and two accelerometers to provide redundancy to sensor failure. Complete specifications of the Pixhawk 1 flight controller are given in [166]. Raw measurement directly from IMU have the following issues:

- Gyroscopes measure the angular rates instead of angular displacements or attitude.
- Accelerometers measurements for linear acceleration have effects due to gravitational acceleration and Coriolis-centrifugal forces.
- IMU measurements have sensor bias and measurement noise due to vibration.

Therefore, these measurements are processed by an estimator to estimate vehicle states. IMU hardware and the estimator software are collectively referred to as Attitude and Heading Reference System (AHRS).

Motion Capture System (MCS)

The UAVs can obtain a position estimate from a GPS during outdoor flights; however, GPS is unavailable for indoor flight operations. Therefore, ANCL has a Motion Capture System in the lab that provides UAV pose estimates with an accuracy down to 1 mm in translational and 1 degree in the rotational. When testing control algorithms, this pose estimate is used either as a ground truth or a system state. ANCL motion capture system consists of 8 Vicon Bonita B3 Cameras with a resolution of 0.3 megapixels and a frame rate of 240 frames per second (fps). The cameras emit infrared light, which is reflected by the infrared markers in the room, and projected on cameras to track the location of markers. The system can detect passive markers of diameter as low as 9 mm, which can be uniquely arranged in a 3D pattern onboard a UAV for identification and tracking. For better reliability, we use 38 mm markers onboard the ANCL quadrotor platforms. The cameras are connected to the lab network using two switches which also provide power to the cameras i.e., Power over Ethernet.

Radios

Communication radios are essential components of a UAV platform. They allow a wireless connection between the UAV and its peripherals. On ANCL platforms, we have three primary radios. First is a pair of LairdTech RM024-P125-M-30 radios that allow communication between the MCS and the UAV. One radio is connected to a Ground computer running the Vicon tracker software. This radio sends data to its paired radio on the UAV side. The other radio on the UAV is connected to the flight controller through a telemetry port.

The second radio is a Roving Networks Wifly RN-171 802.11b radio, which essentially serves as a Wi-Fi Adapter for the flight controller computer, connecting it to the Wi-Fi Network of the lab. This allows us to wirelessly control the UAV and monitor the flight data online during a flight. This radio is only needed when the UAV does not have a computer vision system. When the computer vision system is connected to the UAV, its onboard companion computer has a built-in Wi-Fi that allows connection to the Pixhawk. Both LairdTech and Wifly radios need a base known as XBee Explorer that converts the 20 pin radio ports to a Universal Asynchronous Receiver Transmitter (UART) port.

The third radio is a Spektrum receiver and transmitter pair. The transmitter serves as the remote control for the quadrotor. Its channels are programmed in the autopilot firmware to operate the quadrotor in specific modes and fly manually. This is essential from a safety perspective and allows the user to intervene immediately if an accident is about to happen. In ANCL, we use Spektrum DX8 remote transmitter. More detail on its functions can be found in [79]. This remote control

has safety functions and therefore has the highest priory and uses the most sophisticated equipment. Also, for indoor flight in a confined space, manual attitude control is challenging. Therefore, a position estimate from the MCS is critical for indoor flights. Due to this, UAV has a dedicated radio for this communication. The Wi-Fi connection is at a low priority as it does not perform any safety-related functions.

Motors and Propellers

Electric-powered actuators have become favourable for UAVs due to their high power-to-weight ratio compared to gas-powered internal combustion engines. These electric actuators consist of a motor-propeller pair. In a quadrotor UAV, there are four motor-propeller pairs. The motor is a brushless DC outrunner with three inputs. The motor speed and torque depend upon the voltage levels and phase shift between the input pulses. The motor speeds are rated in revolutions per minute (RPM) per volt at no load, often referred to as Kv rating. The motors used in ANCL quadrotors are Turnigy D2836/8 with a Kv rating of 1100 RPM per volt. Their rated voltage is 7.4 - 14.8V, and their maximum current specification is 18 A. The propellers used in ANCL quadrotor UAVs have 11-inch diameters with a pitch of 4.5 inches. A propeller pitch is defined as the measure of forward distance moved through a soft solid due to one complete rotation of the propeller.

Electronic Speed Controllers

The Electronic Speed Controllers (ESCs) used in the quadrotor receive a Pulse Width Modulated (PWM) signal from the flight controller and produce DC pulses at its three outputs. The ESCs vary the magnitude and phase of these pulses with the variation in the PWM input to control the speed of brushless DC motors. The ESCs used in ANCL Quadrotor UAVs are Arfo 30 A ESCs based on Atmel AVR 8-bit Microcontroller ATmega8. These ESCs have a continuous current rating of 30 A and a voltage rating of 7.4 - 14.8 V.

Onboard Computer and Camera

The onboard computer, often called a companion computer, together with a camera, makes the Computer Vision System (CVS) hardware. The image processing is a computationally expensive task, and the flight controller has limited resources that are focused on the control and safety of the UAV. Therefore, a companion computer is often needed to implement image processing and computer vision algorithms. The computer onboard ANCLQ 2.0 is an NVIDIA Jetson TX1 which has a 256-core Maxwell GPU, Quad-Core ARM Cortex-A57 MPCore CPU and 4GB LP DDR4 64-bit Memory. Since the NVIDIA development board for Jetson TX1 has a large

size for use on a UAV, we use Auvidea J100 as its carrier board that has the same form factor and size as TX1 and provide several ports for connecting peripherals. This board is connected to Pixhawk through a USB to UART cable. The camera used on ANCLQ 2.0 is a Teledyne Flir Chameleon 3 USB 3.0 camera.

2.1.2 Software

Autopilot Firmware

The autopilot firmware runs on the flight controller and performs several functions. It has drivers for various sensors, collects data from sensors, using estimation algorithms estimates the states of a UAV, communicates with external sensors, accepts user commands and allows the switching of flight modes. The control algorithms are also implemented in the autopilot to generates inputs, which are converted to PWM signals for ESCs to actuate the UAV through motors and propellers. The auto pilot also performs a great deal of other safety and control related functions. The most commonly used open-source autopilot firmwares are PX4 [167] and ArduPilot [168].

At ANCL, we use a customized version of the v1.5.5 release of PX4 [169]. PX4 is based on a publish-subscribe structure and has several built-in applications for various UAV models. PX4's primary programming language is C++. It also provides many libraries and tools for developers to implement their algorithms. PX4 uses a MAVLINK protocol [170] for its communication with onboard components e.g., CVS as well as off-board components e.g., Ground station. The customizations of PX4 used at ANCL include incorporating the position estimate from the Vicon MCS and development of additional applications and topics to communicate with the onboard computer for the Computer Vision experiments. More detail on PX4 used at ANCL can be found [79].

As mentioned earlier, the raw IMU measurements are erroneous and contain measurement noise and sensor bias. An Attitude and Heading Reference System (AHRS) is usually used to estimate the vehicle states by fusing the measurements from multiple sensors. For example, a roll angle estimate can be calculated by integrating the roll rate measurement from the gyroscope, as well as from accelerometer measurements using $\hat{\phi}_A = \arctan(\frac{y}{\sqrt{A_x^2+A_z^2}})$, where A_x, A_y and A_z are the accelerometer measurements along corresponding axes. The roll from accelerometer measurements has high noise due to the calculation involving product and division of multiple noisy measurements. On the other hand, the roll from the gyroscope has bias or drift due to integration over time. The two measurements can be fused in a filter to better estimate the roll. The AHRS in PX4 performs Extended Kalman Filter (EKF) to fuse measurements from different sensors and obtain a better estimate for vehicle states. This EKF has 24 states consisting of vehicle attitude (4 quaternion states),

velocity (3 states), position (3 states), IMU and Magnetometer bias (9 states), Earth Magnetic field (3 states) and wind velocity (2 states). The processing delay in the EKF is 100 - 200 milliseconds which is a fairly large delay. Instead of directly using the delayed EKF estimates, they are used to correct the current IMU measurements in a Predictor algorithm, and the AHRS output available on `vehicle_attitude` or `vehicle_local_position` has a delay of less than 100 microseconds.

For the IBVS line following implementation, a module named `mc_ibvsline` was created on PX4, which subscribes to the `img_moments` topic containing the line moment features received from MAVROS and subsequently from CVS. This module runs the outer loop control, and its output is normalized thrust and an attitude reference that serves as a reference signal for the inner loop. Since the main contribution of the IBVS approach presented in Chapter 3 lies in the outer loop control design, the inner loop control from the stock controller implementation in `mc_att_control` application of v.1.5.5 is used.

PX4 computes the attitude using the `attitude_estimator_q` module, which is a quaternion-based attitude estimator and fuses raw IMU measurements. This module provides the estimates of attitude used in the visual servoing control. The IMU sensors are initially calibrated using the QGroundControl software. Despite the calibration, we remark that the attitude estimates include a small bias and measurement noise and can be considered accurate to about 0.5° near hover. Attitude estimates are provided at 93.9 Hz. During IBVS experiments, the Vicon motion capture system is used for ground truth UAV position. This allows us to evaluate the control performance. We also use the position from MCS for emergency control and safety in case the tested approach fails during the experiment.

Ground Station Software

Q-Ground Control (QGC) is a ground station software for MAVLINK enabled UAVs, which provides graphical user interface (GUI) based user-friendly option for complete setup and configuration of PX4 autopilot. It also provides in-flight support and mission planning. The Q-Ground Control displays flight maps and UAV trajectories. The QGC can be used to modify UAV parameters, switch flight modes, download flight data, monitor instruments online during flight, calibrate UAV sensors, configure control, and several other functions to support UAV flight. QGC software is a valuable tool; however, it is not necessary for the UAV flight. At ANCL, we use a customized QGroundControl v3.1.1, which has been modified to include additional topics to monitor various states of the controller during flight.

ESC Firmware

ESC firmware runs inside the ESC microcontroller and, depending upon the PWM input, generates an output signal to provide electronic commutation to the brushless DC motor. The most commonly used open-source firmwares for UAV ESCs are SimonK [171] and BLHeli [172]. ESCs on ANCL quadrotors are programmed with SimonK firmware Afro NFET Version 2013-09-20.

Computer Vision System

The software part of the computer vision system is based on Robot Operating System (ROS) running on a Ubuntu 16.04 operating system (OS) on Jetson TX1. ROS is a flexible open-source robotic framework and a meta-operating system that runs over an existing operating system. It is a set of tools, libraries, and conventions that simplify the development of software programs for robots. ROS can be used on a low level to implement sensor and actuator drivers of a robot and provide low-level device control. It can be used to implement robot mapping, localization, control, and estimation algorithms on a higher level.

ROS has a modular structure, where individual programs called nodes are written to perform specific tasks. Nodes share data between them through message passing. The message streams are called topics. Usually, a node subscribes to one or more topics, processes the data received from those topics, and publishes the result on another topic. For example, a robot control node in ROS would subscribe to a topic containing system states or sensor data and use these states to generate the control inputs and publish these control inputs on another topic. A driver node will subscribe to the topic published by the control node to drive UAV actuators. At ANCL, we have used Kinetic Kame release of the ROS.

In the case of computer vision processing, a camera image is the sensor data. Image from camera is processed in ROS nodes to detect specific targets e.g., lines. Based on the parameters of the target object certain meaningful state variables e.g., line features, are calculated. These states based in the visual data are then used in a control program to achieve visual servoing. In our case, we use functions Open Source Computer Vision Library (OpenCV) for image processing and target tracking. OpenCV provides efficient algorithms and functions for real-time image processing. Since we use the UAV attitude to remove the perspective effects from the image, and the UAV controller is implemented on PX4, two-way communication between ROS and PX4 is needed. This communication is established using MAVLINK protocol. The CVS captures the image, performs several image processing steps, computes the line feature in the virtual camera frame (defined later), and sends it to the PX4 at a rate of 21.7 Hz.

To simplify line detection, we use coloured patches positioned in a straight line

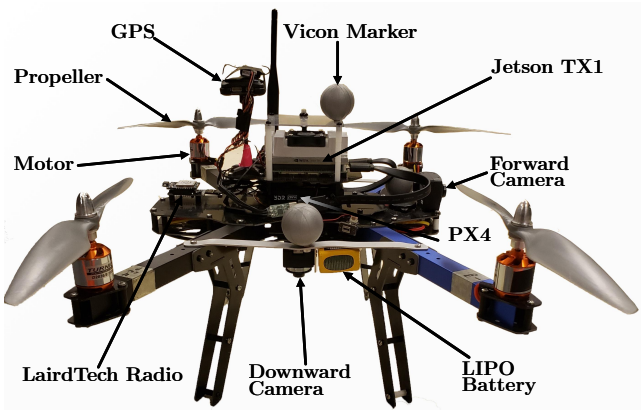


Figure 2.6: Components of ANCLQ 2.0.

to construct the lines for use in IBVS experiments. Patches of Green and Magenta colours are used to form two parallel lines. We detect the coloured patches as points located at their centroids and then fit a line through these points. This allows us to reuse the experimental test system for points target. However, it is discussed later in the experimental results of IBVS in Chapter 3 that this approach introduces additional noise and could be replaced by Hough transform for line detection for practical applications.

2.2 Modelling

In this section, the mathematical modelling of various platform configurations is given. We start by presenting the model of a quadrotor UAV. Standard notations are used throughout this text, with only a few exceptions. An alphabetic superscript above a variable represents the reference frame, except the superscript T, which specifies a transpose.

2.2.1 Quadrotor Model

We use nominal rigid body dynamics in 3D to model the quadrotor UAV shown in Figure 2.7. The model does not consider the effects of motor dynamics, aerodynamics of the rotors and the gyroscopic effects. The detail on modelling the effects of these factors can be found in [150, 173]. As shown in Figure 2.7, we consider a navigation frame $\mathcal{N}$ with basis vectors $\{n_1, n_2, n_3\}$ pointing north, east, down, respectively, and its origin is fixed to point on the earth. A body fixed frame $\mathcal{B}$ with its origin at the Geometric Center (GC) of UAV is also defined. Assuming a symmetric quadrotor, its Center of Mass (COM) is also located at its GC. For greater clarity, the origin of $\mathcal{B}$, COM of the UAV and the GC of UAV refer to the same point on a UAV. The basis vectors for $\mathcal{B}$ are $\{b_1, b_2, b_3\}$ that point forward, right,

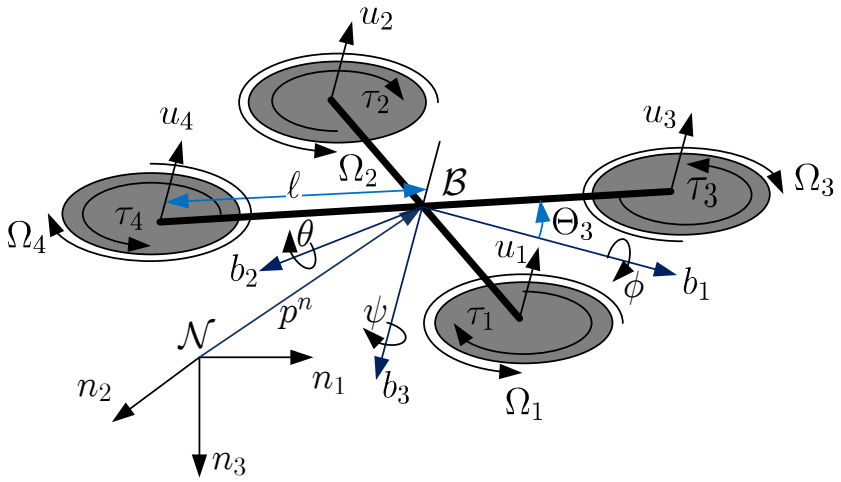


Figure 2.7: Diagram of a quadrotor showing frames, Euler angles and input forces and toques.

and downward, relative to the UAV. The UAV position or the position of the origin of $\mathcal{B}$, relative to the origin of $\mathcal{N}$ is denoted, p^n expressed in $\mathcal{N}$, which is a vector pointing from the origin of $\mathcal{N}$ to the origin of $\mathcal{B}$. The linear velocity of kinematics of the UAV in $\mathcal{N}$ expressed as $v^n = [v_1^n, v_2^n, v_3^n]^T$ is given by

$$\dot{p}^n = v^n. \tag{2.1}$$

The relative orientation of $\mathcal{B}$ and $\mathcal{N}$ is described by the rotation matrix $R_b^n = [b_1, b_2, b_3]^T[n_1, n_2, n_3] \in \mathrm{SO}(3)$. It is convenient to parameterize R_b^n with Tait-Bryan angles which is commonly referred to as Euler angles. These rotations are around three unique axes x, y, and z and are called roll ϕ, pitch θ, and yaw ψ in contrast to the Proper Euler angles which involve three rotations around two unique axis only e.g., zxz. The Euler angle parameterization of rotation matrix consists of three successive intrinsic rotations around axes of a mobile frame i.e., the body frame $\mathcal{B}$ in our case. The order of these intrinsic rotations is ZYX i.e., ψ around b_3, θ around b_2 and then ϕ around b_1, which is equivalent to three extrinsic rotations around fixed frame $\mathcal{N}$ in the reverse order of xyz i.e., ϕ around n_1, θ around n_2 and then ψ around n_3. Here capital ZYX and small xyz notations are used to specify the order in case of intrinsic and extrinsic rotations, respectively. We use the intrinsic angle

definitions of $\psi - \theta - \phi$ in $\mathcal{B}$ as shown in Figure 2.7 and write R_b^n as follows

$$R_b^n = R_{b_3}(\psi)R_{b_2}(\theta)R_{b_1}(\phi) = \begin{bmatrix} \mathrm{c}_\psi & -\mathrm{s}_\psi & 0 \\ \mathrm{s}_\psi & \mathrm{c}_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{c}_\theta & 0 & \mathrm{s}_\theta \\ 0 & 1 & 0 \\ -\mathrm{s}_\theta & 0 & \mathrm{c}_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}_\phi & -\mathrm{s}_\phi \\ 0 & \mathrm{s}_\phi & \mathrm{c}_\phi \end{bmatrix}$$
$$= \begin{bmatrix} \mathrm{c}_\theta\mathrm{c}_\psi & \mathrm{s}_\phi\mathrm{s}_\theta\mathrm{c}_\psi - \mathrm{c}_\phi\mathrm{s}_\psi & \mathrm{c}_\psi\mathrm{s}_\theta\mathrm{c}_\phi + \mathrm{s}_\psi\mathrm{s}_\phi \\ \mathrm{c}_\theta\mathrm{s}_\psi & \mathrm{s}_\psi\mathrm{s}_\theta\mathrm{s}_\phi + \mathrm{c}_\psi\mathrm{c}_\phi & \mathrm{c}_\phi\mathrm{s}_\theta\mathrm{s}_\psi - \mathrm{s}_\phi\mathrm{c}_\psi \\ -\mathrm{s}_\theta & \mathrm{c}_\theta\mathrm{s}_\phi & \mathrm{c}_\theta\mathrm{c}_\phi \end{bmatrix}, \tag{2.2}$$

whereas $\mathrm{c}_\xi = \cos\xi, \mathrm{s}_\xi = \sin\xi$. The UAV attitude i.e., the angular displacement of $\mathcal{B}$ with respect to $\mathcal{N}$ can be written in the vector form

$$\eta = [\phi, \theta, \psi]^T.$$

For greater clarity, the order of angles written in the definition of UAV attitude vector η does not conflict with the definition of $\psi{-}\theta{-}\phi$ as intrinsic angles because the resulting rotation matrix is the same if the angles are defined as extrinsic rotations with reverse order $\phi - \theta - \psi$. The angular velocity $\omega = [\omega_1, \omega_2, \omega_3]^T$ of the UAV with respect to the body-fixed frame $\mathcal{B}$ is related to the R_b^n by

$$\dot{R}_b^n = R_b^n \mathrm{S}(\omega), \tag{2.3}$$

where

$$\mathrm{S}(x) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}, \forall x \in \mathbb{R}^3,$$

is a skew-symmetric matrix. Using (2.3) and the parameterization of R_b^n in terms of η in (2.2), the UAV attitude kinematics can be written as

$$\omega = W\dot{\eta}, \tag{2.4}$$

where

$$W = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}.$$

The parameterization of 3D rotations using Euler angles results in a singularity at $\theta = (2k+1)\frac{\pi}{2}$ where k is an integer. This singularity is commonly referred to as *gimbal lock*. Gimbal lock is not a concern in non-acrobatic flights, where UAV roll and pitch angles are usually small. The complete dynamics of the system with linear velocity dynamics defined in $\mathcal{N}$ and angular velocity dynamics defined in $\mathcal{B}$ are given

by

$$m_q\dot{v} = m_q g e_3 - F + \tau_{\text{ext},p}, \tag{2.5a}$$

$$J_q\dot{\omega} = -\mathrm{S}(\omega)J_q\omega + \tau_q + \tau_{\text{ext},\eta}, \tag{2.5b}$$

where $F = uR_b^n e_3$, $e_3 = [0, 0, 1]^T$ is a unit vector, g is the gravitational acceleration, m_q is the mass of the UAV, $J_q \in \mathbb{R}^{3\times3}$ is inertia of the quadrotor UAV which is a positive definite symmetric matrix by its definition, u is the scalar input thrust, $\tau_q \in \mathbb{R}^3$ is the input torque acting on the UAV, $\tau_{\text{ext},p} \in \mathbb{R}^3$ represents an external force disturbance acting on the UAV e.g., due to wind, and $\tau_{\text{ext},\eta} \in \mathbb{R}^3$ represent external torque disturbance acting on the vehicle. It should be noted that the angular kinematics or dynamics are always defined in body frame $\mathcal{B}$ throughout this thesis. However, the linear velocity dynamics in the body frame $\mathcal{B}$ are useful when the relative position of the vehicle to a target whose coordinates are measured using an onboard sensor, is desired. Therefore, the transnational dynamics of the vehicle in $\mathcal{B}$ are given below.

$$\dot{p}^b = -\mathrm{S}(\omega)p^b - v^b, \tag{2.6a}$$

$$m_q\dot{v}^b = -m_q\mathrm{S}(\omega)v^b + m_q g R_n^b e_3 - F^b + R_n^b\tau_{\text{ext},p}, \tag{2.6b}$$

where p^b and v^b are the position and velocity of a rigid body in $\mathcal{B}$, $R_n^b = (R_b^n)^T$ and $F^b = ue_3$. These dynamics are later used in obtaining a dynamic model for IBVS.

The thrust input u and the torque input τ_q are the inputs that act on the UAV body at its COM and are not the actuator inputs. The quadrotor UAV has four motor-driven propellers as actuators, out of which two propellers $i = 1, 2$ rotate in a clockwise direction while the other two $i = 3, 4$ rotate in the counter-clockwise direction. The actuator inputs and inputs in the rigid body model are related to each other through the UAV's geometry. We consider a symmetric quadrotor UAV in a cross '$\times$' configuration, whose geometry can be defined by two parameters ℓ and Θ_i. Parameter ℓ is the distance of each propeller from the origin of $\mathcal{B}$, while Θ_i is the angle of ith arm with respect to b_1 measured in the counter-clockwise direction looking from the top of the UAV. We consider fixed propellers with the axis of rotation parallel to b_3 that generate thrust in the vertically upward direction of the body frame i.e., along $-b_3$. The thrust generated by the ith propeller is given by $u_i = k_u\Omega_i^2$, where k_u is the aerodynamic coefficient and Ω_i is the rotational speed of ith propeller. The total thrust force due to all propellers is defined as $u = \sum_{i=1}^{4} u_i$ in SI units N. The torque generated by individual propellers is $\tau_i = k_\tau\Omega_i^2$ for $i = 1, 2$ and $\tau_i = -k_\tau\Omega_i^2$ for $i = 3, 4$ which is also measured in SI units Nm. The total input torque acting on the UAV is denoted by τ_q, and can be calculated by projecting the generated thrust forces and torques of each propeller along the direction of b_1,

b_2 and b_3. For a quadrotor in '×' configuration, if we consider $\Theta_3 = \Theta$, we have $\Theta_1 = -\Theta$, $\Theta_2 = \pi - \Theta$, $\Theta_4 = \pi + \Theta$, the relation between the u and τ_q and the rotational speed of each propeller can be written as

$$\begin{bmatrix} u \\ \tau_q \end{bmatrix} = \begin{bmatrix} k_u & k_u & k_u & k_u \\ -k_u\ell\sin\Theta & k_u\ell\sin\Theta & k_u\ell\sin\Theta & -k_u\ell\sin\Theta \\ k_u\ell\cos\Theta & -k_u\ell\cos\Theta & k_u\ell\cos\Theta & -k_u\ell\cos\Theta \\ k_\tau & k_\tau & -k_\tau & -k_\tau \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}. \tag{2.7}$$

Since the inputs provided to the UAV by the controller is a set of PWM signals given to Electronic Speed Controller (ESC) which drive motor and propellers, denoted by $W_i, i = 1, 2, 3, 4$. We rewrite the above equation in terms of PWM signals as

$$\begin{bmatrix} u \\ \tau_q \end{bmatrix} = K_T \begin{bmatrix} f_T \\ \bar{\tau}_q \end{bmatrix} = K_T \begin{bmatrix} k_u & k_u & k_u & k_u \\ -k_u\ell\sin\Theta & k_u\ell\sin\Theta & k_u\ell\sin\Theta & -k_u\ell\sin\Theta \\ k_u\ell\cos\Theta & -k_u\ell\cos\Theta & k_u\ell\cos\Theta & -k_u\ell\cos\Theta \\ k_\tau & k_\tau & -k_\tau & -k_\tau \end{bmatrix} \begin{bmatrix} \tilde{W}_1^2 \\ \tilde{W}_2^2 \\ \tilde{W}_3^2 \\ \tilde{W}_4^2 \end{bmatrix}.$$

where $\tilde{W}_i$ is a normalized PWM signal defined by $\tilde{W}_i = (W_i - W_{\min})/(W_{\max} - W_{\min})$, and $W_{\min}$ and $W_{\max}$ are the minimum and maximum values of W_i, respectively. Therefore, $\tilde{W}_i \in [0, 1]$ and is related to rotor speed by $\Omega_i^2 = K_T\tilde{W}_i^2$, where k_T is a normalizing thrust coefficient that depends upon battery voltage, ESC configuration and motor parameters. ESC configuration and motor properties are relatively static over time, however, battery voltage varies during flight. As the battery weakens and its voltage drops, the motor receive less power and propeller speed is reduced for a constant PWM signal. Therefore, the thrust constant proportional to the battery voltage slowly varies over time and significantly affects vertical control performance during flight. This variation in thrust coefficient is accommodated in the IBVS control in Chapter 3. It should be noted that in (2.2.1), f_T and $\bar{\tau}_q$ are bounded in the interval $[0, 1]$. Since there is an algebraic relationship between PWM, and forces/torques acting on the UAV frame in $\mathcal{B}$ without any dynamic involvement, we take the later i.e., torque τ_q and thrust u as system inputs when designing control for UAM UAVs.

2.2.2 IBVS Modelling

For image-based visual servoing of linear structures, we consider a quadrotor UAV with a downward-facing camera that sees a planar horizontal target containing more than one line. We assume that the camera's optical center O_c i.e., the point where all light rays forming the image intersect coincides with the UAV COM. We also assume that the camera is perfectly aligned with the UAV body frame $\mathcal{B}$ so that the image

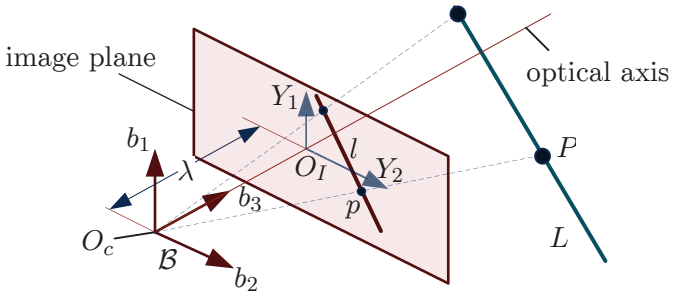


Figure 2.8: Pinhole Camera model showing projection of a 3D point P and 3D line L as a 2D point p and 2D line l in the image plane.

plane with basis $\{Y_1, Y_2\}$ is parallel to the plane defined by $\{b_1, b_2\}$. While principal axis, also known as the camera's optical axis, is the line or axis connecting the optical center O_c and image center O_I, coincides with b_3. This is shown in Figure 2.8 along with the camera model. For greater clarity, in the following discussion, the terms body frame, camera frame or real camera frame all refer to $\mathcal{B}$. In the following section, we describe the projection model of the camera.

Camera model

We use a pinhole camera model for modelling of the downward facing camera attached to UAV as shown in Figure 2.8. Consider a static 3D point P located inside the camera field of view with coordinates $P = [X_1, X_2, X_3]^T$ expressed in $\mathcal{B}$. The projection of P onto the image plane is denoted $p = [y_1, y_2]^T$ and given by

$$p = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \lambda \begin{bmatrix} \frac{X_1}{X_3} \\ \frac{X_2}{X_3} \end{bmatrix}, \tag{2.8}$$

where λ is the focal length of the camera whose value does not need to be known exactly.

A line in 3D is represented by the intersection of two planes with a plane defined by its normal vector $n = [a, b, c]^T$ and a point $P_0 = [X_{01}, X_{02}, X_{03}]^T$ on the plane. For any arbitrary point $P = [X_1, X_2, X_3]^T$ lying on the plane, the vector $\overrightarrow{P_0P}$ which points from P_0 to P must be orthogonal to n. That is, the dot product $\overrightarrow{P_0P} \cdot n$ must be zero:

$$\begin{aligned} \overrightarrow{P_0P} \cdot n &= a(X_1 - X_{10}) + b(X_2 - X_{20}) + c(X_3 - X_{30}) \\ &= n^T P + d = 0, \end{aligned}$$

where $d = -aX_{10} - bX_{20} - cX_{30}$. We consider a 3D line L represented by the intersection of two planes

$$n_i^T P + d_i = a_i X_1 + b_i X_2 + c_i X_3 + d_i = 0, \quad i = 1, 2, \tag{2.9}$$

with $n_1 \times n_2 \neq 0$, where $n_i = [a_i, b_i, c_i]^T$ is the normal vector of ith plane expressed in $\mathcal{B}$. The representation of a 3D line as the intersection of two planes does not

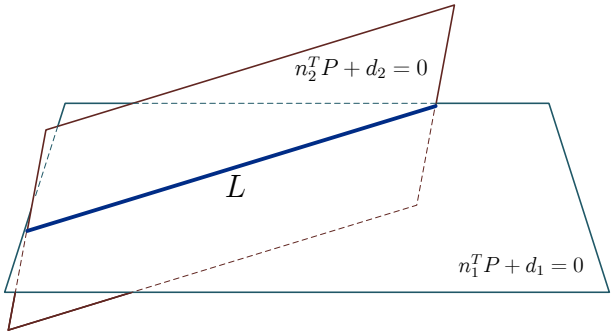


Figure 2.9: A line L represented as the intersection of two planes

define a unique pair of planes. A given 3D line can be represented by an infinite number of pairs of planes. Without loss of generality and as shown in Figure 2.9 we choose one of the planes parallel to the horizontal target in order to simplify the image kinematics below. We exclude the degenerate case $d_1 = d_2 = 0$, which means L does not pass through the focal point or origin of $\mathcal{B}$. For a downward-facing camera, this case is impractical as it corresponds to the 3D line passing through the camera or when the UAV has a 90° roll or pitch. As in [53] the projection of L for a camera of unit focal length with principal point or image center at $(0, 0)$ can be parametrized as

$$l_u^T p_h = Ay_1 + By_2 + C = 0, \tag{2.10}$$

with

$$A = \begin{vmatrix} a_1 & d_1 \\ a_2 & d_2 \end{vmatrix}, \quad B = \begin{vmatrix} b_1 & d_1 \\ b_2 & d_2 \end{vmatrix}, \quad C = \begin{vmatrix} b_1 & d_1 \\ b_2 & d_2 \end{vmatrix},$$

and $l_u = [A, B, C]^T = [n_1, n_2][-d_2, d_1]^T$ being the vector representation of the 2D line in homogeneous form and $p_h = [y_1, y_2, 1]^T$ is the representation of the 2D point in homogeneous form. For a camera of focal length λ and image center at (y_{10}, y_{20}) the projection of L is

$$l_\lambda = H l_u, \tag{2.11}$$

where $H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -y_{10} & -y_{20} & \lambda \end{bmatrix}$. We remark that the representation of the projection of a 3D line into the 2D image plane is non-minimal as any scalar multiple of l_λ or l_u represents the same line, which is evident from (2.10).

As shown in Figure 2.10, a 2D line in the image plane can be parameterized by two parameters α and ρ. The parameter α is the angle the line makes with the Y_1-axis, and ρ is the perpendicular distance between the line and the origin O_I of

the image frame. We have

$$l_\lambda^T p_h = y_1 \sin\alpha + y_2 \cos\alpha - \rho = 0, \tag{2.12}$$

where $l_\lambda = [\sin\alpha, \cos\alpha, -\rho]^T$ where $\alpha \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$. This equation is often referred as normal form of a 2D line. In what follows, we use vector $l = [\alpha, \rho]^T$ to specify a line. Our objective is to find the line kinematics so that we can couple it to the UAV linear velocity dynamics in (2.6b) and design a controller.

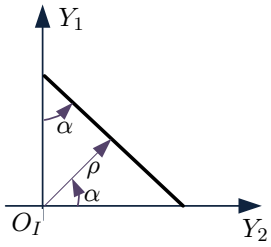


Figure 2.10: Projection of line parameterized by α and ρ

Line kinematics

To find the line kinematics in the image plane, we first need to find the kinematics of a point in image plane because a line is made up of points. We can obtain the kinematics of a point in 3D by substituting $p^b = P$ in (2.6a), we have

$$\begin{aligned}
\dot{X}_1 &= X_2\omega_3 - X_3\omega_2 - v_1^b, \\
\dot{X}_2 &= X_3\omega_1 - X_1\omega_3 - v_2^b, \\
\dot{X}_3 &= X_1\omega_2 - X_2\omega_1 - v_3^b.
\end{aligned}$$

From (2.8), we know $X_1 = \frac{y_1 X_3}{\lambda}, X_2 = \frac{y_2 X_3}{\lambda}$, substituting this in above equation, we have

$$\begin{aligned}
\dot{X}_1 &= \frac{y_1 X_3}{\lambda}\omega_3 - X_3\omega_2 - v_1^b, \\
\dot{X}_2 &= X_3\omega_1 - \frac{y_1 X_3}{\lambda}\omega_3 - v_2^b, \\
\dot{X}_3 &= \frac{y_1 X_3}{\lambda}\omega_2 - \frac{y_2 X_3}{\lambda}\omega_1 - v_3^b.
\end{aligned} \tag{2.13}$$

Taking the derivative of $X_1 = \frac{y_1 X_3}{\lambda}, X_2 = \frac{y_2 X_3}{\lambda}$, which is $\dot{X}_1 = \frac{\dot{y}_1 X_3 + y_1 \dot{X}_3}{\lambda}, \dot{X}_2 = \frac{\dot{y}_2 X_3 + y_2 \dot{X}_3}{\lambda}$, rearranging, we get

$$\begin{aligned}
\dot{y}_1 &= \frac{\lambda\dot{X}_1 - y_1\dot{X}_3}{X_3}, \\
\dot{y}_2 &= \frac{\lambda\dot{X}_2 - y_2\dot{X}_3}{X_3}.
\end{aligned} \tag{2.14}$$

Substituting (2.13) in above equation and simplifying, we have

$$\dot{p} = \begin{bmatrix} -\frac{\lambda}{X_3} & 0 & \frac{y_1}{X_3} \\ 0 & -\frac{\lambda}{X_3} & \frac{y_2}{X_3} \end{bmatrix} \begin{bmatrix} v_1^b \\ v_2^b \\ v_3^b \end{bmatrix} + \begin{bmatrix} \frac{y_1 y_2}{\lambda} & -(\lambda + \frac{y_1^2}{\lambda}) & y_2 \\ (\lambda + \frac{y_2^2}{\lambda}) & -\frac{y_1 y_2}{\lambda} & -y_1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \tag{2.15}$$

The above equation specifies the kinematics of a point p in the image plane in relation to the camera velocity. Since we consider a downward facing camera rigidly attached to the vehicle with image plane parallel to the body frame $b_1 - b_2$ axis, the velocities in above equation are the velocities of the vehicle. To find the line kinematics in the image plane, let us take the derivative of (2.12) on both sides, we have

$$\dot{y}_1 \sin\alpha + \dot{\alpha} y_1 \cos\alpha + \dot{y}_2 \cos\alpha - \dot{\alpha} y_2 \sin\alpha - \dot{\rho} = 0.$$

Rearranging, we get

$$\dot{\rho} + \dot{\alpha}(y_2 \sin\alpha - y_1 \cos\alpha) = \begin{bmatrix} \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix}.$$

From (2.12), we know $y_2 = \frac{\rho}{\cos\alpha} - y_1 \tan\alpha$. Substituting it into above equation and simplifying, we have

$$-\dot{\alpha} y_1 + \cos\alpha(\dot{\rho} + \dot{\alpha}\rho \tan\alpha) = \cos\alpha \begin{bmatrix} \sin\alpha & \cos\alpha \end{bmatrix} \dot{p}. \tag{2.16}$$

Now again consider the point kinematics in (2.15), substituting $y_2 = \frac{\rho}{\cos\alpha} - y_1 \tan\alpha$ from (2.12), and then left multiply by row vector $[\sin\alpha, \cos\alpha]$ to solve for $[\sin\alpha, \cos\alpha]\dot{p}$, we have

$$\begin{aligned} \begin{bmatrix} \sin\alpha & \cos\alpha \end{bmatrix} \dot{p} = \frac{1}{X_3} \begin{bmatrix} -\lambda \sin\alpha & -\lambda\cos\alpha & \rho \end{bmatrix} \begin{bmatrix} v_1^b \\ v_2^b \\ v_3^b \end{bmatrix} \\ + \begin{bmatrix} -\left(\frac{\rho}{\lambda}\tan\alpha\right) y_1 + \lambda\cos\alpha + \frac{\rho^2}{\lambda\cos\alpha} \\ -\left(\frac{\rho}{\lambda}\right) y_1 - \lambda\sin\alpha \\ -\left(\tan\alpha\sin\alpha + \cos\alpha\right) y_1 + \rho\tan\alpha \end{bmatrix}^T \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \end{aligned} \tag{2.17}$$

Recall the equation of a 3D line using two planes in (2.9), substituting $X_1 = y_1 \frac{X_3}{\lambda}, X_2 = (\frac{\rho}{\cos\alpha} - y_1 \tan\alpha)\frac{X_3}{\lambda}$ from (2.8) and (2.12), and solving for $\frac{1}{X_3}$, we have

$$\frac{1}{X_3} = -(f_\alpha y_1 + f_\rho), \tag{2.18}$$

where $f_\alpha = \frac{a_i - b_i \tan\alpha}{d_i \lambda}$, $f_\rho = \frac{\frac{b_i \rho}{\cos\alpha} + c_i \lambda}{d_i \lambda}$. Substituting it into (2.17) we have

$$\begin{bmatrix} \sin\alpha & \cos\alpha \end{bmatrix} \dot{p} = \begin{bmatrix} f_\alpha \lambda \sin\alpha \\ f_\alpha \lambda \cos\alpha \\ -f_\alpha \rho \\ -\frac{\rho}{\lambda}\tan\alpha \\ -\frac{\rho}{\lambda} \\ -\frac{1}{\cos\alpha} \end{bmatrix} \begin{bmatrix} v_1^b \\ v_2^b \\ v_3^b \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} y_1 + \begin{bmatrix} f_\rho \lambda \sin\alpha \\ f_\rho \lambda \cos\alpha \\ -f_\rho \rho \\ \lambda\cos\alpha + \frac{\rho^2}{\lambda\cos\alpha} \\ -\lambda \sin\alpha \\ \rho \tan\alpha \end{bmatrix} \begin{bmatrix} v_1^b \\ v_2^b \\ v_3^b \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \tag{2.19}$$

Substituting this into (2.16), and collecting coefficients of y_1 on both sides of the equation, we have

$$\begin{aligned} &-\dot{\alpha} y_1 + \cos\alpha(\dot{\rho} + \dot{\alpha}\rho\tan\alpha) \\ &= \cos\alpha \begin{bmatrix} f_\alpha \lambda \sin\alpha \\ f_\alpha \lambda \cos\alpha \\ -f_\alpha \rho \\ -\frac{\rho}{\lambda}\tan\alpha \\ -\frac{\rho}{\lambda} \\ -\frac{1}{\cos\alpha} \end{bmatrix} \begin{bmatrix} v_1^b \\ v_2^b \\ v_3^b \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} y_1 + \cos\alpha \begin{bmatrix} f_\rho \lambda \sin\alpha \\ f_\rho \lambda \cos\alpha \\ -f_\rho \rho \\ \lambda\cos\alpha + \frac{\rho^2}{\lambda\cos\alpha} \\ -\lambda \sin\alpha \\ \rho \tan\alpha \end{bmatrix} \begin{bmatrix} v_1^b \\ v_2^b \\ v_3^b \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \end{aligned} \tag{2.20}$$

Equating coefficients of y_1 and others on both sides of the above equation, solving the resulting system of equations for $\dot{\alpha}$ and $\dot{\rho}$, we can write

$$\begin{aligned} \dot{l} = \begin{bmatrix} \dot{\alpha} \\ \dot{\rho} \end{bmatrix} &= \begin{bmatrix} \sigma_\alpha \lambda \sin\alpha & \sigma_\alpha \lambda \cos\alpha & -\sigma_\alpha \rho \\ \sigma_\rho \lambda \sin\alpha & \sigma_\rho \lambda \cos\alpha & -\sigma_\rho \rho \end{bmatrix} v^b \\ &+ \begin{bmatrix} \frac{\rho}{\lambda}\sin\alpha & \frac{\rho}{\lambda}\cos\alpha & 1 \\ (\frac{\rho^2}{\lambda} + \lambda)\cos\alpha & -(\frac{\rho^2}{\lambda} + \lambda)\sin\alpha & 0 \end{bmatrix} \omega, \end{aligned} \tag{2.21}$$

where $l = [\alpha, \rho]^T$, $\sigma_\alpha = -f_\alpha \cos\alpha$, $\sigma_\rho = f_\alpha \rho \sin\alpha + f_\rho$, $f_\alpha = \frac{a_i - b_i \tan\alpha}{d_i \lambda}$, and $f_\rho = \frac{\frac{b_i \rho}{\cos\alpha} + c_i \lambda}{d_i \lambda}$. If we combine above equation with (2.6b), we have a dynamic model of the system where the force acting on the UAV directly affects the line parameters $l = [\alpha, \rho]^T$, this is referred to as Dynamic Image Based Visual Servoing (DIBVS). However, it is challenging and difficult to design a controller for this system due to the coupling with angular velocity ω in both (2.21) and (2.6b). The appearance of ω at this stage of dynamics destroys the triangular structure and passivity is not preserved. This problem arises due to the fact that when the UAV has a nonzero roll of pitch, the camera sees the target from a perspective, and projection of the target on image plane has perspective effects.

Various methods have been employed in the literature to deal with the perspective projection effects. In [55] spherical projection is used to remove the angular

velocity dependence from the image feature kinematics. However, [60] points out the difficulty in defining a feature for yaw control using spherical projection. In [62, 63] homography matrix is used to define feature error between current and desired target view. This method assumes a planar target and a small range of error, particularly for yaw. A virtual spring approach for points features is studied in [66, 67] where the image plane is assumed parallel to the target, which is restrictive because the image plane is not parallel to the target plane for nonzero roll and pitch. We use a virtual camera method, which uses attitude estimates from the IMU of the UAV to define a virtual image plane that remains parallel to the planar horizontal target [45, 68, 69, 71, 74]. The line feature kinematics and the UAV linear velocity dynamics in the virtual camera frame are independent of angular velocity ω, while a strong connection between image feature and yaw is preserved.

Virtual Camera Frame

Here we introduce a virtual camera frame $\mathcal{V}$ that is employed to remove the effects of perspective projection in the image feature kinematics and the translational velocity dynamics. The virtual camera frame is defined by basis $\{\nu_1, \nu_2, \nu_3\}$ whose origin coincides with the origin of real camera frame $\mathcal{B}$ but has zero roll and pitch with respect to fixed navigation $\mathcal{N}$. This implies the $n_1 - n_2$ plane is parallel to $\nu_1 - \nu_2$ plane. Figure 2.11 shows the virtual camera frame and its virtual image plane with respect to other frames. The rotation matrix describing the orientation of $\mathcal{B}$ and $\mathcal{V}$ is

$$R_b^v = R_{b_2}(\theta) R_{b_1}(\phi) = R_{\theta\phi}.$$

This rotation matrix allows us to convert any 3D point or vector from body frame $\mathcal{B}$ to the virtual camera frame $\mathcal{V}$, essentially removing the perspective effect from it.

Quadrotor UAV Dynamics in Virtual camera frame

In this Section we rewrite the dynamics of UAV in $\mathcal{V}$ so that they can be used in the IBVS control design. We can obtain the translational velocity kinematics in $\mathcal{V}$ by left multiplying R_b^v on both sides of (2.6b) and using the fact that roll and pitch angles between $\mathcal{V}$ and $\mathcal{N}$ are zero together with (2.4). The complete dynamics of the UAV with translational dynamics defined in $\mathcal{V}$ are given below

$$\dot{v}^v = -\mathrm{S}(\dot{\psi} e_3) v^v + g e_3 - \frac{F^v}{m_q} + \delta, \tag{2.22a}$$

$$\dot{\eta} = W^{-1} \omega, \tag{2.22b}$$

$$\dot{\omega} = -J_q^{-1} \mathrm{S}(\omega) J_q \omega + J_q^{-1} \tau_q, \tag{2.22c}$$

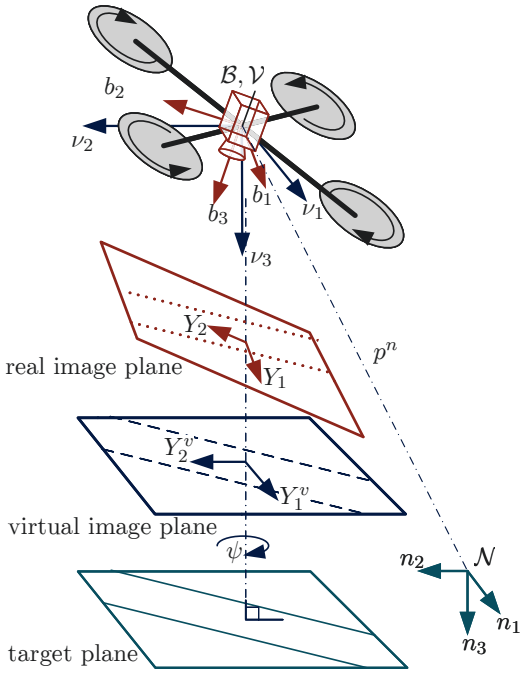


Figure 2.11: Navigation frame $\mathcal{N}$, real camera frame $\mathcal{B}$, virtual camera frame $\mathcal{V}$, real image and virtual image planes.

where $v^v = [v_1^v, v_2^v, v_3^v]^T$ is linear velocity in $\mathcal{V}$, $F^v = R_b^v F^b$ is the force input vector in $\mathcal{V}$,

$$W^{-1} = \begin{bmatrix} 1 & \mathrm{s}_\phi \mathrm{t}_\theta & \mathrm{c}_\phi \mathrm{t}_\theta \\ 0 & \mathrm{c}_\phi & -\mathrm{s}_\phi \\ 0 & \frac{\mathrm{s}_\phi}{\mathrm{c}_\theta} & \frac{\mathrm{c}_\phi}{\mathrm{c}_\theta} \end{bmatrix},$$

where $\mathrm{t}_\xi = \tan\xi$, $\delta = R_{b_3}^T(\psi)\frac{\tau_{\text{ext},p}}{m_q}$ is the disturbance term, $F^b = ue_3$, and $u = K_T f_T$ is the total thrust generated by the propellers, where the normalized thrust input is denoted $f_T \in [0, 1]$.

As shown experimentally in [45, 79], the thrust constant K_T slowly decreases with battery voltage, and significantly affects the UAV's vertical actuation. Over a short time frame, we can effectively treat K_T as an unknown constant parameter, and its slow variation allows the use of adaptive control. The mass of the UAV m_q is also taken as an unknown parameter. This allows for components on the UAV to be changed without affecting motion control performance (e.g., heavier batteries could be used for longer flights). The UAV payload and its mass can vary depending on the application (e.g., additional sensors might be added to achieve obstacle avoidance). In addition, we consider the disturbance term $\delta = [\delta_1, \delta_2, \delta_3]^T$ a slowly varying quantity, acting as a disturbance to the linear velocity dynamics. In addition to constant external forces such as wind, it also accounts for other uncertainties including attitude estimate bias [45] and rotor drag [173].

Transforming Line Features to the Virtual Camera Frame

As discussed above, any point or vector expressed in $\mathcal{B}$ can be transformed to $\mathcal{V}$ using the rotation matrix $R_b^v = R_\theta R_\phi = R_{\theta\phi}$. Therefore, $P^v = R_{\theta\phi}P$ and $n_i^v = R_{\theta\phi}n_i$. So substituting $P = R_{\theta\phi}^T P^v$ in (2.9), we have

$$n_i^T(R_{\theta\phi}^T P^v) + d_i = (R_{\theta\phi}n_i)^T P^v + d_i = n_i^{vT} P^v + d_i^v = 0, \tag{2.23}$$

where $i = 1, 2$. Hence, from (2.10) and (2.23) we have

$$l_u^v = \begin{bmatrix} n_1^v & n_2^v \end{bmatrix} \begin{bmatrix} -d_2 \\ d_1 \end{bmatrix} = R_{\theta\phi} \begin{bmatrix} n_1 & n_2 \end{bmatrix} \begin{bmatrix} -d_2 \\ d_1 \end{bmatrix} = R_{\theta\phi} l_u.$$

Substituting $l_\lambda^v = H l_u^v$ from (2.11), we have

$$l_\lambda^v = \begin{bmatrix} l_{\lambda 1}^v \\ l_{\lambda 2}^v \\ l_{\lambda 3}^v \end{bmatrix} = \begin{bmatrix} \sin\alpha^v \\ \cos\alpha^v \\ -\rho^v \end{bmatrix} = H R_{\theta\phi} H^{-1} l_\lambda = H R_{\theta\phi} H^{-1} \begin{bmatrix} \sin\alpha \\ \cos\alpha \\ -\rho \end{bmatrix}. \tag{2.24}$$

Therefore, the line features in the virtual camera frame are

$$l^v = \begin{bmatrix} \alpha^v \\ \rho^v \end{bmatrix} = \begin{bmatrix} \arctan\frac{l_{\lambda 1}^v}{l_{\lambda 2}^v} \\ \frac{-l_{\lambda 3}^v}{\sqrt{{l_{\lambda 1}^v}^2 + {l_{\lambda 2}^v}^2}}. \end{bmatrix}.$$

The line feature kinematics in virtual camera frame i.e., $l^v = [\alpha^v, \rho^v]^T$ can be written as follows

$$\begin{aligned} \dot{l}^v = \begin{bmatrix} \dot{\alpha}^v \\ \dot{\rho}^v \end{bmatrix} &= \begin{bmatrix} \sigma_{\alpha^v}\lambda\sin\alpha^v & \sigma_{\alpha^v}\lambda\cos\alpha^v & -\sigma_{\alpha^v}\rho^v \\ \sigma_{\rho^v}\lambda\sin\alpha^v & \sigma_{\rho^v}\lambda\cos\alpha^v & -\sigma_{\rho^v}\rho^v \end{bmatrix} \begin{bmatrix} v_1^v \\ v_2^v \\ v_3^v \end{bmatrix} \\ &+ \begin{bmatrix} \frac{\rho^v}{\lambda}\sin\alpha^v & \frac{\rho^v}{\lambda}\cos\alpha^v & 1 \\ (\frac{(\rho^v)^2}{\lambda} + \lambda)\cos\alpha^v & -(\frac{(\rho^v)^2}{\lambda} + \lambda)\sin\alpha^v & 0 \end{bmatrix} \begin{bmatrix} \omega_1^v \\ \omega_2^v \\ \omega_3^v \end{bmatrix}. \end{aligned} \tag{2.25}$$

where $\sigma_{\rho^v} = \frac{{a_i}^v\rho^v\sin\alpha^v + {b_i}^v\rho^v\cos\alpha^v + {c_i}^v\lambda}{{d_i}^v\lambda}$ and $\sigma_{\alpha^v} = -\frac{{a_i}^v\cos\alpha^v - {b_i}^v\sin\alpha^v}{{d_i}^v\lambda}$. Since every point lying on the 2D line ${y_1}^v\sin\alpha^v + {y_2}^v\cos\alpha^v - \rho^v = 0$ satisfies $\frac{1}{{X_3}^v} = -\frac{{a_i}^v{y_1}^v + {b_i}^v{y_2}^v + {c_i}^v\lambda}{{d_i}^v\lambda}$ and the point $({y_1}^v, {y_2}^v) = (\rho^v\sin\alpha^v, \rho^v\cos\alpha^v)$ lies on the line because ${y_1}^v\sin\alpha^v + {y_2}^v\cos\alpha^v - \rho^v = \rho^v\sin^2\alpha^v + \rho^v\cos^2\alpha^v - \rho^v = 0$, therefore

$$\sigma_{\rho^v} = \frac{{a_i}^v\rho^v\sin\alpha^v + {b_i}^v\rho^v\cos\alpha^v + {c_i}^v\lambda}{{d_i}^v\lambda} = \frac{-1}{{X_3}^v}.$$

Also, ${a_i}^v = {b_i}^v = 0$ as in an image plane parallel to the target plane containing lines, the line angel would not be affected by linear motion, only the angular velocities would affect the angel α^v, therefore $\sigma_{\alpha^v} = 0$. Since the image plane of the virtual camera frame is always parallel to the $n_1 - n_2$ plane in the navigation frame, the roll, pitch angles between $\mathcal{N}$ and $\mathcal{V}$ are zero. It can be shown from (2.4) that the respective angular velocities $\omega_1^v = \omega_2^v = 0$, while ω_3^v is equal to the yaw rate $\dot{\psi}$. Substituting these in line kinematics in virtual camera frame we have,

$$\dot{l}^v = \begin{bmatrix} \dot{\alpha}^v \\ \dot{\rho}^v \end{bmatrix} = \frac{-1}{{X_3}^v} \begin{bmatrix} 0 & 0 & 0 \\ \lambda \sin \alpha^v & \lambda \cos \alpha^v & -\rho^v \end{bmatrix} \begin{bmatrix} v_1^v \\ v_2^v \\ v_3^v \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \dot{\psi}. \tag{2.26}$$

Since we consider a horizontal target containing $N > 1$ parallel lines it can be shown

$$\sigma_{\alpha_k^v} = 0, \ \sigma_{\rho_k^v} = \frac{-1}{{X_3}^v},$$

where α_k^v, ρ_k^v denote the line features of the kth line. In the experimental validation of the control law in Section 3.4 we consider a linear target with a change in direction to demonstrate the method's robustness. Using (2.21) the kth line feature kinematics is

$$\begin{aligned} {\dot{\alpha}_k}^v &= \dot{\psi}, \\ {\dot{\rho}_k}^v &= \frac{-1}{X_3^v} \begin{bmatrix} \lambda \sin \alpha_k^v & \lambda \cos \alpha_k^v & -\rho_k^v \end{bmatrix} v^v. \end{aligned} \tag{2.27}$$

It is worth mentioning that using the virtual camera frame greatly simplifies the line feature kinematics. The dynamics for α_k^v is only a function of yaw rate $\dot{\psi}$. This is to be expected because the line angle as seen in the virtual image plane parallel to the target should not be affected by linear velocity or roll and pitch rate. Further, the depth X_3^v of all lines in the horizontal target in the virtual image plane is the same.

2.2.3 Line Moment Features

In this subsection we define line moment features which further simplify the feature kinematics (2.27). We start by defining

$$\rho_m = \frac{1}{N} \sum_{k=1}^{N} \rho_k^v, \qquad \alpha_m = \frac{1}{N} \sum_{k=1}^{N} \alpha_k^v.$$

Since the lines are parallel, the angle α_k^v is independent of k, therefore, $\alpha_k^v = \alpha_m$. Although all lines in the target have the same α_k, in practice different values of α_k are obtained due to measurement error or when targets are not perfectly linear. Using a mean value of α_k helps reduces the effect of these nonidealities. The kinematics

of α_m, ρ_m are given by

$$\begin{aligned}\dot{\alpha}_m &= \dot{\psi}, \\ \dot{\rho}_m &= \frac{-1}{X_3^v}\begin{bmatrix}\lambda \sin\alpha_m & \lambda\cos\alpha_m & -\rho_m\end{bmatrix} v^v.\end{aligned} \tag{2.28}$$

The mean distance ρ_m measures the lateral position of the lines in the image and the relative lateral displacement of the UAV to the target. Next, we define

$$\mu = \sum_{k=1}^{N} (\rho_k^v - \rho_m)^2, \tag{2.29}$$

which is the variance line distances from origin in the virtual image plane. It is a measure of the distance between lines and provides information about the height of the UAV. This is because lines appear closer in the image as UAV height increases. Its dynamics can be obtained by differentiating (2.29) and using (2.27) and (2.28),

$$\dot{\mu} = \frac{2\mu}{X_3^v} v_3^v. \tag{2.30}$$

As in [45] it can be shown that $X_3^v\sqrt{\mu}$ is a constant, and this leads to

$$X_3^v\sqrt{\mu} = X_3^{v*}\sqrt{\mu^*}, \tag{2.31}$$

where X_3^{v*} is desired depth or height above the target and μ^* is the desired value of μ at the UAV's reference configuration. We remark that μ^* is computed directly from the image of the target when the UAV is in its desired configuration. However, in order to improve the usability of the control law, we assume that no value of X_3^{v*} is available. Only an image of the desired goal configuration is needed. This image does not provide a value of X_3^{v*} unless we assume knowledge of accurate camera calibration parameters and target geometry. Since these last two assumptions are impractical, we treat X_3^{v*} as an unknown parameter in the control design.

Next, we define three-line moment features related to height, lateral distance, and the UAV's yaw relative to the target. The height moment feature is

$$s_h = \sqrt{\frac{\mu^*}{\mu}}.$$

Taking its derivative and using (2.31) and (2.30) we have

$$\dot{s}_h = -\frac{1}{X_3^{v*}} v_3^v. \tag{2.32}$$

As discussed earlier, ρ_m is a measure of the lateral position of the UAV. However, its

sensitivity is inversely related to height. To obtain the moment feature kinematics that depend linearly on lateral distance we define

$$s_l = \rho_m s_h. \tag{2.33}$$

Taking the time derivative of s_l and using (2.28), (2.32), and (2.31) we have

$$\begin{aligned}\dot{s}_l &= \frac{-\lambda}{X_3^{v*}} \begin{bmatrix}\sin\alpha_m^v & \cos\alpha_m^v\end{bmatrix} \begin{bmatrix}v_1^v \\ v_2^v\end{bmatrix} \\ &= \frac{-\lambda}{X_3^{v*}}(v_1^v \sin\alpha_m^v + v_2^v \cos\alpha_m^v). \end{aligned} \tag{2.34}$$

The quantity $(v_1^v \sin\alpha_m^v + v_2^v \cos\alpha_m^v)$ in (2.34) is the projection of the linear velocity in $\mathcal{V}$ along the perpendicular direction to the lines.

The yaw angle moment feature is defined as

$$s_\psi = \frac{1}{N}\sum_{k=1}^{N} \alpha_k^v = \alpha_m^v,$$

and its dynamics is

$$\dot{s}_\psi = \dot{\psi}. \tag{2.35}$$

This modelling of the line feature kinematics will be used later in Chapter 3 for IBVS line following.

2.2.4 Unmanned Aerial Manipulator model

In this section, we present the model of an Unmanned Aerial Manipulator (UAM). A UAM is essentially a UAV with a manipulator arm attached to it. We consider a quadrotor UAV with an n_a-DOF robot arm attached to its bottom. We consider that the arm has same number of links as joints. An ith link is connected to ith-joint on the UAV side and $(i+1)$-joint on the end-effector side. The modelling treats any external object temporarily attached to the arm as a link e.g., an object picked by end-effector for a pick and place application can be treated as an additional link attached to the end-effector. In modelling the UAM, we use the UAV and the frame definitions presented in Section 2.2.1 along with other notations defined in the previous sections. The term $\frac{\partial}{\partial z}(.)$ represents a gradient if its argument is a scalar and a Jacobian if the argument is a vector, while the transpose superscript above an operator signifies the transpose is evaluated after evaluating the operator, e.g., $\frac{\partial}{\partial z}^T(.) = \left(\frac{\partial}{\partial z}(.)\right)^T$ while $\frac{\partial}{\partial z}(x)^T = \frac{\partial}{\partial z}(x^T)$.

Kinematics of an arm

In addition, to the navigation frame $\mathcal{N}$ and the body frame $\mathcal{B}$ defined in Section 2.2.1, we consider a frame for each individual link of the robot arm. We consider that an i-th link has a frame $\mathcal{L}_i$ attached to its COM for all $i = 1, 2, ..., n_a$. A rotation matrix describing the relative orientation of $\mathcal{B}$ and $\mathcal{L}_i$ is $R^b_{L_i} \in \mathrm{SO}(3)$ and composed as below

$$R^b_{L_i} = R^b_{L_1} R^{L_1}_{L_2}, ..., R^{L_{i-1}}_{L_i},$$

where any R^k_j specifies a rotation from a frame j to a frame k. The origin of $\mathcal{L}_i$ or the location of the COM of the ith link, in $\mathcal{N}$ is given by

$$p^n_{L_i} = p^n + R^n_b p^b_{L_i}, \tag{2.36}$$

where $p^b_{L_i}$ denote the position of the COM of the ith link in $\mathcal{B}$ and is a function of the joint variables $\alpha_1, \alpha_2, ..., \alpha_i$. The velocity of the i-th link in $\mathcal{B}$ can be obtained by taking the time-derivative of $p^b_{L_i}$, given by

$$\dot{p}^b_{L_i} = J_{v,L_i} \dot{\alpha}, \tag{2.37}$$

where $\alpha \in \mathbb{R}^{n_a}$ is the vector containing the joint variables of the arm. The matrix $J_{v,L_i} \in \mathbb{R}^{3 \times n_a}$ is a linear velocity Jacobian matrix defined as $J_{v,L_i} = \frac{\partial p^b_{L_i}}{\partial \alpha}$. The total angular velocity of the ith link in $\mathcal{B}$ is $(\omega + \omega_{L_i})$ where $\omega_{L_i} \in \mathbb{R}^3$ is the angular velocity of the ith link measured in $\mathcal{B}$, in case of a static base (stationary UAV) and given by

$$\omega_{L_i} = J_{\omega,L_i} \dot{\alpha}, \tag{2.38}$$

where $J_{\omega,L_i} \in \mathbb{R}^{3 \times n_a}$ is the angular velocity Jacobian matrix given by

$$J_{\omega,L_i} = [R^b_{L_1} e_j, R^b_{L_2} e_j, ..., R^b_{L_i} e_j, 0_3, 0_3, ..., 0_3],$$

whereas e_j is one of the three vectors $e_1 = [1, 0, 0]^T$, $e_2 = [0, 1, 0]^T$ or $e_3 = [0, 0, 1]^T$ selected separately and appropriately for each vector entry of J_{ω,L_i}, depending upon the axis of operation of each joint of the arm, and $0_3 = [0, 0, 0]^T$.

Modelling of UAM Dynamics

There are three commonly used methods to derive the equations of motion of a mechanical system: Euler-Lagrange formulation, Newton-Euler formulation, and Kane's method. All three methods result in the same dynamics of the system. However, they vary in approach, calculation structure and complexity. The Euler-Lagrange formation is based on the calculation of the Lagrangian i.e., the difference of kinetic and potential energy of the whole system and then applying the Euler-

Lagrange equations. On the other hand, the Newton-Euler formulation uses a recursive method to derive the equations of motion. In contrast to the Euler-Lagrange, which considers all components of a multi-body system together, the Newton-Euler formulation considers the dynamics of the individual components with coupling forces and toques from the neighbouring components. Since the same coupling terms appear as reaction forces in the dynamics of the neighbouring components along with the coupling forces of their neighbouring components, the coupling effects are propagated throughout the chain of components. All coupling terms are evaluated using a so-called forward-backward recursion process to obtain the complete dynamics of the whole system in terms of generalized coordinates.

The Newton-Euler formulation is a lengthy and recursive process but has simple computations at each step, especially at the start, which become complex towards the end. On the other hand, Euler-Lagrange only involves two steps, i.e., calculating Lagrangian and then applying the Euler-Lagrange equations; however, because it considers the system as a whole, each step is massive and involves complex and lengthy expressions. Nevertheless, Euler-Lagrange is a well-established method that can capture all aspects of the system and have more applications than others. For example, Newton-Euler assumes the rigidity of the components and can not be applied to bodies with elastic deformations.

Kane's method is based on the principle of linear and angular momentum. It uses Kane's dynamical equations, which state that the sum of two quantities called generalized active forces and generalized inertia forces for all bodies in a system is zero. Instead of using position-based variables i.e., generalized coordinates as in Euler-Lagrange or Newton Euler formulation, Kane's method uses velocity-based scalar variables called generalized speeds, which are selected to be equal or more than the total number of degrees of the freedom of the system. After expressing the translational and angular velocity of each body or component of the system in terms of these generalized speeds, their partial derivative is taken with respect to the generalized speeds, called a partial velocity matrix. The generalized active force is the sum of all external forces acting on a body projected through the translational partial velocity matrix plus all external torques acting on the body projected through the angular partial velocity matrix. Similarly, the generalized inertia force of a body is its time rate of change of linear momentum projected through the translational partial velocity matrix plus the time rate of change of angular momentum projected through the angular partial velocity matrix. In Kane's dynamical equations, generalized active forces and generalized inertia forces for all bodies in the system are added and equated to zero to obtain the system dynamics. Kane's formulation is described in detail in [174] and [175] while a summary is given in [176].

Kane's method is a step-by-step procedure to obtain the system dynamics. It is

known to involve less complexity than the Newton-Euler due to the early cancellation of unused force terms. Also, the partial derivatives involved to obtain partial velocity matrices are far less complex than the partial derivative of Lagrangian required in the Euler-Lagrange equations. Nevertheless, Kane's method is generally non-intuitive and complex to understand.

We use Euler-Lagrange formulation to model the UAM system, which is well-established, reliable and most commonly used for modelling mechanical systems. The Euler-Lagrange equations are given by

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i, \quad i = 1, ..., n_s, \tag{2.39}$$

where $L = K - P$ is Lagrangian written as a function of generalized coordinates q_i, K is the kinetic energy of the system, P is the potential energy of the system, τ_i denote the generalized forces or torques which is the sum of external forces acting along the axis of the coordinate q_i, and n_s is the number of generalized coordinates of the system. More detail on Euler-Lagrange method can be found in [177]. In case of a UAM consisting of quadrotor and an n_a degrees of freedom robot arm, the generalized coordinates are

$$q = [p^{nT}, \eta^T, \alpha^T]^T \in \mathbb{R}^{6+n_a}. \tag{2.40}$$

The total potential energy P of the system is given by

$$P = -m_q g e_3^T p^n - \sum_{i=1}^{n_a} m_{L_i} g e_3^T p_{L_i}^n,$$

where m_{L_i} is the mass of the i-th link. Since the vertical basis of $\mathcal{N}$ points downward, the height above the ground is negative of the third component of the position vector defined in $\mathcal{N}$, this is the reason for both terms being negitive in above equation. Using (2.36) we can write

$$P = -e_3^T g((m_q + \sum_{i=1}^{n_a} m_{L_i})p^n + R_b^n \sum_{i=1}^{n_a} m_{L_i} p_{L_i}^b). \tag{2.41}$$

The total Kinetic energy of the system can be written as follows

$$\begin{aligned} K =& \frac{1}{2} m_q \dot{p}^{nT} \dot{p}^n + \frac{1}{2} \omega^T J_q \omega + \frac{1}{2} \sum_{i=1}^{n_a} m_{L_i} \dot{p}_{L_i}^{n\ T} \dot{p}_{L_i}^n \\ &+ \frac{1}{2} \sum_{i=1}^{n_a} (\omega + \omega_{L_i})^T R_{L_i}^b J_{L_i} R_{L_i}^{b\ T} (\omega + \omega_{L_i}). \end{aligned} \tag{2.42}$$

where J_{L_i} is the inertia of the i-th link with respect to its COM. The above equation

is not a function of generalized coordinates as specified in (2.40). To write it as a function of generalized coordinates, we derive the following expressions. Taking derivative of (2.36) and using (2.3), (2.4) and (2.37), we have

$$\dot{p}^n_{L_i} = \dot{p}^n - R^n_b \sum_{i=1}^{n_a} \mathrm{S}(p^b_{L_i}) W \dot{\eta} + R^n_b \sum_{i=1}^{n_a} J_{v,L_i} \dot{\alpha}. \tag{2.43}$$

Using (2.4), (2.38) and (2.43) in (2.42), and simplifying, we get

$$\begin{aligned}
K =& \frac{1}{2}(m_q + \sum_{i=1}^{n_a} m_{L_i}) \dot{p}^{nT} \dot{p}^n \\
&+ \frac{1}{2} \dot{\eta}^T W^T (J_q + \sum_{i=1}^{n_a} (R^b_{L_i} J_{L_i} {R^b_{L_i}}^T - m_{L_i} \mathrm{S}(p^b_{L_i})^2) W \dot{\eta} \\
&+ \frac{1}{2} \dot{\alpha}^T \sum_{i=1}^{n_a} (m_{L_i} J^T_{v,L_i} J_{v,L_i} + J^T_{\omega,L_i} R^b_{L_i} J_{L_i} {R^b_{L_i}}^T J_{\omega,L_i}) \dot{\alpha} \\
&- \frac{1}{2} \dot{\eta}^T W^T \sum_{i=1}^{n_a} (m_{L_i} \mathrm{S}^T(p^b_{L_i})) R^{nT}_b \dot{p}^n - \frac{1}{2} \dot{p}^{nT} R^n_b \sum_{i=1}^{n_a} (m_{L_i} \mathrm{S}(p^b_{L_i})) W \dot{\eta} \\
&+ \frac{1}{2} \dot{\alpha}^T \sum_{i=1}^{n_a} (m_{L_i} J^T_{v,L_i}) R^{nT}_b \dot{p}^n + \frac{1}{2} \dot{p}^{nT} R^n_b \sum_{i=1}^{n_a} (m_{L_i} J_{v,L_i}) \dot{\alpha} \\
&+ \frac{1}{2} \dot{\alpha}^T \sum_{i=1}^{n_a} (J^T_{\omega,L_i} R^b_{L_i} J_{L_i} {R^b_{L_i}}^T - m_{L_i} J_{v,L_i} \mathrm{S}(p^b_{L_i})) W \dot{\eta} \\
&+ \frac{1}{2} \dot{\eta}^T W^T \sum_{i=1}^{n_a} (R^b_{L_i} J_{L_i} {R^b_{L_i}}^T J_{\omega,L_i} - m_{L_i} \mathrm{S}^T(p^b_{L_i}) J_{v,L_i}) \dot{\alpha}.
\end{aligned} \tag{2.44}$$

The Lagrangian function $L = K - P$ can be obtained using (2.41) and (2.44). Since we have found the Lagrangian, we can apply Euler-Lagrange equations in (2.39) on it. The resulting system model has the following form

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + \tau_{ext}, \tag{2.45}$$

where $M(q) \in \mathbb{R}^{n_s \times n_s}$ and $C(q, \dot{q}) \in \mathbb{R}^{n_s \times n_s}$ are Inertial and Coriolis-Centrifugal matrices respectively, while $G(q) \in \mathbb{R}^{n_s}$ is a gravitational vector, $\tau \in \mathbb{R}^{n_s}$ is the input vector defined as $\tau = [(-uR^n_b n_3)^T, (W^T \tau_q)^T, \tau^T_\alpha]^T$ with $\tau_\alpha \in \mathbb{R}^{n_a}$ being the joint torques for the arm, $\tau_{\text{ext}} = [\tau^T_{\text{ext},p}, \tau^T_{\text{ext},\eta}, \tau^T_{\text{ext},\alpha}]^T \in \mathbb{R}^{n_s}$ is a vector consisting of external disturbances acting on the system, and $n_s = 6 + n_a$ i.e., system order is 6 degrees of freedom from the UAV and n_a from the arm. Due to long and complicated expressions for K in (2.44) it is difficult to apply (2.39) on L. Also, the scalar nature of (2.39), results in loss of intuitive understanding of interaction between various states in the resulting expressions associated with the vector representation. An alternative is to directly obtain $M(q)$ from K and obtain $C(q, \dot{q})$ using the entries

of $M(q)$. Using this approach, the matrix $M(q)$ can be obtained by rewriting (2.44) in the quadratic form

$$K = \frac{1}{2}\dot{q}^T M(q)\dot{q},$$

where

$$M(q) = \begin{bmatrix} (m_q + \sum_{i=1}^{n_a} m_{L_i})I_3 & -R_b^n \sum_{i=1}^{n_a}(m_{L_i}\mathrm{S}(p_{L_i}^b))W & R_b^n \sum_{i=1}^{n_a}(m_{L_i}J_{v,L_i}) \\ * & W^T J_{\eta\eta} W & W^T J_{\eta\alpha} \\ * & * & J_{\alpha\alpha} \end{bmatrix}, \tag{2.46}$$

where $J_{\eta\eta} = J_q + \sum_{i=1}^{n_a}(R_{L_i}^b J_{L_i} {R_{L_i}^b}^T - m_{L_i}\mathrm{S}^2(p_{L_i}^b))$, $J_{\eta\alpha} = \sum_{i=1}^{n_a}(R_{L_i}^b J_{L_i} {R_{L_i}^b}^T J_{\omega,L_i} - m_{L_i}\mathrm{S}(p_{L_i}^b)^T J_{v,L_i})$, and $J_{\alpha\alpha} = \sum_{i=1}^{n_a}(m_{L_i} J_{v,L_i}^T J_{v,L_i} + J_{\omega,L_i}^T R_{L_i}^b J_{L_i} {R_{L_i}^b}^T J_{\omega,L_i})$ while an $*$ at ijth entry represents a transpose of jith entry. The matrix $C(q,\dot{q})$ can be obtained by using the elements of $M(q)$

$$c_{kj} = \frac{1}{2}\sum_{i=1}^{n_a}\left(\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\right)\dot{q} \tag{2.47}$$

where c_{kj}, is the kjth scalar entry of the $C(q,\dot{q})$ and known as e Christoffel symbols of the first kind, while m_{kj} is the kjth scalar entry of $M(q)$. Equation (2.47) can be implemented using a symbolic tool on a computer to obtain $C(q,\dot{q})$. Although the vector or matrix representation of quantities involved is preserved in $M(q)$, however, equation (2.47) operates on the individual scalar entries of $M(q)$ and provides individual scalar entries of $C(q,\dot{q})$ the matrix, which once again destroys the matrix structure. The $C(q,\dot{q})$ consist of lengthy scalar expressions and do not provide information on the physics or the structure of coupling between the UAV and the arm for an intuitive understanding. The exiting models in literature including the most recent ones [106–108] do not provide the $C(q,\dot{q})$ matrix. Instead, the equation (2.47) is given. In contrast to the existing literature, we provide the complete model of the system with the Coriolis-Centrifugal matrix written in the form of sub-matrices that provides an intuitive understanding of the coupling between the UAV and arm dynamics.

In obtaining, the matrix $C(q,\dot{q})$, we have used MATLAB symbolic alongside hand calculations of Euler-Lagrange equations. We have calculated (2.39) the individual $i = 1, ..., n_s$ and then combined the result into matrix form for each of the sub-vectors of generalized coordinates i.e., p^n, η, α. We analyzed the result from hand calculation to write it as a vector or matrix differential that would result in the same expression. We then verified the vector or matrix representation by comparing it's result with that obtained from MATLAB symbolic. The resulting $C(q,\dot{q})$ matrix

has the following structure

$$C(q,\dot{q}) = \begin{bmatrix} 0_{3\times3} & C_{12} & C_{13} \\ 0_{3\times3} & C_{22} & C_{23} \\ 0_{n_a\times3} & C_{32} & C_{33} \end{bmatrix}$$

where

$$\begin{aligned}
C_{12} &= -R_b^n\Big(\mathrm{S}(W\dot{\eta})(\sum_{i=1}^{n_a} m_{L_i}\mathrm{S}(p_{L_i}^b))W - (\sum_{i=1}^{n_a} m_{L_i}\mathrm{S}(p_{L_i}^b))\dot{W}\Big),\\
C_{13} &= R_b^n\Big(2\mathrm{S}(W\dot{\eta})\sum_{i=1}^{n_a}(m_{L_i}J_{v,L_i}) + \sum_{i=1}^{n_a}(m_{L_i}\dot{J}_{v,L_i})\Big),\\
C_{22} &= \dot{W}^T J_{\eta\eta}W + W^T\dot{J}_{\eta\eta}W + W^T J_{\eta\eta}\dot{W} - \frac{1}{2}\frac{\partial}{\partial\eta}^T(W^T J_{\eta\eta}W\dot{\eta}),\\
C_{23} &= \dot{W}^T J_{\eta\alpha} + W^T\dot{J}_{\eta\alpha} - \frac{\partial}{\partial\eta}^T(W\dot{\eta})J_{\eta\alpha},\\
C_{32} &= \dot{J}_{\eta\alpha}{}^T W + J_{\eta\alpha}{}^T\dot{W} - \frac{\partial}{\partial\alpha}^T(J_{\eta\alpha}\dot{\alpha})W,\\
C_{33} &= \dot{J}_\alpha - \frac{1}{2}\frac{\partial}{\partial\alpha}^T(J_\alpha\dot{\alpha}).
\end{aligned}$$

It should be noted that the matrix $C(q,\dot{q})$ representation is not unique, however, the vector $C(q,\dot{q})\dot{q}$ is unique. The gravitational vector $G(q)$ is given by

$$G(q) = \frac{\partial P}{\partial q} = \begin{bmatrix} (m_q + \sum_{i=1}^{n_a} m_{L_i})ge_3 \\ -g\frac{\partial}{\partial\eta}^T(e_3^T R_b^n)^T\sum_{i=1}^{n_a}(m_{L_i}p_{L_i}^b) \\ ge_3^T R_b^n\frac{\partial}{\partial\alpha}\sum_{i=1}^{n_a}(m_{L_i}p_{L_i}^b) \end{bmatrix}.$$

2.2.5 Changes in UAV model due to an arm

It is interesting and important to identify how a nominal UAV model (2.5) differs from that of a UAM. It helps us identify how the addition of an arm to a UAV affects its dynamic behavior. After evaluating the matrix-vector multiplication and rearranging, the first two rows of (2.45) can be written as

$$(m_q + \sum_{i=1}^{n_a} m_{L_i})\ddot{p}^n = (m_q + \sum_{i=1}^{n_a} m_{L_i})ge_3 - uR_b^n e_3 + \bar{d}_f, \tag{2.48a}$$

$$W^T J_{\eta\eta}W\ddot{\eta} + W^T J_{\eta\eta}\dot{W}\dot{\eta} = -W^T\mathrm{S}(W\dot{\eta})J_{\eta\eta}W\dot{\eta} + W^T\tau_q + W^T d_\tau, \tag{2.48b}$$

$$J_{\alpha\alpha}\ddot{\alpha} = -\dot{J}_{\alpha\alpha}\dot{\alpha} + \frac{1}{2}\frac{\partial}{\partial\alpha}^T(J_{\alpha\alpha}\dot{\alpha})\dot{\alpha} + \tau_\alpha + d_\alpha, \tag{2.48c}$$

where

$$
\begin{aligned}
\bar{d_f} =& \tau_{\text{ext},p} + W^{-T} R_b^n \sum_{i=1}^{n_a} (m_{L_i} \mathrm{S}(p_{L_i}^b)) W \ddot{\eta} - R_b^n \sum_{i=1}^{n_a} (m_{L_i} J_{v,L_i}) \ddot{\alpha} \\
&+ R_b^n \Big(\mathrm{S}(W\dot{\eta}) (\sum_{i=1}^{n_a} m_{L_i} \mathrm{S}(p_{L_i}^b)) \times W - (\sum_{i=1}^{n_a} m_{L_i} \mathrm{S}(p_{L_i}^b)) \dot{W} \Big) \dot{\eta} \\
&- R_b^n \Big(2\mathrm{S}(W\dot{\eta}) \sum_{i=1}^{n_a} (m_{L_i} J_{v,L_i}) + \sum_{i=1}^{n_a} (m_{L_i} \dot{J}_{v,L_i}) \Big) \dot{\alpha}, \\
d_\tau =& \tau_{\text{ext},\eta} + \sum_{i=1}^{n_a} (m_{L_i} \mathrm{S}^T(p_{L_i}^b)) R_b^{n^T} \ddot{p^n} - W^{-T} R_b^n \sum_{i=1}^{n_a} (m_{L_i} J_{v,L_i}) \ddot{\alpha} \\
&- W^{-T} \dot{W}^T J_{\eta\eta} W \dot{\eta} - \dot{J}_{\eta\eta} W \dot{\eta} + \frac{1}{2} W^{-T} \frac{\partial}{\partial \eta}^T (W^T J_{\eta\eta} W \dot{\eta}) \dot{\eta} \\
&+ W^{-T} \dot{W}^T J_{\eta\alpha} \dot{\alpha} + \dot{J}_{\eta\alpha} \dot{\alpha} - W^{-T} \frac{\partial}{\partial \eta}^T (W\dot{\eta}) J_{\eta\alpha} \dot{\alpha} \\
&+ g \frac{\partial}{\partial \eta}^T (e_3^T R_b^n)^T \sum_{i=1}^{n_a} (m_{L_i} p_{L_i}^b), \\
d_\alpha =& \tau_{\text{ext},\alpha} - \sum_{i=1}^{n_a} (m_{L_i} J_{v,L_i}^T) R_b^{nT} - J_{\eta\alpha}^T W \ddot{\eta} - \dot{J}_{\eta\alpha}{}^T W \dot{\eta} - J_{\eta\alpha}^T \dot{W} \dot{\eta} \\
&+ \frac{\partial}{\partial \alpha}^T (J_{\eta\alpha} \dot{\alpha}) W \dot{\eta}.
\end{aligned}
$$

From (2.4), we have

$$\dot{\omega} = \dot{W}\dot{\eta} + W\ddot{\eta}. \tag{2.49}$$

In the UAM system model (2.48), (2.48a) and (2.48b) form the UAV subsystem referred to as UAM-vehicle (UAM-V) and (2.48c) is the arm subsystem referred to as UAM-Arm (UAM-A) in Chapter 1. These notations are used throughout this text to refer to either of the individual subsystems of a UAM. Let us now re-notate some of the terms in UAV subsystem to write it into a compact form. Defining $m = (m_q + \sum_{i=1}^{n_a} m_{L_i})$ and $d_f = \frac{\bar{d_f}}{m}$, and substituting (2.1), (2.4) and (2.49) in (2.48a) and (2.48b), we have

$$\dot{v}^n = ge_3 - auR_b^n e_3 + d_f, \tag{2.50a}$$

$$J_{\eta\eta}\dot{\omega} = -\mathrm{S}(\omega) J_{\eta\eta} \omega + \tau_q + d_\tau. \tag{2.50b}$$

where $a = \frac{1}{m}$ is the mass admittance. Here we have written the UAM-V dynamics in the same structure as the UAV dynamics in (2.5). This helps us identify the additional terms and factors introduced in the UAV model due to the addition of an n_a DOF robot arm. The change in UAV translational dynamics is represented as a change in the mass parameter and an additive time-varying force term d_f in the

linear velocity dynamics of the UAV, that further consists of the coupling terms due to the arm. Also, the change in UAV inertia matrix that varies with arm motion and an additive time-varying vector term d_τ containing the coupling terms in the angular velocity dynamics of the UAV as shown in (2.50). This representation allows us to look at the variations in the quantities in (2.50) and determine if any assumptions can be made about the UAM system under specific operational configurations. These assumptions will be considered later in Chapter 4.

2.3 Conclusion

In this chapter, we have described various Quadrotor UAV platforms maintained at ANCL. We looked at the essential hardware components of a quadrotor and their functioning and purpose. We also described the different software programs that run onboard the hardware components. We provided the detail on PX4, ROS and the Computer Vision system onboard ANCLQ 2.0 used for IBVS experiments. The modelling of different UAV configurations is also presented in this chapter. We looked at different frame definitions, rotation matrices, kinematic and dynamic modelling of a quadrotor UAV, camera model, image features definitions and their kinematics. We introduced the virtual camera frame to solve the problem due to perspective projection and defined image moment feature and their kinematics in the virtual camera frame that will be used in the next chapter for visual servoing of a quadrotor UAV. In the end, we looked at different modelling approaches used to model a mechanical system and used the Euler-Lagrange formulation to derive the dynamic model of a UAM. We analyzed this model and obtained the UAM dynamics in the same structure as a usual UAV dynamics so that the effects of the addition of a arm to a UAV can be identified. This modelling will be used in future chapters to design control for UAM systems.

Chapter 3

Image Based Visual Servoing: Line Following

In the last chapter, we presented the model of a quadrotor UAV, derived image features for lines, and presented line moment features in a virtual camera frame that decoupled the angular velocity from the outer-loop dynamics. This chapter will use that modelling to design a controller for motion control of a camera-equipped UAV relative to a linear or lines target. We consider a quadrotor UAV with a downward-facing camera that sees a planar horizontal target containing more than one line. Our objective is to control the relative pose of the UAV using IBVS. The pose here refers to the heading or yaw angle, lateral distance from the lines, and the UAV's height above the lines. The motion along the linear target is controlled manually by the user, who assigns the reference value for the UAV's pitch. It is challenging to design a controller for the motion of UAV along the linear target since the linear velocity along the line is not measurable for an unmarked line target on a plain background.

We use an output feedback control method, which eliminates the need for a GPS module, motion capture system or optical flow sensors. For estimation of lateral and vertical velocity relative to the lines, a simple linear observer is employed. The design is motivated by the output feedback approach in [178] and used in [71, 179] for points features. The proposed method considers uncertainty in UAV mass and the thrust constant, which depends on battery voltage and significantly affects vertical motion of the UAV as remarked experimentally in [45, 79]. Uncertainty also includes the unknown desired depth, that appears in the moment feature kinematics. Finally, we include a constant disturbance input to the transnational dynamics. This disturbance models attitude estimate bias and disturbance forces. A robust adaptive control law is proposed to compensate for the model uncertainty and disturbance.

Section 3.1 rearranges the system model and describes the control structure.

The controller and observer design for the outer-loop and the stability analysis of complete closed loop system is presented in Section 3.2. Simulation and experimental results are presented in Sections 3.3 and 3.4, respectively. Section 3.5 winds up the chapter by providing a conclusion.

3.1 Control Structure

To achieve the line tracking objective, we use an inner-outer loop control structure, as shown in Figure 3.1. The image feature kinematics and linear velocity dynamics comprise the open-loop outer subsystem dynamics. The UAV roll and pitch angles, and the input thrust are treated as inputs to the outer-loop. Hence, an outer-loop control design provides thrust input and desired values for roll and pitch angles that stabilize the feature error. These roll and pitch angles are then fed as a reference to an inner-loop controller, that determines vehicle torques for attitude tracking. The thrust computed by the outer-loop control i.e., the normalized thrust input, f_T, is sent directly to the UAV.

The outer-loop dynamics can be divided into three decoupled subsystems: yaw, height, and lateral. We define moment feature errors as $e_\psi = s_\psi$, $e_h = s_h - 1$, and $e_l = s_l$. Using (2.35), (2.32), (2.34), (2.22a), and $\bar{f}_T = f_T \cos\theta \cos\phi$ we have the error dynamics for the outer-loop:

$$\dot{e}_\psi = \dot{\psi}, \tag{3.1a}$$

$$\dot{e}_h = -\frac{1}{X_3^{v*}} v_3^v, \tag{3.1b}$$

$$\dot{v}_3^v = g - \frac{K_T \bar{f}_T}{m_q} + \delta_3, \tag{3.1c}$$

$$\dot{e}_l = \frac{-\lambda}{X_3^{v*}} (v_1^v \sin e_\psi + v_2^v \cos e_\psi), \tag{3.1d}$$

$$\dot{v}_2^v = -\dot{\psi} v_1^v + \frac{K_T \bar{f}_T}{m_q \cos\theta} \tan\phi + \delta_2, \tag{3.1e}$$

where (3.1a) forms the first order yaw subsystem, (3.1b)–(3.1c) is the height subsystem, and (3.1d)–(3.1e) is the lateral subsystem. In the height and lateral subsystems, the linear velocities $v_i^v, i = 1, 2, 3$ are unmeasured. Therefore we design an observer to estimate these states. Furthermore, adaptive control will account for the unknown parameters K_T, X_3^{v*}, m_q and constant disturbance δ.

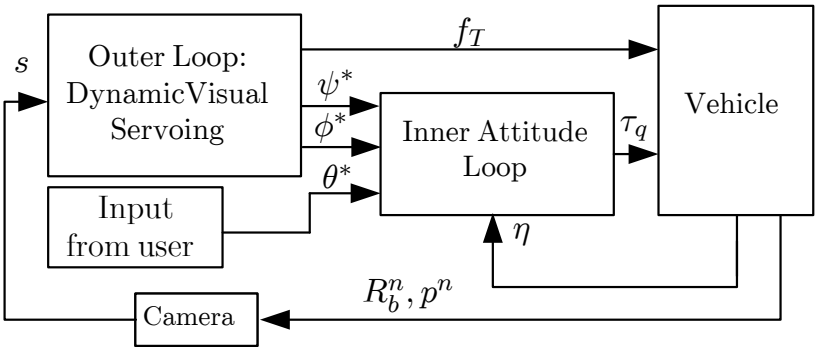


Figure 3.1: Inner-outer loop control structure

3.2 Controller and Observer Design

3.2.1 Outer-loop Control

Defining the inner-loop error for yaw $\tilde{\psi} = \psi - \psi^*$, if we use the yaw reference $\psi^* = -K_\psi \int_0^t e_\psi(\tau)d\tau$ with $K_\psi > 0$, the yaw subsystem in (3.1a) becomes

$$\dot{e}_\psi = -K_\psi e_\psi + \dot{\tilde{\psi}}, \tag{3.2}$$

which is exponentially stable assuming perfect inner-loop tracking (i.e., $\tilde{\psi} = 0$). Before presenting the controller and observer design for the remaining two subsystems we remark that it is not necessary to estimate $v_i^v, i = 1, 2, 3$ to ensure closed-loop convergence. As shown below, the transformed velocities can be estimated and used in the control with the transformation depending on unknown model parameters. For example, for the height subsystem we estimate a scaled relative velocity defined by $v_h = v_3^v / X_3^{v*}$. Expressing (3.1b), (3.1c) in terms of v_h, we rewrite the height subsystem as

$$\begin{aligned} \dot{e}_h &= -v_h, \\ \dot{v}_h &= b_h(D_h - \bar{f}_T), \end{aligned} \tag{3.3}$$

where $b_h = \frac{K_T}{m_q X_3^{v*}}$ and $D_h = \frac{m_q}{K_T}(g + \delta_3)$. We remark that D_h is the value of $\bar{f}_T$ at hover.

Subsystem (3.3) involves a measured output e_h, an unmeasured state v_h, an unknown parameter b_h and an unknown disturbance D_h. Our observer design based on output feedback will estimate D_h and v_h and the controller will be robust to error

in parameter b_h. Consider the following observer

$$\begin{aligned}
\dot{\hat{e}}_h &= -\hat{v}_h + l_{h1}(e_h - \hat{e}_h), \\
\dot{\hat{v}}_h &= -l_{h1}l_{h2}(e_h - \hat{e}_h), \\
\hat{D}_h &= \xi_h - 2\beta_h e_h, \\
\dot{\xi}_h &= -\beta_h((\gamma_h - l_{h2})e_h + l_{h2}\hat{e}_h + \hat{v}_h),
\end{aligned} \tag{3.4}$$

where $\hat{e}_h, \hat{v}_h, \hat{D}_h$ denote estimated quantities, ξ_h is the internal state of the disturbance observer and $l_{h1}, l_{h2}, \gamma_h, \beta_h$ are control gains to be determined. Let us define the output estimation error as $\tilde{e}_h = e_h - \hat{e}_h$, the velocity estimation error as $\tilde{v}_h = v_h - \hat{v}_h$ and disturbance estimation error as $\tilde{D}_h = \hat{D}_h - D_h$. The error dynamics of (3.3), (3.4) is

$$\begin{aligned}
\dot{\tilde{e}}_h &= -v_h + \hat{v}_h - l_{h1}\tilde{e}_h = -\tilde{v}_h - l_{h1}\tilde{e}_h, \\
\dot{\tilde{v}}_h &= b_h(D_h - \bar{f}_T) + l_{h1}l_{h2}\tilde{e}_h, \\
\dot{\tilde{D}}_h &= -\beta_h(\gamma_h e_h - l_{h2}\tilde{e}_h - v_h - \tilde{v}_h).
\end{aligned} \tag{3.5}$$

To stabilize the dynamics in (3.3) and (3.5) the control law is taken as

$$\bar{f}_T = k_h\left(\hat{v}_h - (l_{h2} + \gamma_h)e_h + l_{h2}\hat{e}_h\right) + \hat{D}_h, \tag{3.6}$$

where k_h is a control gain to be determined.

In the lateral subsystem in (3.1d)–(3.1e), we have two unknown states v_1^v, v_2^v, three unknown parameters K_T, m_q, X_3^{v*} and a constant disturbance δ_2. In order to stabilize the lateral feature error e_l we define a scaled version of the component of velocity along the shortest path connecting the origin of $\mathcal{C}$ and the lines:

$$v_l = \frac{\lambda}{X_3^{v*}}(v_1^v \sin e_\psi + v_2^v \cos e_\psi) = \frac{\lambda}{X_3^{v*}}v_2^v - \xi_1(e_\psi),$$

where

$$\xi_1(e_\psi) = \frac{-\lambda}{X_3^{v*}}\left(v_1^v \sin e_\psi + v_2^v(\cos e_\psi - 1)\right),$$

As with the height subsystem, we rewrite the lateral error dynamics (3.1d)-(3.1e) as

$$\begin{aligned}
\dot{e}_l &= -v_l, \\
\dot{v}_l &= b_l\left(\frac{\tan(\phi^* + e_\phi)}{\cos\theta} - D_l\right) + \xi(t),
\end{aligned} \tag{3.7}$$

where $b_l = \frac{\lambda K_T D_h}{m_q X_3^{v*}}$ and $D_l = \frac{-m}{K_T D_h}\delta_2$ are unknown constants, v_l is an unmeasured state,

$$\xi(t) = \xi_2(e_\psi, \tilde{f}_T) - \dot{\xi}_1(e_\psi),$$

with

$$\xi_2(e_\psi, \tilde{f}_T) = \frac{\lambda}{X_3^{v*}} \left(\frac{K_T \tan\phi}{m_q \cos\theta} \tilde{f}_T - \dot{\psi} v_1^v \right),$$

$\tilde{f}_T = \bar{f}_T - D_h$, and $e_\phi = \phi - \phi^*$. We note that the above equation depends on yaw subsystem variable $\dot{\psi}$ and height subsystem variable f_T. Since the closed-loop height and yaw subsystems will be proven exponentially stable, we treat this coupling as an exponentially decaying disturbance. The exponential convergence of $\dot{\psi}$ and $\tilde{f}_T$ to origin with the exponential convergence of yaw and height subsystem errors can be seen from (3.2) and (3.3), respectively. We propose the following observer for estimation of lateral velocity v_l, and disturbance D_l,

$$\begin{aligned}
\dot{\hat{e}}_l &= -\hat{v}_l + l_{l1}(e_l - \hat{e}_l), \\
\dot{\hat{v}}_l &= -l_{l1} l_{l2} (e_l - \hat{e}_l), \\
\hat{D}_l &= \xi_l + 2\beta_l e_l, \\
\dot{\xi}_l &= \beta_l((\gamma_l - l_{l2}) e_l + l_{l2} \hat{e}_l + \hat{v}_l),
\end{aligned} \tag{3.8}$$

where $\hat{e}_l, \hat{v}_l, \hat{D}_l$ denote estimated quantities, $l_{l1},\ l_{l2},\ \gamma_l,\ \beta_l$ are controller gains to be determined, and ξ_l is the internal state of the disturbance observer. Defining the estimation errors for output $\tilde{e}_l = e_l - \hat{e}_l$, velocity $\tilde{v}_l = v_l - \hat{v}_l$ and the disturbance as $\tilde{D}_l = \hat{D}_l - D_l$, the error dynamics can be written as

$$\begin{aligned}
\dot{\tilde{e}}_l &= -v_l + \hat{v}_l - l_{l1}\tilde{e}_l = -\tilde{v}_l - l_{l1}\tilde{e}_l \\
\dot{\tilde{v}}_l &= b_l \left(\frac{\tan(\phi^* + e_\phi)}{\cos\theta} - D_l \right) + l_{l1} l_{l2} \tilde{e}_l + \xi(t) + b_l e_\phi \\
\dot{\tilde{D}}_l &= \beta_l(\gamma_l e_l - l_{l2}\tilde{e}_l - v_l - \tilde{v}_l)
\end{aligned} \tag{3.9}$$

To stabilize the lateral subsystem dynamics (3.7) and (3.9), a control law is taken as

$$\phi^* = \arctan(\cos\theta(-k_l\,(\hat{v}_l - (l_{l2} + \gamma_l) e_l + l_{l2}\hat{e}_l) + \hat{D}_l)), \tag{3.10}$$

where k_l is a controller gain to be determined.

Theorem 3.2.1. *Assuming perfect inner-loop tracking, i.e., $[e_\phi, e_\theta, \tilde{\psi}]^T = 0$, and consider the height and lateral subsystems (3.3) and (3.7), their respective observers (3.4) and (3.8), the estimation error dynamics (3.5) and (3.9), and the control laws (3.6) and (3.10). If the control gains satisfy*

$$l_{h1} > l_{h2} > b_h k_h > \gamma_h > 0, \tag{3.11}$$

and

$$l_{l1} > l_{l2} > b_l k_l > \gamma_l > 0, \tag{3.12}$$

then the equilibrium $[e_h, v_h, \tilde{e}_h, \tilde{v}_h, \tilde{D}_h, e_l, v_l, \tilde{e}_l, \tilde{v}_l, \tilde{D}_l]^T = 0$ *is globally exponentially stable.*

Proof. Consider the change of state coordinates

$$\begin{bmatrix} z_{h1} \\ z_{h2} \\ z_{h3} \\ z_{h4} \\ z_{h5} \end{bmatrix} = \begin{bmatrix} \gamma_h & -1 & 0 & 0 & 0 \\ 0 & 0 & -l_{h2} & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_h \\ v_h \\ \tilde{e}_h \\ \tilde{v}_h \\ \tilde{D}_h \end{bmatrix}, \tag{3.13}$$

which transforms the control (3.6) into

$$\bar{f}_T = k_h(z_{h2} - z_{h1}) + \hat{D}_h. \tag{3.14}$$

Transforming the dynamics (3.3), (3.5) into the new state coordinates and substituting the control (3.14) gives the closed-loop

$$\dot{z}_{h1} = -(b_h k_h - \gamma_h) z_{h1} + b_h k_h z_{h2} - \gamma_h^2 z_{h3} + b_h z_{h5}, \tag{3.15a}$$

$$\dot{z}_{h2} = -b_h k_h z_{h1} - (l_{h2} - b_h k_h) z_{h2} - l_{h2}^2 z_{h4} + b_h z_{h5}, \tag{3.15b}$$

$$\dot{z}_{h3} = z_{h1} - \gamma_h z_{h3}, \tag{3.15c}$$

$$\dot{z}_{h4} = z_{h2} - (l_{h1} - l_{h2}) z_{h4}, \tag{3.15d}$$

$$\dot{z}_{h5} = -\beta_h (z_{h1} + z_{h2}). \tag{3.15e}$$

Consider a Lyapunov function candidate

$$V = \frac{1}{2}\left(z_{h1}^2 + z_{h2}^2 + \gamma_h^2 z_{h3}^2 + l_{h2}^2 z_{h4}^2 + \frac{b_h}{\beta_h} z_{h5}^2 \right),$$

which is radially unbounded. Taking derivative and substituting (3.15), we have

$$\dot{V} = -(b_h k_h - \gamma_h) z_{h1}^2 - (l_{h2} - b_h k_h) z_{h2}^2 - \gamma_h^3 z_{h3}^2 - l_{h2}^2 (l_{h1} - l_{h2}) z_{h4}^2,$$

which is negative semi-definite if (3.11) are satisfied. This implies that the vector consisting of $[z_{h1}, z_{h2}, z_{h3}, z_{h4}]^T$ converges to the origin. Using LaSalle's invariance principle, (3.15a) and the fact that V is radially unbounded, implies the global asymptotic stability (GAS) of $[z_{h1}, z_{h2}, z_{h3}, z_{h4}, z_{h5}]^T = 0$. Since (3.15) is linear, GAS implies global exponentially stability (GES). Since GAS is preserved under the linear transformation (3.13), we have proven GES of $[e_h, v_h, \tilde{e}_h, \tilde{v}_h, \tilde{D}_h]^T = 0$.

Now consider a similar transformation for the lateral subsystem

$$\begin{bmatrix} z_{l1} \\ z_{l2} \\ z_{l3} \\ z_{l4} \\ z_{l5} \end{bmatrix} = \begin{bmatrix} \gamma_l & -1 & 0 & 0 & 0 \\ 0 & 0 & -l_{l2} & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_l \\ v_l \\ \tilde{e}_l \\ \tilde{v}_l \\ \tilde{D}_l \end{bmatrix},$$

which transforms the control in (3.10) into

$$\phi^* = \arctan(\cos\theta(-k_l(z_{l2} - z_{l1}) + \hat{D}_l)). \tag{3.16}$$

Since we have assumed $e_\phi = 0$, the closed loop dynamics in the new coordinates are

$$\dot{z}_l = A_l z_l + B_\xi \xi(t), \tag{3.17}$$

where

$$A_l = \begin{bmatrix} -(b_l k_l - \gamma_l) & b_l k_l & -\gamma_l^2 & 0 & -b_l \\ -b_l k_l & -(l_{l2} - b_l k_l) & 0 & -l_{l2}^2 & -b_l \\ 1 & 0 & -\gamma_l & 0 & 0 \\ 0 & 1 & 0 & -(l_{l1} - l_{l2}) & 0 \\ \beta_l & \beta_l & 0 & 0 & 0 \end{bmatrix}, B_\xi = \begin{bmatrix} -1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

and $z_l = [z_{l1}, z_{l2}, z_{l3}, z_{l4}, z_{l5}]^T$ The unforced part of the dynamics (3.17) has the same structure as (3.15) and is therefore A_l is Hurwitz provided (3.12) holds. The signal $\xi(t)$ is exponentially convergent since $\xi_1(e_\psi), \xi_2(e_\psi, \tilde{f}_T)$ are exponentially convergent as the closed-loop yaw and height subsystems are are globally exponentially stable. Therefore, the equilibrium $[e_h, v_h, \tilde{e}_h, \tilde{v}_h, \tilde{D}_h, e_l, v_l, \tilde{e}_l, \tilde{v}_l, \tilde{D}_l]^T = 0$ of the entire closed-lop is globally exponentially stable. □

We remark that although conditions (3.11) and (3.12) depend on unknown parameters b_h and b_l, sufficiently small $k_h, k_l, \gamma_h, \gamma_l$ and sufficiently large $l_{l1}, l_{l2}, l_{h1}, l_{h2}$ can always be chosen so that the conditions are satisfied. Estimates on the range of the unknown parameters can determine worst case values for b_l, b_h and help in selecting appropriate controller gains. Care must be taken in selecting gains as selecting values for gains that are too large could lead to the saturation of the vehicle actuators and could lead to lose of vehicle control and instability.

3.2.2 Inner-loop Control and Entire Closed-loop Stability

As discussed earlier, the thrust input for height control is algebraically related to the motor inputs; however, in the case of lateral and yaw subsystems, we need to

design control so that the reference set-points generated from the outer-loop can be achieved.

Using small angle assumption i.e., $\tan\phi \approx \phi$ and $\cos\theta \approx 1$, (3.16) can be written as

$$\phi^* = -k_l(z_{l2} - z_{l1}) + \hat{D}_l = \bar{K}_l z_l + \hat{D}_l, \tag{3.18}$$

where $\bar{K}_l = [k_l, -k_l, 0, 0, 0]$. Also, the dynamics in (3.17) for $e_\phi \neq 0$ become

$$\dot{z}_l = A_l z_l + B_\xi \xi(t) + B_l e_\phi, \tag{3.19}$$

where $B_l = [b_l, b_l, 0, 0, 0]^T$. Taking derivative of (3.18) w.r.t time and substituting values for $\dot{z}_l$ from (3.19), $\dot{\hat{D}}_l = \beta_l(z_{l1} + z_{l2}) = e_5^T A_l Z_l$ where $e_5 = [0, 0, 0, 0, 1]^T$ and $\bar{K}_l B_l = \bar{K}_l B_\xi = 0$, we have

$$\dot{\phi}^* = (\bar{K}_l + e_5^T) A_l z_l = K_l A_l z_l.$$

Using the small angle assumption, the inner-loop dynamics of a quadrotor (2.22b)-(2.22c) can be approximated as

$$\ddot{\eta} = J^{-1}\tau^c.$$

The inner-loop dynamics for the lateral subsystem are therefore given by

$$\begin{aligned} \dot{e}_\phi &= \dot{\phi} - \dot{\phi}^*, \\ \ddot{\phi} &= \frac{1}{J_1}\tau_1. \end{aligned} \tag{3.20}$$

Consider the controller

$$\tau_1 = -\frac{k_{3,\phi}}{k_{2,\phi}}\dot{\phi} - \frac{k_{3,\phi}}{k_{1,\phi}}e_\phi - k_{3,\phi}\int_0^t e_\phi(\tau)d\tau, \tag{3.21}$$

and a linear transformation

$$\begin{aligned} x_{1,\phi} &= \int_0^t e_\phi(\tau)d\tau, \\ x_{2,\phi} &= \frac{1}{k_{1,\phi}}e_\phi + x_{1,\phi}, \\ x_{3,\phi} &= \frac{1}{k_{2,\phi}}\dot{\phi} + x_{2,\phi}, \end{aligned} \tag{3.22}$$

which transforms the controller (3.21) into $\tau_1 = -k_{3,\phi}x_{3,\phi}$ and system in (3.20) with

control substituted becomes

$$\dot{x}_\phi = A_\phi x_\phi - \frac{1}{k_{1,\phi}} B_{\phi l} z_l, \tag{3.23}$$

where

$$A_\phi = \begin{bmatrix} -k_{1,\phi} & k_{1,\phi} & 0 \\ -k_{1,\phi} & k_{1,\phi} - \frac{k_{2,\phi}}{k_{1,\phi}} & \frac{k_{2,\phi}}{k_{1,\phi}} \\ -k_{1,\phi} & k_{1,\phi} - \frac{k_{2,\phi}}{k_{1,\phi}} & \frac{k_{2,\phi}}{k_{1,\phi}} - \frac{k_{3,\phi}}{J_1 k_{2,\phi}} \end{bmatrix}, B_{\phi l} = \begin{bmatrix} 0 \\ K_l A_l \\ K_l A_l \end{bmatrix}, x_\phi = \begin{bmatrix} x_{1,\phi} \\ x_{2,\phi} \\ x_{3,\phi} \end{bmatrix}.$$

Again consider the outer-loop lateral subsystem (3.19) with $e_\phi \neq 0$ together with the inner-loop dynamics (3.23), using $e_\phi = B_\phi x_\phi$ from (3.22) where $B_\phi = [-k_{1,\phi}, k_{1,\phi}, 0]$. The closed loop can be written as

$$\dot{z}_l = A_l z_l + B_d \xi(t) + B_{l\phi} x_\phi, \tag{3.24a}$$

$$\dot{x}_\phi = A_\phi x_\phi - \frac{1}{k_{1,\phi}} B_{\phi l} z_l. \tag{3.24b}$$

where $B_{l\phi} = B_l B_\phi$. We remark that $\xi(t)$ exponentially converges to zero due to the exponential stability of the height and yaw subsystems. From the Theorem above, since the system in (3.19) with $e_\phi = 0$ is globally exponentially stable at origin. Using the Converse Lyapunov Theorem [180, Thm. 4.14], there exists a Lyapunov function $V_{1,\phi}(z_l, t)$ defined on $\mathbb{R}^5 \times \mathbb{R}$ that satisfies the inequalities

$$\begin{aligned} c_1 \|z_l\|^2 \leqslant V_{1,\phi}(z_l, t) \leqslant c_2 \|z_l\|^2, \\ \frac{\partial V_{1,\phi}}{\partial t} + \frac{\partial V_{1,\phi}}{\partial z_l} \left(A_l z_l + B_\xi \xi(t)\right) \leqslant - c_3 \|z_l\|^2, \\ \left\| \frac{\partial V_{1,\phi}}{\partial z_l} \right\| \leqslant c_4 \left\| z_l \right\|, \end{aligned}$$

for all $z_l \in \mathbb{R}^5, t \geq 0$ for some positive constants c_1, c_2, c_3 and c_4. We consider the following Lyapunov function candidate

$$V_\phi(t, x_\phi, z_l) = V_{1,\phi}(z_l, t) + \frac{1}{2} x_\phi^T x_\phi,$$

and its time derivative is given as

$$\begin{aligned}
\dot{V}_\phi &= \frac{\partial V_{1,\phi}}{\partial t} + \frac{\partial V_{1,\phi}}{\partial z_l}\left(A_l z_l + B_{l\phi}x_\phi + B_\xi \xi(t)\right) + \frac{1}{2}\dot{x}_\phi^T x_\phi + \frac{1}{2}x_\phi^T \dot{x}_\phi \\
&= \frac{\partial V_{1,\phi}}{\partial t} + \frac{\partial V_{1,\phi}}{\partial z_l}\left(A_l z_l + B_\xi \xi(t)\right) + \frac{\partial V_{1,\phi}}{\partial z_l}B_{l\phi}x_\phi + \frac{1}{2}(A_\phi x_\phi - \frac{1}{k_{1,\phi}}B_{\phi l}z_l)^T x_\phi \\
&+ \frac{1}{2}x_\phi^T(A_\phi x_\phi - \frac{1}{k_{1,\phi}}B_{\phi l}z_l) \\
&\leqslant -c_3\|z_l\|^2 + \left\|\frac{\partial V_{1,\phi}}{\partial z_l}B_{l\phi}x_\phi\right\| + \frac{1}{2}x_\phi^T A_\phi^T x_\phi - \frac{1}{2k_{1,\phi}}z_l^T B_{\phi l}^T x_\phi \\
&+ \frac{1}{2}x_\phi^T A_\phi x_\phi - \frac{1}{2k_{1,\phi}}x_\phi^T B_{\phi l} z_l \\
&\leqslant -c_3\|z_l\|^2 + \left\|\frac{\partial V_{1,\phi}}{\partial z_l}\right\| \|B_{l\phi}\| \|x_\phi\| + \frac{1}{2}x_\phi^T\left(A_\phi^T + A_\phi\right)x_\phi - \frac{1}{k_{1,\phi}}z_l^T B_{\phi l}^T x_\phi. \quad (3.25)
\end{aligned}$$

Since we can choose A_ϕ, we take

$$A_\phi^T + A_\phi = -Q_\phi$$

where Q_ϕ satisfies

$$\lambda_{\min}(Q_\phi)\|x_\phi\|^2 \leqslant x_\phi^T Q_\phi x_\phi \leqslant \lambda_{\max}(Q_\phi)\|x_\phi\|^2.$$

Also ,

$$-\frac{1}{k_{1,\phi}}z_l^T B_{\phi l}^T x_\phi \leqslant \left\|\frac{1}{k_{1,\phi}}z_l^T B_{\phi l}^T x_\phi\right\| \leqslant \|z_l\|\left\|\frac{1}{k_{1,\phi}}B_{\phi l}^T\right\|\|x_\phi\|.$$

The last inequality in (3.25) becomes

$$\begin{aligned}
\dot{V}_\phi &\leqslant -c_3\|z_l\|^2 + c_4\|z_l\|\|B_{l\phi}\|\|x_\phi\| - \frac{1}{2}x_\phi^T Q_\phi x_\phi + \|z_l\|\left\|\frac{1}{k_{1,\phi}}B_{\phi l}^T\right\|\|x_\phi\| \\
&\leqslant -c_3\|z_l\|^2 - \frac{1}{2}\lambda_{\min}(Q_\phi)\|x_\phi\|^2 + \|z_l\|\left(c_4\|B_{l\phi}\| + \left\|\frac{1}{k_{1,\phi}}B_{\phi l}^T\right\|\right)\|x_\phi\|. \quad (3.26)
\end{aligned}$$

If the controller gains $k_{1,\phi}, k_{2,\phi}, k_{3,\phi}$ in (3.21) are selected so that $\lambda_{\min}(Q_\phi)$ is large enough such that $\sqrt{2c_3\lambda_{\min}(Q_\phi)} > c_4\|B_{l\phi}\| + \left\|\frac{1}{k_{1,\phi}}B_{\phi l}^T\right\|$ is satisfied then re-writing (3.26) gives

$$\begin{aligned}
\dot{V}_\phi \leqslant &-\left(\sqrt{c_3}\|z_l\| - \sqrt{\frac{\lambda_{\min}(Q_\phi)}{2}}\|x_\phi\|\right)^2 - \left(\sqrt{2c_3\lambda_{\min}(Q_\phi)}\right. \\
&\left. - c_4\|B_{l\phi}\| - \left\|\frac{1}{k_{1,\phi}}B_{\phi l}^T\right\|\right)\|z_l\|\|x_\phi\| \\
&< 0.
\end{aligned}$$

This proves the closed-loop system is GAS at the origin. Given that system in

(3.24) is linear, GAS implies GES. We remark that if we remove the small-angle assumption in (3.24), only local exponential stability can be shown. It is important to note that the small-angle assumption has been used to simplify the analysis, and it does not necessarily mean that the system will diverge if the assumption is not valid. Since inspection tasks do not require acrobatic maneuvers, the absolute value of vehicle roll and pith references can be saturated at $\frac{\pi}{4}$ rad which is a reasonably large angle and can provide sufficient actuation for translational motion. For roll and pitch angles in the range $[-\pi/4, \pi/4]$, there is a bounded difference between the model's original and approximated values of nonlinear terms. Since exponential stability is known for its robustness to disturbances, the bounded effects of the model approximation can be treated as a disturbance and would lead to a bounded error in the worst case.

Similarly, the inner-outer closed loop yaw subsystem is

$$\begin{aligned}\dot{\tilde{\psi}} &= \dot{\psi} - \dot{\psi}^*, \\ \ddot{\psi} &= \frac{1}{J_3}\tau_3.\end{aligned}$$

which can be exponentially stabilized using

$$\tau_3 = -\frac{k_{3,\psi}}{k_{2,\psi}}\dot{\psi} - \frac{k_{3,\psi}}{k_{1,\psi}}\tilde{\psi} - k_{3,\psi}\int_0^t \tilde{\psi}(\tau)d\tau. \tag{3.27}$$

Evidently the velocity along the line can not be measured. Therefore, UAV pitch is controlled from a pitch reference θ^* given by the user. We define the pitch tracking error $e_\theta = \theta - \theta^*$ and its dynamics

$$\begin{aligned}\dot{e}_\theta &= \dot{\theta}, \\ \ddot{\theta} &= \frac{1}{J_2}\tau_2,\end{aligned}$$

which can be controlled using a PID controller as in (3.21) and (3.27):

$$\tau_2 = -\frac{k_{3,\theta}}{k_{2,\theta}}\dot{\theta} - \frac{k_{3,\theta}}{k_{1,\theta}}e_\theta - k_{3,\theta}\int_0^t e_\theta(\tau)d\tau. \tag{3.28}$$

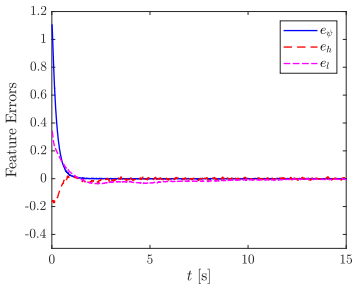
3.3 Simulation Results

In this section, we present the simulation results for the inner-outer closed-loop control. The rigid body dynamics of the UAV are implemented in MATLAB Simulink and the camera is modelled using Peter Corke's Machine Vision Toolbox [181].

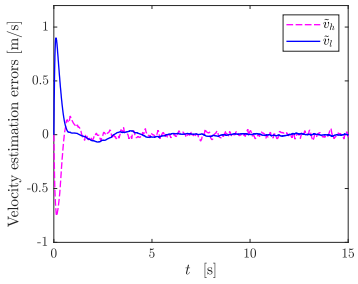
Table 3.1: Outer-loop measurement noise

Signal	Mean	Standard Deviation	Mean PSD [dB]	Sampling Rate [Hz]
e_h	7.85×10^{-4}	0.0051	-45.0	21.7
e_l	-0.0498	0.1785	-36.3	21.7
e_ψ [rad]	7.14×10^{-4}	6.62×10^{-4}	-76.5	21.7
ϕ [rad]	-0.0026	2.62×10^{-4}	-109	93.9
θ [rad]	0.0159	1.83×10^{-4}	-94.6	93.9

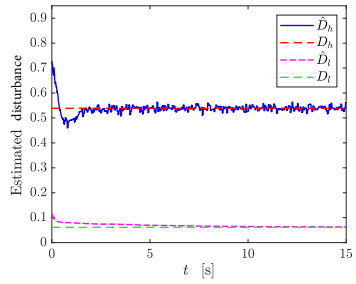
The proposed inner-outer loop control laws (3.2), (3.6), (3.10), (3.21), (3.28) and (3.27) are implemented. The simulation uses quadrotor and camera parameters given in Table 3.2 which correspond to the ANCLQ 2.0 experimental platform used in Section 3.4. These parameters of the UAV and the camera are determined using system identification. The UAV mass is measured using a weight scale and its inertia is obtained using a pendulum swing test. The camera parameters are obtained from the camera's specifications and using camera calibration. No image processing is performed in the simulation. Rather, the pinhole model (2.8) is used to project 3D points which comprise the linear target onto the image plane. The controller gains are listed in Table 3.3. In the simulation, the lateral feature in (2.33) is scaled by a factor of $\varepsilon = 1/100$ so that it has a similar range as the other feature components. This helps with gain tuning since both lateral and height controllers have a similar structure, and if the values of the feature errors have the same order, the same gains can be used as a starting point when tuning. The only effect this scaling has on controller design is that b_l is replaced by εb_l in the conditions (3.12). The controller gains are appropriately selected to avoid a large overshoot in lateral motion to ensure the target remains in the camera's field of view. The simulation employs two parallel lines in a horizontal plane and spaced apart by $1/\sqrt{3}\,\mathrm{m}$. The initial conditions for the UAV are such that $e_\psi = 0.35\pi\,\mathrm{rad}$ and $\rho_m = 770$ pixels. The initial position of the UAV in the navigation frame is $p^n = [-0.3, 0, -5]^T\,\mathrm{m}$. The initial conditions for v^v, η, and ω are zero. This results in initial conditions for $[s_{\psi 0}, s_{h0}, s_{l0}] = [0.35\pi, 0.82, 0.35]$. We take $K_T = 37.6\,\mathrm{N/ms^2}$, $X_3^{v*} = 6\,\mathrm{m}$, $[\delta_1, \delta_2, \delta_3] = [0, -1, -1]\,\mathrm{m/s^2}$ which results in $D_h = 0.539$, $D_l = 0.0612$, $b_h = 2.725$ and $b_l = 6.12$. The desired moment features are $[s_\psi^*, s_h^*, s_l^*] = [0, 1, 0]$. The initial conditions for the observers are $\hat{e}_h = 0$, $\hat{e}_l = 0$, $\hat{v}_h = 0$, $\hat{v}_l = 0$. The initial values of estimated disturbances depend on initial conditions for output and velocity estimation errors and given as $\hat{D}_h = 0.731$, $\hat{D}_l = 0.171$. In order to simulate the approximate measurement noise in the actual platform, Gaussian white noise is added to the measurements of the outer-loop according to Table 3.1. The noise powers and means were obtained by collecting measurement data on a fixed UAV. A nominal value for focal length was chosen with a 20% error. Such a large error could arise when no calibration is performed and a rough estimate for λ is chosen.



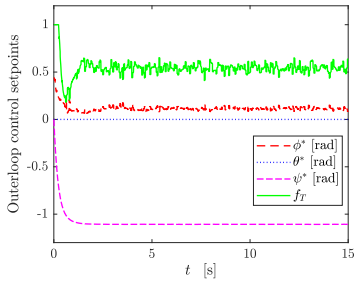
(a) Feature error



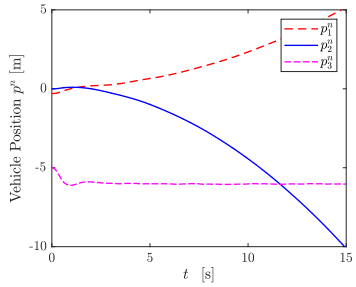
(b) Velocity estimation error



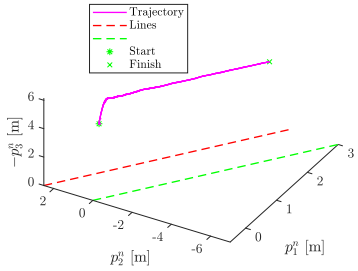
(c) Disturbance estimates



(d) Outer loop control inputs



(e) Vehicle position



(f) Top view of vehicle trajectory

Figure 3.2: Line following simulation results

Table 3.2: UAV and Camera Parameters

Parameter	Value
Inertial Matrix J	$\text{diag}([0.03, 0.03, 0.05])\,\text{kgm}^2$
Mass m_q	$2.3\,\text{kg}$
Focal Length λ	415 pixels
Pixel size	$4.8\,\mu\text{m}$
Image size	[640, 480]
Image centre $[y_{10}, y_{20}]$	[320, 240]

Table 3.3: Controller Gains

Gain	Value
K_ψ	4
l_{h1}	16
l_{h2}	5
k_h	0.94
γ_h	1.25
β_h	0.9
l_{l1}	14
l_{l2}	4.37
k_l	0.36
γ_l	1.093
β_l	0.15

Gain	Value
$k_{1,\psi}$	0.0008
$k_{2,\psi}$	0.002
$k_{3,\psi}$	0.1
$k_{1,\phi}$	0.0003
$k_{2,\phi}$	0.0078
$k_{3,\phi}$	0.1
$k_{1,\theta}$	0.0008
$k_{2,\theta}$	0.003
$k_{3,\theta}$	0.1

The trajectories of the system states, disturbance estimates, and control inputs are shown in Figure 3.2. The line features in Figure 3.2 converge to a small neighbourhood of the origin. When no noise in the attitude is present, convergence to zero is obtained. As expected from the theory, the unknown disturbance estimates $\hat{D}_h$ and $\hat{D}_l$ converge to their actual values in Figure 3.2c while the state estimation errors $\tilde{v}_h$ and $\tilde{v}_l$ converge to zero in Figure 3.2b. The plots in Figure 3.2d show the trajectories of ϕ^*, θ^*, ψ^* and f_T. The normalized thrust f_T saturates at 1 initially and eventually settles to the value of D_h once the desired height is achieved. The plot for ϕ^* shows more noise than other plots. This is because the lateral feature s_l is defined, involving the product of ρ_m and s_h.

In steady-state the absolute values of feature errors remain bounded i.e., $|e_\psi| < 0.003$, $|e_l| < 0.012$ and $|e_h| < 0.02$ after 10 seconds. Although the features and the inputs are affected by noise, they remain bounded to practically useful levels. Moreover, UAV position is relatively unaffected by noise as shown in Figure 3.2e. Here we observe that p_3^n converges to $-6\,\text{m}$. This satisfies the desired height requirement $X_3^* = 6\,\text{m}$. The 3D trajectory of the UAV is shown in Figure 3.2f. The lateral position error at steady state is bounded to $0.035\,\text{m}$ while the height error is bounded to $0.02\,\text{m}$. The simulation results demonstrate accurate motion control in the face of the unmodelled uncertainty in focal length and attitude measurement

noise and bias.

3.4 Experimental Results

The experimental validation of the proposed algorithm is performed using the Applied Nonlinear Control Laboratory (ANCL) quadrotor UAV platform ANCLQ 2.0 described in Section 2.1. The Computer Vision System onboard ANCLQ 2.0 captures the image, performs several image processing steps, computes the moment feature in $\mathcal{V}$, and sends it to the PX4, where they are used in the controller to generate UAV inputs. To simplify line detection in the experiment and keep the implementation portable to other applications e.g., those involving point features, we construct a target consisting of two lines by using two sets of different coloured patches positioned along the straight lines. We detect the coloured patches as points located at their centroids and then fit a line through these points. However, the coloured patches appear to be along two curves instead of two parallel straight lines due to camera distortion. This introduces additional noise in the image feature measurements. Direct line detection algorithms such as the Hough transform, and camera distortion models as shown in [79] could also be used to reduce this noise. For each

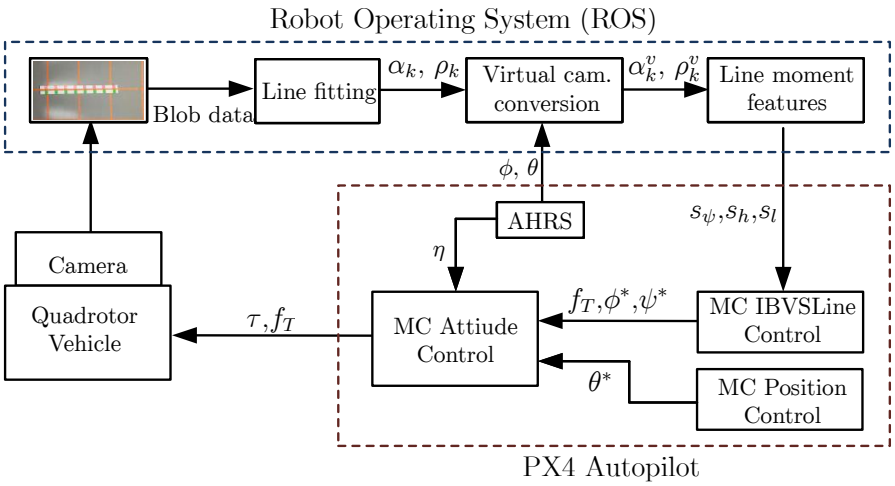


Figure 3.3: Block diagram of the controller showing implementation details

image two lines are determined, one for each colour. This determines α, ρ, and l_λ. Using l_λ and the UAV attitude received from the PX4 through a MAVROS topic in ROS, we use (2.24) to obtain l_λ^v. Using l_λ^v we can compute line moment features which are then sent back to the PX4. The block diagram of this implementation is shown in Figure 3.3.

On the PX4 side, `mc_ibvsline` module is implemented, which subscribes to the `img_moments` topic containing the line moment features received from MAVROS.

This module runs the outer-loop control, and its output is normalized thrust and an attitude reference that serves as a reference signal for the inner-loop. These reference values are obtained by implementing the control laws (3.6) and (3.10) together with the estimators (3.4) and (3.8). Since the main contribution of the paper lies in the outer-loop design, the inner-loop control is implemented using the stock v.1.5.5 PX4 firmware module `mc_att_control`. During the experiment, we employ a Vicon motion capture system for ground truth UAV position in $\mathcal{N}$. This allows us to evaluate the relative position error of the control. We set the desired height above the target to 1 m. For convenience, position along the line is controlled using the built-in position controller of the PX4 so that the UAV changes direction when it reaches 0.5 m from the origin. Vicon data for p_1^n is used to generate an appropriate θ^* to simulate a user manually controlling linear velocity along the line via θ^*.

The experimental results are shown in Figures 3.4-3.7. The shaded area in each plot represents the time during which IBVS is enabled. Figure 3.4 shows the convergence of the image feature errors to a practically small region of the origin. The lateral feature error e_l has a relatively large variation since the lateral feature is generally more sensitive to change in position than the other two features. There are two primary reasons that all feature errors exhibit some level of variation in steady-state. First, due to the relatively small ratio between patch length (i.e., 10 cm) and the spacing between patches (i.e., 25 cm). This leads to a low density of points used to create a line, and any errors in the positions of these points lead to a large error in feature. Secondly, since there is a change in the UAV direction along the line every 160 cm, this leads to a disturbance torque which periodically disturbs the feature errors away from the origin. Figure 3.5 shows the control input for the outerloop. The plot for θ^* includes periodic "spikes" corresponding to the change in UAV direction while moving along the line. The inner-loop control error is in Figure 3.6. Clearly, roll and pitch errors remain close to the origin. Yaw error convergence is slower due to the relatively low bound of torque about the c_3 axis. The 3D UAV position is shown in Figure 3.7. The data shows that the UAV remains within about ± 10 cm of the desired position for both lateral and height control. This is an acceptable accuracy for transmission line inspection as outlined in [182]. Also, the performance is significantly better than a consumer or civilian GNSS with an accuracy of a few meters. The statistics of the steady-state performance are given in Table 3.4. Here, e_{p_2} and e_{p_3} denote lateral and height position errors, respectively.

The experiment described above tests the proposed method's robustness to a range of model uncertainties and measurement noises. For example, image features include noise due to camera distortion, time delay for image processing, implementation requiring controller discretization, and many system variables are bounded in

Table 3.4: Statistics of experimental results

Parameter	Mean	Standard Deviation
e_h	0.004	0.026
e_l	0.108	0.294
e_ψ [rad]	0.005	0.010
e_{p_2} [m]	0.009	0.018
e_{p_3} [m]	0.005	0.031

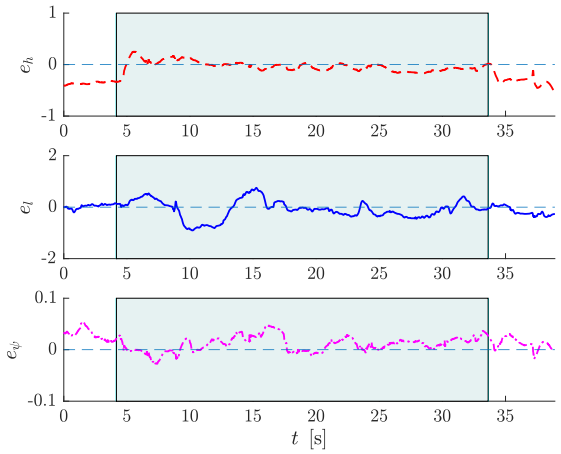


Figure 3.4: Image moment feature errors for line following experiment

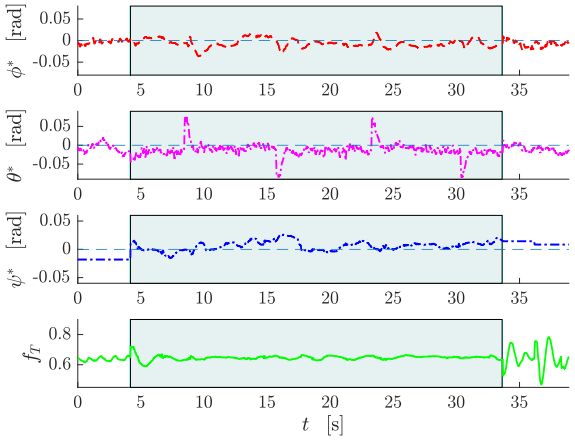


Figure 3.5: Attitude and thrust reference inputs

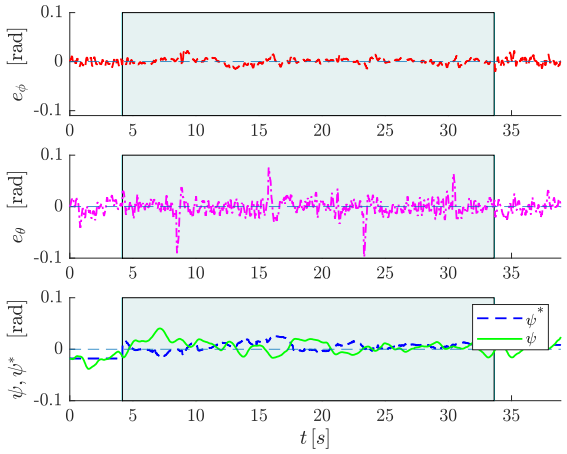


Figure 3.6: Attitude control errors

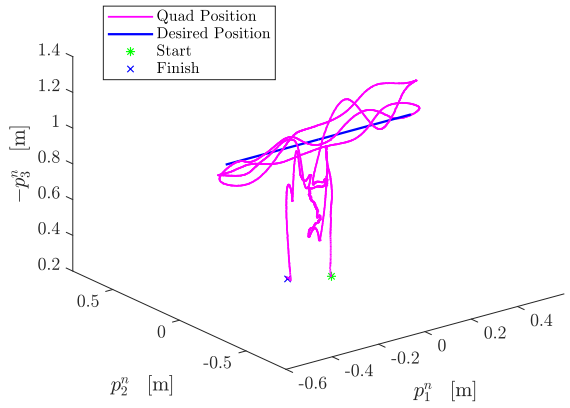


Figure 3.7: 3D position trajectory of the UAV during experiment

practice. We remark that the indoor lab environment has good lighting conditions, and the visual targets used are not necessarily representative of linear targets found in the field. In real outdoor environments, image processing could introduce error, leading to performance limitations in the motion control.

In the following experiment, we relax the assumption that the target consists of straight lines. We also initialize the UAV away from the line to demonstrate the convergence of the feature error. We consider a visual target that is piecewise linear, including a 30^o change of direction. We apply an initial lateral displacement error of about 35 cm. The results for this case are in Figs. 3.8-3.10 and a video of the experiment is available online [1]. We observe from e_l in Figure 3.8 that the initial lateral feature error when the IBVS is engaged is about $e_l = 2$ and convergence to the origin is similar to that in simulation in Figure 3.2a. The plot of the yaw feature in Figure 3.2a has intervals where the error is non-zero; this is when the UAV performs a change of direction, and there are two non-parallel sets of lines in the image. In this case, α is a weighted average with more weight given to the line with more visible points. So as the UAV moves above the turn, α is constantly changing. Once it is past the change of direction, the yaw feature error stabilizes to zero. The attitude and thrust references are shown in Figure 3.9 and are similar to Figure 3.5. The 3D position of UAV is shown in Figure 3.10.

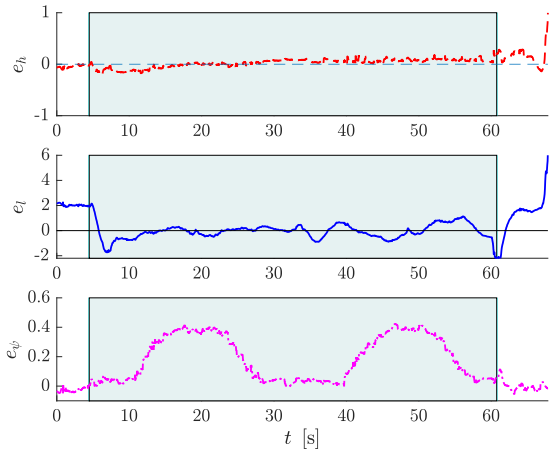


Figure 3.8: Image moment feature errors for a piecewise linear target

3.5 Conclusion

In this chapter, we proposed an IBVS for a quadrotor UAV for line following. Output feedback is used to eliminate the need for linear velocity measurements. We consider

[1] IBVS Experiment video: `https://youtu.be/Nkaf59vUjKM`

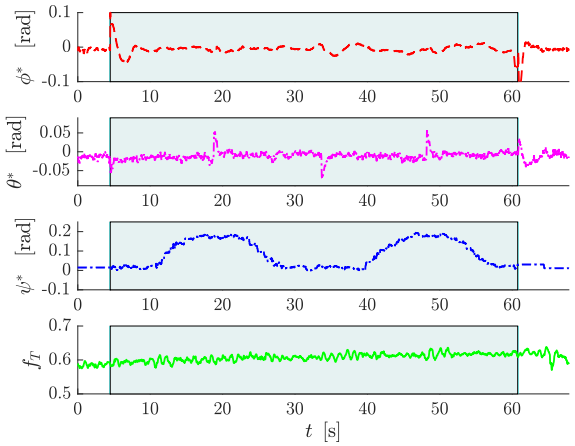


Figure 3.9: Attitude and thrust reference inputs for a piecewise linear target

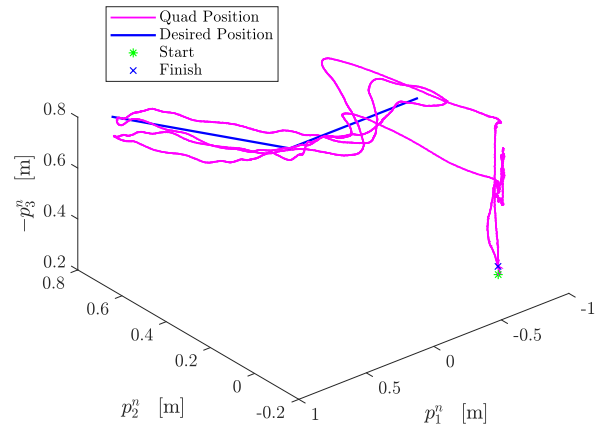


Figure 3.10: 3D position trajectory of the UAV for a piecewise linear target

model uncertainty in thrust constant, mass, desired depth, and linear acceleration disturbance. An inner-outer loop structure is used, and the global exponential stability of the outer-loop is proven. The inner-outer loop exponential stability is proven. Simulation and experimental results demonstrate the effectiveness of the method. Future work focuses on compensating for camera field of view constraints and relaxing the small-angle assumption in the inner-outer loop stability proof. Future work will improve experimental validation by considering outdoor environments using real-world linear targets such as transmission lines. A possible limitation of the work is that linear velocity along the lines must be controlled manually with a user-supplied pitch set point. Manual control is required since the velocity cannot be estimated if the lines have no distinctive features along their length and their background is plain. However, in practice, the background will usually have texture (e.g., due to vegetation and non-uniform terrain). Therefore, an optical flow measurement could be used to estimate scaled linear velocity along the line. Alternately, accelerometer measurements could be fused with known distances between features that a camera can detect (e.g., joints between pipes of known length).

Chapter 4

Inner-outer Loop UAM Motion Control

In the previous chapter, we designed an IBVS control for line-following applications. This chapter shifts topics to a traditional (i.e., where the full UAV state is assumed measured) motion control problem for a UAM consisting of a UAV equipped with a robot arm. This UAM was modelled in Section 2.2.4. An example of a UAM which is used in our simulations below is given in Figure 4.1. UAMs increase the capability of UAVs as they can be used for new applications such as load transport as described in Chapter 1. In the interest of increased UAV capability, future work could naturally combine UAM and visual servoing technologies.

Similar to the IBVS approach discussed in Chapter 3, we employ an inner-outer loop control design in this chapter. The benefits of an inner-outer loop design are described in Section 1.2.2. Such designs are simpler to implement and tune. They usually have robust performance and are therefore used in the field e.g., in the PX4 firmware, an inner-outer loop motion control is used [169]. In some commercial UAVs e.g., the DJI Matrice 100, [183] only the outer-loop can be modified and an inner-loop design is not required. An inner-outer loop structure is modular

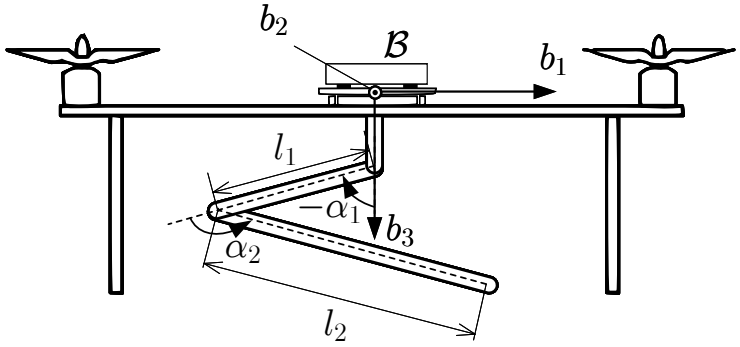


Figure 4.1: Configuration of UAM arm.

and different designs for each loop can generally be mixed. Evidently, closed-loop analysis of the entire error dynamics is affected by the combination chosen.

In Section 4.1 we describe how a UAM model can be approximated by a traditional UAV dynamics with parametric uncertainty and disturbance inputs. Based on this uncertain dynamics we describe an adaptive inner-outer motion controller in Sections 4.2 and 4.3. In Section 4.4 we present a simulation of the proposed design under various conditions including a pick and place operation.

4.1 UAM Model Approximation

In this section, we explain how the UAM model presented in Section 2.2.4 can be approximated so that it can be used for motion control. The approach involves lumping the effect of the arm subsystem (UAM-A) on the UAV subsystem (UAM-V) into disturbances and parametric uncertainty. We do not consider the control of the UAM-A and assume an independent controller is running which can accurately track joint variable references.

To motivate the model approximation, we consider a general pick and place task where the UAM picks an object from one location and moves it to another location where it is placed. We divide the pick and place operation into three phases.

Phase 1: Stationary Arm and Moving Vehicle In the first phase, a UAM-V takes off and moves to a location near the target object. The appropriate location is selected through trajectory planning so that the target object is within reach of the arm. During this phase, the arm remains retracted and stationary with respect to the UAM-V.

Phase 2: Stationary Arm and Vehicle Hovering In the second phase, the UAM-V has reached a desired location near the target object, and the arm has not yet started moving. Therefore, both the UAM-V and the arm are stationary with respect to the navigation frame.

Phase 3: Moving arm Vehicle Hovering In this phase, the arm slowly extends to pick up an object of interest. Once the end-effector has reached the target object, the arm stops moving. The end-effector picks up the object, which causes a change in UAM mass and inertia. However, the mass remains constant before and after this change. Once the object is picked up, the arm retracts close to the UAM-V.

Phase 1-3 are repeated after the UAM has picked the object in order to complete a place operation.

Now we discuss the variations of UAM mass m, inertia $J_{\eta\eta}$, force d_f and torque d_τ from the model in (2.50) for all three phases. The mass m and the inertia $J_{\eta\eta}$

of the UAM remain constant in both Phase 1 and Phase 2. In Phase 3, the value of UAM mass m and inertia $J_{\eta\eta}$ changes to a new constant value when an object is picked or placed. The inertia $J_{\eta\eta}$ also varies slowly with the slow arm movement in Phase 3. To generalize it for all phases of a pick and place operation, we can assume that both m and $J_{\eta\eta}$ remain constant or vary slowly in all three phases. In addition, to account for their slow variation and a shift at the moment of picking or placing the object, we consider them as unknown parameters.

Let us now discuss the variation of the force coupling $d_f = \bar{d}_f/m$ in the UAM-V due to arm, for a pick and place application. Let us rewrite the equation for $\bar{d}_f$ from (2.48), we have

$$\begin{aligned}\bar{d}_f =&\tau_{\text{ext},p} + W^{-T}R_b^n\sum_{i=1}^{n_a}(m_{L_i}\mathrm{S}(p_{L_i}^b))W\ddot{\eta} - R_b^n\sum_{i=1}^{n_a}(m_{L_i}J_{v,L_i})\ddot{\alpha}\\ &+ R_b^n\Big(\mathrm{S}(W\dot{\eta})(\sum_{i=1}^{n_a}m_{L_i}\mathrm{S}(p_{L1}^b))W - (\sum_{i=1}^{n_a}m_{L_i}\mathrm{S}(p_{Li}^b))\dot{W}\Big)\dot{\eta}\\ &- R_b^n\Big(2\mathrm{S}(W\dot{\eta})\sum_{i=1}^{n_a}(m_{L_i}J_{v,Li}) + \sum_{i=1}^{n_a}(m_{L_i}\dot{J}_{v,Li})\Big)\dot{\alpha}.\end{aligned} \tag{4.1}$$

In Phase 2, if we consider the external disturbance $\tau_{\text{ext},p}$ as constant e.g., due to wind, UAM-V maintains a constant attitude η to compensate for constant external forces to remain in hover. Also, since the arm is stationary in Phase 2, α is also constant. The constant η and α result in $\ddot{\eta} = \dot{\eta} = \ddot{\alpha} = \dot{\alpha} = 0$ and (4.1) reduces to $d_f = \tau_{\text{ext},p}/m$. Therefore, we conclude d_f remains constant throughout Phase 2.

In Phase 1, the arm is stationary with respect to UAM-V, while the UAM-V moves, therefore $\ddot{\alpha} = \dot{\alpha} = 0$ and the (4.1) reduces to

$$\begin{aligned}\bar{d}_f =&\tau_{\text{ext},p} + W^{-T}R_b^n\sum_{i=1}^{n_a}(m_{L_i}\mathrm{S}(p_{L_i}^b))W\ddot{\eta} + R_b^n\Big(\mathrm{S}(W\dot{\eta})(\sum_{i=1}^{n_a}m_{L_i}\mathrm{S}(p_{L1}^b))W\\ &- (\sum_{i=1}^{n_a}m_{L_i}\mathrm{S}(p_{Li}^b))\dot{W}\Big)\dot{\eta}.\end{aligned} \tag{4.2}$$

Although $\bar{d}_f$ depends on UAV attitude, however, its effect is minimal. The actuation for UAV yaw is known to be limited. Therefore, UAV yaw varies slowly, allowing us to approximate $\ddot{\psi} = \dot{\psi} = 0$. The changes in UAV roll ϕ and pitch θ are for a minimal period. This is because in an aerial transport mission, a UAM-V flies the shortest distance between the two points, which is usually a straight line when no obstacle is present. If we ignore drag force and assume constant external force disturbance, then UAM-V roll and pitch are constant except at the endpoints of the motion. Therefore, we can assume that UAM-V attitude η remains constant in Phase 1. In case when a UAM-V tracks a time-varying trajectory where roll or pitch angles are

not constant, we have the ability to minimize their effects on d_f. This is because the robot arm's retracted state can be configured to minimize the effects of change in roll and pitch. If the joint variables of the arm are selected as below

$$\alpha_h = \arg\min_{\alpha} \left\| \sum_{i=1}^{n_a} m_{L_i} p^b_{L_i}(\alpha) \right\|,$$

which minimizes the first moment of mass $\sum_{i=1}^{n_a} m_{L_i} p^b_{L_i}(\alpha)$. This effectively reduces the term $\sum_{i=1}^{n_a} m_{L_i} \mathrm{S}(p^b_{L_i})$ in (4.2) because for any vector $\delta \in \mathbb{R}^3$, $\sum_{i=1}^{n_a} m_{L_i} \mathrm{S}(p^b_{L_i})\delta = -\mathrm{S}(\delta) \sum_{i=1}^{n_a} m_{L_i} p^b_{L_i}$. For a sufficiently high DOF robot arm, if the arm angles can be controlled arbitrarily, it is possible to reduce $\sum_{i=1}^{n_a} m_{L_i} p^b_{L_i}(\alpha)$ to null vector. Let us combine the assumption of η mostly constant when moving on a straight line with constant speed resulting in $\ddot{\eta} = \dot{\eta} = 0$ and arm angles in α_h configuration, when moving along time-varying trajectory, we assume that $d_f = \tau_{\mathrm{ext},p}/m$ and hence constant in Phase 1.

In Phase 3, the UAM-V is stationary, while the arm moves, resulting

$$d_f = \tau_{\mathrm{ext},p} - R^n_b \sum_{i=1}^{n_a} (m_{L_i} J_{v,L_i})\ddot{\alpha} - R^n_b \Big(2\mathrm{S}(W\dot{\eta}) \sum_{i=1}^{n_a} (m_{L_i} J_{v,Li}) + \sum_{i=1}^{n_a} (m_{L_i} \dot{J}_{v,Li}) \Big) \dot{\alpha}.$$

If we assume that the arm moves slowly to extend or retract, we can approximate $\ddot{\alpha} = \dot{\alpha} = 0$, which again results in $d_f = \tau_{\mathrm{ext},p}$.

Following the above discussion, we conclude that it is reasonable to assume that d_f is constant or slowly varying in all three phases of pick and place operation. Using a similar logic, d_τ can be approximated as constant as well, i.e., $d_\tau = \tau_{\mathrm{ext},\eta}$. Even if we assume that the coupling force d_f and torque d_τ due to arm slowly vary, they can attain large values; therefore, in the following analysis, they are considered unknown. This discussion leads to the following assumption.

Assumption 1. *The parameter m, $J_{\eta\eta}$ and the disturbance vectors d_f and d_τ are considered unknown, however, it is assumed that their values are either slowly varying or constant so that $\dot{a} \approx 0$, $\dot{J}_{\eta\eta} \approx 0$, $\dot{d}_f \approx 0$ and $\dot{d}_\tau \approx 0$, where $a = \frac{1}{m}$.*

The reason for the above discussion is to motivate this assumption, which allows us to apply adaptive control techniques for the estimation of m, $J_{\eta\eta}$, d_f and d_τ. In the following section, we will see that this assumption gets used and allows us to derive the control and estimation laws.

4.2 Control Design

In this section, we design a controller for the UAM-V considering the system parameters and coupling force and torque terms due to the arm as unknown constants.

The motion control objective for a UAM-V is to track a time-varying pose (position and heading) trajectory while compensating for the effects due to arm. As discussed earlier, we use an inner-outer loop control structure, which is further explained in the following section.

4.2.1 Control Structure

We use a cascade control structure known as inner-outer loop control, where the model of a quadrotor UAM-V is further divided into two subsystems, translational and rotational. The outer-loop control consists of UAM-V position kinematics and UAM-V linear velocity dynamics. A UAM-V thrust along with roll and pitch angles are considered inputs to the outer-loop dynamics. The thrust input goes to the UAM-V, while roll and pitch, being virtual inputs, serve as reference trajectories for the inner-loop dynamics.

The inner-loop dynamics consist of UAM-V attitude kinematics and angular velocity dynamics controlled by the UAM-V torques. Therefore, in the inner-loop, we design a controller for UAM-V torques. The UAM-V attitude tracks the reference roll and pitch angle trajectories received from the outer-loop and the yaw reference received from the user. In order to make the presentation simple, let us re-notate some parameters and variables defined in the model of a UAM in Chapter 2. We notate $p = p^n$, $R = R_b^n$, $v = v^n$, $J = J_{\eta\eta}$, and write the complete system dynamics as follows

$$\dot{p} = v, \tag{4.3a}$$
$$\dot{v} = ge_3 - aF + d_f, \tag{4.3b}$$
$$\dot{\eta} = W^{-1}(\eta)\omega, \tag{4.3c}$$
$$J\dot{\omega} = -\mathrm{S}(\omega)J\omega + \tau_q + d_\tau, \tag{4.3d}$$

where $F = uRe_3 = [F_1, F_2, F_3]^T$ which gives $F_1 = u(\mathrm{c}_\psi \mathrm{s}_\theta \mathrm{c}_\phi + \mathrm{s}_\psi \mathrm{s}_\phi)$, $F_2 = u(\mathrm{c}_\phi \mathrm{s}_\theta \mathrm{s}_\psi - \mathrm{s}_\phi \mathrm{c}_\psi)$, $F_3 = uc_\theta c_\phi$. Here (4.3a) and (4.3b) form the outer-loop dynamics, where as (4.3c) and (4.3d) are the inner-loop dynamics. The UAM-V thrust input u, roll ϕ and pitch θ are treated as the inputs to the outer-loop subsystem and can be calculated from components of the force vector F using following equations.

$$\begin{aligned} u &= \sqrt{F_1^2 + F_2^2 + F_3^2}, \\ \phi &= \arcsin \frac{\mathrm{s}_\psi F_1 - \mathrm{c}_\psi F_2}{u}, \\ \theta &= \arctan \frac{\mathrm{c}_\psi F_1 + \mathrm{s}_\psi F_2}{F_3}. \end{aligned} \tag{4.4}$$

The inner-loop control inputs are UAM-V torques τ_q. It is reiterated that parameters

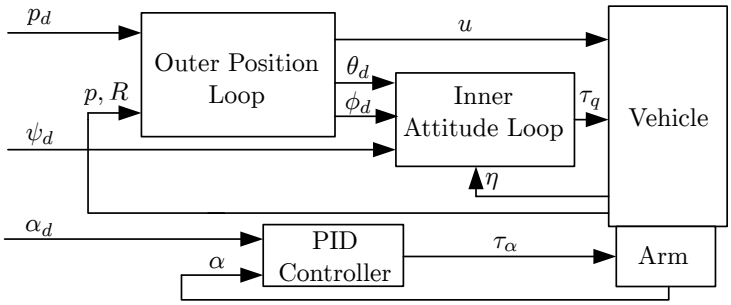


Figure 4.2: Inner-outer loop control structure

a, d_f, J, d_τ are treated as unknowns constants that are adapted in the control design to accommodate the motion of robotic arm. The control structure is shown in Figure 4.2. We use a simple PID control to control the arm position described by its generalized coordinates α.

4.2.2 Outer-loop Control

Let us define the position error as $z_1 = p - p_d$, its time derivative is given by

$$\dot{z}_1 = \dot{p} - \dot{p}_d = v - v_d. \tag{4.5}$$

Consider the Lyapunov function as $V_1 = \frac{1}{2} z_1^T z_1$, whose time-derivative after substitution of (4.5) is given by

$$\dot{V}_1 = z_1^T \dot{z}_1 = z_1^T (v - v_d).$$

Adding and subtracting $k_1 z_1^T z_1$ in above equation, we have

$$\dot{V}_1 = -k_1 \left\| z_1 \right\|^2 + z_1^T (v - v_d + k_1 z_1). \tag{4.6}$$

Let us define $z_2 = v - v_d + k_1 z_1$ which allows us to rewrite (4.5) as follows

$$\dot{z}_1 = v - v_d = -k_1 z_1 + z_2, \tag{4.7}$$

while the time-derivative of z_2 after substituting (4.3b) is given by

$$\dot{z}_2 = g e_3 - aF + d_f - \dot{v}_d + k_1 (v - v_d). \tag{4.8}$$

Since the parameters a and disturbance d_f are unknown, let us consider $\hat{a} \in \mathbb{R} > 0$ and $\hat{d}_f \in \mathbb{R}^3$ as their respective estimate, that will be determined later. Consider the estimation errors for a and d_f as $\tilde{a} = a - \hat{a}$ and $\tilde{d}_f = d_f - \hat{d}_f$, let us substitute

the values a and d_f in terms of their estimate and estimation errors i.e., $a = \hat{a} + \tilde{a}$ and $d_f = \hat{d}_f + \tilde{d}_f$ along with expression for $v - v_d$ in terms on z_1 and z_2 from (4.7), we have

$$\dot{z}_2 = ge_3 - \hat{a}F + \hat{d}_f - \dot{v}_d - k_1^2 z_1 + k_1 z_2 - \tilde{a}F + \tilde{d}_f. \tag{4.9}$$

Recall $\dot{V}_1$ from (4.6), after substitution of the definition of z_2, it can be written as follows

$$\dot{V}_1 = -k_1 \|z_1\|^2 + z_2^T z_1. \tag{4.10}$$

Consider a second Lyapunov function as

$$V_2 = V_1 + \frac{1}{2} z_2^T z_2 + \frac{1}{2k_{d_f}} \tilde{d}_f^T \tilde{d}_f + \frac{1}{2\lambda} \tilde{a}^2. \tag{4.11}$$

Taking its time-derivative, we have

$$\dot{V}_2 = \dot{V}_1 + z_2^T \dot{z}_2 + \frac{1}{k_{d_f}} \tilde{d}_f^T \dot{\tilde{d}}_f + \frac{1}{\lambda} \tilde{a} \dot{\tilde{a}}. \tag{4.12}$$

Substituting value of $\dot{V}_1$ from (4.10) and $\dot{z}_2$ from (4.9), we have

$$\begin{aligned} \dot{V}_2 = & - k_1 \|z_1\|^2 + z_1^T z_2 + z_2^T (ge_3 - \hat{a}F + \hat{d}_f - \dot{v}_d - k_1^2 z_1 + k_1 z_2 - \tilde{a}F + \tilde{d}_f) \\ & + \frac{1}{k_{d_f}} \tilde{d}_f^T \dot{\tilde{d}}_f + \frac{1}{\lambda} \tilde{a} \dot{\tilde{a}}. \end{aligned}$$

Collecting terms containing $\tilde{d}_f$ and $\tilde{a}$, we have

$$\begin{aligned} \dot{V}_2 = & - k_1 \|z_1\|^2 + z_2^T (z_1 + ge_3 - \hat{a}F + \hat{d}_f - \dot{v}_d - k_1^2 z_1 + k_1 z_2) \\ & + \tilde{d}_f^T (\frac{1}{k_{d_f}} \dot{\tilde{d}}_f + z_2) + \tilde{a} (\frac{1}{\lambda} \dot{\tilde{a}} - z_2^T F). \end{aligned} \tag{4.13}$$

Consider the following adaptation laws

$$\begin{aligned} \dot{\hat{a}} = & - \lambda z_2^T F, \\ \dot{\hat{d}}_f = & k_{d_f} z_2. \end{aligned} \tag{4.14}$$

Substituting these adaptation laws, along with assumption $\dot{a} = 0$ and $\dot{d}_f = 0$, in (4.13), we have

$$\dot{V}_2 = -k_1 \|z_1\|^2 + z_2^T (ge_3 - \hat{a}F + \hat{d}_f - \dot{v}_d + (1 - k_1^2) z_1 + k_1 z_2).$$

Adding and subtracting $k_2 z_2^T z_2$ in above equation, we have

$$\dot{V}_2 = -k_1 \|z_1\|^2 - k_2 \|z_2\|^2 + z_2^T (ge_3 - \hat{a}F + \hat{d}_f - \dot{v}_d + (1 - k_1^2) z_1 + (k_1 + k_2) z_2). \tag{4.15}$$

From this equation, we can design the force vector F as follows

$$F = \frac{1}{\hat{a}}(ge_3 + \hat{d}_f - \dot{v}_d + (1 - k_1^2)z_1 + (k_1 + k_2)z_2). \tag{4.16}$$

Substituting this into $\dot{V}_2$ from (4.15), we have

$$\dot{V}_2 = -k_1 \|z_1\|^2 - k_2 \|z_2\|^2 \leq 0, \tag{4.17}$$

The expression for $\dot{V}_2$ is negative semi-definite because for $\tilde{a} \neq 0$ and $\tilde{d_f} \neq 0$, $\dot{V}_2 = 0$ when $z_1 = z_2 = 0$.

Theorem 4.2.1. *Consider the outer-loop dynamics of UAM-V consisting of* (4.3a) *and* (4.3b) *under the assumption that* $\dot{d}_f = 0$ *and* $\dot{a} = 0$*, with a smooth and bounded time-varying trajectory consisting of* $\{p_d(t), v_d(t), \dot{v}_d(t)\}$*. Assuming a perfect inner-loop tracking, if the UAM-V thrust* u*, and references for roll* ϕ *and pitch* θ *angles are selected according to* (4.4)*, where the vector* F *is given by* (4.16) *with gains* $k_1 > 0, k_2 > 0$*, and estimates* $\hat{a}$ *and* $\hat{d}_f$ *are given by the adaptive laws in* (4.14) *with gains* $\lambda > 0, k_{d_f} > 0$*, then the tracking error* $z_1 = p - p_d$ *is asymptotically stable at origin while the estimates* $\hat{a}$ *and* $\hat{d}_f$ *are bounded.*

Proof. Consider the Lyapunov function V_2 in (4.11) which is a positive definite function under $k_1 > 0, k_2 > 0, \lambda > 0, k_{d_f} > 0$, and there exist two positive definite functions that upper and lower bound V_2. The derivative of V_2 after substitution of control and adaptation laws, given in (4.17) is negative semi-definite due to absence $\tilde{d_f}$ and $\tilde{a}$ terms. However, using Barbalat's lemma [184, Theorem 8.4], we can conclude that the vector $[z_1^T, z_2^T]^T \to 0$ as $t \to \infty$ and the estimates $\hat{a}$ and $\hat{d}_f$ are bounded. □

Remark 1. *It is worth mentioning that using this adaptive approach,* $[z_1^T, z_2^T]^T \to 0$*, while* $\hat{a}, \hat{d}_f$ *estimate are not guaranteed to converge to their actual values. From a control perspective, the values attained by* $\hat{a}, \hat{d}_f$ *are not important, as long as they are bounded and the tracking error is convergent. Therefore, convergence of estimates* $\hat{a}, \hat{d}_f$ *to their actual values is not a requirement.*

Remark 2. *If we compare the outer-loop control presented above with that presented for IBVS in Chapter 2, we notice that the IBVS outer-loop provides exponential stability of feature errors together with the estimation of state and disturbance whereas the approach in this chapter provides only an asymptotic stability of position tracking errors while all states are required to be known. This limitation arises due to the tracking of a time-varying trajectory. In case of IBVS, only the stabilization of the feature errors is required, that allow us to combine the unknown parameters with the disturbance term and design a controller where gain can be selected sufficiently high*

to provide robustness to variation in unknown parameters. However, for tracking a time-varying trajectory, the term $\dot{v}_d$ in the error dynamics $\dot{z}_2$ prevent us from combing the unknown mass with the disturbance d_f. If we consider $k_1 = \bar{k}_1 a$ in (4.6) and define $ge_3 + d_f = D_f$ in (4.8) then we can write

$$\begin{aligned}\dot{z}_2 =& D_f - aF - \dot{v}_d + \bar{k}_1 a(v - v_d) \\ =& a(\frac{1}{a}D_f - F - \frac{1}{a}\dot{v}_d + \bar{k}_1(v - v_d)).\end{aligned} \tag{4.18}$$

This equation shows that if we try to make the term F independent of the mass parameter a and combine it with D_f, it shows up with $\dot{v}_d$. If $\dot{v}_d = 0$, then the mass parameter a only appears with D_f which is already being estimated and we can estimate $\frac{1}{a}D_f$ instead of D_f, while the control can be made robust to the mass parameter a outside the parenthesis. Therefore, it is the time-varying trajectory tracking prevents the robustness.

Also, it is worth mentioning that the outer-loop tracking result presented here is stronger than that presented for the IBVS in case of a time-varying reference trajectory. If IBVS outer-loop control is used for a time-varying trajectory, it would only result in the bounded tracking error considering an input-to-state stability of an exponentially stable linear system under a bounded input $\dot{v}_d$. However, the outer-loop control presented here guarantees an asymptotic convergence of tracking errors for a time-varying trajectory.

4.2.3 Exponential stability and parameter convergence

The traditional back-stepping design results in a state transformation that makes the closed-loop dynamics of the system linear. In the adaptive backstepping design above, the non-linearity is introduced by the parameter estimation required to compensate for unknown parameters. Using (4.7), (4.9), (4.16) and (4.14) we can write the closed loop dynamics of the system as follows

$$\begin{aligned}\dot{z}_1 &= -k_1 z_1 + z_2, \\ \dot{z}_2 &= -z_1 - k_2 z_2 - \tilde{a}F + \tilde{d}_f, \\ \dot{\tilde{d}}_f &= -k_{d_f} z_2, \\ \dot{\tilde{a}} &= \lambda {z_2}^T F.\end{aligned} \tag{4.19}$$

Here to keep the presentation concise the input F is substituted with the terms containing $\hat{a}$ and left with the $\tilde{a}$ term. We know from Theorem 4.2.1 that the tracking errors $[z_1^T, z_2^T]$ are asymptotically stable while the estimation errors $[\tilde{d}_f^T, \tilde{a}]$ are bounded. If we substitute $z_1 = z_2 = 0$ in (4.19), it results in $\tilde{a}F = \tilde{d}_f$. This does not provide additional information except that if $\tilde{a} = 0$, then $\tilde{d}_f = 0$ as well.

To investigate parameter convergence, we write the first three equations of (4.19) as a reduced order linear system

$$\dot{x}_{r_1} = A_{r_1} x_{r_1} + B_{r_1} u_{r_1}, \tag{4.20}$$

where $x_{r_1} = [z_1^T, z_2^T, \tilde{d}_f^T]^T$, $u_{r_1} = -\tilde{a}F$,

$$A_{r_1} = \begin{bmatrix} -k_1 I_3 & I_3 & 0_3 \\ -I_3 & -k_2 I_3 & I_3 \\ 0_3 & -k_{d_f} I_3 & 0_3 \end{bmatrix}, \quad B_{r_1} = \begin{bmatrix} 0_3 \\ I_3 \\ 0_3 \end{bmatrix},$$

while I_3 is a 3×3 identity matrix and 0_3 is matrix of zeros with size 3×3. The system matrix A_{r_1} is Hurwitz if $k_1 > 0$, $k_2 > 0$ and $K_{d_f} > 0$.

If we consider UAM mass known, $\hat{a} = a$ can be substituted in the control resulting in $u_{r_1} = 0$. This results in exponential convergence of $\tilde{d}_f$ and $[z_1^T, z_2^T]$ to zero. This exponential stability is equivalent to the exponential convergence result in the case of disturbance observer based backstepping control that assumes mass is known [1]. If we consider a unknown, we expect the term $\tilde{a}F$ to restrict the convergence of $[z_1^T, z_2^T]$ to a bound, considering it as a bounded input to a linear system that results in bounded-input-bounded-state (BIBS). However, from Theorem 4.2.1, we know that it does not effect the convergence of $[z_1^T, z_2^T]$ to the origin, but results in a bounded $\tilde{d}_f$. Hence, it is important to note that term $\tilde{a}F$ does not destabilize the system.

From the above discussion we know that $\tilde{a}$ and $\tilde{d}_f$ may or may not converge to the origin. In adaptive control the convergence is usually on a need-to-know basis as the main goal is the convergence of tracking errors. Once the tracking errors are close to the equilibrium, the adaptation or learning is negligibly small. However, when the exponential convergence of the tracking errors is desired to achieve robustness guarantees such as BIBS. This can be achieved with persistence of excitation (PE) [185, Theorem 3.5.1]. PE provides exponential convergence of the tracking and estimation errors. To derive the PE condition, we rewrite (4.19) as

$$\begin{bmatrix} \dot{z} \\ \dot{\zeta} \end{bmatrix} = \begin{bmatrix} A_z & C\mu^T(t) \\ -K\mu(t)C^T & 0 \end{bmatrix} \begin{bmatrix} z \\ \tilde{\zeta} \end{bmatrix}, \tag{4.21}$$

where $z = [z_1^T, z_2^T]^T$, $\mu(t) = [I_3, -F]^T$, $\zeta = [\tilde{d}_f^T, \tilde{a}]^T$, $C = [0_3, I_3]^T$,

$$A_z = \begin{bmatrix} -k_1 I_3 & I_3 \\ -I_3 & -k_2 I_3 \end{bmatrix}, \quad K = \begin{bmatrix} k_{d_f} I_3 & 0 \\ 0 & \lambda \end{bmatrix}.$$

The function $\mu(t)$ is known as a regressor. From [186], the condition for PE of this

system is

$$\int_{t_0}^{t+t_0} \mu(\delta)\mu^T(\delta)d\delta \geq \beta I_4, \quad t > 0, \beta > 0. \tag{4.22}$$

Intuitively, the PE condition ensures the only solution to $\mu^T(t)\tilde{\zeta} = [I_3, -F][\tilde{d}_f^T, \tilde{a}]^T = 0$ is $\tilde{\zeta} = [\tilde{d}_f^T, \tilde{a}]^T = 0$. This can be be seen from

$$\int_{t_0}^{t+t_0} \mu(\delta)\mu^T(\delta)d\delta\tilde{\zeta} = 0, \tag{4.23}$$

whose only solution is $\tilde{\zeta} = 0$ due to (4.22). From the definition of $\mu(t)$ the PE condition reduces to

$$\int_{t_0}^{t+t_0} F(\delta)F^T(\delta)d\delta \geq \beta I_3, \quad t > 0. \tag{4.24}$$

Although, F is dependent on system states, parameter estimates, and the desired trajectory, it is the desired or reference trajectory that is the ultimate driver of all dynamical variables of the system. Intuitively, the persistent excitation condition requires the reference trajectory to be of sufficient richness or complexity that it can only be tracked by the controller when the system parameter estimates have converged to their true values.

4.2.4 Effect of time-varying disturbance d_f and mass parameter a

Let us first consider the effect of time-varying force disturbance d_f. We write the closed loop dynamics in case of $\dot{d}_f \neq 0$ from (4.20) as

$$\dot{x}_{r_2} = A_{r_2}x_{r_2} + B_{r_2}u_{r_2} \tag{4.25}$$

where $x_{r_2} = [z_1^T, z_2^T, \tilde{d}_f^T]^T$, $u_{r_2} = [\dot{d}_f^T, -\tilde{a}F^T]^T$,

$$A_{r_2} = \begin{bmatrix} -k_1 I_3 & I_3 & 0_3 \\ -I_3 & -k_2 I_3 & I_3 \\ 0_3 & -k_{d_f} I_3 & 0_3 \end{bmatrix}, \quad B_{r_2} = \begin{bmatrix} 0_3 & 0_3 \\ 0_3 & I_3 \\ I_3 & 0_3 \end{bmatrix},$$

As discussed in the previous section in relation to (4.20), the input $\tilde{a}F$ does not prevent the convergence of the tracking errors $[z_1^T, z_2^T]$ to the origin. The only difference between (4.20) and (4.25) is that we have an additional input $\dot{d}_f$. The system matrix A_{r_2} is Hurwitz if $k_1 > 0$, $k_2 > 0$ and $K_{d_f} > 0$. Therefore, the system

is BIBS and a bound on the system state is

$$\begin{aligned}\|x_{r_2}(t)\| =& x_{r_2}(t_0)e^{A_{r_2}(t-t_0)} + \int_{t_0}^{t} e^{A_{r_2}(t-t_0)} B_{r_2} u_{r_2}(\delta) d\delta \\ \leq& \|x_{r_2}(t_0)\| k e^{-\sigma(t-t_0)} + k\frac{\|B_{r_2}\|}{\sigma} \sup_{t_0\leq\delta\leq t} \|u_{r_2}(\delta)\|,\end{aligned} \tag{4.26}$$

where $-\sigma$ is the real part of the largest eigenvalue of A_{r_2}. If we assume $\tilde{a}F$ and $\dot{d}_f$ are bounded then $x_{r_2}(t)$ is bounded. Further, the bound on the system can be made smaller by selecting a larger σ with control gains k_1, k_2 and k_{d_f}.

Let us now consider the effect of time-varying mass parameter a and the time-varying force disturbance d_f. If $\dot{d}_f \neq 0$ and $\dot{a} \neq 0$, the system in (4.21) can be written as

$$\dot{x} = Ax + Bu_d, \tag{4.27}$$

where $x = [z^T, \tilde{\zeta}^T]^T$, $u_d = [\dot{d}_f^T, \dot{a}]^T$,

$$A = \begin{bmatrix} A_z & C\mu^T(t) \\ -K\mu(t)C^T & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0_3 & 0 \\ 0_3 & 0 \\ I_3 & 0 \\ 0 & 1 \end{bmatrix}.$$

We know from previous section that the system in (4.27) is exponentially convergent under the PE condition. Exponential convergence implies the existence of an exponential rate of convergence. In the above equation we have an exponentially stable system perturbed by input u_d. Using the BIBS framework we obtain the bound

$$\|x(t)\| \leq \|x(t_0)\| k e^{-\sigma_{pe}(t-t_0)} + k\frac{\|B\|}{\sigma_{pe}} \sup_{t_0\leq\delta\leq t} \|u_d(\delta)\|, \tag{4.28}$$

where σ_{pe} is a minimum rate of exponential convergence. If $\dot{d}_f$ and $\dot{a}$ are bounded, the resulting x is bounded. Further, the bound can be decreased by increasing σ_{pe} which depends on the controller gains. The effect of time-varying d_f and a is further studied in simulation below.

Figure 4.3 shows the plots for the norm of position error. These plots are generated while the vehicle is tracking a figure-8 trajectory and a time-varying triangular wave is applied in the form of force disturbance d_f, given by

$$d_f(t) = 2|(t - 2\ \text{floor}(0.5t + 0.5))| \cdot [1, 1, 1]^T + [0, -3, -2]^T. \tag{4.29}$$

The vehicle mass is kept constant during this simulation. The plots are generated for different values of gains. It can be seen that the bound on the position error decreases as the gains are increased.

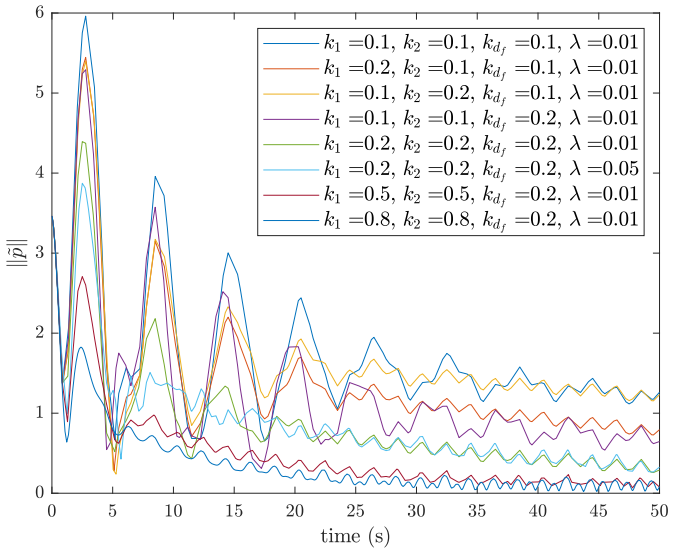


Figure 4.3: Bounded $\|\tilde{p}\|$ in the presence of time-varying d_f.

Let us now consider the effect of both a and d_f time-varying. We consider same triangular function for the disturbance d_f, while a sinusoidal variation in mass is considered.

$$a(t) = \frac{1}{m(t)} = \frac{1}{1 + 0.5\sin(\frac{2\pi t}{6})}. \tag{4.30}$$

The resulting plots for the norm of position error are shown in Figure 4.4. A figure-8 trajectory is used as reference to obtain these simulation plots. The controller gains are varied and it can be seen that the vehicle position error is bounded when both mass and the force disturbance are considered time-varying. Further, the bound can be made smaller by selecting appropriate values of gains in control and adaptation laws.

4.3 Inner-loop Control

As mentioned earlier, the inner-loop receives a reference trajectory $\eta_d = [\phi_d, \theta_d, \psi_d]^T$ for UAM-V attitude whereas ϕ_d and θ_d are provided by outer-loop controller while ψ_d is given by the user as a reference heading angle. The purpose of the inner-loop control is to make the UAM-V attitude track the reference attitude trajectory. Let us now start the inner-loop control design by defining the inner-loop tracking error $z_\eta = \eta - \eta_d$, taking its time-derivative and substituting (4.3c), we have

$$\dot{z}_\eta = \dot{\eta} - \dot{\eta}_d = W^{-1}\omega - \dot{\eta}_d. \tag{4.31}$$

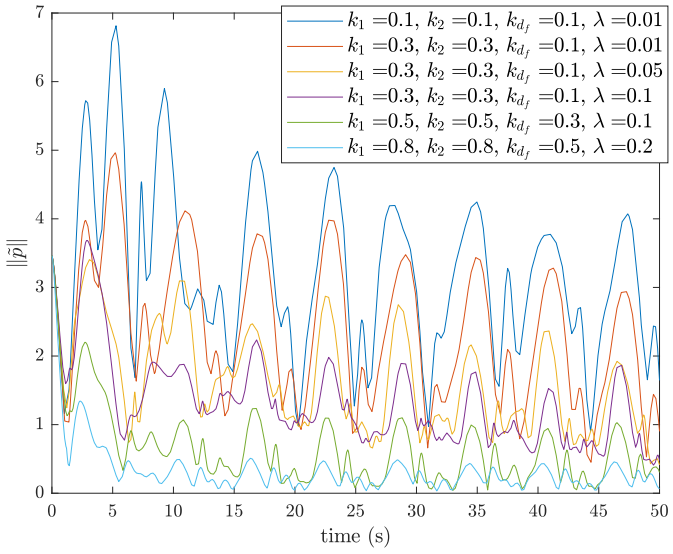


Figure 4.4: Bounded $\|\tilde{p}\|$ in the presence of time-varying d_f and a.

Consider the following Lyapunov function

$$V_\eta = \frac{1}{2} z_\eta^T z_\eta.$$

Its time-derivative is given by

$$\dot{V}_\eta = z_\eta^T \dot{z}_\eta.$$

Adding and subtracting $k_\eta z_\eta^T z_\eta$, and substituting $\dot{z}_\eta$ from (4.31), we have

$$\dot{V}_\eta = -k_\eta z_\eta^T z_\eta + z_\eta^T (W^{-1}\omega - \dot{\eta}_d + k_\eta z_\eta).$$

Let us define $z_\omega = \omega - W(\dot{\eta}_d - k_\eta z_\eta)$, substituting in above equation, we have

$$\dot{V}_\eta = -k_\eta z_\eta^T z_\eta + z_\eta^T W^{-1} z_\omega, \tag{4.32}$$

while we can rewrite $\dot{z}_\eta$ in (4.31) as

$$\dot{z}_\eta = W^{-1} z_\omega - k_\eta z_\eta, \tag{4.33}$$

whereas the derivative of z_ω is given by

$$\dot{z}_\omega = \dot{\omega} - \dot{W}(\dot{\eta}_d - k_\eta z_\eta) - W(\ddot{\eta}_d - k_\eta \dot{z}_\eta).$$

Substituting $\dot{z}_\eta$ from (4.33) and simplifying, we have

$$\dot{z}_\omega = \dot{\omega} - \dot{W}(\dot{\eta}_d - k_\eta z_\eta) - W(\ddot{\eta}_d + k_\eta^2 z_\eta) + k_\eta z_\omega. \tag{4.34}$$

Consider another Lynapunov function candidate

$$V_\omega = V_\eta + \frac{1}{2} z_\omega^T J z_\omega.$$

whose time-derivative after substitution of $\dot{V}_\eta$ from (4.32) is given by

$$\dot{V}_\omega = -k_\eta z_\eta^T z_\eta + z_\eta^T W^{-1} z_\omega + z_\omega^T J \dot{z}_\omega.$$

Adding and subtracting $k_\omega z_\omega^T z_\omega$ in above equation, we have

$$\dot{V}_\omega = -k_\eta z_\eta^T z_\eta - k_\omega z_\omega^T z_\omega + z_\omega^T (k_\omega z_\omega + W^{-T} z_\eta + J \dot{z}_\omega).$$

Substituting expression for $\dot{z}_\omega$ from (4.34), we have

$$\begin{aligned}\dot{V}_\omega = &- k_\eta z_\eta^T z_\eta - k_\omega z_\omega^T z_\omega + z_\omega^T \Big(k_\omega z_\omega + W^{-T} z_\eta \\ &+ J\big(\dot{\omega} - \dot{W}(\dot{\eta}_d - k_\eta z_\eta) - W(\ddot{\eta}_d + k_\eta^2 z_\eta) + k_\eta z_\omega\big)\Big).\end{aligned}$$

Substituting dynamics for $J\dot{\omega}$ from (4.3d), we have

$$\begin{aligned}\dot{V}_\omega = &- k_\eta z_\eta^T z_\eta - k_\omega z_\omega^T z_\omega + z_\omega^T \Big(k_\omega z_\omega + W^{-T} z_\eta - \mathrm{S}(\omega) J\omega + \tau_q + d_\tau \\ &- J\big(\dot{W}(\dot{\eta}_d - k_\eta z_\eta) + W(\ddot{\eta}_d + k_\eta^2 z_\eta) - k_\eta z_\omega\big)\Big).\end{aligned}$$

Let us define $\bar{\omega}_d = \dot{W}(\dot{\eta}_d - k_\eta z_\eta) + W(\ddot{\eta}_d + k_\eta^2 z_\eta) - k_\eta z_\omega$, using this in above, we have

$$\dot{V}_\omega = -k_\eta z_\eta^T z_\eta - k_\omega z_\omega^T z_\omega + z_\omega^T \Big(k_\omega z_\omega + W^{-T} z_\eta - \mathrm{S}(\omega) J\omega + \tau_q + d_\tau - J\bar{\omega}_d \Big) \tag{4.35}$$

The inertia matrix J is a positive definite symmetric matrix, and it has only six unique entries. If we consider

$$J = \begin{bmatrix} J_{xx} & J_{xy} & J_{zx} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{zx} & J_{yx} & J_{zz} \end{bmatrix},$$

then we can write

$$\begin{aligned}\mathrm{S}(\omega) J\omega &= \Phi(\omega) J_v, \\ J\bar{\omega}_d &= \Psi(\bar{\omega}_d) J_v,\end{aligned} \tag{4.36}$$

where

$$\Phi(\omega) = \begin{bmatrix} 0 & -\omega_2\omega_3 & \omega_2\omega_3 & -\omega_1\omega_3 & \omega_2^2 - \omega_3^2 & \omega_1\omega_2 \\ \omega_1\omega_3 & 0 & -\omega_1\omega_3 & \omega_2\omega_3 & -\omega_1\omega_2 & \omega_3^2 - \omega_1^2 \\ -\omega_1\omega_2 & \omega_1\omega_2 & 0 & \omega_1^2 - \omega_2^2 & \omega_1\omega_3 & -\omega_2\omega_3 \end{bmatrix},$$

$$\Psi(\bar{\omega}_d) = \begin{bmatrix} \bar{\omega}_{d1} & 0 & 0 & \bar{\omega}_{d2} & 0 & \bar{\omega}_{d3} \\ 0 & \bar{\omega}_{d2} & 0 & \bar{\omega}_{d1} & \bar{\omega}_{d3} & 0 \\ 0 & 0 & \bar{\omega}_{d3} & 0 & \bar{\omega}_{d2} & \bar{\omega}_{d1} \end{bmatrix},$$

and $J_v = [J_{xx}, J_{yy}, J_{zz}, J_{xy}, J_{yz}, J_{zx}]^T$. Substituting (4.36) in $\dot{V}_\omega$ expression from (4.35), we have

$$\begin{aligned} \dot{V}_\omega = -k_\eta z_\eta^T z_\eta - k_\omega z_\omega^T z_\omega + z_\omega^T (k_\omega z_\omega + W^{-T} z_\eta \\ -(\Psi(\bar{\omega}_d) + \Phi(\omega)) J_v + \tau_q + d_\tau). \end{aligned} \tag{4.37}$$

Since J_v and d_τ are unknown constants, we define $\hat{J}_v \in \mathbb{R}^6$ and $\hat{d}_\tau \in \mathbb{R}^3$ as their respective estimates, to be determined later, and, $\tilde{J}_v = J_v - \hat{J}_v$ and $\tilde{d}_\tau = d_\tau - \hat{d}_\tau$ as their respective estimation errors. Let us substitute $J_v = \hat{J}_v + \tilde{J}_v$ and $d_\tau = \hat{d}_\tau + \tilde{d}_\tau$ in (4.37), we have

$$\begin{aligned} \dot{V}_\omega = &- k_\eta z_\eta^T z_\eta - k_\omega z_\omega^T z_\omega + z_\omega^T \Big(k_\omega z_\omega + W^{-T} z_\eta \\ &- \big(\Psi(\bar{\omega}_d) + \Phi(\omega)\big)(\hat{J}_v + \tilde{J}_v) + \tau_q + \hat{d}_\tau + \tilde{d}_\tau \Big). \end{aligned}$$

Rearranging, we have

$$\begin{aligned} \dot{V}_\omega = &- k_\eta z_\eta^T z_\eta - k_\omega z_\omega^T z_\omega + z_\omega^T \Big(k_\omega z_\omega + W^{-T} z_\eta - (\Psi(\bar{\omega}_d) + \Phi(\omega)) \hat{J}_v \\ &+ \hat{d}_\tau + \tau_q \Big) - z_\omega^T (\Psi(\bar{\omega}_d) + \Phi(\omega)) \tilde{J}_v + z_\omega^T \tilde{d}_\tau. \end{aligned} \tag{4.38}$$

Consider a new Lyapunov function,

$$V_{\omega,2} = V_\omega + \frac{1}{2} \tilde{J}_v^T \Gamma^{-1} \tilde{J}_v + \frac{1}{2} \tilde{d}_\tau^T \Upsilon \tilde{d}_\tau. \tag{4.39}$$

Its time derivative is given by

$$\dot{V}_{\omega,2} = \dot{V}_\omega + \tilde{J}_v^T \Gamma^{-1} \dot{\tilde{J}}_v + \tilde{d}_\tau^T \Upsilon \dot{\tilde{d}}_\tau.$$

Substituting the expression for $\dot{V}_\omega$ from (4.38)

$$\begin{aligned} \dot{V}_{\omega,2} = &- k_\eta z_\eta^T z_\eta - k_\omega z_\omega^T z_\omega + z_\omega^T \Big(k_\omega z_\omega + W^{-T} z_\eta - (\Psi(\bar{\omega}_d) + \Phi(\omega)) \hat{J}_v \\ &+ \hat{d}_\tau + \tau_q \Big) - z_\omega^T (\Psi(\bar{\omega}_d) + \Phi(\omega)) \tilde{J}_v + z_\omega^T \tilde{d}_\tau + \tilde{J}_v^T \Gamma^{-1} \dot{\tilde{J}}_v + \tilde{d}_\tau^T \Upsilon^{-1} \dot{\tilde{d}}_\tau. \end{aligned} \tag{4.40}$$

Consider the following control input

$$\tau_d = -k_\omega z_\omega - W^{-T} z_\eta + (\Psi(\bar{\omega}_d) + \Phi(\omega))\hat{J}_v - \hat{d}_\tau. \tag{4.41}$$

Substituting this in (4.40) and collecting coefficients of $\tilde{J}_v^T$ and $\tilde{d}_\tau^T$, we have

$$\dot{V}_{\omega,2} = -k_\eta z_\eta^T z_\eta - k_\omega z_\omega^T z_\omega - \tilde{J}_v^T((\Psi(\bar{\omega}_d)+\Phi(\omega))z_\omega - \Gamma^{-1}\dot{\hat{J}}_v) + \tilde{d}_\tau^T(z_\omega + \Upsilon^{-1}\dot{\hat{d}}_\tau). \tag{4.42}$$

Consider the following adaptation laws

$$\begin{aligned} \dot{\hat{J}}_v &= -\Gamma(\Psi(\bar{\omega}_d) + \Phi(\omega))z_\omega, \\ \dot{\hat{d}}_\tau &= \Upsilon z_\omega. \end{aligned} \tag{4.43}$$

Substituting these along with the assumption that J_v and d_τ are constants, i.e., $\dot{J}_v = 0$ and $\dot{d}_\tau = 0$, in (4.40) we have

$$\dot{V}_{\omega,2} = -k_\eta z_\eta^T z_\eta - k_\omega z_\omega^T z_\omega, \tag{4.44}$$

which is semi-negative definite because $\tilde{J}_v$ and $\tilde{d}_\tau$ do not appear in above equation.

Theorem 4.3.1. *Consider the inner-loop dynamics of a UAM-V consisting of* (4.3c) *and* (4.3d) *under the assumption that* $\dot{d}_\tau = 0$ *and* $\dot{J}_v = 0$*, with a smooth time-varying trajectory consisting of* $\{\eta_d(t), \dot{\eta}_d(t), \ddot{\eta}_d(t)\}$*. If the UAM-V torque inputs are selected according to* (4.41) *with gains* $k_\eta > 0, k_\omega > 0$*, while the estimates* $\hat{d}_\tau$ *and* $\hat{J}_v$ *are given by the adaptive laws in* (4.43) *with gain matrices* $\Gamma > 0, \Upsilon > 0$*, then the tracking error* $z_\eta = \eta - \eta_d$ *is asymptotically stable at origin while the estimates* $\hat{d}_\tau$ *and* $\hat{J}_v$ *are bounded.*

Proof. Consider the Lyapunov function $V_{\omega,2}$ in (4.39) which is a positive definite function under $k_\eta > 0, k_\omega > 0, \Gamma > 0, \Upsilon > 0$, and its derivative after substitution of control and adaptation laws, given in (4.44), is semi-negative definite due to absence $\tilde{d}_\tau$ and $\tilde{J}_v$ term. However, using Barbalat's lemma in [184, Theorem 8.5] we can conclude that the vector $[z_\eta^T, z_\omega^T]^T \to 0$ as $t \to \infty$ while the parameter estimates $\hat{d}_\tau$ and $\hat{J}_v$ are bounded. □

It should be noted that the inner-loop control requires the first two derivatives of the reference attitude trajectory. We will see in the next chapter that an algebraic derivative leads to the appearance of unknown terms, which are difficult to manage in the stability proof. Therefore, numerical derivatives are recommended. The noise amplification effects due to these derivatives can be reduced using a low-pass filter.

4.4 Simulation Results

Computer simulations provide important insight into the dynamic behaviour of a system in response to a control algorithm. They are an essential tool in testing new control designs and provide flexibility in generating ideal and non-ideal scenarios. Although simulations do not replace the practical implementation and testing on a physical platform, they allow performing theoretical analysis, which is otherwise difficult in the case of physical experiments. Simulations are instrumental in Aerial Robotics, where frequent crashes could lead to delays and reduce research productivity.

Computer simulations of mechanical systems could be performed in two different ways, depending upon how the physical system is modelled in the computer. One way to represent is to implement the dynamic equations from the mathematical model of the system. This approach is simplistic and does not provide a visual insight into the behaviour of the mechanical system, which is important especially in case of robotic systems. The other approach is the multi-body approach, where the CAD model of the mechanical system is created, and multiple rigid bodies are connected with various joints. The user provides the inputs at the specified joints of the system. Here, the mathematical model of the system is not implemented; instead, the physics engine of the simulation software automatically figures out the dynamic behaviour of the whole system depending upon the properties of the individual bodies in the system and their interconnection. The multi-body approach is more elegant and closer to the actual physical system. It allows visualization of system behavior through an animation.

There are various tools used in academia and industry for multi-body simulation. The most commonly used simulation environment among the robotics community is Gazebo, which is used in combination with ROS. In an earlier phase of this research, a UAM model was implemented in Gazebo. However, due to its open-source nature, less support is available for troubleshooting in Gazebo. In particular, Gazebo does not provide an easy interface for measuring states and controlling the system's joints. Also, any change in the model requires a significant amount of development work. Due to this, we moved to MATLAB's Simscape multi-body and the result presented below are implemented using this platform. Simscape multi-body allows modelling 3D mechanical systems using various blocks representing rigid bodies, joints, input forces and torques, and sensor elements. Simscape generates a visual 3D animation of the system that shows the UAM's motion in its environment, much like in the real world. Simscape models are user-friendly and easier to configure. Figure 4.5 shows a UAM i.e., Quadrotor with 2-DOF robotic arm having two revolute joints, picking up a red coloured object from the table.

The quadrotor UAV design used in the UAM simulations is based on ANCLQ

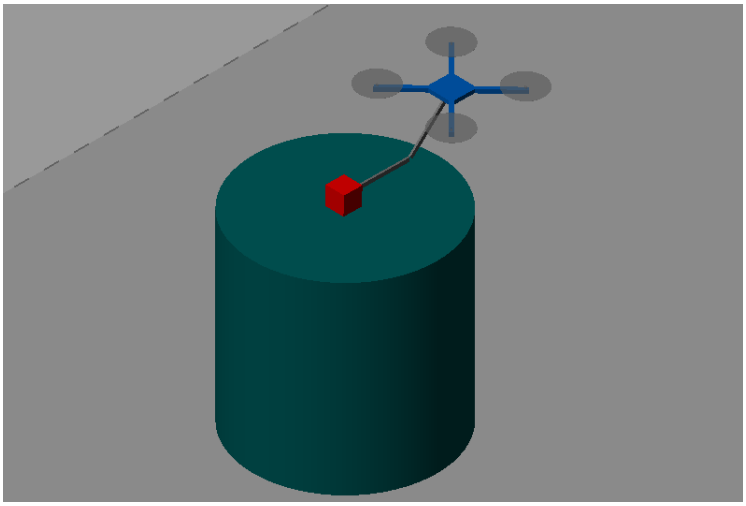


Figure 4.5: Implementation of a Quadrotor UAV with a 2-DOF robot arm in Simscape Multibody

1.0 experimental platform, and its parameters are given in Table 4.1. The arm in this UAM configuration is considered to be a 2-DOF arm with two revolute joints as shown in Figure 4.1. The axis of rotation of each revolute joint is parallel to b_2, which is perpendicular to the plane of paper and points out of the paper, as in Figure 4.1. The arm angles α_1 and α_2 are measured in the counterclockwise direction. The negative sign with α_1 in Figure 4.1 is due to clockwise rotation of the Link 1. The parameters of the arm are given in Table 4.2. Here Link 1 and Link 2 describe the parameters of the two links of the arm, while Link 3 describes a mass that is used in the simulations for a pick and place application. The external mass is defined as a link to be consistent with the arm modelling in Section 2.2.4.

To test the performance of the UAM-V control for its ability to track a time-varying trajectory in the presence of a robotic arm, we consider two scenarios. In the first scenario, the arm is fixed to the UAM-V where the angles $\alpha_1 = \alpha_2 = 0$ and locked. Although the disturbances on the UAM-V due to arm in this configuration are not constant due to the roll and pitch of the UAM-V, they are minimal and slowly varying and can be considered as constant. In addition, we add constant external disturbances to the UAM-V i.e., $\tau_{\text{ext},p} = \tau_{\text{ext},\eta} = [0.2, 0.2, 0.2]^T$. In the second scenario, we consider the arm is tracking a trajectory and continuously moves.

In both scenarios, the desired UAM-V trajectory in the shape of figure-8 is considered and is given by $p_d(t) = [A\sin(\frac{2\pi t}{T}), B\sin(\frac{4\pi t}{T}), -1]$ m. The parameter A and B correspond to the length and width of the figure-8, respectively, whereas T is the time-period of the trajectory i.e., time required to complete one cycle of figure-8. The trajectory for yaw angle is $\psi_d(t) = \frac{3\pi}{4}\cos(\frac{2\pi}{T}t)$ rad. We use choose $A = 1.5$ m, $B = 0.75$ m, and $T = 10$ s which results in a bounded, smooth and adequately

dynamic trajectory to test the performance of the proposed controller.

The initial conditions of the quadrotor UAM-V are $p(0) = [-2, 2, 0]^T$ m, $v(0) = [0, 0, 0]^T \mathrm{m\,s^{-1}}$, $\eta(0) = [0, 0, 0]^T$ rad and $\omega(0) = [0, 0, 0]^T \mathrm{rad\,s^{-1}}$. The initial conditions for adaptive parameters are $\hat{a}(0) = \frac{1}{0.7m}$ and $\hat{J}_v(0) = 0.6J_v$ i.e., considering a 30% error in mass and a 40% error in inertia initially. The initial conditions for disturbance estimates are considered $\hat{d}_f(0) = \hat{d}_\tau(0) = [0, 0, 0]^T$.

Table 4.1: Quadrotor UAM-V parameters.

Parameter	Value	Parameter	Value
m_q	1.6 kg	$J_{q,xx}$	0.03 kg m^2
$J_{q,yy}$	0.03 kg m^2	$J_{q,zz}$	0.05 kg m^2
$J_{q,xy}$	0.0 kg m^2	$J_{q,yz}$	0.0 kg m^2
$J_{q,zx}$	0.0 kg m^2	ℓ	0.25 m
Θ	$\pi/4$ rad		

Table 4.2: Arm parameters.

Parameter	Link 1	Link 2	Link 3
Joint Type	Revolute	Revolute	Fixed
Mass m_{L_i} (kg)	0.1	0.1	0.1
Length l_i (m)	0.2	0.3	0.1
Inertia J_{L_i} (kg m^2)	$[7.508, 7.533, 0.0417] \times 10^{-4}$	$[7.508, 7.533, 0.0417] \times 10^{-4}$	$[1.667, 1.667, 1.667] \times 10^{-4}$

Table 4.3: Control Gains

Parameter	Value	Parameter	Value
k_1	1.6	k_2	2
λ	0.0055	k_{d_f}	1
k_η	3.5	k_ω	5
Γ	2×10^{-4}	Υ	5

The simulation results for the fixed-arm case are given below. Figure 4.6 shows the desired and the actual trajectory for UAM-V position and yaw. The outer-loop tracking errors i.e., the errors in position tracking are shown in Figure 4.7. Although the position errors appear to be convergent towards zero as time goes to infinity, in fact they are reaching a small bound which is not noticeable in the plots. The figure-8 trajectory results in a time-varying trajectory attitude as shown in Figure 4.9 resulting in $\ddot{\eta} \neq 0, \dot{\eta} \neq 0$ in (4.1). Therefore the d_f is time-varying hence a bounded convergence. The bounded error result is adequate for practical stability of the system. The estimates for force disturbance and mass parameter $a = \frac{1}{m}$ are shown in Figure 4.8 which are bounded with minor oscillations. The

first two components of $\hat{d}_f$ are convergent towards their actual values while $\hat{d}_{f_3}$ and $\hat{a}$ converge to a constant bound. The inner-loop tracking errors i.e., the attitude tracking errors shown in Figure 4.10 are bounded as well. The torque disturbance estimates are shown in Figure 4.11, which are bounded and appear to oscillate around their true value. The inertia estimates of the UAM are shown in Figure 4.12 and have a bounded error in relation to their true values. This is consistent with the proposed adaptive control theory as the convergence of estimation errors is not guaranteed. In practice, the convergence of estimation errors is not required if the tracking errors are convergent. The thrust and torque inputs applied to the UAM-V are shown in Figure 4.13. Figure 4.14 and 4.15 show 2D and 3D position plots for the UAM-V that give an overall picture of the tracking performance of the adaptive inner-outer loop control.

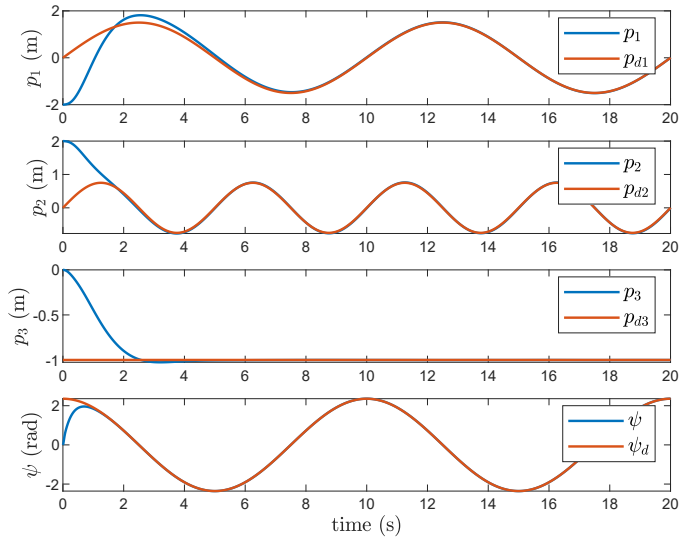


Figure 4.6: UAM-V pose in case of fixed arm i.e., $\alpha_1 = \alpha_2 = 0$

In the above discussion, the arm has been fixed, which shows the tracking performance when the disturbance is small and slowly time-varying, while inertia is constant. In what follows, we continuously move the arm to see the position tracking performance of the UAM-V in case of a time-varying force and torque disturbance and time-varying inertia of the UAM-V. Time-varying inertia is a consequence of its dependence on arm configuration $J = J_q + \sum_{i=1}^{n_a}(R^b_{L_i} J_{L_i} {R^b_{L_i}}^T - m_{L_i} \mathrm{S}^2(p^b_{L_i}))$ from (2.46). The arm motion is selected so that Link 2 rotates in a circle while Link 1 oscillates. The trajectory of arm angles is selected to be $\alpha_1(t) = \frac{\pi}{3}\sin(\pi t)$ and $\alpha_2(t) = \frac{3\pi}{2}t$.

The resulting position and yaw plots are shown in Figure 4.16 while the position errors are shown in Figure 4.17. The position errors exhibit small oscillations close

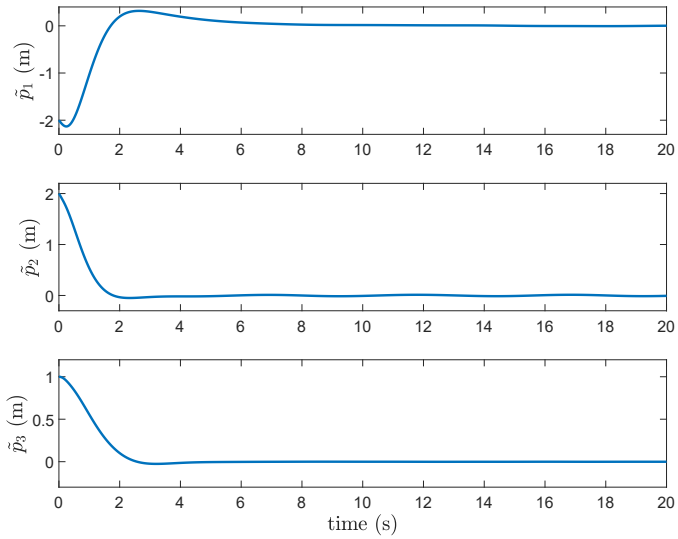


Figure 4.7: Outer-loop tracking errors in case of fixed arm.

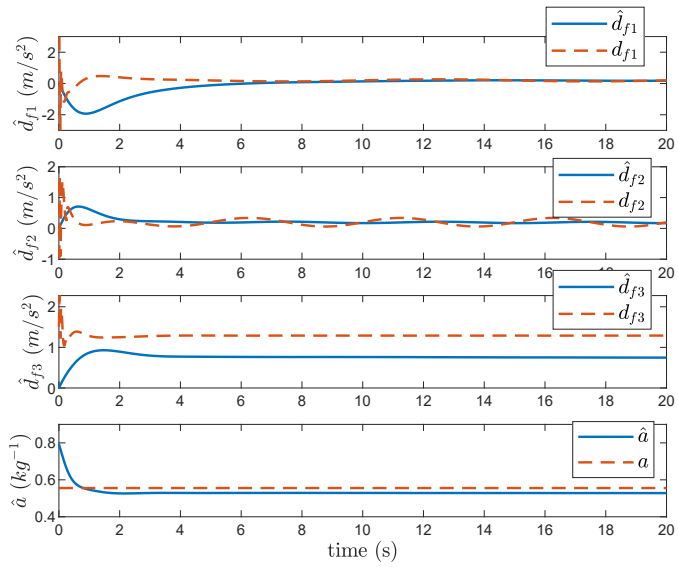


Figure 4.8: Outer-loop estimates for d_f and a in case of fixed arm.

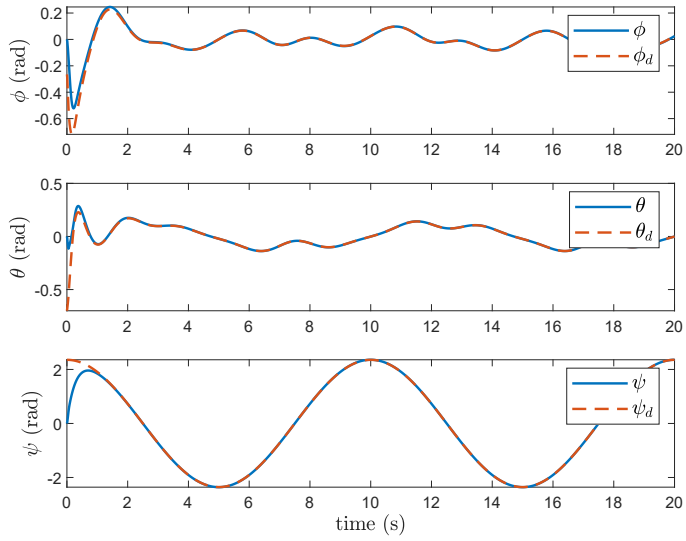


Figure 4.9: Inner-loop attitude tracking errors in case of fixed arm.

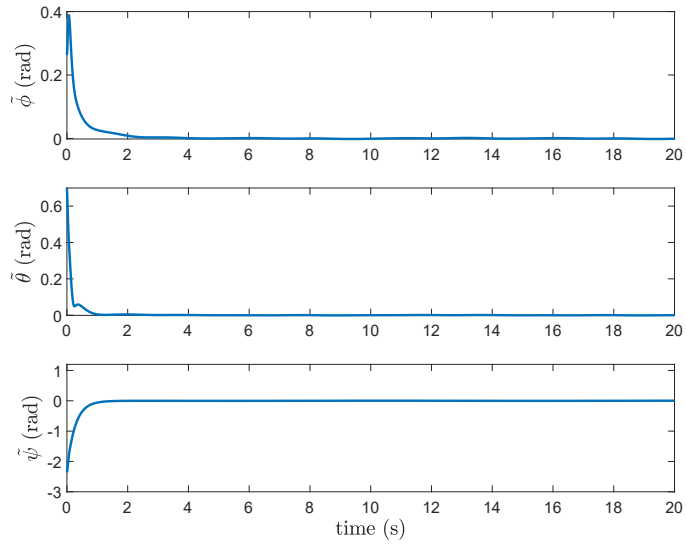


Figure 4.10: Inner-loop attitude tracking errors in case of fixed arm.

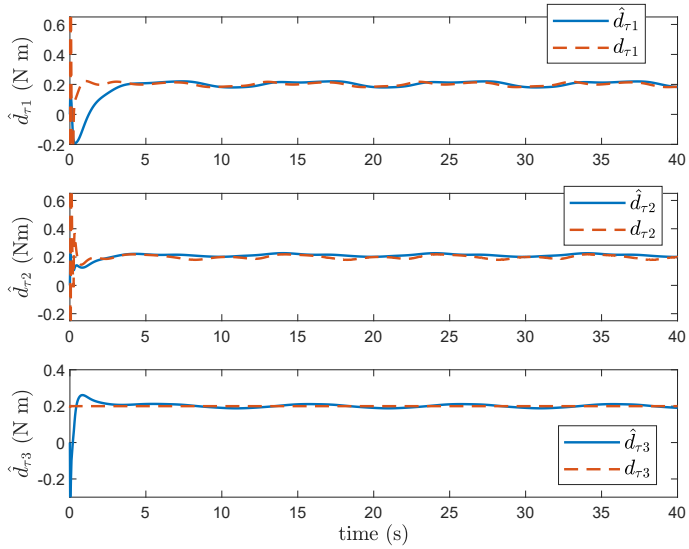


Figure 4.11: Estimates for torque disturbance d_τ in case of fixed arm.

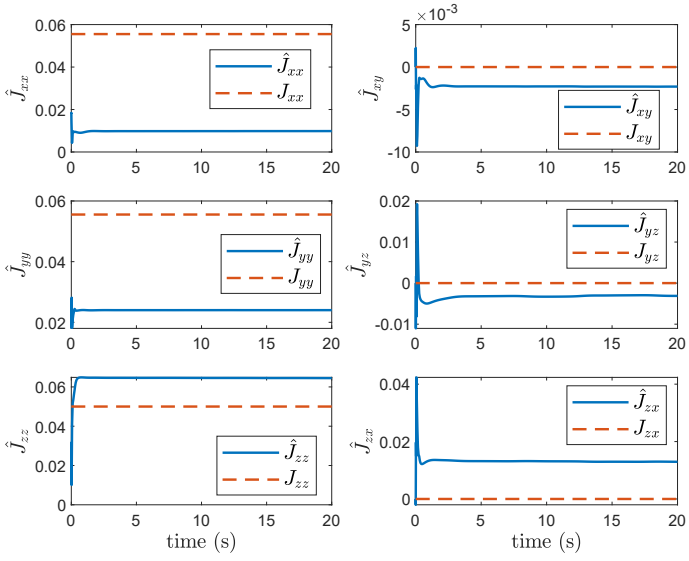


Figure 4.12: Estimate for Interia matrix J in case of fixed arm.

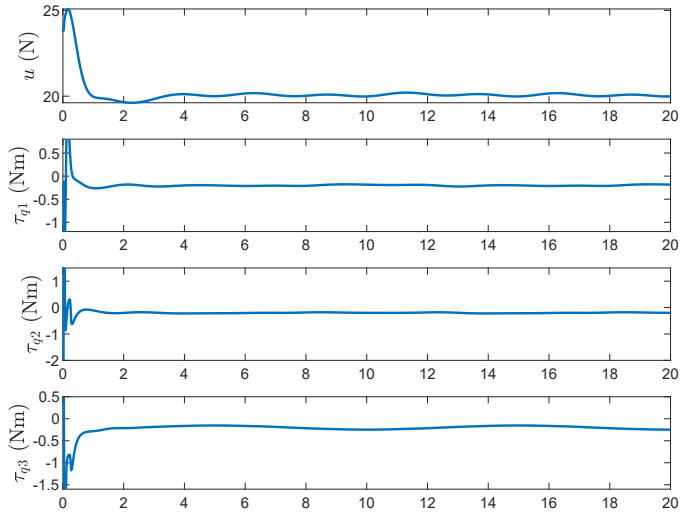


Figure 4.13: UAM-V inputs in case of fixed arm.

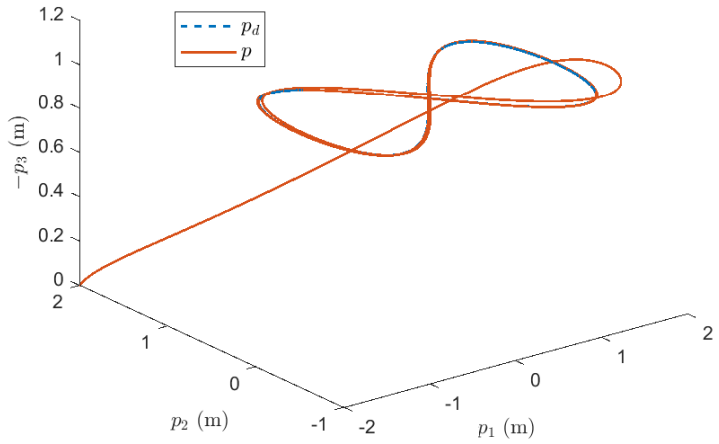


Figure 4.14: 3D UAM-V position in case of fixed arm.

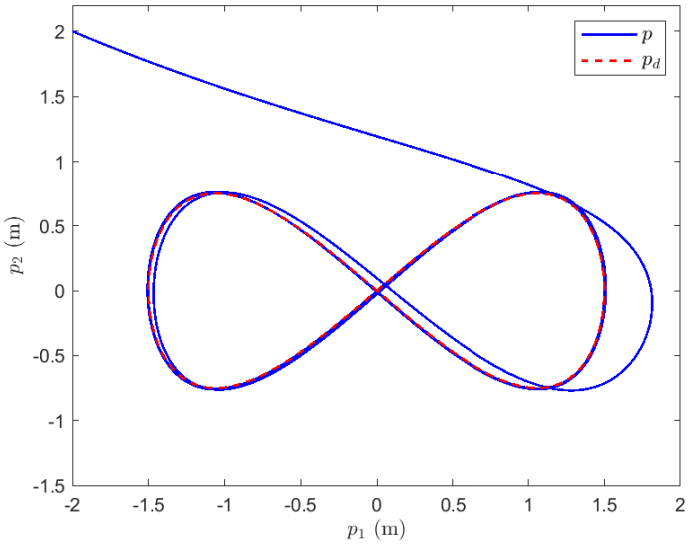


Figure 4.15: 2D UAM-V position in case of fixed arm.

to zero and show the robustness of the controller in the presence of time-varying disturbances due to arm motion. The plots for force disturbance and its estimate are shown in Figure 4.18, which are bounded. The inner-loop tracking errors are shown in Figure 4.20 which show convergence of attitude error to a bounded region around the origin. The torque disturbance and its estimate are shown in Figures 4.21. Since axis of rotation of arm joints is parallel to b_2, a relatively larger variation can be seen in d_{τ_2} as compared to other two components. The disturbance estimate is less oscillatory than its true value but have similar mean as the true value. The plots for the inertia parameters are given in 4.22. There is a bounded difference between the estimated and true values of inertia. The UAM-V inputs for the moving arm case are shown in Figure 4.23. The 3D and 2D plots of the UAM-V are shown in Figures 4.24 and 4.25, respectively, and provide a bigger picture on the tracking performance for the two cases. It can be seen from these plots are that the UAM-V position tracks the desired trajectory with the bounded error.

Now we present the results for a pick and place application where a Quadrotor flies to a target, extends its arm, picks up the mass and then flies to another location and places a mass on the other location. This is done by creating a virtual environment in Simscape Multibody, consisting of two tables and a small cube of mass 0.1 kg and dimension of 0.1 m on each side. The specifications of this cubic object are listed under Link 3 in Table 4.2. The UAM-V starts at its initial condition, $p(0) = [-2, 2, 0]^T$ and the desired pose of the UAM-V at Table 1 is $p_d^1 = [1.4379, 2, -1.4121]^T$, $\psi_d^1 = 0$ rad, and at Table 2 is $p_d^2 = [-2, -1.4379, -0.9121]^T$, $\psi_d^2 = -\frac{\pi}{2}$ rad. The superscripts with these symbols

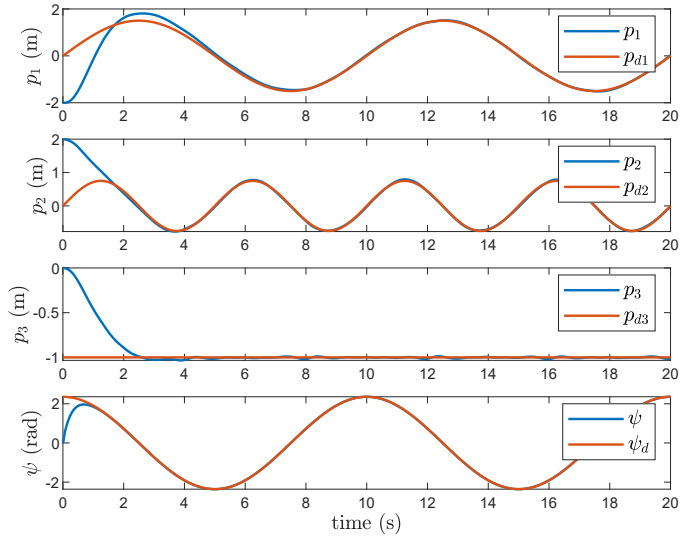


Figure 4.16: UAM-V pose in case of moving arm.

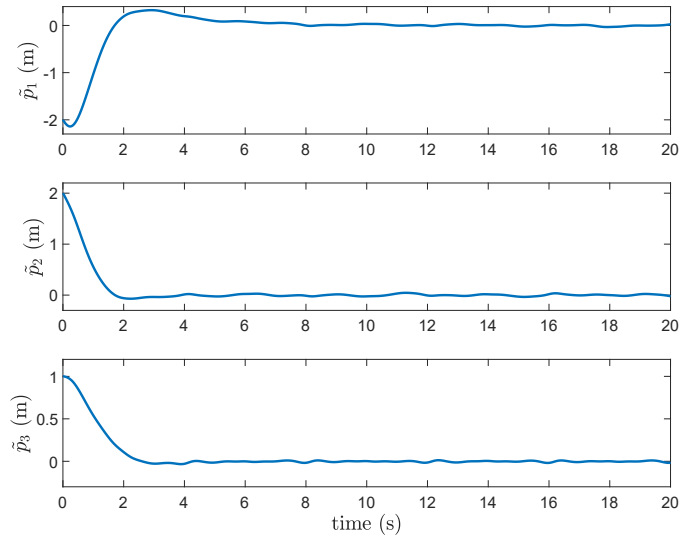


Figure 4.17: Outer-loop tracking errors in case of moving arm.

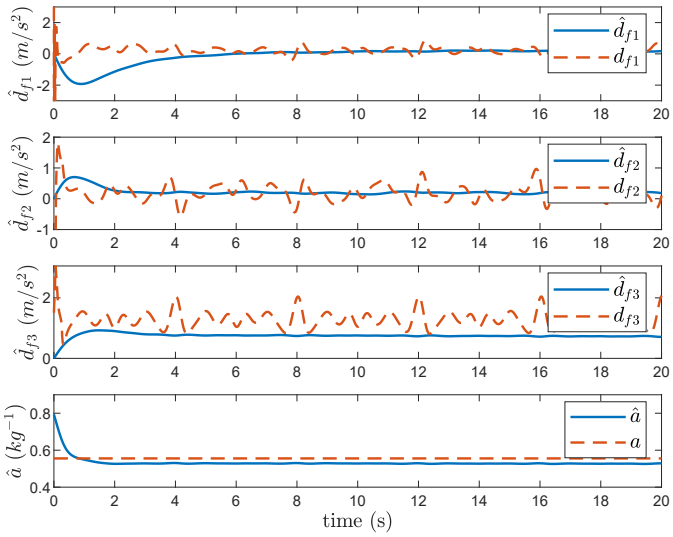


Figure 4.18: Outer-loop estimates for d_f and a in case of moving arm.

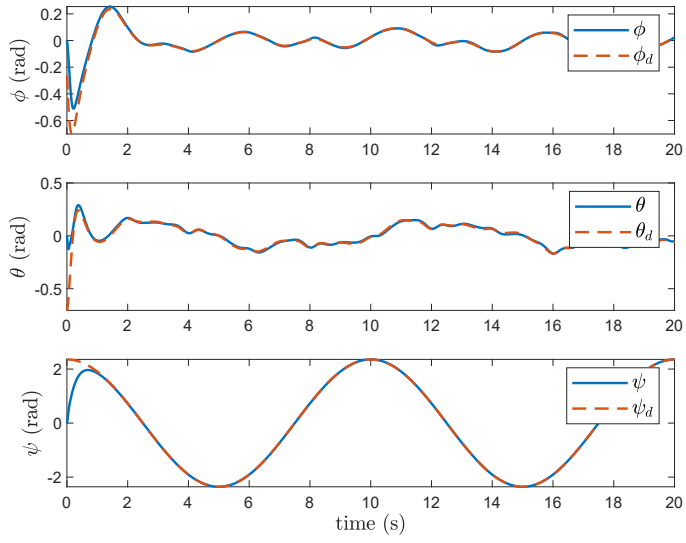


Figure 4.19: UAM-V reference and measured attitude in case of moving arm.

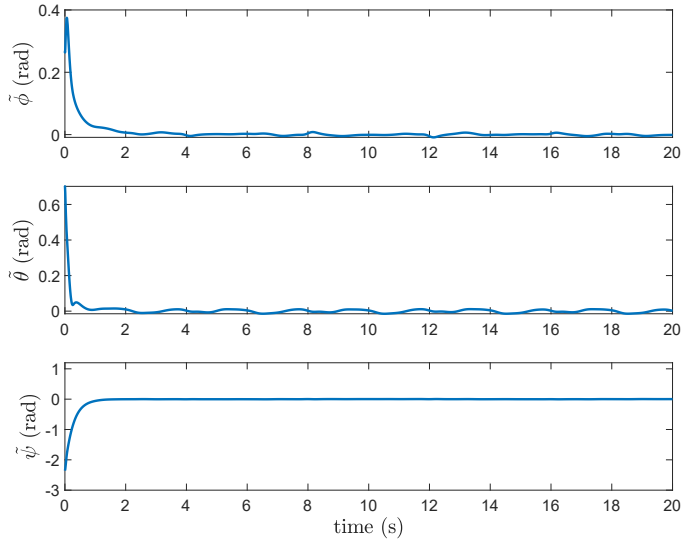


Figure 4.20: Inner-loop attitude tracking errors in case of moving arm.

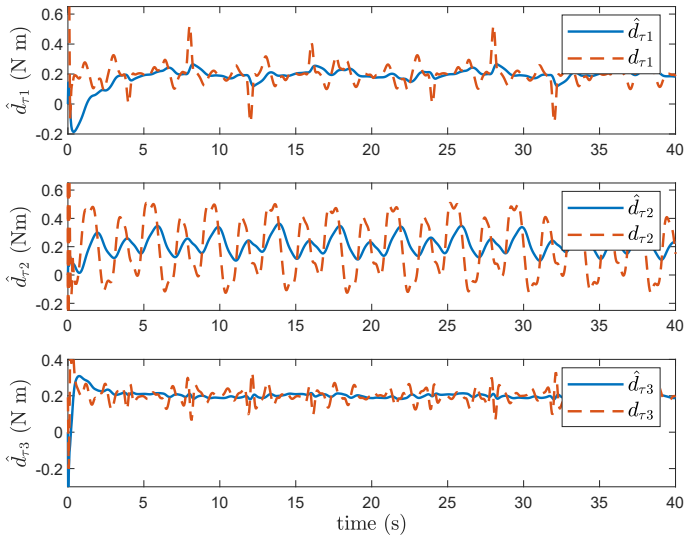


Figure 4.21: Actual and estimated torque disturbance d_τ in case of a moving arm.

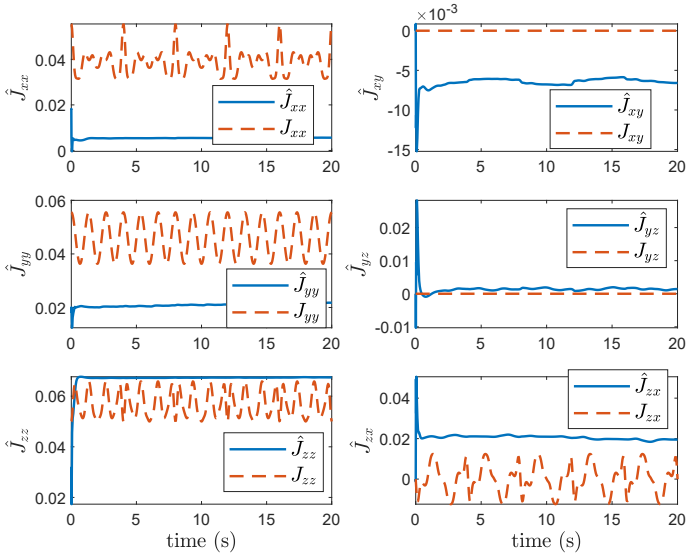


Figure 4.22: Actual and estimated Interia matrix J in case of a moving arm.

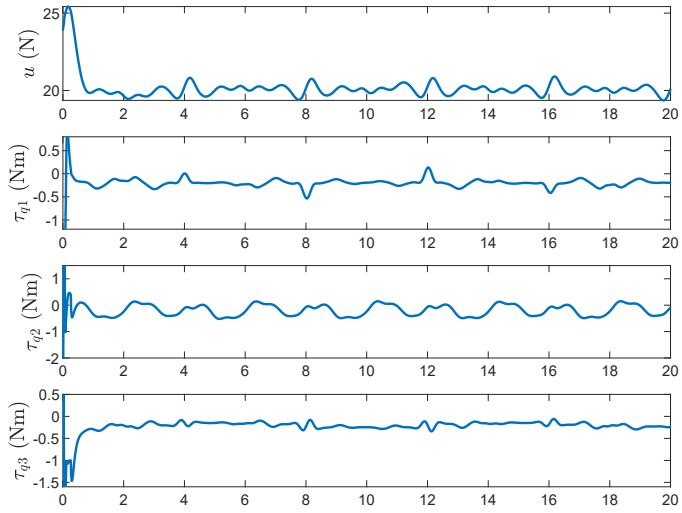


Figure 4.23: UAM-V inputs in case of moving arm.

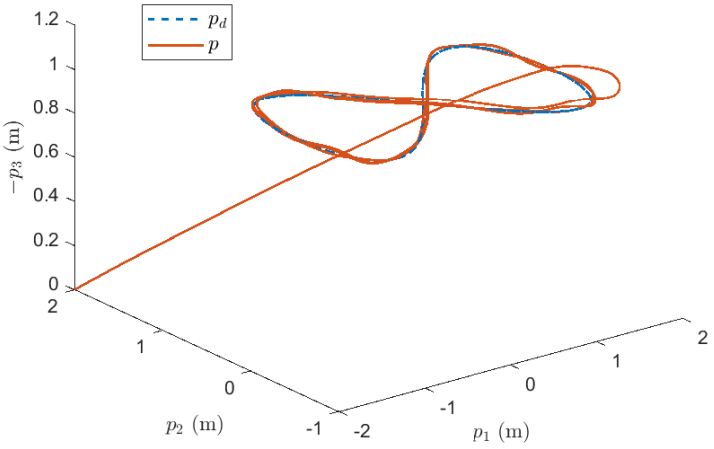


Figure 4.24: 3D UAM-V position in case of moving arm.

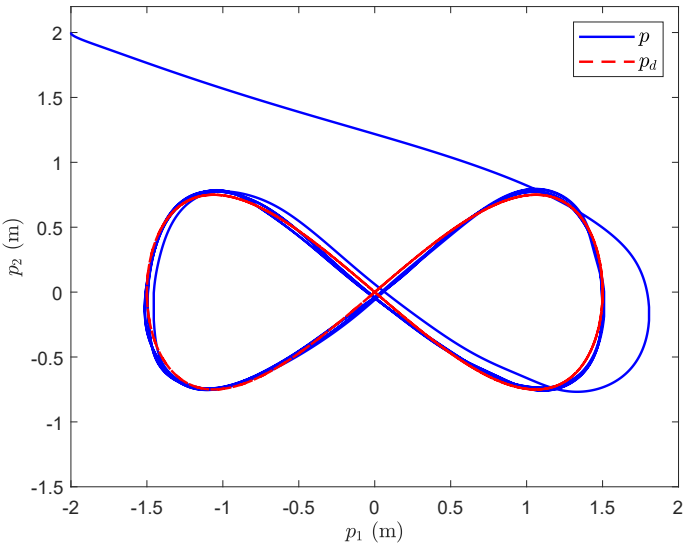


Figure 4.25: 2D UAM-V position in case of moving arm.

correspond to the table number. To avoid the collision of arm with the table during extension or retraction, UAM-V altitude is kept higher than desired. The UAM-V position is lowered to desired value using a function $h(t_0) = [0, 0, 0.1\sin(\pi(t_0)/5))]^T$, when the arm is fully extended. The UAM-V trajectory is designed in a way that it tests the control performance for both trajectory tracking and reference set point tracking. Table 4.4 shows the trajectory plan for the UAM-V and arms. The home configuration in the table refers to the retracted state of the arm with $\alpha_1 = -\pi/4, \alpha_2 = 3\pi/4$. The desired and reference values of UAM-V pose are shown

Table 4.4: Object pick and place sequence

Time t	Desired Position p_d	Desired Yaw ψ_d	Arm configuration	Cube Location
0	p_d^1	ψ_d^1	Home	Table 1
20	p_d^1	ψ_d^1	Extending	Table 1
27	$p_d^1 + h(t-27)$	ψ_d^1	Extended	Table 1
29	$p_d^1 + h(t-27)$	ψ_d^1	Extended	UAM
32	p_d^1	ψ_d^1	Retracting	UAM
40	$p_d^1 + \frac{p_d^2 - p_d^1}{1+\exp(-(t-45))}$	$\frac{\psi_d^2}{1+\exp(-(t-45))}$	Home	UAM
60	p_d^2	ψ_d^2	Extending	UAM
67	$p_d^2 + h(t-67)$	ψ_d^2	Extending	UAM
70	$p_d^2 + h(t-67)$	ψ_d^2	Extended	Table 2
73	p_d^2	ψ_d^2	Retracting	Table 2
80	$[0, 0, 0]^T$	0	Home	Table 2

in Figure 4.26 while the position errors are given in Figure 4.27 with labelling for each phase. A video of the simulation is available online [1]. It is evident that the UAM-V closely tracks the desired references for position. There are discontinuities in the plots. These discontinuities are due to shift in reference set point for the UAM-V position. This tests the ability of control to respond to the changes in set points after it has reached the steady state. Clearly, the system behavior remains convergent and the large set point changes do not lead to instability.

4.5 Conclusion

In this chapter, a control scheme for the control of Unmanned Aerial Manipulator is presented. A decentralized approach has been used for control, where the effect of Arm motion has been considered in the form of unknown parameters and disturbance on the UAM-V dynamics. A backstepping-based control design has been presented for the position tracking of a time-varying trajectory. An inner-outer loop control structure is used where the UAM-V dynamics are subdivided into inner and outer loops. UAM-V outer-loop or translational dynamics are controlled by selecting

[1] Pick and place simulation video: `https://youtu.be/GHHQEnf4kgE`

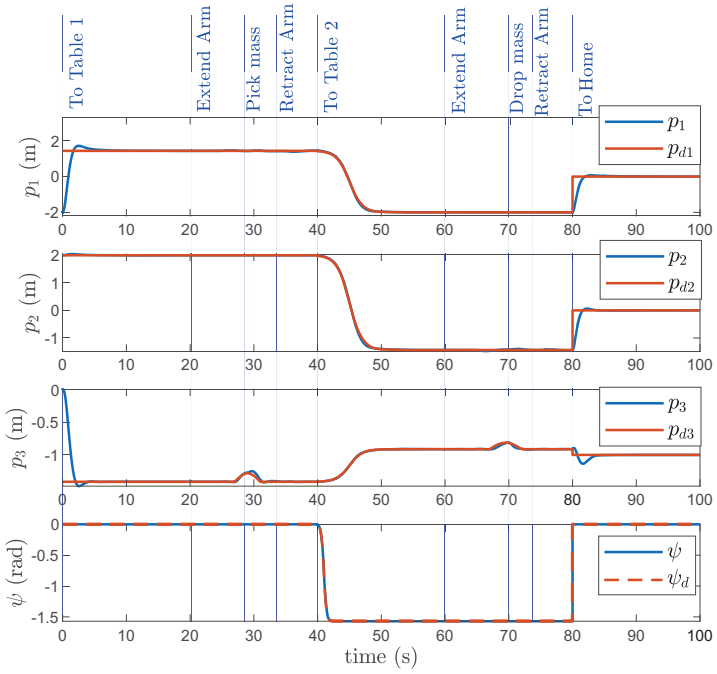


Figure 4.26: UAM-V pose for pick and place task.

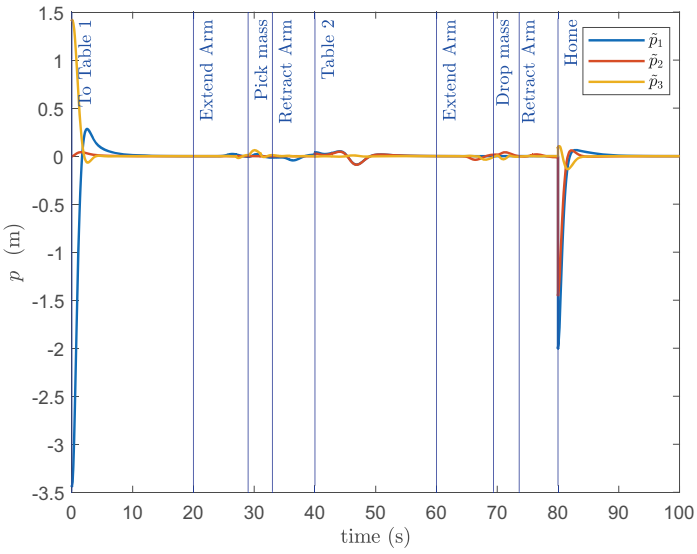


Figure 4.27: UAM-V position errors for pick and place task.

appropriate thrust, roll, and pitch references. The references from the outer-loop are fed as reference trajectories to the inner-loop. The asymptotic convergence of both inner and outer loop tracking errors is proved through Lyapunov stability analysis. This is proved in the presence of unknown parameters and unknown disturbances due to arm motion. The adaptive estimates in the controller adapt to the changes in UAM-V parameters and disturbance due to arm motion. The approach has been rigorously tested using a Multi-body simulation platform in Simscape. The simulation results show fast convergence of both inner and outer loop tracking errors in the presence of constant unknown parameters and disturbances, while a convergence to a bounded region in the presence of time-varying disturbance. The effectiveness of the proposed approach has also been tested for a proper UAM application, where the proposed controller is tested for a pick and place operation.

Chapter 5

UAM Motion Control using entire UAM-V Dynamics

In the previous chapter, we looked at the variation of system parameters and coupling terms in the dynamics of the UAV subsystem (UAM-V) in a UAM. The parameters include the mass and the inertia of the UAM-V, while the coupling terms include the force and torque disturbances due to the arm. We made simplifying assumptions of parameters and disturbances due to arm, and proposed an inner-outer loop control approach. We analysed the stability of inner and outer loops individually, and the stability of the complete closed loop was not considered. In this chapter, we present a control approach that removes this limitation.

We develop a control approach that considers the entire UAM-V dynamics in a single control loop. To be specific, instead of designing thrust and attitude reference as inputs for position control and then torque inputs for attitude control, this approach directly provides UAM-V thrust and torque for position control. The approach is later modified to avoid numerical derivatives in the presence of measurement noise.

This chapter is organized as follows. Section 5.1 reiterates the assumptions on UAM model and the objective of the control design. Section 5.1 presents the control structure and controller design and the stability analysis. Section 5.4 discusses the problems with numerical derivatives and provides a modified approach that does not require numerical derivatives.

5.1 UAM Model and Control Structure

Similar to the previous chapter, we consider a decentralized control approach where the UAV and the arm are treated as separate subsystems. Again, we use the modelling assumptions discussed in Section 4.1 for UAM-V, treating the effects of arm

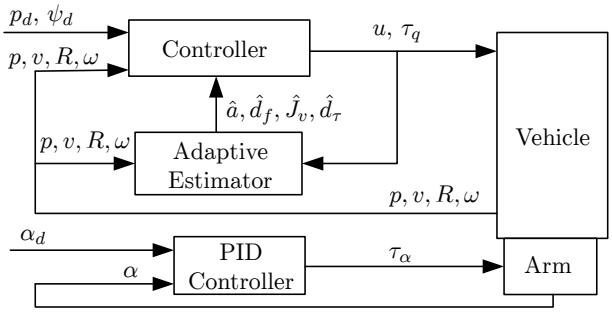


Figure 5.1: Control Structure using complete UAM-V dynamics in one loop

in the form of unknown parameters and disturbances. We assume that the arm attached to the UAV has its independent controller, which can control the joint variables to any arbitrary value.

The aim of this control design is to propose an adaptive controller for torque and thrust inputs of the UAM-V that can track a desired time-varying trajectory for position and yaw. We use a single loop control structure, where the full UAM-V dynamics are used to directly design a controller for UAM-V thrust and torque that tracks a desired position and yaw trajectory. The control structure is shown in Figure 5.1.

Similar to the last chapter, we re-notate some parameters in the UAM model from Chapter 2, to make the presentation simple. We notate $p^n = p$, $R_b^n = R$, $v^n = v$ and $J_{\eta\eta} = J$, in (2.1), (2.3) and (2.50), and write the complete system dynamics as follows

$$\dot{p} = v, \tag{5.1a}$$

$$\dot{v} = ge_3 - auRe_3 + d_f, \tag{5.1b}$$

$$\dot{R} = R\mathrm{S}(\omega), \tag{5.1c}$$

$$J\dot{\omega} = -\mathrm{S}(\omega)J\omega + \tau_q + d_\tau. \tag{5.1d}$$

Here the inputs to the UAM-V are thrust u and torque τ_q. It is noticeable that the UAM-V dynamics (5.1), in contrast to the UAM-V dynamics used in Chapter 4, employ the full rotation matrix to represent attitude dynamics instead of the Euler angle parameterization in (2.4). This allows us to avoid the singularity due to the Gimbal lock and a *global* stability result can be proved.

We know that J in (5.1) is a symmetric positive definite matrix and consists of only six unique entries. Similar to (4.36), we can collect all the unique entries of J in a vector form called $J_v \in \mathbb{R}^6$. In this model we treat a, J_v, d_f and d_τ as unknown constants whereas $\hat{a} \in \mathbb{R} > 0$, $\hat{J}_v \in \mathbb{R}^6$, $\hat{d}_f \in \mathbb{R}^3$, and $\hat{d}_\tau \in \mathbb{R}^3$ are their estimates, respectively. The estimation errors for these quantities are defined as $\tilde{a} = a - \hat{a}$,

$\tilde{J}_v = J_v - \hat{J}_v$, $\tilde{d}_f = d_f - \hat{d}_f$ and $\tilde{d}_\tau = d_\tau - \hat{d}_\tau$.

The objective of this control design is to make the UAM-V position p and yaw ψ track a desired position $p_d \in \mathbb{R}^3$ and $\psi_d \in \mathbb{R}$ trajectory in the presence of parameter uncertainties and disturbances arising due to the robot arm attached to the UAM-V. Similar to Chapter 4, we use a backstepping approach to design an adaptive controller to achieve the control objective. We start with the position tracking first and then move to both position and yaw tracking control.

5.2 Position Tracking Control

Let us start the backstepping procedure by defining a position error as $z_1 = p - p_d$. Its time-derivative is given by

$$\dot{z}_1 = \dot{p} - \dot{p}_d = v - v_d. \tag{5.2}$$

Consider the first Lyapunov function $V_1 = \frac{1}{2} z_1^T z_1$, taking its time-derivative and adding and subtracting $k_1 z_1^T z_1$, we get

$$\dot{V}_1 = -k_1 \|z_1\|^2 + z_1^T (v - v_d + k_1 z_1). \tag{5.3}$$

We define the quantity in the parentheses in the above equation as $z_2 = v - v_d + k_1 z_1$, which allows us to rewrite (5.2) as

$$\dot{z}_1 = -k_1 z_1 + z_2. \tag{5.4}$$

After taking the time-derivative of z_2, substituting the linear velocity dynamics from (5.1b), $a = \hat{a} + \tilde{a}$ and $d_f = \hat{d}_f + \tilde{d}_f$ we get

$$\dot{z}_2 = g e_3 - \hat{a} u R e_3 + \hat{d}_f - \dot{v}_d - k_1^2 z_1 + k_1 z_2 - \tilde{a} u R e_3 + \tilde{d}_f. \tag{5.5}$$

Consider the 2nd Lyapunov function as $V_2 = V_1 + \frac{1}{2} z_2^T z_2$. Taking its time derivative, substituting $\dot{V}_1$ from (5.3) and simplifying, we have

$$\dot{V}_2 = -k_1 \|z_1\|^2 + z_2^T (g e_3 - \hat{a} u R e_3 + \hat{d}_f - \dot{v}_d - k_1^2 z_1 + k_1 z_2 - \tilde{a} u R e_3 + \tilde{d}_f).$$

After adding and subtracting $k_2 z_2^T z_2$ in above equation and simplifying, we get

$$\begin{aligned} \dot{V}_2 = &- k_1 \|z_1\|^2 - k_2 \|z_2\|^2 + z_2^T (g e_3 - \hat{a} u R e_3 + \hat{d}_f - \dot{v}_d + (1 - k_1^2) z_1 \\ &+ (k_1 + k_2) z_2) - \tilde{a} z_2^T u R e_3 + \tilde{d}_f^T z_2. \end{aligned} \tag{5.6}$$

Let us define the coefficient of z_2^T in above equation as

$$z_3 = ge_3 - \hat{a}uRe_3 + \hat{d}_f - \dot{v}_d + (1 - k_1^2)z_1 + (k_1 + k_2)z_2. \tag{5.7}$$

which can be rearranged as

$$z_3 - z_1 - k_2 z_2 = ge_3 - \hat{a}uRe_3 + \hat{d}_f - \dot{v}_d - k_1^2 z_1 + k_1 z_2,$$

and substituted into (5.5) to get

$$\dot{z}_2 = z_3 - z_1 - k_2 z_2 - \tilde{a}uRe_3 + \tilde{d}_f. \tag{5.8}$$

Taking the time-derivative of (5.7), we have

$$\dot{z}_3 = -\dot{\hat{a}}uRe_3 - \hat{a}\dot{u}Re_3 - \hat{a}u\dot{R}e_3 + \dot{\hat{d}}_f - \ddot{v}_d + (1 - k_1^2)\dot{z}_1 + (k_1 + k_2)\dot{z}_2.$$

Substituting (5.4) and (5.8), we have

$$\begin{aligned}\dot{z}_3 = &- \dot{\hat{a}}uRe_3 - \hat{a}\dot{u}Re_3 - \hat{a}uR\mathrm{S}(\omega)e_3 + \dot{\hat{d}}_f - \ddot{v}_d + (1 - k_1^2)(-k_1 z_1 + z_2) \\ &+ (k_1 + k_2)(z_3 - z_1 - k_2 z_2 - \tilde{a}uRe_3 + \tilde{d}_f).\end{aligned}$$

Simplifying and rearranging the above equation, we get

$$\begin{aligned}\dot{z}_3 = &- \dot{\hat{a}}uRe_3 - \hat{a}\dot{u}Re_3 - \hat{a}uR\mathrm{S}(\omega)e_3 + \dot{\hat{d}}_f - \ddot{v}_d \\ &+ (k_1(k_1^2 - 1) - k_2 - k_1)z_1 + (1 - k_2(k_1 + k_2) - k_1^2)z_2 \\ &+ (k_1 + k_2)z_3 - (k_1 + k_2)\tilde{a}uRe_3 + (k_1 + k_2)\tilde{d}_f.\end{aligned} \tag{5.9}$$

Consider the 3rd Lyapunov function $V_3 = V_2 + \frac{1}{2}z_3^T z_3$ whose time derivative is given by

$$\dot{V}_3 = -k_1\|z_1\|^2 - k_2\|z_2\|^2 + z_2^T z_3 + z_3^T \dot{z}_3 - \tilde{a}z_2^T uRe_3 + \tilde{d}_f^T z_2.$$

Substituting (5.9), we have

$$\begin{aligned}\dot{V}_3 = &- k_1\|z_1\|^2 - k_2\|z_2\|^2 + z_3^T\big(z_2 - \dot{\hat{a}}uRe_3 - \hat{a}\dot{u}Re_3 - \hat{a}uR\mathrm{S}(\omega)e_3 + \dot{\hat{d}}_f - \ddot{v}_d \\ &+ (k_1(k_1^2 - 1) - k_2 - k_1)z_1 + (1 - k_2(k_1 + k_2) - k_1^2)z_2 + (k_1 + k_2)z_3 \\ &- (k_1 + k_2)\tilde{a}uRe_3 + (k_1 + k_2)\tilde{d}_f\big) - \tilde{a}z_2^T uRe_3 + \tilde{d}_f^T z_2.\end{aligned}$$

Adding and subtracting $k_3 z_3^T z_3$ and simplifying

$$\begin{aligned}\dot{V}_3 =& - k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T\big(- \dot{\hat{a}}uRe_3 - \hat{a}\dot{u}Re_3 - \hat{a}uR\mathrm{S}(\omega)e_3 + \dot{\tilde{d}}_f \\ & - \ddot{v}_d + (k_1(k_1^2-1) - k_2 - k_1)z_1 + (2 - k_2(k_1+k_2) - k_1^2)z_2 + (k_1+k_2+k_3)z_3\big) \\ & - \tilde{a}(z_2^T + (k_1+k_2)z_3^T)uRe_3 + \tilde{d}_f^T(z_2 + (k_1+k_2)z_3).\end{aligned}$$

Consider a modified Lyapunov function

$$V_{3b} = V_3 + \frac{1}{2\lambda}\tilde{a}^2 + \frac{1}{2k_{d_f}}\tilde{d}_f^T\tilde{d}_f, \tag{5.10}$$

where $k_{d_f} > 0, \lambda > 0$, and the time-derivative of above equation is

$$\begin{aligned}\dot{V}_{3b} =& - k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T\big(- \dot{\hat{a}}uRe_3 - \hat{a}\dot{u}Re_3 - \hat{a}uR\mathrm{S}(\omega)e_3 + \dot{\tilde{d}}_f \\ & - \ddot{v}_d + (k_1(k_1^2-1) - k_2 - k_1)z_1 + (2 - k_2(k_1+k_2) - k_1^2)z_2 + (k_1+k_2+k_3) \\ & \times z_3\big) - \tilde{a}(z_2^T + (k_1+k_2)z_3^T)uRe_3 + \tilde{d}_f^T(z_2 + (k_1+k_2)z_3) + \frac{1}{\lambda}\tilde{a}\dot{\tilde{a}} + \frac{1}{k_{d_f}}\tilde{d}_f^T\dot{\tilde{d}}_f.\end{aligned}$$

Simplifying, we get

$$\begin{aligned}\dot{V}_{3b} =& - k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T\big(- \dot{\hat{a}}uRe_3 - \hat{a}\dot{u}Re_3 - \hat{a}uR\mathrm{S}(\omega)e_3 \\ & + \dot{\tilde{d}}_f - \ddot{v}_d + (k_1(k_1^2-1) - k_2 - k_1)z_1 + (2 - k_2(k_1+k_2) - k_1^2)z_2 \\ & + (k_1+k_2+k_3)z_3\big) + \tilde{a}\big(\lambda\dot{\tilde{a}} - (z_2^T + (k_1+k_2)z_3^T)uRe_3\big) \\ & + \tilde{d}_f^T\big(\frac{1}{k_{d_f}}\dot{\tilde{d}}_f + z_2 + (k_1+k_2)z_3\big).\end{aligned} \tag{5.11}$$

To reduce the lengthy expressions, let us define $\sigma_1 \in \mathbb{R}^3$ and $\sigma_2 \in \mathbb{R}^3$ as functions of known quantities

$$\begin{aligned}\sigma_1 =& (k_1(k_1^2-1) - k_2 - k_1)z_1 + (2 - k_2(k_1+k_2) - k_1^2)z_2 \\ & + (k_1+k_2+k_3)z_3 - \ddot{v}_d, \qquad (5.12)\\ \sigma_2 =& z_2 + (k_1+k_2)z_3. \qquad (5.13)\end{aligned}$$

Substituting these equations in (5.11) along with $\dot{\tilde{d}}_f = -\dot{\hat{d}}_f$, which can be written because d_f is approximated as constant rendering $\dot{d}_f = 0$, we have

$$\begin{aligned}\dot{V}_{3b} =& - k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T\big(- \dot{\hat{a}}uRe_3 - \hat{a}\dot{u}Re_3 \\ & - \hat{a}uR\mathrm{S}(\omega)e_3 + \dot{\hat{d}}_f + \sigma_1\big) + \tilde{a}(\frac{1}{\lambda}\dot{\tilde{a}} - \sigma_2^T uRe_3) + \tilde{d}_f^T(-\frac{1}{k_{d_f}}\dot{\hat{d}}_f + \sigma_2).\end{aligned}$$

Adding and subtracting $k_{d_f} z_3^T \sigma_2$ in above equation

$$\begin{aligned}\dot{V}_{3b} = &- k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T\big(- \dot{\hat{a}} u R e_3 - \hat{a}\dot{u} R e_3 - \hat{a} u R \mathrm{S}(\omega) e_3 \\ &+ \dot{\tilde{d}}_f - k_{d_f}\sigma_2 + k_{d_f}\sigma_2 - \ddot{v}_d + \sigma_1\big) + \tilde{a}(\frac{1}{\lambda}\dot{\tilde{a}} - \sigma_2^T u R e_3) + \tilde{d}_f^T(-\frac{1}{k_{d_f}}\dot{\tilde{d}}_f + \sigma_2),\end{aligned}$$

and simplifying

$$\begin{aligned}\dot{V}_{3b} = &- k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T(-\dot{\hat{a}} u R e_3 - \hat{a}\dot{u} R e_3 \\ &- \hat{a} u R \mathrm{S}(\omega) e_3 + \sigma_1 + k_{d_f}\sigma_2) - k_{d_f} z_3^T(-\frac{1}{k_{d_f}}\dot{\tilde{d}}_f + \sigma_2) \\ &+ \tilde{a}(\frac{1}{\lambda}\dot{\tilde{a}} - \sigma_2^T u R e_3) + \tilde{d}_f^T(-\frac{1}{k_{d_f}}\dot{\tilde{d}}_f + \sigma_2).\end{aligned} \tag{5.14}$$

We know that

$$\begin{aligned}\sigma_1 + k_{d_f}\sigma_2 =& R R^T(\sigma_1 + k_{d_f}\sigma_2) = R I R^T(\sigma_1 + k_{d_f}\sigma_2) \\ =& R(I - e_3 e_3^T + e_3 e_3^T) R^T(\sigma_1 + k_{d_f}\sigma_2) \\ =& \big(R(I - e_3 e_3^T) R^T + R e_3 e_3^T R^T\big)(\sigma_1 + k_{d_f}\sigma_2).\end{aligned}$$

Therefore,

$$\begin{aligned}\dot{V}_{3b} = &- k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T\big(- \dot{\hat{a}} u R e_3 - \hat{a}\dot{u} R e_3 \\ &+ R e_3 e_3^T R^T(\sigma_1 + k_{d_f}\sigma_2) - \hat{a} u R \mathrm{S}(\omega) e_3 + R(I - e_3 e_3^T) R^T(\sigma_1 + k_{d_f}\sigma_2)\big) \\ &- k_{d_f} z_3^T(-\frac{1}{k_{d_f}}\dot{\tilde{d}}_f + \sigma_2) + \tilde{a}(\frac{1}{\lambda}\dot{\tilde{a}} - \sigma_2^T u R e_3) + \tilde{d}_f^T(-\frac{1}{k_{d_f}}\dot{\tilde{d}}_f + \sigma_2).\end{aligned}$$

By factoring out Re_3 from first three terms inside the parentheses of z_3^T coefficient and equating them to zero, we have

$$\dot{\hat{a}}\dot{u} + \dot{\hat{a}} u = e_3^T R^T(\sigma_1 + k_{d_f}\sigma_2), \tag{5.15}$$

while the remaining $\dot{V}_{3b}$ is

$$\begin{aligned}\dot{V}_{3b} = &- k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T R\big(- \hat{a} u \mathrm{S}(\omega) e_3 \\ &+ (I - e_3 e_3^T) R^T(\sigma_1 + k_{d_f}\sigma_2)\big) + (\tilde{d}_f^T - k_{d_f} z_3^T)(-\frac{1}{k_{d_f}}\dot{\tilde{d}}_f + \sigma_2) \\ &+ \tilde{a}(\frac{1}{\lambda}\dot{\tilde{a}} - \sigma_2^T u R e_3).\end{aligned} \tag{5.16}$$

In this equation, we consider the term $\hat{a} u \mathrm{S}(\omega_d) e_3$ as the virtual control, whose desired value is defined as

$$\hat{a} u \mathrm{S}(\omega_d) e_3 = (I - e_3 e_3^T) R^T(\sigma_1 + k_{d_f}\sigma_2),$$

which can be further written as

$$\hat{a}u\mathrm{S}(e_3)\omega_d = -(I - e_3e_3^T)R^T(\sigma_1 + k_{d_f}\sigma_2). \tag{5.17}$$

Extracting individual components of ω_d, we have

$$\omega_{d1} = -\frac{e_2^T}{\hat{a}u}(I - e_3e_3^T)R^T(\sigma_1 + k_{d_f}\sigma_2), \tag{5.18}$$

$$\omega_{d2} = \frac{e_1^T}{\hat{a}u}(I - e_3e_3^T)R^T(\sigma_1 + k_{d_f}\sigma_2), \tag{5.19}$$

while (5.16) becomes

$$\begin{aligned}\dot{V}_{3b} = &- k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T\big(-\hat{a}uR\mathrm{S}(\omega)e_3 + \hat{a}uR\mathrm{S}(\omega_d)e_3\big)\\ &+ (\tilde{d}_f^T - k_{d_f}z_3^T)(-\frac{1}{k_{d_f}}\dot{\hat{d}}_f + \sigma_2) + \tilde{a}(\frac{1}{\lambda}\dot{\tilde{a}} - \sigma_2^T uRe_3).\end{aligned}$$

Simplifying and rearranging above equation, we have

$$\begin{aligned}\dot{V}_{3b} = &- k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T\hat{a}uR\mathrm{S}(e_3)\big(\omega - \omega_d\big)\\ &+ (\tilde{d}_f^T - k_{d_f}z_3^T)(-\frac{1}{k_{d_f}}\dot{\hat{d}}_f + \sigma_2) + \tilde{a}(\frac{1}{\lambda}\dot{\tilde{a}} - \sigma_2^T uRe_3).\end{aligned} \tag{5.20}$$

Consider the following adaptive laws,

$$\begin{aligned}\dot{\hat{d}}_f &= k_{d_f}\sigma_2,\\ \dot{\hat{a}} &= -\lambda\sigma_2^T uRe_3,\end{aligned} \tag{5.21}$$

Substituting them in (5.20), we have

$$\dot{V}_{3b} = - k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T\hat{a}uR\mathrm{S}(e_3)\big(\omega - \omega_d\big). \tag{5.22}$$

Let $z_4 = \omega - \omega_d$. Using (2.50b), we can write

$$J\dot{z}_4 = J\dot{\omega} - J\dot{\omega}_d = -\mathrm{S}(\omega)J\omega + \tau_q + d_\tau - J\dot{\omega}_d. \tag{5.23}$$

From (4.36), we know that

$$\begin{aligned}\mathrm{S}(\omega)J\omega &= \Phi(\omega)J_v,\\ J\bar{\omega}_d &= \Psi(\bar{\omega}_d)J_v,\end{aligned} \tag{5.24}$$

where

$$\Phi(\omega) = \begin{bmatrix} 0 & -\omega_2\omega_3 & \omega_2\omega_3 & -\omega_1\omega_3 & \omega_2^2 - \omega_3^2 & \omega_1\omega_2 \\ \omega_1\omega_3 & 0 & -\omega_1\omega_3 & \omega_2\omega_3 & -\omega_1\omega_2 & \omega_3^2 - \omega_1^2 \\ -\omega_1\omega_2 & \omega_1\omega_2 & 0 & \omega_1^2 - \omega_2^2 & \omega_1\omega_3 & -\omega_2\omega_3 \end{bmatrix},$$

$$\Psi(\bar{\omega}_d) = \begin{bmatrix} \bar{\omega}_{d1} & 0 & 0 & \bar{\omega}_{d2} & 0 & \bar{\omega}_{d3} \\ 0 & \bar{\omega}_{d2} & 0 & \bar{\omega}_{d1} & \bar{\omega}_{d3} & 0 \\ 0 & 0 & \bar{\omega}_{d3} & 0 & \bar{\omega}_{d2} & \bar{\omega}_{d1} \end{bmatrix},$$

and $J_v = [J_{xx}, J_{yy}, J_{zz}, J_{xy}, J_{yz}, J_{zx}]^T$. Therefore, we can rewrite (5.23) as follows

$$J\dot{z}_4 = -(\Phi(\omega) + \Psi(\dot{\omega}_d))J_v + \tau_q + d_\tau. \tag{5.25}$$

Let us substitute $J_v = \hat{J}_v + \tilde{J}_v$ and $d_\tau = \hat{d}_\tau + \tilde{d}_\tau$ in (5.25), we have

$$J\dot{z}_4 = -(\Phi(\omega) + \Psi(\dot{\omega}_d))(\hat{J}_v + \tilde{J}_v) + \tau_q + \hat{d}_\tau + \tilde{d}_\tau. \tag{5.26}$$

Consider the following Lyapunov function

$$V_4 = V_{3b} + \frac{1}{2} z_4^T J z_4 + \frac{1}{2k_{d_\tau}} \tilde{d}_\tau^T \tilde{d}_\tau + \frac{1}{2} \tilde{J}_v^T \Gamma^{-1} \tilde{J}_v, \tag{5.27}$$

where $k_{d_\tau} > 0$, $\Gamma \in \mathbb{R}^{6\times 6} > 0$. Taking the time-derivative of above equation, we have

$$\dot{V}_4 = \dot{V}_{3b} + z_4^T J \dot{z}_4 + \frac{1}{k_{d_\tau}} \tilde{d}_\tau^T \dot{\tilde{d}}_\tau + \tilde{J}_v^T \Gamma^{-1} \dot{\tilde{J}}_v. \tag{5.28}$$

Substituting (5.22) and (5.26) in above equation, we get

$$\begin{aligned} \dot{V}_4 = & -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T \hat{a} u R \mathrm{S}(e_3) z_4 \\ & + z_4^T(-(\Phi(\omega) + \Psi(\dot{\omega}_d))(\hat{J}_v + \tilde{J}_v) + \tau_q + \hat{d}_\tau + \tilde{d}_\tau) \\ & + \frac{1}{k_{d_\tau}} \tilde{d}_\tau^T \dot{\tilde{d}}_\tau + \tilde{J}_v^T \Gamma^{-1} \dot{\tilde{J}}_v. \end{aligned} \tag{5.29}$$

Adding and subtracting $k_4 z_4^T z_4$ in above and rearranging, we have

$$\begin{aligned} \dot{V}_4 = & -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 - k_4\|z_4\|^2 + z_4^T \hat{a} u R_b^{n^T} \mathrm{S}^T(e_3) z_3 \\ & + z_4^T(k_4 z_4 - (\Phi(\omega) + \Psi(\dot{\omega}_d))\hat{J}_v + \tau_q + \hat{d}_\tau) + -\tilde{J}_v^T(\Phi(\omega) + \Psi(\dot{\omega}_d))^T z_4 \\ & + \tilde{d}_\tau^T z_4 + \frac{1}{k_{d_\tau}} \tilde{d}_\tau^T \dot{\tilde{d}}_\tau + \tilde{J}_v^T \Gamma^{-1} \dot{\tilde{J}}_v. \end{aligned} \tag{5.30}$$

Rearranging, we get

$$\begin{aligned} \dot{V}_4 = & -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 - k_4\|z_4\|^2 + z_4^T(\hat{a} u R_b^{n^T} \mathrm{S}^T(e_3) z_3 \\ & + k_4 z_4 - (\Phi(\omega) + \Psi(\dot{\omega}_d))\hat{J}_v + \tau_q + \hat{d}_\tau) \\ & + \tilde{J}_v^T(\Gamma^{-1}\dot{\tilde{J}}_v - (\Phi(\omega) + \Psi(\dot{\omega}_d))^T z_4) + \tilde{d}_\tau^T(\frac{1}{k_{d_\tau}} \dot{\tilde{d}}_\tau + z_4). \end{aligned} \tag{5.31}$$

Consider the following control law

$$\tau_q = -\hat{a}uR_b^{n^T}\mathrm{S}^T(e_3)z_3 - k_4 z_4 + (\Phi(\omega) + \Psi(\dot{\omega}_d))\hat{J}_v - \hat{d}_\tau, \tag{5.32}$$

together with the adaptation laws

$$\begin{aligned}\dot{\hat{J}}_v =& -\Gamma(\Phi(\omega) + \Psi(\dot{\omega}_d))^T z_4,\\ \dot{\hat{d}}_\tau =& k_{d_\tau} z_4.\end{aligned} \tag{5.33}$$

Substituting, the control law (5.32) and adaptation laws (5.33) along with the assumption that J_v and d_τ are constant or slowly varying to satisfy $\dot{J}_v = 0, \dot{d}_\tau = 0$, we have

$$\dot{V}_4 = -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 - k_4\|z_4\|^2, \tag{5.34}$$

which is negative semi-definite.

Theorem 5.2.1. *Consider the dynamics of a UAM-V under the assumption that $\dot{a} = 0$, $\dot{d}_f = 0$, $\dot{d}_\tau = 0$, $\dot{J}_v = 0$, and $u > 0$, $\hat{a} > 0$, with a smooth time-varying trajectory consisting of $\{p_d(t), v_d(t), \dot{v}_d(t), \ddot{v}_d(t)\}$. If the UAM-V inputs consisting of thrust and torques are selected according to* (5.32) *with $k_1 > 0, k_2 > 0, k_3 > 0, k_4 > 0$, while the estimates $\hat{a}, \hat{d}_f, \hat{d}_\tau$ and $\hat{J}_v$ are given by the adaptive laws in* (5.21) *and* (5.33)*, then the tracking errors $[z_1^T, z_2^T, z_3^T, z_4^T]$ are globally asymptotically stable at origin while the estimates $\hat{a}, \hat{d}_f, \hat{d}_\tau$ and $\hat{J}_v$ are bounded.*

Proof. Consider the quadratic positive definite Lyapunov function V_4 in (5.27). According to [184, Lemma 4.3], there exist two positive definite functions that lower and upper bound V_4. The time-derivative of V_4 after substitution of input and control laws, given in (5.34), is a negative semi-definite function. However, using Barbalat's lemma [184, Theorem 8.4], we can conclude that the vector $z^T = [z_1^T, z_2^T, z_3^T, z_4^T]$ $\to 0$ as $t \to \infty$, and parameter and disturbance estimates $\hat{a}$, $\hat{d}_f$, $\hat{d}_\tau$ and $\hat{J}_v$ are bounded. Since, V_4 is radially unbounded, we conclude that the equilibrium point $z = 0$ is globally asymptotically stable. □

5.3 Position and Yaw Tracking Control

The control design proposed above only deals with the tracking of a time-varying position trajectory. The heading or yaw angle of a UAM-V provides an extra degree of freedom in aerial manipulation and allows the end-effector to be aligned with the target from different directions. Therefore, it is essential to design the controller for yaw tracking. Consider the desired yaw trajectory of the UAM-V $\psi_d \in (-\pi, \pi]$ and the yaw tracking error $z_\psi = \psi - \psi_d$. We start with the following Lyapunov function

for the stability yaw tracking error.

$$V_{\psi_1} = \frac{1}{2} z_{\psi_1}^2 \tag{5.35}$$

Taking its time derivative, we have

$$\dot{V}_{\psi_1} = z_{\psi_1} \dot{z}_{\psi_1} = z_{\psi_1} (\dot{\psi} - \dot{\psi}_d)$$

From (2.4), we know $\dot{\psi} = e_3^T W^{-1} \omega$, therefore, we can write

$$\dot{V}_{\psi_1} = z_{\psi_1} \dot{z}_{\psi_1} = z_{\psi_1} (e_3^T W^{-1} \omega - \dot{\psi}_d)$$

Adding and subtracting $k_{\psi_1} z_{\psi_1}^2$, we have

$$\dot{V}_{\psi_1} = -k_{\psi_1} z_{\psi_1}^2 + z_{\psi_1} (k_{\psi_1} z_{\psi_1} + e_3^T W^{-1} \omega - \dot{\psi}_d) \tag{5.36}$$

Let us define

$$e_3^T W^{-1} \omega_d = -k_{\psi_1} z_{\psi_1} + \dot{\psi}_d \tag{5.37}$$

which results in

$$\omega_{d3} = \frac{\mathrm{c}_\theta}{\mathrm{c}_\phi} (-k_{\psi_1} z_{\psi_1} + \dot{\psi}_d + \frac{\mathrm{s}_\phi}{\mathrm{c}_\theta} \omega_{d2}) \tag{5.38}$$

where ω_{d2} is given in (5.19). Substituting (5.37) in (5.36), we have

$$\dot{V}_{\psi_1} = -k_{\psi_1} z_{\psi_1}^2 + z_{\psi_1} (e_3^T W^{-1} (\omega - \omega_d)) \tag{5.39}$$

Since $z_4 = \omega - \omega_d$, we can write

$$\dot{V}_{\psi_1} = -k_{\psi_1} z_{\psi_1}^2 + z_4^T z_{\psi_1} W^{-1^T} e_3 \tag{5.40}$$

Theorem 5.3.1. *Consider the dynamics of a UAM-V in* (5.1) *under the assumption that* $\dot{a} = 0$, $\dot{d}_f = 0$, $\dot{d}_\tau = 0$ *and* $\dot{J}_v = 0$ *and* $u > 0$, $\hat{a} > 0$, *with a smooth time-varying trajectory consisting of* $\{p_d(t), v_d(t), \dot{v}_d(t), \ddot{v}_d(t), \psi_d(t) \dot{\psi}_d(t)\}$. *If the UAM-V input thrust is selected according to* (5.64), *the input torque is selected as below*

$$\tau_q = -z_{\psi_1} W^{-T} e_3 - \hat{a} u R_b^{n^T} S^T(e_3) z_3 - k_4 z_4 + (\Phi(\omega) + \Psi(\dot{\omega}_d)) \hat{J}_v - \hat{d}_\tau \tag{5.41}$$

with $k_1 > 0, k_2 > 0, k_3 > 0, k_4 > 0$, *and the estimates* $\hat{a}, \hat{d}_f, \hat{d}_\tau$ *and* $\hat{J}_v$ *are given by the adaptive laws in* (5.21) *and* (5.33), *then the tracking error* $[z_1^T, z_2^T, z_3^T, z_4^T, z_{\psi_1}]$ *is asymptotically stable at origin while the estimates* $\hat{a}, \hat{d}_f, \hat{d}_\tau$ *and* $\hat{J}_v$ *are bounded.*

Proof. Consider the following Lypunov function $V_{4,\psi} = V_4 + V_{\psi_1}$ and its time deriva-

tive is given by

$$\dot{V}_{4,\psi} = \dot{V}_4 + \dot{V}_{\psi,1} \tag{5.42}$$

Substituting, $\dot{V}_4$ from (5.31) together with adaptive estimation laws in (5.33) and the $\dot{V}_{\psi_1}$ from (5.40), we have

$$\begin{aligned}\dot{V}_{4,\psi} =& -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 - k_4\|z_4\|^2 - k_{\psi_1} z_{\psi_1}^2 + z_4^T z_{\psi_1} W^{-1^T} e_3 \\ &+ z_4^T(\hat{a}u R_b^{n^T} \mathrm{S}^T(e_3) z_3 + k_4 z_4 - (\Phi(\omega) + \Psi(\dot{\omega}_d))\hat{J}_v + \tau_q + \hat{d}_\tau)\end{aligned} \tag{5.43}$$

Substituting the torque control input from (5.41), we have

$$\dot{V}_{4,\psi} = -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 - k_4\|z_4\|^2 - k_{\psi_1}\|z_{\psi_1}\|^2 \tag{5.44}$$

which is negitive semi-definite. Using the similar discussion as in the proof of Theorem 5.2.1, we conclude that the tracking error $[z_1^T, z_2^T, z_3^T, z_4^T, z_{\psi_1}] \to 0$ as $t \to \infty$. □

Remark 3. *The controllers presented in* (5.32) *and* (5.41) *depend upon* $\dot{\omega}_d$*, however, the above derivation gives expression for* ω_d *only. The* $\dot{\omega}_d$ *can be obtained numerically or using a low-pass filter approximation. The algebraic differentiation of* ω_d *results in the reappearance of an unknown term* $\tilde{d}_f$*, which couples with unknown inertia vector* J_v *in the angular velocity dynamics of the UAM-V and results in a difficulty in proving system stability. This is shown in the following discussion.*

Algebraic differentiation of ω_d

in this section, we try to take the algebraic derivative of ω_d and discuss the diificuly that arises due to the algebraic derivative of ω_d. Taking the time-derivative of (5.17), we have

$$\begin{aligned}\hat{a}u\mathrm{S}(e_3)\dot{\omega}_d =& -(\dot{\hat{a}}u + \hat{a}\dot{u})\mathrm{S}(e_3)\omega_d + (I - e_3e_3^T)\mathrm{S}(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2) \\ &- (I - e_3e_3^T)R^T(\dot{\sigma}_1 + k_{d_f}\dot{\sigma}_2).\end{aligned} \tag{5.45}$$

Above equation has $\dot{\sigma}_1$ and $\dot{\sigma}_2$. Let us evaluate $\dot{\sigma}_1$ by taking the time derivative of (5.12), we get

$$\dot{\sigma}_1 = (k_1(k_1^2 - 1) - k_2 - k_1)\dot{z}_1 + (2 - k_2(k_1 + k_2) - k_1^2)\dot{z}_2 + (k_1 + k_2 + k_3)\dot{z}_3 - \ddot{v}_d.$$

Substituting (5.4), (5.8) and (5.9), we have

$$\begin{aligned}\dot{\sigma}_1 =&(k_1(k_1^2-1)-k_2-k_1)(-k_1z_1+z_2)+(2-k_2(k_1+k_2)-k_1^2)\\&\times\big(z_3-z_1-k_2z_2-\tilde{a}uRe_3+\tilde{d}_f\big)+(k_1+k_2+k_3)\big(-(\dot{\hat{a}}u+\hat{a}\dot{u})Re_3\\&-\hat{a}uR\mathrm{S}(\omega)e_3+\dot{\tilde{d}}_f-\ddot{v}_d+(k_1(k_1^2-1)-k_2-k_1)z_1+(k_1+k_2)z_3\\&+(1-k_2(k_1+k_2)-k_1^2)z_2-(k_1+k_2)\tilde{a}uRe_3+(k_1+k_2)\tilde{d}_f\big)-\ddot{v}_d.\end{aligned}\tag{5.46}$$

From (5.12) we have

$$(k_1(k_1^2-1)-k_2-k_1)z_1+(1-k_2(k_1+k_2)-k_1^2)z_2+(k_1+k_2)z_3-\ddot{v}_d=\sigma_1-z_2-k_3z_3.$$

Substituting above and (5.15) in (5.46), we get

$$\begin{aligned}\dot{\sigma}_1 =&(k_1(k_1^2-1)-k_2-k_1)(-k_1z_1+z_2)+(2-k_2(k_1+k_2)-k_1^2)\big(z_3-z_1\\&-k_2z_2-\tilde{a}uRe_3+\tilde{d}_f\big)+(k_1+k_2+k_3)\big(-(\dot{\hat{a}}u+\hat{a}\dot{u})Re_3-\hat{a}uR\mathrm{S}(\omega)e_3\\&+\dot{\tilde{d}}_f+\sigma_1-z_2-k_3z_3-(k_1+k_2)\tilde{a}uRe_3+(k_1+k_2)\tilde{d}_f\big)-\ddot{v}_d\\&-(2-k_2(k_1+k_2)-k_1^2)\tilde{a}uRe_3+(2-k_2(k_1+k_2)-k_1^2)\tilde{d}_f.\end{aligned}\tag{5.47}$$

Collecting coefficients of $\tilde{d}_f$ and $\tilde{a}$

$$\begin{aligned}\dot{\sigma}_1 =&(k_1(k_1^2-1)-k_2-k_1)(-k_1z_1+z_2)+(2-k_2(k_1+k_2)-k_1^2)(z_3-z_1-k_2z_2)\\&+(k_1+k_2+k_3)\big(-(\dot{\hat{a}}u+\hat{a}\dot{u})Re_3-\hat{a}uR\mathrm{S}(\omega)e_3+\dot{\tilde{d}}_f+\sigma_1-z_2-k_3z_3\big)\\&-\ddot{v}_d-\big(2-k_2(k_1+k_2)-k_1^2+(k_1+k_2+k_3)(k_1+k_2)\big)\tilde{a}uRe_3\\&+\big(2-k_2(k_1+k_2)-k_1^2+(k_1+k_2+k_3)(k_1+k_2)\big)\tilde{d}_f.\end{aligned}$$

Simplifying, we have

$$\begin{aligned}\dot{\sigma}_1 =&(k_1^2+k_1k_2+k_2^2-2)(z_1-z_3+k_2z_2)-(-k_1^3+2k_1+k_2)(z_2-k_1z_1)\\&+(k_1+k_2+k_3)\big(-(\dot{\hat{a}}u+\hat{a}\dot{u})Re_3-\hat{a}uR\mathrm{S}(\omega)e_3+\dot{\tilde{d}}_f+\sigma_1-z_2-k_3z_3\big)\\&-\ddot{v}_d-(k_1k_2+k_2k_3+k_3k_1+2)\tilde{a}uRe_3+(k_1k_2+k_2k_3+k_3k_1+2)\tilde{d}_f.\end{aligned}\tag{5.48}$$

Let us define known quantities in the above equation as

$$\begin{aligned}\bar{\sigma}_1 =&(k_1^2+k_1k_2+k_2^2-2)(z_1-z_3+k_2z_2)-(-k_1^3+2k_1+k_2)\\&\times(z_2-k_1z_1)-\ddot{v}_d+(k_1+k_2+k_3)(-(\dot{\hat{a}}u+\hat{a}\dot{u})Re_3\\&-\hat{a}uR\mathrm{S}(\omega)e_3+\dot{\tilde{d}}_f+\sigma_1-z_2-k_3z_3),\end{aligned}\tag{5.49}$$

which reduces (5.48) as follows

$$\dot{\sigma}_1 = \bar{\sigma}_1 - (k_1k_2 + k_2k_3 + k_3k_1 + 2)\tilde{a}uRe_3 + (k_1k_2 + k_2k_3 + k_3k_1 + 2)\tilde{d}_f. \quad (5.50)$$

To obtain $\dot{\sigma}_2$, let us take the time-derivative of (5.13) and substitute (5.8) and (5.9), we have

$$\begin{aligned}\dot{\sigma}_2 =& z_3 - z_1 - k_2z_2 - \tilde{a}uRe_3 + \tilde{d}_f + (k_1 + k_2)\big(- (\dot{\hat{a}}u + \hat{a}\dot{u})Re_3 \\ & - \hat{a}uR\mathrm{S}(\omega)e_3 + \dot{\tilde{d}}_f + \sigma_1 - z_2 - k_3z_3 - (k_1 + k_2)\tilde{a}uRe_3 + (k_1 + k_2)\tilde{d}_f\big).\end{aligned}$$

Simplifying and rearranging

$$\begin{aligned}\dot{\sigma}_2 =& z_3 - z_1 - k_2z_2 + (k_1 + k_2)(-z_2 - k_3z_3) + (k_1 + k_2) \\ & \times \big(- (\dot{\hat{a}}u + \hat{a}\dot{u})Re_3 - \hat{a}uR\mathrm{S}(\omega)e_3 + \dot{\tilde{d}}_f + \sigma_1\big) \\ & - (1 + (k_1 + k_2)^2)\tilde{a}uRe_3 + (1 + (k_1 + k_2)^2)\tilde{d}_f. \end{aligned} \quad (5.51)$$

Let us define the known quantities in above equation as

$$\begin{aligned}\bar{\sigma}_2 =& - z_1 - (k_1 + 2k_2)z_2 + (1 - k_1k_3 - k_2k_3)z_3 + (k_1 + k_2) \\ & \times \big(- (\dot{\hat{a}}u + \hat{a}\dot{u})Re_3 - \hat{a}uR\mathrm{S}(\omega)e_3 + \dot{\tilde{d}}_f + \sigma_1\big), \end{aligned} \quad (5.52)$$

which reduces (5.51) to

$$\dot{\sigma}_2 = \bar{\sigma}_2 - (1 + (k_1 + k_2)^2)\tilde{a}uRe_3 + (1 + (k_1 + k_2)^2)^2\tilde{d}_f. \quad (5.53)$$

Substituting (5.50) and (5.53) in (5.45), we have

$$\begin{aligned}\hat{a}u\mathrm{S}(e_3)\dot{\omega}_d =& - (\dot{\hat{a}}u + \hat{a}\dot{u})\mathrm{S}(e_3)\omega_d + (I - e_3e_3^T)\mathrm{S}(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2) \\ & - (I - e_3e_3^T)R^T\Big(\bar{\sigma}_1 - (k_1k_2 + k_2k_3 + k_3k_1 + 2)\tilde{a}uRe_3 \\ & + (k_1k_2 + k_2k_3 + k_3k_1 + 2)\tilde{d}_f + k_{d_f}\bar{\sigma}_2 - k_{d_f}(1 + (k_1 + k_2)^2)\tilde{a}uRe_3 \\ & + k_{d_f}(1 + (k_1 + k_2)^2)^2\tilde{d}_f\Big).\end{aligned}$$

Let $\bar{k} = k_1k_2 + k_2k_3 + k_3k_1 + 2 + k_{d_f}(1 + (k_1 + k_2)^2)^2$, then

$$\begin{aligned}\hat{a}u\mathrm{S}(e_3)\dot{\omega}_d =& - (\dot{\hat{a}}u + \hat{a}\dot{u})\mathrm{S}(e_3)\omega_d + (I - e_3e_3^T)\mathrm{S}(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2) \\ & - (I - e_3e_3^T)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2) - (I - e_3e_3^T)R^T(-\bar{k}\tilde{a}uRe_3 + \bar{k}\tilde{d}_f).\end{aligned}$$

Since $\mathrm{S}(e_3) = [-e_2, e_1, 0_3]^T$, we can multiply both sides with $\mathrm{S}^T(e_3)$ to obtain $\dot{\omega}_{d1}$ and $\dot{\omega}_{d2}$ as follows

$$\begin{aligned}\hat{a}u \begin{bmatrix}\dot{\omega}_{d1}\\ \dot{\omega}_{d2}\\ 0\end{bmatrix} =& -\mathrm{S}^T(e_3)(\dot{\hat{a}}u + \hat{a}\dot{u})\mathrm{S}(e_3)\omega_d + \mathrm{S}^T(e_3)(I - e_3e_3^T)\mathrm{S}(\omega)\\ &\times R^T(\sigma_1 + k_{d_f}\sigma_2) - \mathrm{S}^T(e_3)(I - e_3e_3^T)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2)\\ &- \mathrm{S}^T(e_3)(I - e_3e_3^T)R^T(-\bar{k}\tilde{a}uRe_3 + \bar{k}\tilde{d}_f).\end{aligned}$$

Using $\mathrm{S}^T(e_3)(I - e_3e_3^T) = \mathrm{S}^T(e_3)$, above equation can further be simplified as

$$\begin{aligned}\begin{bmatrix}\dot{\omega}_{d1}\\ \dot{\omega}_{d2}\\ 0\end{bmatrix} =& -\frac{1}{\hat{a}u}(\dot{\hat{a}}u + \hat{a}\dot{u})(I - e_3e_3^T)\omega_d + \frac{1}{\hat{a}u}\mathrm{S}^T(e_3)\mathrm{S}(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2)\\ &- \frac{1}{\hat{a}u}\mathrm{S}^T(e_3)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2) - \frac{1}{\hat{a}u}\mathrm{S}^T(e_3)R^T(-\bar{k}\tilde{a}uRe_3 + \bar{k}\tilde{d}_f).\end{aligned}$$

All terms in the above equation have their last row zero. Therefore we can combine them together $\begin{bmatrix}\dot{\omega}_{d1}\\ \dot{\omega}_{d2}\\ 0\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \dot{\omega}_{d3}\end{bmatrix} = \begin{bmatrix}\dot{\omega}_{d1}\\ \dot{\omega}_{d2}\\ \dot{\omega}_{d3}\end{bmatrix}$ and write

$$\begin{aligned}\dot{\omega}_d =& -\frac{1}{\hat{a}u}(\dot{\hat{a}}u + \hat{a}\dot{u})(I - e_3e_3^T)\omega_d + \frac{1}{\hat{a}u}\mathrm{S}^T(e_3)\mathrm{S}(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2)\\ &- \frac{1}{\hat{a}u}\mathrm{S}^T(e_3)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2) - \frac{1}{\hat{a}u}\mathrm{S}^T(e_3)R^T(-\bar{k}\tilde{a}uRe_3 + \bar{k}\tilde{d}_f\Big) + \dot{\omega}_{d3}e_3,\end{aligned}$$

where $\dot{\omega}_{d3}$ will be obtained later through yaw control. Let us define

$$\begin{aligned}\bar{\omega}_d =& -\frac{1}{\hat{a}u}(\dot{\hat{a}}u + \hat{a}\dot{u})(I - e_3e_3^T)\omega_d + \frac{1}{\hat{a}u}\mathrm{S}^T(e_3)\mathrm{S}(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2)\\ &- \frac{1}{\hat{a}u}\mathrm{S}^T(e_3)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2) + \dot{\omega}_{d3}e_3,\end{aligned} \tag{5.54}$$

consisting of all the known quantities then

$$\dot{\omega}_d = \bar{\omega}_d - \frac{1}{\hat{a}u}\mathrm{S}^T(e_3)R^T(-\bar{k}\tilde{a}uRe_3 + \bar{k}\tilde{d}_f).$$

Since $\mathrm{S}^T(e_3)R^TRe_3 = 0_3$, the $\tilde{a}$ term vanishes and we have

$$\dot{\omega}_d = \bar{\omega}_d - \frac{\bar{k}}{\hat{a}u}\mathrm{S}^T(e_3)R^T\tilde{d}_f. \tag{5.55}$$

Now that we have $\dot{\omega}_d$, let us substitute it in $\dot{V}_4$ from along with $\dot{V}_{3b}$ from (5.20) and $J\dot{z}_4$ from (5.25) defined by

$$\begin{aligned}\dot{V}_4 = &- k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 - k_4\|z_4\|^2 + z_4^T(k_4 z_4 + \hat{a}u\mathrm{S}^T(e_3){R_b^n}^T z_3 \\ &- \mathrm{S}(\omega)J\omega + \tau_q + d_\tau - J(\bar{\omega}_d - \frac{\bar{k}}{\hat{a}u}\mathrm{S}^T(e_3)R^T\tilde{d}_f)) - k_{d_f} z_3^T(-\frac{1}{k_{d_f}}\dot{\hat{d}}_f + \sigma_2) \\ &+ \tilde{d}_f^T(-\frac{1}{k_{d_f}}\dot{\hat{d}}_f + \sigma_2) + \tilde{a}(\frac{1}{\lambda}\dot{\tilde{a}} - \sigma_2^T uRe_3) + \frac{1}{k_{d_\tau}}\tilde{d}_\tau^T\dot{\tilde{d}}_\tau + \tilde{J}_v^T\Gamma^{-1}\dot{\tilde{J}}_v.\end{aligned}$$

In the above equation, we see that $\tilde{d}_f$ appears with the inertia matrix J. If we consider $\frac{1}{2}z_4^T z_4$ to avoid multiplication with J, a J^{-1} would appear with input τ_q and with $\mathrm{S}(\omega)J\omega$ preventing it from writing it into the affine form $\Phi(\omega)J_v$, making adaptive control inapplicable for estimation of inertia matrix J. Also, if we avoid multiplication of $\tilde{d}_f$ with J by using $\frac{1}{2}z_4^T J^{-1}Jz_4$, and substituting $J\dot{z}_4$ in $\dot{V}_4$, we need to multiply $\hat{a}u\mathrm{S}^T(e_3){R_b^n}^T z_3$ with J to bring inside parenthesis i.e., coefficient of z_4^T to cancel with the control. This problem is unavoidable as one of the unknowns J, J_v or $\tilde{J}_v$ keep appearing. Since J is unknown, it is not available for control design or adaptive estimation law.

This issue arises if we try taking the algebraic derivative of ω_d; therefore, using a numerical derivative is suitable to avoid this difficulty. In what follows, we look at another approach that does not define ω_d at the third stage of backstepping. Instead, we continue the backstepping a stage forward and then define $\dot{\omega}_d$ algebraically, after segregating $\tilde{d}_f$ from its definition in the above analysis, and then use its integral as ω_d in the controller. This results in new adaptation laws for force disturbance, inertia matrix, torque disturbance and the control law for torque.

5.4 Modified Control Approach

The control presented in the previous section requires the numerical derivative of the expression used for virtual control of angular velocity. Numerical derivatives usually result in amplification of high-frequency noise in the measurement, e.g., $\frac{d}{dt}(\sin(300t)) = 300 \times \cos(300t)$. The algebraic derivatives, on the other hand, involve direct measurements instead of the derivative of measurements. Only the derivatives of reference trajectories are required, which are noise-free and usually smooth. Also, in the presence of a discontinuity in reference, such as a new reference set point in case of constant reference tracking, a derivative could lead to singularity. In the case of algebraic derivatives, derivatives of constant references are uniformly set to zero even when the reference changes to avoid discontinuity propagation to the controller. This, however, is difficult to implement in the case of numerical derivatives.

In this section, we propose another approach where the numerical derivative of

ω_d is avoided by continuing the backstepping procedure past the 3rd stage until $\dot{\omega}$ shows up. In this derivation, we do not extract the adaptation laws for $\hat{d}_f$ at the third stage of backstepping. This allows us to manage the reappearing unknown $\tilde{d}_f$ terms by cancelling them with an adaptive law expression at a later stage. This results in modified adaptation laws for force disturbance, inertia matrix, torque disturbance and modified control law for torque input.

5.4.1 Position Control

To start, let us recall the definitions of z_1, z_2, z_3, σ_1, σ_2 in Section 5.2, the Lyapunov function V_{3b} from (5.10) and the derivation up to its derivative after substitution of the control law from (5.15), given in (5.16), which is re-written below

$$\begin{aligned}\dot{V}_{3b} =& -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T R\big(-\hat{a}u\mathrm{S}(\omega)e_3 \\ &+ (I - e_3e_3^T)R^T(\sigma_1 + k_{d_f}\sigma_2)\big) + (\tilde{d}_f^T - k_{d_f}z_3^T)(-\frac{1}{k_{d_f}}\dot{\hat{d}}_f + \sigma_2) \\ &+ \tilde{a}(\frac{1}{\lambda}\dot{\tilde{a}} - \sigma_2^T uRe_3)\end{aligned}$$

the remaining coefficient of $z_3^T R$ inside parenthesis in above equation is defined as

$$z_\omega = -\hat{a}u\mathrm{S}(\omega)e_3 + (I - e_3e_3^T)R^T(\sigma_1 + k_{d_f}\sigma_2) \tag{5.56}$$

Taking its time derivative, we have

$$\begin{aligned}\dot{z}_\omega =& -\dot{\hat{a}}u\mathrm{S}(\omega)e_3 - \hat{a}\dot{u}\mathrm{S}(\omega)e_3 - \hat{a}u\mathrm{S}(\dot{\omega})e_3 + (I - e_3e_3^T)\dot{R}^T(\sigma_1 + k_{d_f}\sigma_2) \\ &+ (I - e_3e_3^T)R^T(\dot{\sigma}_1 + \dot{\sigma}_2)\end{aligned}$$

Substituting (2.3) and (5.15), we have

$$\begin{aligned}\dot{z}_\omega =& -(\dot{\hat{a}}u + \hat{a}\dot{u})\mathrm{S}(\omega)e_3 - \hat{a}u\mathrm{S}(\dot{\omega})e_3 + (I - e_3e_3^T)\mathrm{S}^T(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2) \\ &+ (I - e_3e_3^T)R^T(\dot{\sigma}_1 + \dot{\sigma}_2)\end{aligned} \tag{5.57}$$

Substituting (5.50) and (5.53) in (5.57), we have

$$\begin{aligned}\dot{z}_\omega =& -(\dot{\hat{a}}u + \hat{a}\dot{u})\mathrm{S}(\omega)e_3 - \hat{a}u\mathrm{S}(\dot{\omega})e_3 + (I - e_3e_3^T)\mathrm{S}^T(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2) \\ &+ (I - e_3e_3^T)R^T\big(\bar{\sigma}_1 - (k_1k_2 + k_2k_3 + k_3k_1 + 2)\tilde{a}uRe_3 + (k_1k_2 + k_2k_3 \\ &+ k_3k_1 + 2)\tilde{d}_f + \bar{\sigma}_2 - k_{d_f}(1 + (k_1 + k_2)^2)\tilde{a}uRe_3 + k_{d_f}(1 + (k_1 + k_2)^2)\tilde{d}_f\big)\end{aligned}$$

Let us define

$$k = k_1k_2 + k_2k_3 + k_3k_1 + 2 + k_{d_f}(1 + (k_1 + k_2)^2)$$

Using this definition in $\dot{z}_\omega$ and rearranging, we have

$$\begin{aligned}\dot{z}_\omega =& -e_3^T R^T(\sigma_1 + k_{d_f}\sigma_2)\mathrm{S}(\omega)e_3 - \hat{a}u\mathrm{S}(\dot{\omega})e_3 + (I - e_3e_3^T)\mathrm{S}^T(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2)\\ &+ (I - e_3e_3^T)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2) + (I - e_3e_3^T)R^T k\tilde{d}_f - (I - e_3e_3^T)R^T k\tilde{a}uRe_3\end{aligned} \tag{5.58}$$

Recall (5.16) and substitute (5.56), we have

$$\begin{aligned}\dot{V}_{3b} =& -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T R z_\omega + (\tilde{d}_f^T - k_{d_f}z_3^T)(-\frac{1}{k_{d_f}}\dot{\hat{d}}_f + \sigma_2)\\ &+ \tilde{a}(\frac{1}{\lambda}\dot{\tilde{a}} - \sigma_2^T uRe_3)\end{aligned} \tag{5.59}$$

Let us define the 4th Lyapunov function

$$V_{4a} = V_{3b} + \frac{1}{2}z_\omega^T z_\omega \tag{5.60}$$

whose time derivative is given by

$$\dot{V}_{4a} = -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_3^T R z_\omega + z_\omega^T\dot{z}_\omega + z_3^T(\dot{\hat{d}}_f - \sigma_2)$$

Substituting (5.58), we have

$$\begin{aligned}\dot{V}_{4a} =& -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_\omega^T\big(R^T z_3 - (\dot{\hat{a}}u + \hat{a}\dot{u})\mathrm{S}(\omega)e_3 - \hat{a}u\mathrm{S}(\dot{\omega})e_3\\ &+ (I - e_3e_3^T)\mathrm{S}^T(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2) + (I - e_3e_3^T)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2)\\ &+ k(I - e_3e_3^T)R^T\tilde{d}_f - k(I - e_3e_3^T)R^T\tilde{a}uRe_3\big)\\ &+ (\tilde{d}_f^T - k_{d_f}z_3^T)(-\frac{1}{k_{d_f}}\dot{\hat{d}}_f + \sigma_2) + \tilde{a}(\frac{1}{\lambda}\dot{\tilde{a}} - \sigma_2^T uRe_3)\end{aligned}$$

Since

$$k(I - e_3e_3^T)R^T\tilde{a}uRe_3 = \tilde{a}uk(I - e_3e_3^T)R^T Re_3 = \tilde{a}uk(I - e_3e_3^T)e_3 = \tilde{a}uk0_3 = 0_3$$

we have

$$\begin{aligned}\dot{V}_{4a} =& -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_\omega^T\big(R^T z_3 - e_3^T R^T(\sigma_1 + k_{d_f}\sigma_2)\mathrm{S}(\omega)e_3\\ &- \hat{a}u\mathrm{S}(\dot{\omega})e_3 + (I - e_3e_3^T)\mathrm{S}^T(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2) + (I - e_3e_3^T)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2)\\ &+ k(I - e_3e_3^T)R^T\tilde{d}_f\big) + (\tilde{d}_f^T - k_{d_f}z_3^T)(-\frac{1}{k_{d_f}}\dot{\hat{d}}_f + \sigma_2) + \tilde{a}(\frac{1}{\lambda}\dot{\tilde{a}} - \sigma_2^T uRe_3)\end{aligned}$$

Collecting coefficients of $\tilde{d}_f^T$, we have

$$\begin{aligned}\dot{V}_{4a} =& - k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_\omega^T\big(R^T z_3 - e_3^T R^T(\sigma_1 + k_{d_f}\sigma_2)\\ &\times \mathrm{S}(\omega)e_3 - \hat{a}u\mathrm{S}(\dot{\omega})e_3 + (I - e_3e_3^T)\mathrm{S}^T(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2)\\ &+ (I - e_3e_3^T)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2)\big) - k_{d_f}z_3^T(-\frac{1}{k_{d_f}}\dot{\hat{d}}_f + \sigma_2)\\ &+ \tilde{a}(\frac{1}{\lambda}\dot{\hat{a}} - \sigma_2^T uRe_3) + \tilde{d}_f^T\big(-\frac{1}{k_{d_f}}\dot{\hat{d}}_f + \sigma_2 + kR(I - e_3e_3^T)z_\omega\big). \end{aligned} \tag{5.61}$$

We can extract the estimation laws from above equation as follows

$$\dot{\hat{a}} = -\lambda\sigma_2^T uRe_3 \tag{5.62}$$

$$\dot{\hat{d}}_f = k_{d_f}(\sigma_2 + kR(I - e_3e_3^T)z_\omega) \tag{5.63}$$

Using (5.62) in (5.15), we get the control law for thrust input

$$\dot{u} = \frac{1}{\hat{a}}e_3^T R^T(\lambda u^2\sigma_2 + \sigma_1 + k_{d_f}\sigma_2) \tag{5.64}$$

Substituting (5.62) and (5.63) into (5.61), we get

$$\begin{aligned}\dot{V}_{4a} =& - k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 + z_\omega^T\big(R^T z_3 - e_3^T R^T(\sigma_1 + k_{d_f}\sigma_2)\mathrm{S}(\omega)e_3\\ &- \hat{a}u\mathrm{S}(\dot{\omega})e_3 + (I - e_3e_3^T)\mathrm{S}^T(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2)\\ &+ (I - e_3e_3^T)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2)\big) + k_{d_f}z_3^T kR(I - e_3e_3^T)z_\omega\end{aligned}$$

Adding and subtracting $k_\omega z_\omega^T z_\omega$ and simplifying

$$\begin{aligned}\dot{V}_{4a} =& - k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 - k_\omega\|z_\omega\|^2 + z_\omega^T\big(k_\omega z_\omega + R^T z_3\\ &+ k_{d_f}k(I - e_3e_3^T)R^T z_3 - e_3^T R^T(\sigma_1 + k_{d_f}\sigma_2)\mathrm{S}(\omega)e_3 - \hat{a}u\mathrm{S}(\dot{\omega})e_3\\ &+ (I - e_3e_3^T)\mathrm{S}^T(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2) + (I - e_3e_3^T)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2)\big)\end{aligned} \tag{5.65}$$

Let us define $\dot{\omega}_d$ as follows

$$\begin{aligned}\hat{a}u\mathrm{S}(\dot{\omega}_d)e_3 =& k_\omega z_\omega + R^T z_3 + k_{d_f}k(I - e_3e_3^T)R^T z_3 - e_3^T R^T(\sigma_1 + k_{d_f}\sigma_2)\mathrm{S}(\omega)e_3\\ &+ (I - e_3e_3^T)\mathrm{S}^T(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2) + (I - e_3e_3^T)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2)\end{aligned} \tag{5.66}$$

It is worth noticing that the vector $\mathrm{S}(\dot{\omega}_d)e_3 = -\mathrm{S}(e_3)\dot{\omega}_d = [\dot{\omega}_{d2}, -\dot{\omega}_{d1}, 0]^T$ has its third component zero. This shows that the position control is independent of $\dot{\omega}_{d3}$. Therefore, $\dot{\omega}_{d3}$ will be designed separately for the yaw or ψ control. By left multi-

plying (5.66) with $\frac{1}{\hat{a}u}R^T$ followed by left multiplication of e_2^T or e_1^T, we can obtain

$$\begin{aligned}\dot{\omega}_{d1} =& -\frac{e_2^T}{\hat{a}u}(k_\omega z_\omega + R^T z_3 + k_{d_f} k(I - e_3 e_3^T)R^T z_3 - e_3^T R^T(\sigma_1 + k_{d_f}\sigma_2)\mathrm{S}(\omega)e_3 \\ &+ (I - e_3 e_3^T)\mathrm{S}^T(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2) + (I - e_3 e_3^T)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2))\end{aligned} \tag{5.67}$$

$$\begin{aligned}\dot{\omega}_{d2} =& \frac{e_1^T}{\hat{a}u}(k_\omega z_\omega + R^T z_3 + k_{d_f} k(I - e_3 e_3^T)R^T z_3 - e_3^T R^T(\sigma_1 + k_{d_f}\sigma_2)\mathrm{S}(\omega)e_3 \\ &+ (I - e_3 e_3^T)\mathrm{S}^T(\omega)R^T(\sigma_1 + k_{d_f}\sigma_2) + (I - e_3 e_3^T)R^T(\bar{\sigma}_1 + k_{d_f}\bar{\sigma}_2))\end{aligned} \tag{5.68}$$

Substituting (5.66) into (5.65)

$$\dot{V}_{4a} = -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 - k_\omega\|z_\omega\|^2 + \hat{a}u z_\omega^T \mathrm{S}(e_3)(\dot{\omega} - \dot{\omega}_d) \tag{5.69}$$

In the above equation, yaw control is absent, therefore, we introduce it here.

5.4.2 Yaw Control

Lets us now find $\dot{\omega}_{d3}$ for controlling the yaw angle of the UAM-V. Recall from Section 5.3, the definitions of z_{ψ_1} and V_{ψ_1}, their respective derivatives and the derivation up to (5.36). The equation (5.36) is re-written below

$$\dot{V}_{\psi_1} = -k_{\psi_1} z_{\psi_1}^2 + z_{\psi_1}(k_{\psi_1} z_{\psi_1} + e_3^T W^{-1}\omega - \dot{\psi}_d) \tag{5.70}$$

Let us define $z_{\psi_2} = k_{\psi_1} z_{\psi_1} + e_3^T W^{-1}\omega - \dot{\psi}_d$, which reduces (5.70) to

$$\dot{V}_{\psi_1} = -k_{\psi_1} z_{\psi_1}^2 + z_{\psi_1} z_{\psi_2} \tag{5.71}$$

while the derivative of z_{ψ_2} is given by

$$\dot{z}_{\psi_2} = k_{\psi_1}\dot{z}_{\psi_1} + e_3^T \dot{W}^{-1}\omega + e_3^T W^{-1}\dot{\omega} - \ddot{\psi}_d$$

since $\dot{z}_{\psi_1} = -k_{\psi_1} z_{\psi_1} + z_{\psi_2}$, we have

$$\dot{z}_{\psi_2} = -k_{\psi_1}^2 z_{\psi_1} + k_{\psi_1} z_{\psi_2} + e_3^T \dot{W}^{-1}\omega + e_3^T W^{-1}\dot{\omega} - \ddot{\psi}_d$$

Let us consider the following Lyapunov function

$$V_{\psi_2} = V_{\psi_1} + \frac{1}{2} z_{\psi_2}^2 \tag{5.72}$$

whose derivative is given by

$$\dot{V}_{\psi_2} = -k_{\psi_1} z_{\psi_1}^2 + z_{\psi_2}(z_{\psi_1} + \dot{z}_{\psi_2})$$

Adding and subtracting $k_{\psi_2} z_{\psi_2}^2$ and substituting $\dot{z}_{\psi_2}$ from above, we have

$$\begin{aligned}\dot{V}_{\psi_2} =& - k_{\psi_1} z_{\psi_1}^2 - k_{\psi_2} z_{\psi_2}^2 + z_{\psi_2}((1 - k_{\psi_1}^2) z_{\psi_1} + (k_{\psi_1} + k_{\psi_2}) z_{\psi_2} \\ &+ e_3^T \dot{W}^{-1} \omega + e_3^T W^{-1} \dot{\omega} - \ddot{\psi}_d)\end{aligned} \tag{5.73}$$

Let us define

$$e_3^T W^{-1} \dot{\omega}_d = -(1 - k_{\psi_1}^2) z_{\psi_1} - (k_{\psi_1} + k_{\psi_2}) z_{\psi_2} - e_3^T \dot{W}^{-1} \omega + \ddot{\psi}_d \tag{5.74}$$

which results in

$$\dot{\omega}_{d3} = \frac{\mathrm{c}_\theta}{\mathrm{c}_\phi}(-(1 - k_{\psi_1}^2) z_{\psi_1} - (k_{\psi_1} + k_{\psi_2}) z_{\psi_2} - e_3^T \dot{W}^{-1} \omega + \ddot{\psi}_d + \frac{\mathrm{s}_\phi}{\mathrm{c}_\theta} \dot{\omega}_{d2}) \tag{5.75}$$

where $\dot{\omega}_{d2}$ is given in (5.68). Substituting (5.75) in (5.73), we have

$$\dot{V}_{\psi_2} = -k_{\psi_1} z_{\psi_1}^2 - k_{\psi_2} z_{\psi_2}^2 + z_{\psi_2} e_3^T W^{-1} (\dot{\omega} - \dot{\omega}_d) \tag{5.76}$$

5.4.3 Position and Yaw control

Recall (5.60) and (5.72), let us combine them to define

$$V_{4b} = V_{4a} + V_{\psi_2} \tag{5.77}$$

Taking derivative of above and substituting (5.69) and (5.76), we have

$$\begin{aligned}\dot{V}_{4b} =& - k_1 \|z_1\|^2 - k_2 \|z_2\|^2 - k_3 \|z_3\|^2 - k_\omega \|z_\omega\|^2 - k_{\psi_1} z_{\psi_1}^2 - k_{\psi_2} z_{\psi_2}^2 \\ &+ \Big(\hat{a} u z_\omega^T \mathrm{S}(e_3) + z_{\psi_2} e_3^T W^{-1}\Big)(\dot{\omega} - \dot{\omega}_d)\end{aligned} \tag{5.78}$$

Introducing $J^{-1}J$ in above equation, we have

$$\begin{aligned}\dot{V}_{4b} =& - k_1 \|z_1\|^2 - k_2 \|z_2\|^2 - k_3 \|z_3\|^2 - k_\omega \|z_\omega\|^2 - k_{\psi_1} z_{\psi_1}^2 - k_{\psi_2} z_{\psi_2}^2 \\ &+ \Big(\hat{a} u z_\omega^T \mathrm{S}(e_3) + z_{\psi_2} e_3^T W^{-1}\Big) J^{-1} (J\dot{\omega} - J\dot{\omega}_d)\end{aligned} \tag{5.79}$$

If the inertia matrix J was known, we could have algebraically cancelled the last term of the above equation by using (2.50b) and designing the torque input τ_q as done in [152]. However, due to J unknown, we are unable to design τ_q to algebraically cancel the last term. Therefore, at this point, will assume that $z_4 = \omega - \omega_d$ asymptotically converges to zero.

Theorem 5.4.1. *Consider the UAM system* (2.50) *and its approximation for the UAM-V* (5.1) *under Assumption 1 that the effect of attached robot arm is considered in the form of unknown constant parameter a and an unknown constant force d_f on*

the UAM-V, and assuming that ω, perfectly tracks ω_d which is defined by (5.66)*, then UAM-V position asymptotically converges to the desired position p_d.*

Proof. Since the theorem assumes that ω, perfectly tracks ω_d, therefore $z_4 = 0$ which implies $\dot{z}_4 = \dot{\omega} - \dot{\omega}_d = 0$. This reduces the (5.78) to

$$\dot{V}_{4b} = -k_1\|z_1\|^2 - k_2\|z_2\|^2 - k_3\|z_3\|^2 - k_\omega\|z_\omega\|^2 - k_{\psi_1}z_{\psi_1}^2 - k_{\psi_2}z_{\psi_2}^2$$

which is negative semi-definite. Using Barbalat's lemma, we can find that $\hat{a}$, d_f and u are bounded and the vector $[z_1^T, z_2^T, z_3^T, z_\omega^T]$ is asymptotically stable at origin. □

Let us now consider the convergence of z_4 defined above. Let us recall the definitions of $J_v, \tilde{J}_v, \hat{J}_v$ and the derivation of (5.26) from Section 5.2, which is re-written below

$$J\dot{z}_4 = -(\Phi(\omega) + \Psi(\dot{\omega}_d))(\hat{J}_v + \tilde{J}_v) + \tau_q + \hat{d}_\tau + \tilde{d}_\tau \tag{5.80}$$

Let $\hat{J}_v \in \mathbb{R}^6$ be the estimate of J_v, then estimation error is defined as $\tilde{J}_v = J_v - \hat{J}_v$. Consider the following Lyapunov function

$$V_{4c} = \frac{1}{2}z_4^T J z_4 + \frac{1}{2}\tilde{J}_v^T\Gamma^{-1}\tilde{J}_v + \frac{1}{2k_{d_\tau}}\tilde{d}_\tau^T\tilde{d}_\tau \tag{5.81}$$

Its time-derivative is given by

$$\dot{V}_{4c} = z_4^T J\dot{z}_4 + \tilde{J}_v^T\Gamma^{-1}\dot{\tilde{J}}_v + \frac{1}{k_{d_\tau}}\tilde{d}_\tau^T\dot{\tilde{d}}_\tau$$

substituting (5.80) and using $\dot{\tilde{J}}_v = -\dot{\hat{J}}_v$ and $\dot{\tilde{d}}_\tau = -\dot{\hat{d}}_\tau$ due to the constant assumption on J and d_η, we have

$$\dot{V}_{4c} = z_4^T\big(-(\Phi(\omega) + \Psi(\dot{\omega}_d))(\hat{J}_v + \tilde{J}_v) + \tau_q + \hat{d}_\tau + \tilde{d}_\tau\big) - \tilde{J}_v^T\Gamma^{-1}\dot{\hat{J}}_v - \frac{1}{k_{d_\tau}}\tilde{d}_\tau^T\dot{\hat{d}}_\tau$$

Adding and subtracting $k_4 z_4^T z_4$, and substituting $J_v = \hat{J}_v + \tilde{J}_v$

$$\begin{aligned}\dot{V}_{4c} = &- k_4\|z_4\|^2 + z_4^T\big(k_4 z_4 - (\Phi(\omega) + \Psi(\dot{\omega}_d))(\hat{J}_v + \tilde{J}_v) + \tau_q + \hat{d}_\tau + \tilde{d}_\tau\big)\\ &- \tilde{J}_v^T\Gamma^{-1}\dot{\hat{J}}_v - \frac{1}{k_{d_\tau}}\tilde{d}_\tau^T\dot{\hat{d}}_\tau\end{aligned} \tag{5.82}$$

Consider the following torque input

$$\tau_q = -k_4 z_4 + (\Phi(\omega) + \Psi(\dot{\omega}_d))\hat{J}_v - \hat{d}_\tau \tag{5.83}$$

along with the adaptation laws for estimation of unknowns are given below

$$\dot{\hat{J}}_v = -\Gamma(\Phi^T(\omega) + \Psi^T(\dot{\omega}_d))z_4 \tag{5.84a}$$

$$\dot{\hat{d}}_\tau = -k_{d_\tau} z_4 \tag{5.84b}$$

substituting them in (5.82), we get

$$\dot{V}_{4c} = -k_4 \|z_4\|^2 \tag{5.85}$$

Theorem 5.4.2. *Consider the angular velocity dynamics of a UAM-V* (5.1d) *under the assumption that* $\dot{d_\tau} = 0$ *and* $\dot{J}_v = 0$*, with a smooth time-varying trajectory consisting of* $\{\omega_d(t), \dot{\omega}_d(t)\}$ *given by* (5.67),(5.68) *and* (5.75)*. If the UAM-V torques are selected according to* (5.83)*, while the estimates* $\hat{d}_\tau$ *and* $\hat{J}_v$ *are given by the adaptive laws in* (5.84)*, then the tracking error* $z_4 = \omega - \omega_d$ *is asymptotically stable at origin while the estimates* $\hat{d}_\tau$ *and* $\hat{J}_v$ *and bounded.*

Proof. Consider the Lyapunov function V_{4c} in (5.81) which is a positive definite function and its derivative after substitution of control and adaptation laws, given in (5.85), is semi-negative definite due to absence $\tilde{d_\tau}$ and $\tilde{J}_v$ term in $\dot{V}_{4c}$. However, using Barbalat's lemma, we conclude that $z_4 \to 0$ as $t \to \infty$ while the estimates $\hat{d}_\tau$ and $\hat{J}_v$ are bounded. □

Remark 4. *It is worth noting that* z_4 *has some level of redundancy with respect to* z_ω*. However, they are both needed in this design. We saw in the previous control design in this chapter that we can select* ω_d *at* z_3 *level in* (5.16) *to avoid* z_ω*, and directly jump to* z_4*. However, at* z_4 *level of the backstepping, we need to know* $\dot{\omega}_d$*, which, if we want to avoid obtaining numerically, will again introduce unknown terms* $\tilde{d}_f$ *as they show up in* (5.58)*. Therefore, the solution is to obtain* $\dot{\omega}_d$ *after going to the level of* z_ω *and cancelling unknown terms and then integrate* $\dot{\omega}_d$ *to get* ω_d*. Also, another consideration would be to substitute* (5.1d) *in* (5.79) *and design for* τ_q*. This would significantly complicate the control design due to the introduction of* J^{-1} *with input* τ_q *and* $S(\omega)J\omega$ *which can no longer be written into the affine form i.e.,* $\Phi(\omega)J_v$*, leaving adaptive control technique inapplicable.*

Remark 5. *We notice that this approach, somewhat similar to the inner-outer loop design presented in the previous chapter, considers a perfect tracking for* ω *i.e.,* $z_4 = 0$ *in Theorem 5.4.1, which is later shown to be asymptotically stable in Theorem 5.4.2. Also, it assumes the angular velocity dynamics separately from the linear dynamics. However, it eliminates the need to take numerical derivative of UAM-V attitude reference trajectory* $\eta_d(t)$ *which is needed up to second derivative in case of inner-outer loop control design i.e.,* $\dot{\eta}_d(t), \ddot{\eta}_d(t)$ *and the need to obtain*

one numerical derivative $\dot{\omega}_d(t)$ as required in the approach used in Theorem 5.2.1, presented earlier in this chapter.

5.5 Simulation Results

Similar to Chapter 4, we perform simulation to test the effectiveness of the proposed methodologies. We again use the Simscape Multi-body environment with exact same model and its governing parameters and a 2-DOF robotic arm as descried in Section 4.4. We use exact same simulation scenarios as discussed in Section 4.4, i.e., figure-8 trajectory tracking for with a fixed arm, moving arm and a pick and place application. In addition to the coupling forces and torques due to the arm, we consider constant external disturbances acting on the UAM-V i.e., $\tau_{\text{ext},p} = \tau_{\text{ext},\eta} = [0.3, 0.3, 0.3]^T$. The model parameters and initial conditions are given in Section 4.4. The control gains used for this approach are lower and provided in Table 5.1.

Table 5.1: Control Gains

Parameter	Value	Parameter	Value
k_1	0.4	k_2	0.5
k_3	0.5	k_4	7
k_{d_f}	1	λ	3
Γ	0.001	k_{d_τ}	5
k_{ψ_1}	1	k_{ψ_2}	1

We performed simulation, for each of the control approaches presented in Section 5.1 and 5.4. However, for the same UAM-V and arm trajectory, model parameters, initial conditions, and control gains, there is no noticeable difference between the simulation results of two approaches. This is because both the approaches have a similar theory with the main difference in the method of obtaining $\dot{\omega}_d$ the term used in the torque controller. To be precise, the simulation results presented below are obtained using the Modified Control approach presented in Section 5.4.

The simulation results for the case where the arm is fixed are given in Figures 5.2-5.9. The position tracking errors shown in Figure 5.3 converge to a small neighbourhood around origin as time goes to infinity. The estimates for UAM-V mass admittance a, force disturbance d_f, inertia J_v and torque disturbance d_τ shown in Figures 5.4, 5.6 and 5.5, respectively, are bounded predicted by theory. The plot of 3D position in Figure 5.8 gives an overall picture of the tracking performance of the proposed control algorithm.

The detailed plots in case of figure-8 trajectory for UAM-V with moving arm are given in Figures 5.10-5.17. The position errors shown in Figure 5.11 exhibit

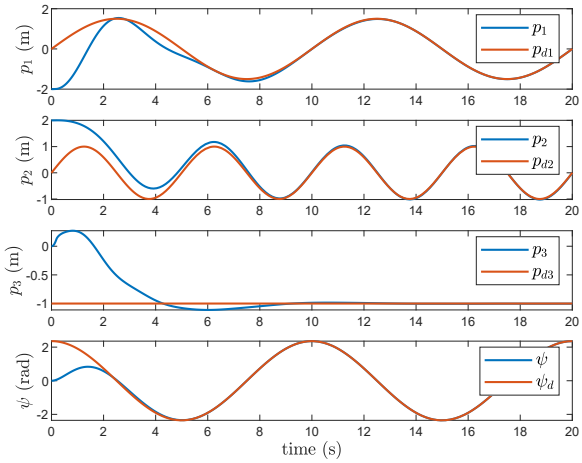


Figure 5.2: UAM-V pose in case of a fixed arm i.e., $\alpha_1 = \alpha_2 = 0$

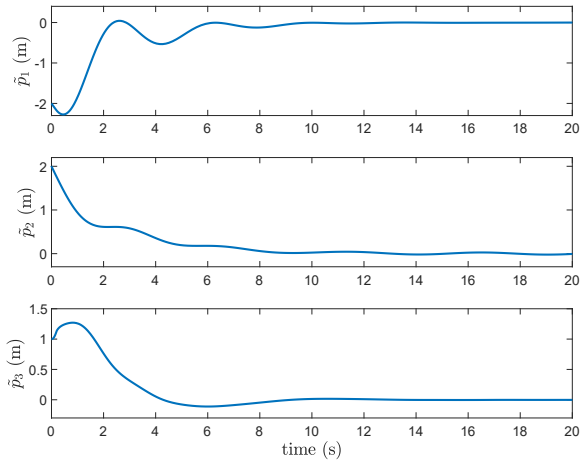


Figure 5.3: Position tracking errors in case of a fixed arm.

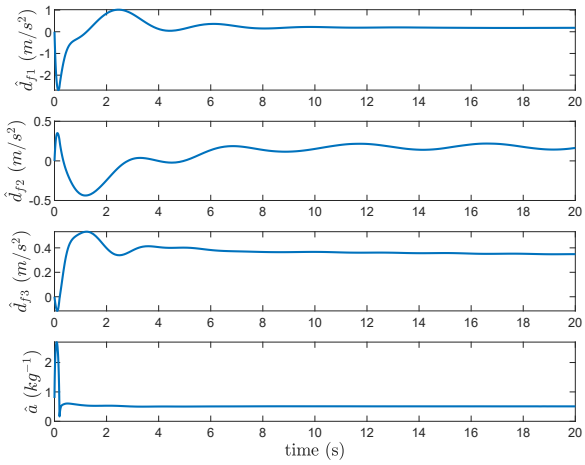


Figure 5.4: Estimates for force disturbance d_f and mass admittance a in case of a fixed arm.

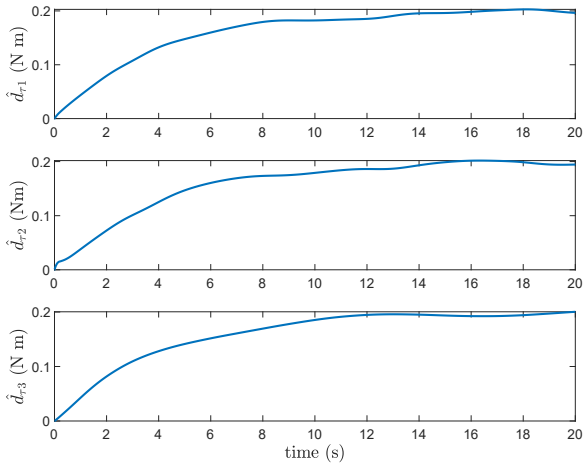


Figure 5.5: Estimates for torque disturbance d_τ in case of a fixed arm.

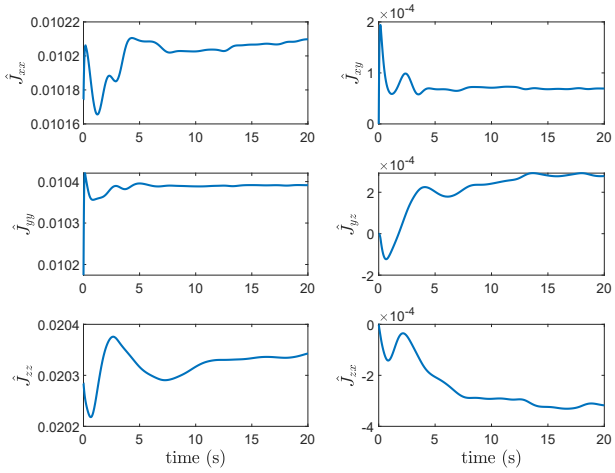


Figure 5.6: Estimate for Inertia matrix J in case of a fixed arm.

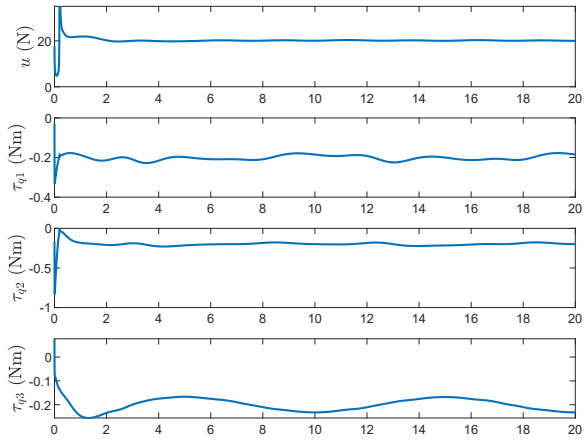


Figure 5.7: UAM-V inputs in case of a fixed arm.

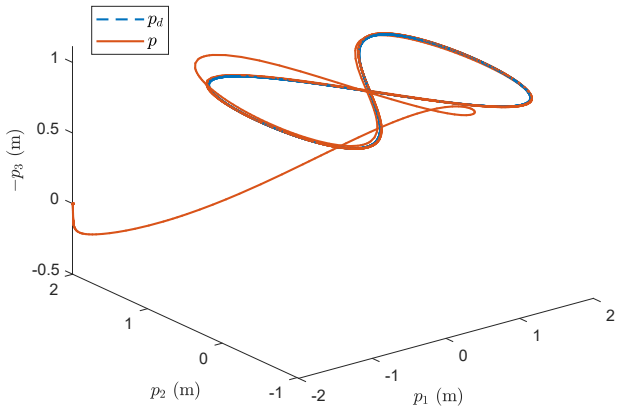


Figure 5.8: 3D UAM-V position in case of a fixed arm.

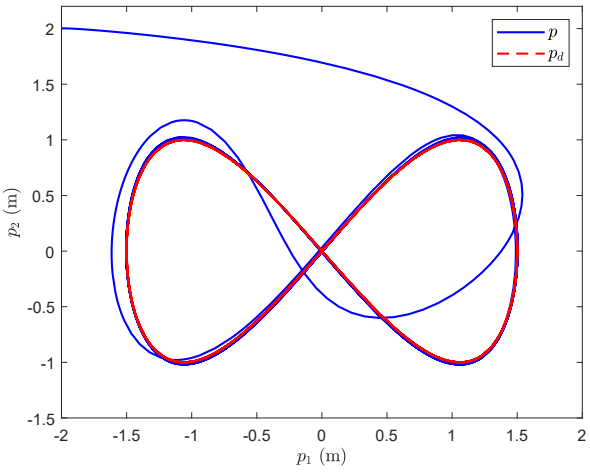


Figure 5.9: 2D UAM-V position in case of a fixed arm.

small oscillations close to origin. The plots in Figures 5.12, 5.13 and 5.14 show that estimated parameters and disturbances are bounded.

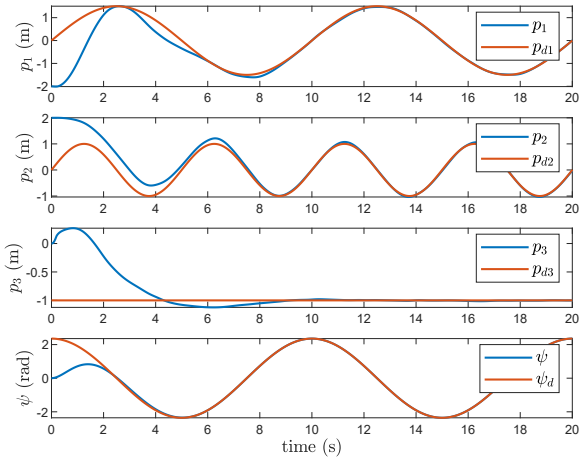


Figure 5.10: UAM-V pose in case of a moving arm.

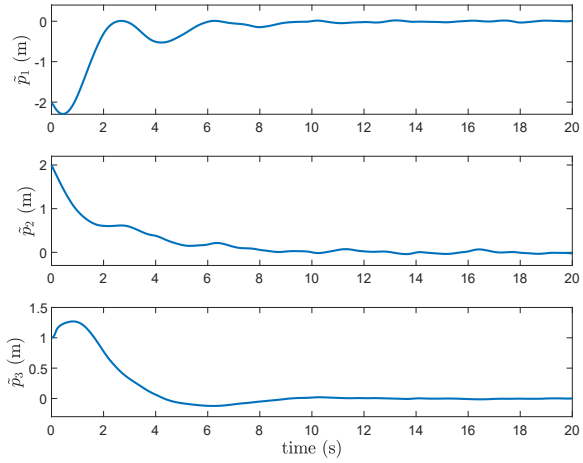


Figure 5.11: Position tracking errors in case of a moving arm.

Similar to the previous chapter, we now present the results for a practical application where a Quadrotor flies to a target, extends its arm, picks up the mass and then flies to another location and places a mass at the desired location. Pick and place trajectory, parameters and initial conditions are given in Section 4.4. The desired and reference values of UAM-V pose are shown in Figure 4.26 while the position errors are given in Figure 4.27 with labelling for each phase. It is evident that the tracking errors are convergent for the UAM-V tracking

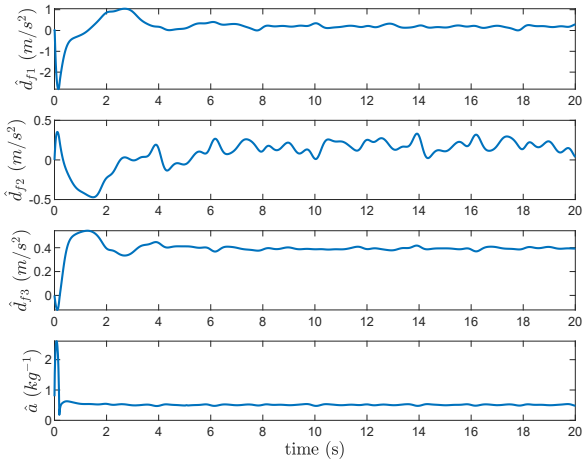


Figure 5.12: Estimates for d_f and a in case of a moving arm.

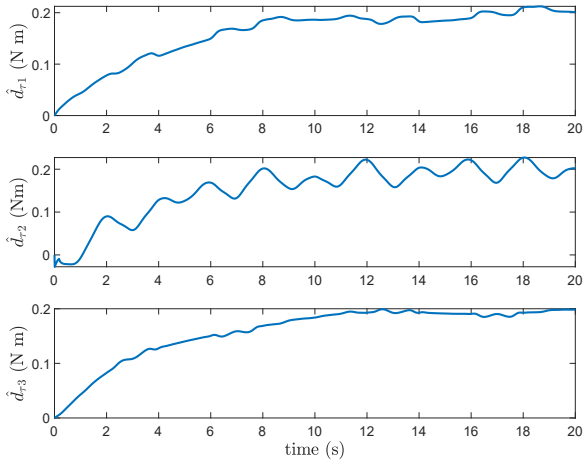


Figure 5.13: Estimates for torque disturbance d_τ in case of a moving arm.

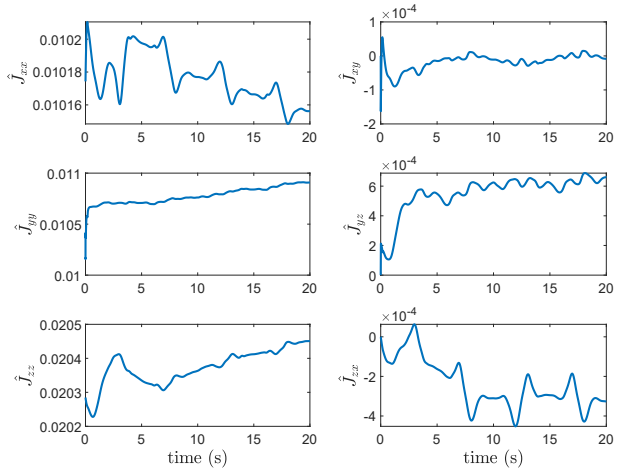


Figure 5.14: Estimate for Inertia matrix J in case of a moving arm.

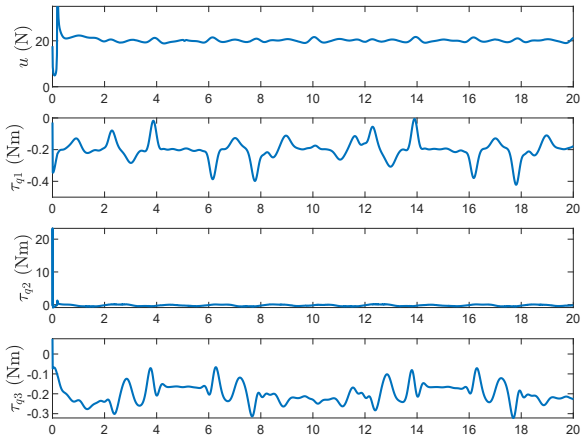


Figure 5.15: UAM-V inputs in case of a moving arm.

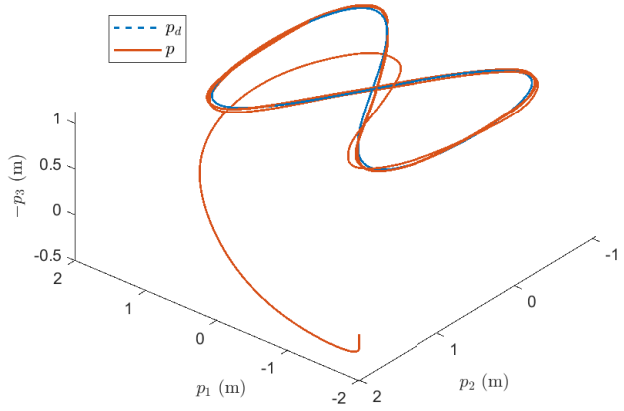


Figure 5.16: 3D UAM-V position in case of a moving arm.

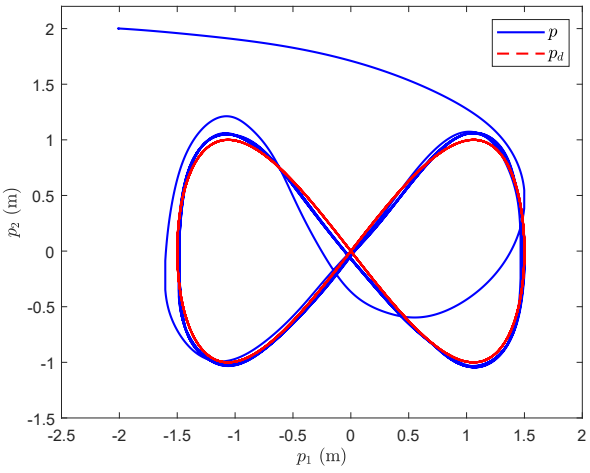


Figure 5.17: 2D UAM-V position in case of a moving arm.

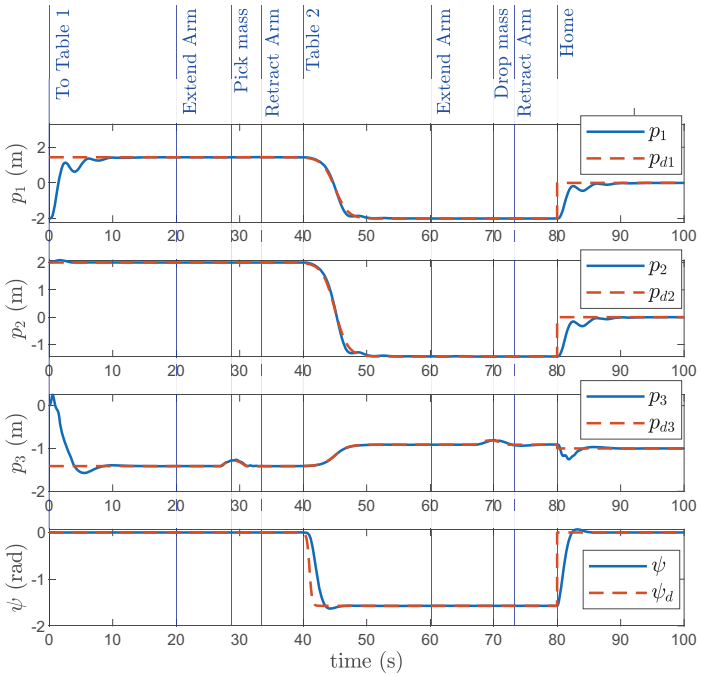


Figure 5.18: UAM-V pose for a pick and place application.

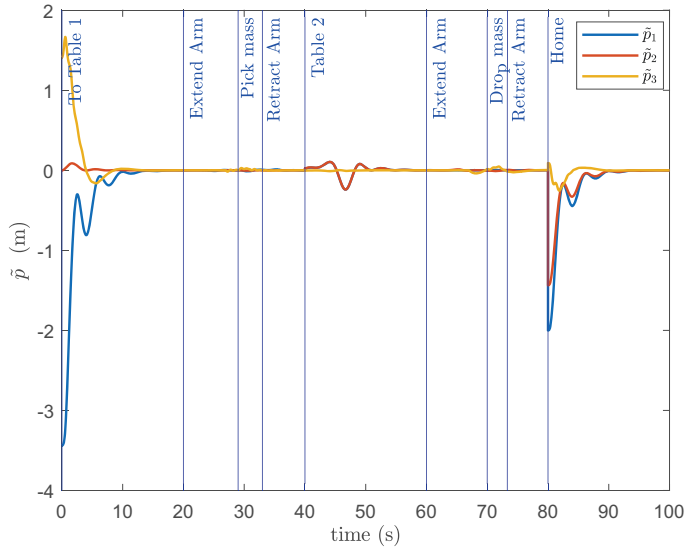


Figure 5.19: Position errors for pick and place application.

5.6 Simulation Comparison with Other Methods

This section compares different approaches by selecting similar controller gains for each approach and testing their performance for a figure-8 as desired trajectory (DT) with a continuously moving arm. This compares the performance of the proposed Inner-outer loop Adaptive Backstepping Control (IOABSC) approach presented in the previous chapter and the closed-loop Adaptive Backstepping Control (ABSC) presented in this chapter with another approach from the literature. All these approaches are backstepping based and test the control performance in the presence of unknown time-varying forces and torques acting on the UAM-V due to the arm, an unknown mass and unknown time-varying inertia of the UAM system.

First, we set a benchmark using a simple backstepping control approach where all parameters and disturbances are perfectly known and included in the controller for compensation. We call it All Known Backstepping Control (AKBSC). Its 2D position plot is shown in Figure 5.20a. Next, we consider the same approach, but we consider a time-varying disturbance applied to the system without compensating it in the controller. We call this controller Unknown Disturbance Backstepping Control (UDBSC). Its 2D plot is shown in Figure 5.20b which is unstable for any practical use. From Figure 5.21, although its absolute position error remains in a bounded region, the region is too large that it is even bigger than the whole flight envelope of the desired trajectory. This shows what it could lead to if arm motion is not compensated in the control design.

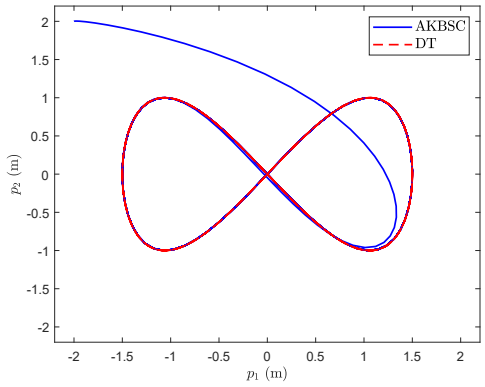
Next, we consider that the disturbance is observed and compensated in the controller. However, the parameters are considered unknown. We use the Disturbance observer-based backstepping control (DOBSC) approach from [1] without the consideration that a robot arm is attached to the system. That means we use UAM-V mass instead of the complete UAM in the controller, which has a 15% error with respect to the actual mass of the UAM. Also, the inertia used in the controller has a 10% error from the nominal value of UAM inertia which is a time-varying quantity when the arm is moving. The resulting 2D position plot is shown in Figure 5.20c.

We also compare the Inner-outer loop control approach presented in Chapter 4 and label it as Inner-outer loop adaptive Backstepping Control (IOABSC). The non-cascade approach presented in this chapter, which considers the dynamics of the UAM-V as a whole, is labelled as Adaptive Backstepping Control (ABSC) in the following figures. As mentioned earlier, we use the same initial conditions and controller gains for all these approaches, which are given in Table 3.3. For performance comparison, we use 2D plots for all approaches are shown in Figure 5.20 as well as the sum of the absolute values of tracking errors of the individual position coordinates, which are plotted in Figure 5.21. It is clear that ABSC performs better than the other approaches in the presence of parameter variation and time-varying

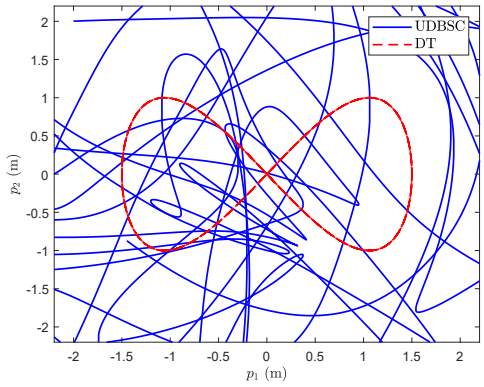
forces and torques acting on the UAM-V due to arm motion.

5.7 Conclusion

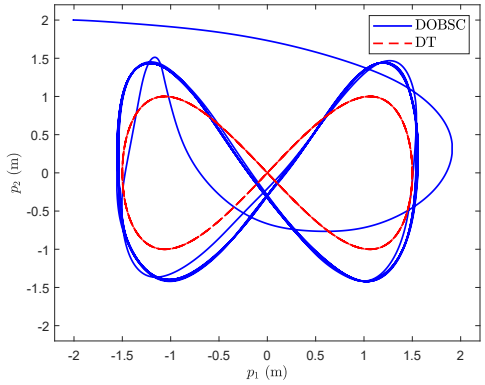
In this chapter, we have presented a control approach for a time-varying trajectory tracking of UAM-V position and yaw, which considers the entire dynamics of the UAM-V and directly provides thrust and torque inputs. A Lyapunov stability analysis guarantees the global asymptotic convergence of position errors at origin and local asymptotic convergence for yaw tracking error. We analyzed the approach and discussed the difficulty due to an algebraic derivative of a reference to a virtual control term in the backstepping process. We modified the approach to obtain the derivative algebraically, which results in a control design similar to the inner-outer loop control approach presented in Chapter 4 i.e., it provides a reference to be tracked by the angular velocity of the UAM-V and makes a perfect tracking assumption for the stability of the UAM-V position. However, in contrast to the inner-outer loop control approach, or the earlier approach presented in Section 5.1, it does not require taking any of the numerical derivatives. Simulations test the proposed approach on a UAM system for three scenarios i.e., figure-8 trajectory tracking with fixed arm, moving arm, and an object pick and place scenario. We also compare the simulations for five backstepping control designs based on the accommodations for unknown parameters and disturbances in control. We conclude that the approach presented in this chapter performs well for tracking a time-varying trajectory with a moving robot arm attached to the UAM-V.



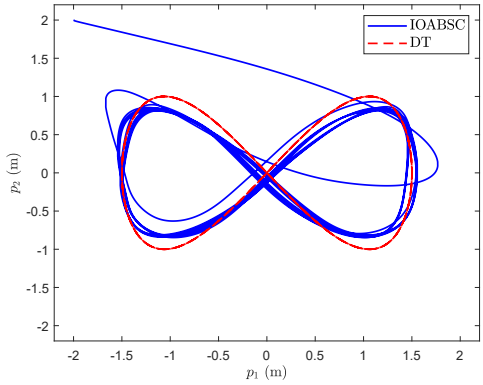
(a) Simple backstepping with all parameters and disturbances perfectly known



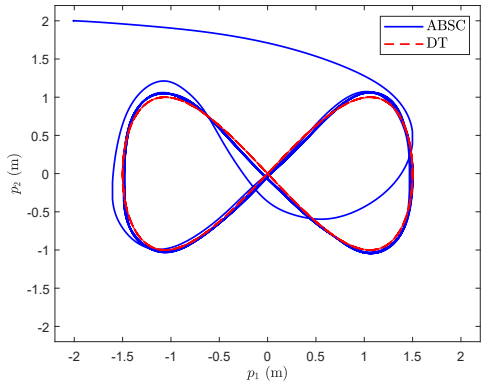
(b) Simple backstepping with known parameters but unknown time-varying disturbance



(c) Backstepping with disturbance observer [1] but unknown parameters



(d) Inner-outer loop Adaptive for unknown parameters and disturbances from Chapter 4



(e) Closed loop Adaptive for unknown parameters and disturbance presented in Section 5.4

Figure 5.20: 2D position plots for a figure-8 trajectory

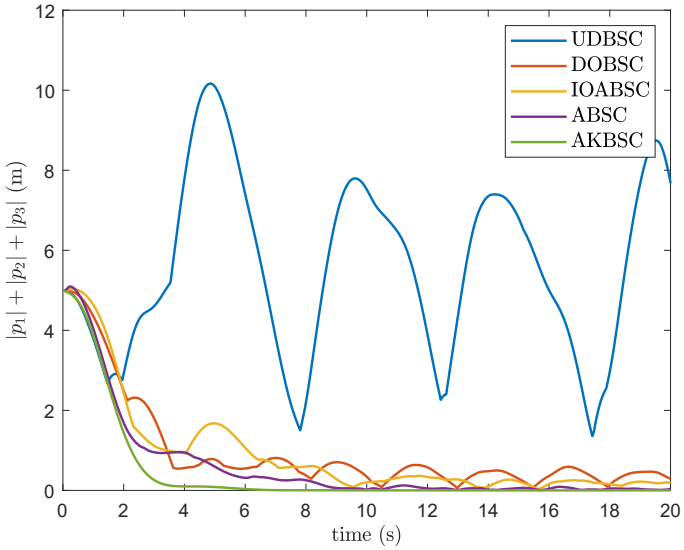


Figure 5.21: Comparison between the sum of the absolute position errors for different backstepping approaches.

Chapter 6

Conclusion

6.1 Summary and Conclusion

In this text, nonlinear and adaptive control has been used to develop new motion control algorithms for Unmanned Aerial Vehicles (UAVs). Specifically, a control design for Visual Servoing of a Camera equipped UAV for line following and two control designs for motion control of Unmanned Aerial Manipulators (UAMs) have been presented. The proposed approaches have been tested in simulation or experimentally to gauge their effectiveness.

This document opened with a thorough literature survey on Visual Servoing and UAVs. The context, purpose and application of the research presented in this thesis are outlined. After a detailed introduction, essential hardware and software aspects of a Quadrotor UAV have been detailed. In particular, the ANCL Quadrotor platforms that have been assembled, built and maintained during this Ph.D. have been described. After describing the physical platforms, the model of the quadrotor UAV was presented. Since the visual servoing approach uses a camera as a sensor, the pinhole camera model is employed to model the relation between a 3D point in space and its projection on a 2D image plane. A similar relationship for a 3D line is stated. Since the relative motion between the camera and a 3D object in space is reflected in the form of motion of its 2D projection in the image plane, the point motion kinematics in the image plane are derived in terms of the camera velocity. A line comprises points; therefore, the point motion kinematics are used to derive the line motion kinematics in the image plane. Since the angular motion of the camera results in a perspective projection of the target, the resulting line kinematics depend upon angular velocity in addition to their evident dependence on the translational velocity. This coupling of translational and rotational dynamics complicates control design. The problem is decoupled using a virtual camera frame. The line kinematics are re-derived in the virtual image plane, and the resulting line kinematics becomes independent of angular velocity. The line parameters measured in the image plane

are mathematically combined to define meaningful quantities related to the height, lateral distance and angle with target lines. These mathematical expressions called image moment features are expressed in the virtual camera frame. The image moment feature kinematics for lines (or simply line feature kinematics) depend on the camera velocity. Since the camera is rigidly attached to the UAV, the camera's velocity is, in fact, the velocity of the UAV. The line feature kinematics are then combined with the UAV dynamics to form the mathematical model used to derive a dynamic image-based visual servoing control.

An output feedback visual servoing control is designed for a linear target using the line feature kinematics. Due to the dynamic coupling of the line feature kinematics with the UAV through the camera velocity, UAV thrust and attitude are designed to directly control the feature error defined in the image plane. A horizontal target consisting of more than one line is considered, which is viewed by a downward-facing camera on the UAV.

The proposed IBVS has a number of important features. First, to make the UAV flyable in a GPS-denied environment, the UAV linear velocity is estimated from the feature errors. Second, the effect of a constant sensor bias is considered in the design and a constant external disturbance, such as wind, is also considered. Third, the control design is made robust against the variation of mass, which could change due to the addition of an unknown payload e.g., a sensor. Fourth, the design is also robust against changes in the thrust constant that depends on the battery voltage and results in reducing the vertical actuation of the UAV with the decrease in battery voltage during the flight. Lastly, the proposed design is robust to the changes in camera focal length.

The proposed controller for thrust and the references for UAV roll and pitch exponentially stabilize the feature and estimation errors. The proof uses a Lyapunov function analysis and LaSalle's invariance principle. Since inspection applications do not require acrobatic maneuvers, a small angle assumption simplifies the inner-loop control design. Also, a symmetric UAV body is considered, resulting in the inertia matrix that can be assumed diagonal. A controller similar to PID has been proposed for the inner-loop to provide UAV torques. The closed-loop stability of the complete system controlled by separate inner and outer loop controls is also considered. The exponential stability of the system is shown using the converse Lyapunov theorem in the stability analysis.

Both simulation and experiments have been performed to evaluate the effectiveness of the proposed controller. Computer simulations verify the theory and show exponential convergence of feature and estimation errors. Experiments have been performed using the ANCLQ 2.0 platform, where image processing to obtain features has been implemented in the onboard companion computer using ROS and

OpenCV, while the inner and outer controls are implemented on the flight controller. The performance of control is tested for a straight line as well as a line with a turn. The experiments show satisfactory performance in both cases.

Work on motion control of UAMs has also been presented in this thesis. A model of a UAM consisting of a multi-DOF robotic arm attached to a quadrotor UAV has been derived using the Euler-Lagrange approach. Unlike other models presented in the literature, the presented model preserves the vector or matrix representations i.e., vectors for position, attitude and velocities, and matrices for rotation or Jacobians in kinematic relationships are preserved. This provides a better understanding of the UAM model and allows identifying the effects of arm motion on the UAV and vice versa. The model is further analyzed and written in the structure of a nominal UAV model. The changes in UAM parameters and the coupling forces and torques exerted on the UAV due to the arm are identified. The forces and torques are also segregated and written as additive terms.

The important problem of "pick and place" involving aerial manipulation is considered in this thesis. Since a UAM system model is highly complicated and harder to control, a decentralized approach is used where the UAM model is divided into a UAV and an arm subsystem. The focus is kept on designing the controller for motion control of the UAV subsystem. Therefore, the UAV subsystem from the UAM model is further analyzed for the "pick and place" problem to make simplifying assumptions. It is concluded that the inertia matrix, forces and torques acting on the UAV subsystem model are either slowly varying or constant during most of the "pick and plane" operation and can be considered as constants in the control design process. This constant assumption allows the application of adaptive control in estimating these parameters, forces and torques.

The goal of the control design is to make the UAV position and heading track a time-varying trajectory in the presence of slowly varying parameters, disturbances and torques. Therefore, an adaptive control approach is proposed, considering only the states of the UAV subsystem are known while all system parameters, including mass and inertia matrix and forces and torques acting on the UAV, are unknown. Moreover, contrary to the common practice of taking inertia as a diagonal matrix found in literature, the full inertia matrix with all diagonal terms is considered. This is fundamentally important in the case of a UAM, where the attachment of a robot arm results in a shift in the UAM centre of mass. A UAV's inertia could be diagonal due to a symmetrical frame; however, a UAM due to its arm is generally not symmetric. Hence its inertia can not be generalized as a diagonal matrix.

Two adaptive nonlinear control approaches based on the backstepping design method are proposed for motion control of the UAM using the simplified dynamics of its UAV subsystem. The first approach considers an inner-outer loop structure

for controlling the UAV, where translational and rotational dynamics are treated separately. An outer-loop controller is designed to track a position trajectory, which provides mass and force disturbance estimates and the UAV thrust input and the reference values for roll and pitch angles. These references for roll and pitch, along with the desired heading from the user, serve as desired reference trajectories for UAV attitude. An inner-loop controller is designed for attitude tracking that provides estimates for the full inertia matrix and the torque disturbance along with the torque inputs of the UAV.

The tracking errors for the individual inner and outer loops are shown to be asymptotically stable at the origin using Lyapunov stability theory. Also, the parameter estimates are shown to be bounded. Simulation in the Simscape Multibody simulation environment has been performed to test the effectiveness of the proposed approach. Three different scenarios have been considered for simulation. First, a UAM system with the arm hanging straight down with its joint locked to zero angles is considered. Here the forces and torques acting on the UAV due to the arm, although time-varying, are very small, and a fast asymptotic convergence can be seen. In the second scenario, the arm continuously moves with its first link oscillating and the second link revolving. This results in time-varying inertia, forces and torques acting on the UAV and tests the control performance. In both these scenarios, the UAV position tracks a figure-8 trajectory.

The third simulation scenario is a 'pick and place' operation, where the UAM starts with a non-zero error in the initial condition. It controls its position to reach the desired reference location and extends its arm to pick an object. After picking the object, it retracts the arm and follows a time-varying trajectory to reach a new location. At the new location, it places the object using the same arm extension and retraction sequence and then returns to the origin. This tests the effectiveness of the proposed pose tracking control approach for a practical application. The simulation results show that tracking error is convergent in the case of figure-8 trajectory with fixed arm and the pick and place application, while it is bounded in case of figure-8 trajectory with moving arm.

The inner-outer loop control does not analyze the complete close loop stability; therefore, the second approach is geared towards removing this gap. Again, an adaptive backstepping-based approach is considered. In contrast to the inner-outer loop, this approach deals with the complete dynamics of the UAV subsystem at once and designs the UAV thrust and torques to track a time-varying position trajectory. In addition, it considers the full rotation matrix in the dynamics of the UAV instead of Euler angles and prevents the gimbal lock singularity at least in the position control. However, to complete the UAV's pose tracking, a yaw tracking control is introduced using Euler angles due to their intuitive understanding; the yaw itself is

an Euler angle and simple design.

The stability of the proposed control is proved using the Lyapunov theory, and the pose tracking errors are proved to be asymptotically stable at the origin. At the same time, the parameter estimations are shown to be bounded. However, this approach requires a derivative of a virtual control expression. The virtual control expression originates from the third level of the backstepping procedure when the UAV's angular velocity is considered as virtual control. Taking its algebraic derivative results in the reappearance of unknown parameters in the stability analysis and coupling between the unknown parameters that prevents deriving the adaptive parameter estimation laws. Therefore, a numerical derivative is recommended.

A numerical derivative could lead to noise amplification or result in singularity when a discontinuity is present in the reference variable. Therefore, the above approach may not be suitable where position, velocity or rotation estimates have significant noise or discontinuity in the reference. Another version of the approach is presented to resolve this, where a backstepping process is continued until the derivative of angular velocity shows up. The derivative of angular velocity is selected as a virtual control, and the expression of this virtual control is integrated to obtain a virtual control expression for angular velocity. However, this approach makes the design structure similar to inner-outer loop control and assumes perfect angular velocity tracking. Despite this structure, the approach is still better than the inner-outer loop approach, requiring taking two derivatives of the attitude reference trajectory. In contrast, this approach does not require taking any derivatives.

Similar to the inner-outer loop simulations, this approach is tested for three scenarios, two involving figure-8 trajectory with a fixed arm and a moving arm, and the third for a 'pick and place application. The simulation results show asymptotic stability of the tracking error in the case of a fixed arm, bounded error in the case of a time-varying arm motion, and asymptotic stability in the case of pick and place operation. Also, the simulation results in the case of a figure-8 trajectory and moving arm are compared with those of the inner-outer loop approach and other approaches in the literature. The results show that this approach performs better than other approaches in the case of a time-varying trajectory for both the UAV and the arm.

6.2 Limitations and Future Work

In the case of Visual Servoing for a line following case, a horizontal planar horizontal target is assumed. However, in reality, the target may not be horizontal or planar, e.g., a section of a pipeline at a 45-degree angle with respect to the ground. This assumption could be removed in future work by taking into account the perspective

projection of lines on the image, when the roll and pitch of the UAV are perfectly zero. The camera field of view is an essential constraint of the proposed visual servoing approach. The approach may fail when the target moves outside the camera's field of view because the vision system could only detect the target if it is present in the image. This problem can be addressed in the future by using an artificial intelligence-based supervisory of the control that could undo a control action that resulted in a target location outside the camera's field of view.

The proposed approach only estimates the lateral velocity with respect to a line's target, while the user manually adjusts the velocity along the lines by selecting the appropriate pitch of the UAV. This limitation could be removed using an optical flow sensor to estimate the velocity along the line using the texture or vegetation in the background. An alternate approach could be to fuse the accelerometer measurements with known distances between features of a target detectable by a camera, e.g., , joints between pipes of known length in case of a pipeline or distances between the transmission line towers. The visual servoing control makes a small angle assumption in the inner-loop control design; this assumption guarantees only the local exponential stability. This assumption could be removed, and a better control could be designed with global stability. The experimental validation of this approach is only done in a lab environment using lines made up of points. The experimental evaluation can also be improved in future work, and it can be tested on a real transmission line or a pipeline.

A fundamental limitation of the UAM control using the decentralized structure is the assumption that time-varying forces and torques acting on the UAV due to arm motion are considered constants. This assumption is required in adaptive control, which results in bounded errors in the case of time-varying forces and torques. The assumption is practically valid for a lighter load and a slowly moving arm. A fast-moving arm with a heavy load could have large variations in disturbances and inertia, which could make the practical stability of the system futile. However, a control design resulting in asymptotic convergence for time-varying disturbances could be explored in future work for acrobatic flight and fast manipulation tasks. The inner-outer loop control approach presented only considers the stability of inner and outer loops individually, while the outer-loop assumes perfect inner-loop tracking. This assumption could be removed in future work, and the closed-loop stability of the system when the inner and outer loops are acting together could be analyzed.

Another limitation of the inner-outer loop control design is that the inner-loop control requires a second derivative of the reference attitude trajectory. Since the outer-loop provides the attitude reference, the attitude reference depends upon the measured UAV position and velocity. If the position or the velocity measurement is noisy, the resulting attitude reference from the outer-loop has noise. The first and

second derivatives of the attitude reference would amplify the noise. Therefore, a low pass filter is used in series with each derivative to reduce the noise. However, it is not considered in the control design. The dynamics of the low pass filter should be included in the model in future work. Also, the inner-loop implementation is based on attitude expressed in Euler angles, which involve the gimbal lock problem. In future, this could be removed by using unit quaternions instead of Euler angles.

The closed-loop design that considers the full UAV subsystem dynamics at once addresses some limitations of the inner-outer loop control. However, it still requires one derivative of the reference of virtual angular velocity. This has been modified to remove the derivative requirement. However, the resulting control approach is similar to the inner-outer loop approach. This could be further addressed in future work where complete closed-loop stability is proved without the need for numerical derivatives. This control also assumes the UAV thrust and mass admittance are greater than zero. Although it has been experimentally tested that these parameters estimates remain positive even for aggressive trajectories, there is no theory to support it. The stability dynamics should be analyzed to confirm their maximum and minimum possible values. Although gimbal lock is prevented in the position tracking control, the yaw control is based on Euler angles, which could again result in gimbal lock. In future work, the yaw control could be replaced with unit quaternions.

The effects of UAV motion on the arm have not been addressed, and only a simple PID controller is used to control the arm motion in simulations. This could be replaced with a more sophisticated control, or a similar controller could be designed for arm motion. The closed-loop stability of the decentralized control can be evaluated in future work. Also, the contact dynamics of the UAM during an interaction are neither considered in theory nor implemented in simulations. Future work includes the study of contact dynamics and impedance-based force/torque control. Lastly, the UAM control has only been tested through simulations. The future work involves its implementation on the ANCLQ 3.0 quadrotor platform using a 3D printed robotic arm.

Bibliography

[1] A. Moeini, A. Lynch, and Q. Zhao, "Disturbance observer-based nonlinear control of a quadrotor UAV," *Advanced Control for Applications: Engineering and Industrial Systems*, vol. 2, no. 1, pp. 1–20, 2019.

[2] H. Cui, Z. Lin, and J. Zhang, "Research on low altitude image acquisition system," in *Computer And Computing Technologies In Agriculture, Volume I*, ser. The International Federation for Information Processing, D. Li, Ed. Springer US, 2008, vol. 258, pp. 95–102.

[3] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner, "Networked UAVs as aerial sensor network for disaster management applications," *e & i Elektrotechnik und Informationstechnik*, vol. 127, no. 3, pp. 56–63, 2010.

[4] Y. Zhang, X. Yuan, W. Li, and S. Chen, "Automatic power line inspection using uav images," *Remote Sensing*, vol. 9, no. 8, 2017.

[5] C. Gómez and D. R. Green, "Small unmanned airborne systems to support oil and gas pipeline monitoring and mapping," *Arabian Journal of Geosciences*, vol. 10, no. 9, p. 202, May 2017.

[6] M. F. da Silva, L. M. Honório, A. L. M. Marcato, V. F. Vidal, and M. F. Santos, "Unmanned aerial vehicle for transmission line inspection using an extended kalman filter with colored electromagnetic interference," *ISA Transactions*, vol. 100, pp. 322–333, 2020.

[7] X. Li, Z. Li, H. Wang, and W. Li, "Unmanned aerial vehicle for transmission line inspection: Status, standardization, and perspectives," *Frontiers in Energy Research*, vol. 9, p. 336, 2021.

[8] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw Hill, 2008.

[9] A. D. Wu, E. N. Johnson, M. Kaess, F. Dellaert, and G. Chowdhary, "Autonomous flight in GPS-denied environments using monocular vision and inertial sensors," *Journal of Aerospace Information Systems*, vol. 10, no. 4, pp. 172–186, 2013.

[10] I. F. M. Bernal, "On-board visual control algorithms for unmanned aerial vehicles," Ph.D. dissertation, Universidad Politécnica de Madrid, Madrid, Spain, 2011.

[11] F. Kendoul, K. Nonami, I. Fantoni, and R. Lozano, "An adaptive vision-based autopilot for mini flying machines guidance, navigation and control," *Advanced Robotics*, vol. 27, no. 3, pp. 165–188, Oct. 2009.

[12] J. Artieda, J. M. Sebastian, P. Campoy, J. F. Correa, I. F. Mondragon, C. Martinez, and M. Olivares, "Visual 3-D SLAM from UAVs," *Journal of Intelligent Robotic Systems*, vol. 55, no. 4-5, pp. 299–321, Aug. 2009.

[13] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[14] M. Przystupa, M. Dehghan, M. Jagersand, and A. R. Mahmood, "Analyzing neural jacobian methods in applications of visual servoing and kinematic control," 2021. [Online]. Available: https://arxiv.org/abs/2106.06083

[15] C. Kanellakis and G. Nikolakopoulos, "Survey on computer vision for UAVs: Current developments and trends," *Journal of Intelligent Robotic Systems*, vol. 87, no. 1, pp. 141–168, Jul. 2017.

[16] P. Pounds, D. Bersak, and A. Dollar, "Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control," *Autonomous Robots*, vol. 33, no. 1–2, pp. 129–142, 2012.

[17] S. Kim, S. Choi, H. Kim, J. Shin, H. Shim, and H. J. Kim, "Robust control of an equipment-added multirotor using disturbance observer," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1524–1531, July 2018.

[18] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 2668–2673.

[19] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 46–64, Aug 2014.

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

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

[22] "AIRobots EU collaboration," 2010 - 2013, accessed Mar 1, 2019. [Online]. Available: http://airobots.dei.unibo.it/

[23] L. Marconi, R. Naldi, A. Torre, J. Nikolic, C. Huerzeler, G. Caprari, E. Zwicker, B. Siciliano, V. Lippiello, R. Carloni, and S. Stramigioli, "Aerial

service robots: An overview of the AIRobots activity," in *Proceedings of the 2012 2nd International Conference on Applied Robotics for the Power Industry (CARPI)*, Sep. 2012, pp. 76–77.

[24] C. Huerzeler, R. Naldi, V. Lippiello, R. Carloni, J. Nikolic, K. Alexis, L. Marconi, and R. Siegwart, "AIRobots: Innovative aerial service robots for remote inspection by contact," in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 2080–2080.

[25] L. Marconi, F. Basile, G. Caprari, R. Carloni, P. Chiacchio, C. Hurzeler, V. Lippiello, R. Naldi, J. Nikolic, B. Siciliano, S. Stramigioli, and E. Zwicker, "Aerial service robotics: The AIRobots perspective," in *Proceedings of the 2012 2nd International Conference on Applied Robotics for the Power Industry (CARPI)*, Sep. 2012, pp. 64–69.

[26] "ARCAS EU collaboration," 2011 - 2015, accessed Mar 1, 2019. [Online]. Available: http://www.arcas-project.eu/

[27] "AEROARMS EU collaboration," 2015 - 2019, accessed Mar 1, 2019. [Online]. Available: https://aeroarms-project.eu/

[28] A. Ollero, G. Heredia, A. Franchi, G. Antonelli, K. Kondak, A. Sanfeliu, A. Viguria, J. R. Martinez-de Dios, F. Pierri, J. Cortes, A. Santamaria-Navarro, M. A. Trujillo Soto, R. Balachandran, J. Andrade-Cetto, and A. Rodriguez, "The AEROARMS project: Aerial robots with advanced manipulation capabilities for inspection and maintenance," *IEEE Robotics and Automation Magazine*, vol. 25, no. 4, pp. 12–23, Dec 2018.

[29] W. Dong, G.-Y. Gu, X. Zhu, and H. Ding, "High-performance trajectory tracking control of a quadrotor with disturbance observer," *Sensors and Actuators A: Physical*, vol. 211, pp. 67–77, May 2014.

[30] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles," *IEEE Control Systems Magazine*, vol. 33, no. 1, pp. 61–75, 2013.

[31] N. Cao and A. F. Lynch, "Inner-outer loop control with constraints for rotary-wing UAVs," in *Proceedings of the 2015 International Conference on Unmanned Aircraft Systems*, Denver, CO, Jun. 2015, pp. 294–302.

[32] D. Cabecinhas, R. Cunha, and C. Silverstre, "A nonlinear quadrotor trajectory tracking controller with disturbance rejection," *Control Engineering Practice*, vol. 26, pp. 1–10, May 2014.

[33] D. Jones, "Power line inspection - a UAV concept," in *Proceedings of the 2005 The IEE Forum on Autonomous Systems (Ref. No. 2005/11271)*, Nov 2005, p. 8.

[34] L. I. Kochetkova, "Pipeline monitoring with unmanned aerial vehicles," *Journal of Physics: Conference Series*, vol. 1015, no. 4, p. 042021, 2018.

[35] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *Journal of Field Robotics*, vol. 29, no. 2, pp. 315–378, Mar./Apr. 2012.

[36] "Drones - a revolution in transmission line inspection and maintenance," MIR Innovation, Hydro-Quebec, accessed 14 August 2018. [Online]. Available: http://mir-innovation.hydroquebec.com/mir-innovation/en/transmission-solutions-uav.html

[37] "Unmanned aerial inspection," ULC Robotics, accessed 14 August 2018. [Online]. Available: http://ulcrobotics.com/services/unmanned-aerial-utility-inspection-services/

[38] J. Oh and C. Lee, "3D power line extraction from multiple aerial images," *Sensors*, vol. 17, no. 10, 2017.

[39] C. Deng, S. Wang, Z. Huang, Z. Tan, and J. Liu, "Unmanned aerial vehicles for power line inspection: A cooperative way in platforms and communications," *JCM*, vol. 9, pp. 687–692, 2014.

[40] L. Matikainen, M. Lehtomäki, E. Ahokas, J. Hyyppä, M. Karjalainen, A. Jaakkola, A. Kukko, and T. Heinonen, "Remote sensing methods for power line corridor surveys," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 119, pp. 10 – 31, 2016.

[41] D. Hausamann, W. Zirnig, G. Schreier, and P. Strobl, "Monitoring of gas pipelines : a civil UAV application," *Aircraft Engineering and Aerospace Technology*, vol. 77, no. 5, pp. 352–360, 2005.

[42] F. Chaumette and S. Hutchinson, "Visual servo control. II. Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, Mar. 2007.

[43] L. Carrillo, G. Flores Colunga, G. Sanahuja, and R. Lozano, "Quad rotorcraft switching control: An application for the task of path following," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1255–1267, Jul. 2014.

[44] E. Rondon, L.-R. Garcia-Carrillo, and I. Fantoni, "Vision-based altitude, position and speed regulation of a quadrotor rotorcraft," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 628–633.

[45] H. Xie, "Dynamic visual servoing of rotary wing unmanned aerial vehicles," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Feb. 2016.

[46] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, Oct. 1996.

[47] B. Espiau, "Effect of camera calibration errors on visual servoing in robotics," in *Experimental Robotics III*, T. Yoshikawa and F. Miyazaki, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 182–192.

[48] O. A. Ramirez and M. Jagersand, "Practical considerations of uncalibrated visual servoing," in *Proceedings of the 2016 13th Conference on Computer and Robot Vision (CRV)*, 2016, pp. 164–169.

[49] M. Gridseth, O. Ramirez, C. P. Quintero, and M. Jagersand, "Vita: Visual task specification interface for manipulation with uncalibrated visual servoing," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*, 2016, pp. 3434–3440.

[50] F. Janabi-Sharifi, L. Deng, and W. J. Wilson, "Comparison of basic visual servoing methods," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 5, pp. 967–983, 2011.

[51] H. Michel and P. Rives, "Singularities in the determination of the situation of a robot effector from the perspective view of 3 points," INRIA, Research Report RR-1850, 1993. [Online]. Available: https://hal.inria.fr/inria-00074822

[52] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, p. 381–395, jun 1981.

[53] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, Jun. 1992.

[54] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 713–723, Aug. 2004.

[55] T. Hamel and R. Mahony, "Visual servoing of an under-actuated dynamic rigid-body system: an image-based approach," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 187–198, 2002.

[56] O. Bourquardez, R. Mahony, T. Hamel, and F. Chaumette, "Stability and performance of image based visual servo control using first order spherical image moments," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 2006, pp. 4304–4309.

[57] T. Hamel and R. Mahony, "Image based visual servo control for a class of aerial robotic systems," *Automatica*, vol. 43, no. 11, pp. 1975–1983, Nov. 2007.

[58] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Transactions on Robotics and Automation*, vol. 24, no. 2, pp. 331–340, 2008.

[59] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck, "Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 743–749, Jun. 2009.

[60] H. Xie and A. F. Lynch, "State transformation-based dynamic visual servoing for an unmanned aerial vehicle," *International Journal of Control*, vol. 89, no. 5, pp. 892–908, 2016.

[61] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., ser. Cambridge Books Online. Cambridge, England: Cambridge University Press, 2003, vol. 1.

[62] N. Metni, T. Hamel, and F. Derkx, "Visual tracking control of aerial robotic systems with adaptive depth estimation," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 6078–6084.

[63] S. Benhimane and E. Malis, "Homography-based 2D visual tracking and servoing," *International Journal of Robotics Research*, vol. 26, no. 7, pp. 661–676, 2007.

[64] H. de Plinval, P. Morin, P. Mouyon, and T. Hamel, "Visual servoing for underactuated VTOL UAVs: A linear, homography-based approach," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 3004–3010.

[65] ——, "Visual servoing for underactuated VTOL UAVs: a linear, homography-based framework," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 16, pp. 2285–2308, Apr. 2013.

[66] R. Ozawa and F. Chaumette, "Dynamic visual servoing with image moments for an unmanned aerial vehicle using a virtual spring approach," *Advanced Robotics*, vol. 27, no. 9, pp. 683–696, 2013.

[67] ——, "Dynamic visual servoing with image moments for a quadrotor using a virtual spring approach," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 5670–5676.

[68] D. Lee, T. Ryan, and H. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, Saint Paul, MN, 2012, pp. 971–976.

[69] D. Lee, H. Lim, H. Kim, Y. Kim, and K. Seong, "Adaptive image-based visual servoing for an underactuated quadrotor system," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 4, pp. 1335–1353, 2012.

[70] H. Jabbari, G. Oriolo, and H. Bolandi, "Dynamic IBVS control of an underactuated UAV," in *Proceedings of the 2012 IEEE International Conference on Robotics and Biomimetics*, Dec 2012, pp. 1158–1163.

[71] H. Jabbari Asl, G. Oriolo, and H. Bolandi, "An adaptive scheme for image-based visual servoing of an underactuated UAV," *International Journal of Robotics and Automation*, vol. 29, no. 1, pp. 92–104, 2014.

[72] H. Xie, A. F. Lynch, and M. Jagersand, "Dynamic IBVS of a rotary wing UAV using line features," *Robotica*, vol. 34, no. 9, pp. 2009–2026, 2014.

[73] H. Xie, G. Fink, A. F. Lynch, and M. Jagersand, "Adaptive dynamic visual servoing of a UAV," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 5, pp. 2529–2538, 2016.

[74] J. Li, H. Xie, R. Ma, and K. H. Low, "Output feedback image-based visual servoing of rotorcrafts," *Journal of Intelligent Robotic Systems*, Apr 2018. [Online]. Available: https://doi.org/10.1007/s10846-018-0826-4

[75] J. R. Azinheira and P. Rives, "Image-based visual servoing for vanishing features and ground lines tracking: Application to a UAV automatic landing," *International Journal of Optomechatronics*, vol. 2, no. 3, pp. 275–295, Sep. 2008.

[76] R. Mahony and T. Hamel, "Visual servoing using linear features for underactuated rigid body dynamics," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 2001, pp. 1153–1158.

[77] ——, "Image-based visual servo control of aerial robotic systems using linear image features," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 227–239, 2005.

[78] S. Cho and D. H. Shim, "Sampling-based visual path planning framework for a multirotor uav," *International Journal of Aeronautical and Space Sciences*, Mar 2019.

[79] G. Fink, "Computer vision-based motion control and state estimation for unmanned aerial vehicles (UAVs)," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, 2018.

[80] D. Zheng, H. Wang, J. Wang, S. Chen, W. Chen, and X. Liang, "Image-based visual servoing of a quadrotor using virtual camera approach," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 972–982, 2017.

[81] F. Castelli, S. Michieletto, S. Ghidoni, and E. Pagello, "A machine learning-based visual servoing approach for fast robot control in industrial setting," *International Journal of Advanced Robotic Systems*, vol. 14, no. 6, p. 10, 2017.

[82] Z. Miljković, M. Mitić, M. Lazarević, and B. Babić, "Neural network reinforcement learning for visual control of robot manipulators," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1721–1736, 2013.

[83] M. Kang, H. Chen, and J. Dong, "Adaptive visual servoing with an uncalibrated camera using extreme learning machine and q-leaning," *Neurocomputing*, vol. 402, pp. 384–394, 2020.

[84] Y. Ghandi and M. Davoudi, "Visually guided manipulator based on artificial neural networks," *IETE Journal of Research*, vol. 65, no. 2, pp. 275–283, 2019.

[85] P. Durdevic and D. Ortiz-Arroyo, "A deep neural network sensor for visual servoing in 3d spaces," *Sensors*, vol. 20, no. 5, 2020.

[86] P. M. Wyder, Y.-S. Chen, A. J. Lasrado, R. J. Pelles, R. Kwiatkowski, E. O. A. Comas, R. Kennedy, A. Mangla, Z. Huang, X. Hu, Z. Xiong, T. Aharoni, T.-C. Chuang, and H. Lipson, "Autonomous drone hunter operating by deep learning and all-onboard computations in GPS-denied environments," *PLOS ONE*, vol. 14, no. 11, pp. 1–18, 11 2019.

[87] A. Albers, S. Trautmann, T. Howard, , M. Frietsch, and C. Sauter, "Semi-autonomous flying robot for physical interaction with environment," in *Proceedings of the 2010 IEEE Conference on Robotics, Automation and Mechatronics*, June 2010, pp. 441–446.

[88] H. Nguyen and D. Lee, "Hybrid force/motion control and internal dynamics of quadrotors for tool operation," in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 3458–3464.

[89] S. Bellens, J. De Schutter, and H. Bruyninckx, "A hybrid pose / wrench control framework for quadrotor helicopters," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 2269–2274.

[90] A. Y. Mersha, S. Stramigioli, and R. Carloni, "Variable impedance control for aerial interaction," in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 3435–3440.

[91] H.-N. Nguyen, C. Ha, and D. Lee, "Mechanics, control and internal dynamics of quadrotor tool operation," *Automatica*, vol. 61, pp. 289 – 301, 2015.

[92] C. Papachristos, K. Alexis, and A. Tzes, "Efficient force exertion for aerial robotic manipulation: Exploiting the thrust-vectoring authority of a tri-tiltrotor uav," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, May 2014, pp. 4500–4505.

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

[94] G. Darivianakis, K. Alexis, M. Burri, and R. Siegwart, "Hybrid predictive control for aerial robotic physical interaction towards inspection operations," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, May 2014, pp. 53–58.

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

[96] S. B. Backus, L. U. Odhner, and A. M. Dollar, "Design of hands for aerial manipulation: Actuator number and routing for grasping and perching," in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 34–40.

[97] V. Ghadiok, J. Goldin, and W. Ren, "Autonomous indoor aerial gripping using a quadrotor," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 4645–4651.

[98] V. Lippiello, G. A. Fontanelli, and F. Ruggiero, "Image-based visual-impedance control of a dual-arm aerial manipulator," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1856–1863, July 2018.

[99] M. Orsag, C. Korpela, S. Bogdan, and P. Oh, "Valve turning using a dual-arm aerial manipulator," in *Proceedings of the 2014 International Conference on Unmanned Aircraft Systems*, May 2014, pp. 836–841.

[100] T. W. Danko, K. P. Chaney, and P. Y. Oh, "A parallel manipulator for mobile manipulating uavs," in *Proceedings of the 2015 IEEE International Conference on Technologies for Practical Robot Applications*, May 2015, pp. 1–6.

[101] P. Grau, A. Suarez, V. M. Vega, A. Rodriguez-Castaño, and A. Ollero, "Design of a high performance dual arm aerial manipulator," in *Proceedings of the ROBOT 2017: Third Iberian Robotics Conference*, A. Ollero, A. Sanfeliu, L. Montano, N. Lau, and C. Cardeira, Eds. Cham: Springer International Publishing, 2018, pp. 730–741.

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

[103] V. Lippiello and F. Ruggiero, "Cartesian impedance control of a UAV with a robotic arm," in *Proceedings of the 10th International IFAC Symposium on Robot Control*, 2012, pp. 704–709.

[104] A. Khalifa, M. Fanni, A. Ramadan, and A. Abo-Ismail, "Modeling and control of a new quadrotor manipulation system," in *Proceedings of the 2012 First International Conference on Innovative Engineering Systems*, Dec 2012, pp. 109–114.

[105] M. Orsag, C. Korpela, and P. Oh, "Modeling and control of mm-uav: Mobile manipulating unmanned aerial vehicle," *Journal of Intelligent Robotic Systems*, vol. 69, no. 1, pp. 227–240, Jan 2013.

[106] C. Coulombe, D. Saussié, and S. Achiche, "Modeling and gain-scheduled control of an aerial manipulator," *International Journal of Dynamics and Control*, 2021.

[107] K. Bouzgou, L. Benchikh, L. Nouveliere, Y. Bestaoui, and Z. Ahmed-foitih, "A novel aerial manipulation design, modelling and control for geometric COM compensation," in *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics*, vol. 2, Prague, Czech Republic, Jul. 2019, pp. 475–482.

[108] J. Serafini, L. Ding, and H. Wu, "Dynamical modelling and robust control for an unmanned aerial robot using hexarotor with 2-DOF manipulator," *International Journal of Aerospace Engineering*, vol. 2019, p. 5483073, 2019.

[109] A. E. 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 2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 4916–4921.

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

[111] S. Kannan, M. A. Olivares-Mendez, and H. Voos, "Modeling and control of aerial manipulation vehicle with visual sensor," *IFAC Proceedings Volumes*, vol. 46, no. 30, pp. 303 – 309, 2013.

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

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

[114] H. Yang and D. Lee, "Dynamics and control of quadrotor with robotic manipulator," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, May 2014, pp. 5544–5549.

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

[116] Z. Samadikhoshkho, S. Ghorbani, F. Janabi-Sharifi, and K. Zareinia, "Nonlinear control of aerial manipulation systems," *Aerospace Science and Technology*, vol. 104, p. 105945, 2020.

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

[118] 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, pp. 693–707, Jan 2014.

[119] B. Yang, Y. He, J. Han, and G. Liu, "Rotor-flying manipulator: Modeling, analysis, and control," *Mathematical Problems in Engineering*, vol. 2014, p. 13, 2014.

[120] E. Cataldi, G. Muscio, M. A. Trujillo, Y. Rodriguez, 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 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2016, pp. 3848–3853.

[121] G. Arleo, F. Caccavale, G. Muscio, and F. Pierri, "Control of quadrotor aerial vehicles equipped with a robotic arm," in *Proceedings of the 21st Mediterranean Conference on Control and Automation*, June 2013, pp. 1174–1180.

[122] A. Y. Mersha, S. Stramigioli, and R. Carloni, "Exploiting the dynamics of a robotic manipulator for control of UAVs," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, May 2014, pp. 1741–1746.

[123] K. Lee, J. Back, and I. Choy, "Nonlinear disturbance observer based robust attitude tracking controller for quadrotor UAVs," *International Journal of Control, Automation and Systems*, vol. 12, no. 6, pp. 1266–1275, 2014.

[124] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005.

[125] J. Thomas, J. Polin, K. Sreenath, and V. Kumar, "Avian-Inspired Grasping for Quadrotor Micro UAVs," in *Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 6A: 37th Mechanisms and Robotics Conference*, 08 2013.

[126] F. Ruggiero, M. A. 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 2015 IEEE International Conference on Robotics and Automation*, May 2015, pp. 4014–4020.

[127] R. Mebarki and V. Lippiello, "Image-based control for aerial manipulation," *Asian Journal of Control*, vol. 16, no. 3, pp. 646–656, 2014.

[128] 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, Aug 2016.

[129] F. Augugliaro and R. D'Andrea, "Admittance control for physical human-quadrocopter interaction," in *Proceedings of the 2013 European Control Conference*, July 2013, pp. 1805–1810.

[130] 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 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2016, pp. 1631–1636.

[131] M. Orsag, C. Korpela, P. Oh, and S. Bogdan, *Sensors and Control*. Cham: Springer International Publishing, 2018, pp. 165–208.

[132] A. Ollero, M. Tognon, A. Suarez, D. Lee, and A. Franchi, "Past, present, and future of aerial robotic manipulators," *IEEE Transactions on Robotics*, pp. 1–20, 2021.

[133] A. Mohiuddin, T. Tarek, Y. Zweiri, and D. Gan, "A survey of single and multi-uav aerial manipulation," *Unmanned Systems*, vol. 08, no. 02, pp. 119–147, 2020.

[134] F. Ruggiero, V. Lippiello, and A. Ollero, "Aerial manipulation: A literature review," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1957–1964, Jul. 2018.

[135] H. B. Khamseh, F. Janabi-Sharifi, and A. Abdessameud, "Aerial manipulation—a literature survey," *Robotics and Autonomous Systems*, vol. 107, pp. 221 – 235, 2018.

[136] X. Meng, Y. He, and J. Han, "Survey on aerial manipulator: System, modeling, and control," *Robotica*, vol. 38, no. 7, p. 1288–1317, 2020.

[137] J. Mendoza-Mendoza, V. J. Gonzalez-Villela, C. Aguilar-Ibañez, M. S. Suarez-Castañon, and L. Fonseca-Ruiz, "Snake aerial manipulators: A review," *IEEE Access*, vol. 8, pp. 28 222–28 241, 2020.

[138] X. Ding, P. Guo, K. Xu, and Y. Yu, "A review of aerial manipulation of small-scale rotorcraft unmanned robotic systems," *Chinese Journal of Aeronautics*, vol. 32, no. 1, pp. 200–214, 2019.

[139] M. Labbadi and M. Cherkaoui, "Robust adaptive backstepping fast terminal sliding mode controller for uncertain quadrotor UAV," *Aerospace Science and Technology*, vol. 93, p. 105306, 2019.

[140] X. Shao, J. Liu, H. Cao, C. Shen, and H. Wang, "Robust dynamic surface trajectory tracking control for a quadrotor uav via extended state observer," *International Journal of Robust and Nonlinear Control*, vol. 28, pp. 2700–2719, 2018.

[141] A. Poutney, C. R. Kennedy, G. M. Clayton, and H. Ashrafiuon, "Robust tracking control of quadrotors based on differential flatness: Simulations and experiments," *IEEE/ASME Transactions on Mechatronics*, p. 1, 2018.

[142] W. Xie, D. Cabecinhas, R. Cunha, and C. Silvestre, "Adaptive backstepping control of a quadcopter with uncertain vehicle mass, moment of inertia, and disturbances," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2021.

[143] Y. Zou, "Nonlinear robust adaptive hierarchical sliding mode control approach for quadrotors," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 6, pp. 925–941, 2017.

[144] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, "Dynamic modelling and configuration stabilization for an X4-Flyer," in *Proceedings of the 2002 IFAC World Congress*, vol. 35, Barcelona, Spain, 2002, pp. 217–222.

[145] M. Kristic, I. Kanellakopoulos, and P. Kokotovic, *Nonlinear and Adaptive Control Design*. New York: John Wiley and Sons, 1995.

[146] A. Dzul, T. Hamel, and R. Lozano, "Helicopter's nonlinear control via backstepping techniques," in *Proceedings of the 2001 European Control Conference*, Porto, Portugal, Sep. 2001, pp. 463–468.

[147] T. Hamel and R. Mahony, "Pure 2D visual servo control for a class of underactuated dynamic systems," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, vol. 3, 2004, pp. 2229–2235.

[148] P. Castillo, R. Lozano, and A. Dzul, *Modelling and control of mini flying machines*. New York City, USA: Springer-Verlag, 2005.

[149] T. J. Koo and S. Sastry, "Output tracking control design of a helicopter model based on approximate linearization," in *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, FL, Dec. 1998, pp. 3635–3640.

[150] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Ph.D. dissertation, Ecole Polytechnique Federale de Luasanne, Lausanne, Switzerland, 2007.

[151] A. Moeini, A. F. Lynch, and Q. Zhao, "A backstepping disturbance observer control for multirotor UAVs: theory and experiment," *International Journal of Control*, vol. 0, no. 0, pp. 1–15, 2021.

[152] ——, "Exponentially stable motion control for multirotor uavs with rotor drag and disturbance compensation," *Journal of Intelligent Robotic Systems*, vol. 103, no. 1, p. 15, 2021.

[153] G. Baraban, M. Sheckells, S. Kim, and M. Kobilarov, "Adaptive parameter estimation for aerial manipulation," in *American Control Conference*, 2020, pp. 614–619.

[154] J. Alvarez-Munoz, N. Marchand, J. F. Guerrero-Castellanos, J. J. Tellez-Guzman, J. Escareno, and M. Rakotondrabe, "Rotorcraft with a 3-DOF rigid manipulator: Quaternion-based modeling and real-time control tolerant to multi-body couplings," *International Journal of Automation and Computing*, vol. 15, no. 5, pp. 547–558, 2018.

[155] J. Alvarez-Munoz, J. Escareno, N. Marchand, J. Guerrero-Castellanos, T. Raharijaona, and M. Rakotondrabe, "Quaternion modeling and observer-based torque compensation of an aerial manipulator," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 543–548, 2018, proceedings of the 2018 2nd IFAC Conference on Modelling, Identification and Control of Nonlinear Systems.

[156] G. Antonelli, E. Cataldi, F. Arrichiello, P. R. Giordano, S. Chiaverini, and A. Franchi, "Adaptive trajectory tracking for quadrotor MAVs in presence of parameter uncertainties and external disturbances," *IEEE Transactions on Control System Technology*, vol. 26, no. 1, pp. 248–254, Jan. 2018.

[157] D. Cabecinhas, R. Cunha, and C. Silvestre, "A globally stabilizing path following controller for rotorcraft with wind disturbance rejection," *IEEE Transactions on Control System Technology*, vol. 23, pp. 708–714, 2015.

[158] F. A. Goodarzi, D. Lee, and T. Lee, "Geometric adaptive tracking control of a quadrotor unmanned aerial vehicle on SE(3) for agile maneuvers," *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, vol. 137, no. 9, pp. 091 007–1–12, Jun. 2015.

[159] A. Moeini, "Disturbance observer-based motion control and Visual-Inertial-Actuator odometry for UAVs," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, 2021.

[160] A. Moeini, M. A. Rafique, Z. Xue, A. F. Lynch, and Q. Zhao, "Disturbance observer-based integral backstepping control for uavs," in *Proceedings of the 2020 International Conference on Unmanned Aircraft Systems*, 2020, pp. 382–388.

[161] M. A. Rafique and A. F. Lynch, "Vision-based motion control of UAVs for GPS-denied environments," in *Proceedings of the 2018 Unmanned Systems Canada Conference*, Vancouver, BC, Oct. 2018, pp. 1–16, student paper award.

[162] ——, "Output-feedback image-based visual servoing for multirotor unmanned aerial vehicle line following," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 4, pp. 3182–3196, 2020.

[163] "Flight contoller hardware," Dronecode Project, Inc., accessed 10 October 2021. [Online]. Available: https://docs.px4.io/v1.12/en/flight_controller/

[164] "Autopilot hardware options," ArduPilot Development Team and Community, accessed 1 Mar 2021. [Online]. Available: https://ardupilot.org/copter/docs/common-autopilots.html

[165] L. Meier, P. Tanskanen, L. Heng, G. Lee, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.

[166] "Pxhawk 1 autopilot," Institute for Visual Computing, Swiss Federal Institute of Technology Zurich, accessed 1 Mar 2019. [Online]. Available: https://docs.px4.io/en/flight_controller/pixhawk.html

[167] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation*, Seattle, WA, May 2015, pp. 6235–6240.

[168] "Ardupilot project," ArduPilot Development Team and Community, accessed 10 Oct 2021. [Online]. Available: https://ardupilot.org/

[169] L. Meier, "PX4 autopilot," http://dev.px4.io/en/ [accessed 01Mar 2019], Institute for Visual Computing, Swiss Federal Institute of Technology Zurich, 2019. [Online]. Available: http://dev.px4.io/en/

[170] ——, "Mavlink: Micro air vehicle communication protocol," 2015, accessed 5 May 2015. [Online]. Available: http://qgroundcontrol.org/mavlink/start

[171] S. Kirby, "Open source firmware for atmega-based brushless escs." [Online]. Available: https://github.com/sim-/tgy

[172] S. Skaug, "BLHeli for brushless ESC firmware." [Online]. Available: https://github.com/bitdump/BLHeli

[173] M. Bangura, "Aerodynamics and control of quadrotors," Ph.D. dissertation, College of Engineering and Computer Science, The Australian National University, 2017.

[174] T. Kane and D. Levinson, *Dynamics, Theory and Applications*, ser. McGraw-Hill series in mechanical engineering. McGraw-Hill, 1985.

[175] C. Roithmayr and D. Hodges, *Dynamics: Theory and Application of Kane's Method*, ser. Titolo collana. Cambridge University Press, 2016. [Online]. Available: https://books.google.ca/books?id=clNyCwAAQBAJ

[176] A. J. Knutson, "Application of Kane's method to incorporate attitude dynamics into the circular restricted three-body problem," Ph.D. dissertation, Purdue University, West Lafayette, Indiana, West Lafayette, Indiana, 2012.

[177] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modelling and Control*. New York, NY: Wiley, 2006.

[178] H. Berghuis and H. Nijmeijer, "Robust control of robots via linear estimated state feedback," *IEEE Transactions on Automatic Control*, vol. 39, no. 10, pp. 2159–2162, Oct 1994.

[179] H. Xie, A. F. Lynch, K. H. Low, and S. Mao, "Adaptive output-feedback image-based visual servoing for quadrotor unmanned aerial vehicles," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 1034–1041, 2020.

[180] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, USA: Prentice Hall, 2001.

[181] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, ser. Spring Tracts in Advanced Robotics. New York City, USA: Springer-Verlag, 2011.

[182] T. He, Y. Zeng, and Z. Hu, "Research of multi-rotor UAVs detailed autonomous inspection technology of transmission lines based on route planning," *IEEE Access*, vol. 7, pp. 114 955–114 965, 2019.

[183] DJI matrice 100 quadcopter for developers. Shenzhen DJI Sciences and Technologies Ltd. Accessed 10 Oct 2021. [Online]. Available: https://www.dji.com/ca/matrice100

[184] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[185] S. S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence, and Robustness*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[186] K. S. Narendra and A. M. Annaswamy, "Persistent excitation in adaptive systems," *International Journal of Control*, vol. 45, no. 1, pp. 127–160, Jan. 1987.