Dynamic Visual Servoing of Rotary Wing Unmanned Aerial Vehicles

by

Hui Xie

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

Visual servoing uses camera feedback to control a robot's motion. In particular, Image-based Visual Servoing (IBVS) laws minimize an error signal defined in the image space. This thesis considers IBVS design for rotary-wing Unmanned Aerial Vehicles (UAVs). This is a challenging problem due to the nonlinear perspective projection of image formation and the underactuated nonlinear dynamics of the UAV. Traditional IBVS approaches calculate the six Degrees of Freedom (DoF) velocity reference and feed it into an inner velocity tracking loop. This approach cannot be directly applied to IBVS of an underactuated UAV. To address this issue, all IBVS laws developed in this thesis consider the vehicle's dynamics, and this approach is termed as Dynamic Image-based Visual Servoing (DIBVS). As compared to traditional position regulation or tracking of UAVs, the nonlinear image kinematics leads to systems structure which makes the control problem challenging. In this thesis a state transformation is proposed to eliminate the time derivative of attitude in the image kinematics. This leads to a simpler system structure for control design. The state transformation is obtained by solving a system of first-order homogeneous Partial Differential Equations (PDEs). The existence of solutions is proven and the general solution provided. Using the transformed state, image moment features for a planar target with multiple points or parallel lines are proposed. The state transformation approach requires an attitude measurement which is commonly available for UAVs. However, this measurement contains small bias which results in state-transformation error and introduces a disturbance into the translational velocity dynamics. The state-transformation error is proven to be negligible. In this thesis, a quadrotor is adopted as the UAV platform to validate the proposed DIBVS

schemes. The external force or torque is assumed to be the product of an aerodynamic gain and the square of the Pulse Width Modulation (PWM) signal to the Electronic Speed Controller (ESC). This gain gradually decreases as battery voltage drops, and the resulting reduction in thrust has a noticeable effect on UAV motion in practise. The attitude bias, aerodynamic gain, and mass of the UAV are treated as unknown constants. Both Proportional-Integral-Derivative (PID) and adaptive DIBVS schemes are proposed to stabilize image feature error and are robust to this system uncertainty. Although in the transformed image state the convergence appears to remain in the camera Field of View (FoV), this may not be case for the actual camera image. To address this issue, a visual servoing scheme based on a nested saturation law is proposed to constrain the attitude of the quadrotor.

Preface

Section 3.2 and Chapter 4 was published online as H. Xie and A. Lynch, "State transformation based dynamic visual servoing for an unmanned aerial vehicle", *International Journal of Control*, 2015. I was responsible for the scientific content in this paper and for writing the manuscript. Professors Lynch acted as supervisor for this work.

Section 3.3 was published as G. Fink, H. Xie, A.F. Lynch, and M. Jagersand, "Nonlinear dynamic image-based visual servoing of a quadrotor," *Journal of Unmanned Vehicle Systems*, vol. 3, no. 1, pp. 1-21, 2015. I was responsible for the initial concept, mathematical analysis, and part of the implementation work. G. Fink wrote the article with my assistance. Professors Lynch and Jagersand supervised the work.

Section 5.1 was accepted for publication as H. Xie, G. Fink, A.F. Lynch, and M. Jagersand, "Adaptive dynamic visual servoing of a UAV," *IEEE Transactions on Aerospace and Electronic Systems*, 2015. I developed and implemented the design. G. Fink assisted me with the implementation work. Professors Lynch and Jagersand acted as supervisors.

Section 5.2 was published in two articles: H. Xie, A.F. Lynch, and M. Jagersand, "Dynamic IBVS of a rotary wing UAV using line features," *Robotica*, 2014, 18 pages, available on FirstView. and H. Xie, A.F. Lynch, and M. Jagersand, "IBVS of a rotary wing UAV using line features," in *Proceedings of the 27th IEEE Canadian Conference on Electrical and Computer Engineering*, 2014, Toronto, ON, May 2014, 6 pages. In this work I provided the content with G. Fink providing assistance with the experimental work. Professors Lynch and Jagersand provided supervision.

The analysis in Chapter 6, the concluding remarks in Chapter 7, and the literature survey in Chapter 1 are my original work.

Table of Contents

List of Tables

List of Figures

List of Acronyms

AHRS Attitude and Heading Reference System.

ANCL Applied Nonlinear Control Lab.

CAD Computer-Aided Design.

DIBVS Dynamic Image-based Visual Servoing.

DoF Degrees of Freedom.

EKF Extended Kalman Filter.

ESC Electronic Speed Controller.

FoV Field of View.

GAS Globally Asymptotically Stable.

GES Globally Exponentially Stable.

GPS Global Positioning Systems.

IBVS Image-based Visual Servoing.

IMU Inertial Measurement Units.

LADAR Laser Detection and Ranging.

LAS Locally Asymptotically Stable.

LES Local Exponentially Stable.

LIDAR Light Detection and Ranging.

PBVS Position-based Visual Servoing.

PDE Partial Differential Equation.

PID Proportional-Integral-Derivative.

PPM Pulse Position Modulation.

PTAM Parallel Tracking and Mapping.

PVTOL Planar Vertical Take-off and Landing.

PWM Pulse Width Modulation.

RTOS Real-Time Operating System.

SLAM Simultaneous Localization and Mapping.

UART Universal Asynchronous Receiver/Transmitter.

UAV Unmanned Aerial Vehicle.

VTOL Vertical Take-off and Landing.

Chapter 1

Introduction

1.1 Background

Rotary-wing Unmanned Aerial Vehicles (UAVs) such as ducted fans, quadrotors, and traditional helicopters have unique flight characteristics including Vertical Take-off and Landing (VTOL), hover, and low-speed flight while retaining cruising abilities. This maneuverability makes them suitable for applications such as search and rescue, infrastructure inspection, and surveillance. A recent survey of rotary wing UAVs is in [1]. UAVs provide a number of benefits for inspection applications such as improved safety, reduced cost, and improved inspection data quality. The motivation of the work in this thesis is to automate small UAVs to inspect the infrastructure used for the transmission of electricity, e.g. [2]. Electrical utilities are required to frequently inspect transmission lines which can span thousands of kilometres of adverse terrain. Inspection is required by law and is important to ensure the safety and reliability of the power system. A traditional solution to inspection involves costly, inefficient, and potentially dangerous manned flights. Given the magnitude and importance of the inspection problem, even small improvements in efficiency using UAVs can lead to significant benefits.

UAVs have a complicated, nonlinear, and underactuated dynamic model which makes their motion control an interesting challenge. Motion control depends on accurate pose estimation. Traditionally, UAV pose is estimated with an aided navigation system which fuses inertial measurements with various sensors such as ultrasonic rangefinders, barometers, magnetometers, and Global Positioning Systems (GPS) [3]. For outdoor applications most commercially available autopilots rely on GPS for maintaining a stable position estimate which is used to autonomously control the vehicle's position trajectory. However, GPS signals are often unavailable due to obstructions or intentional jamming, e.g. urban canyons [4]. Furthermore, GPS cannot provide the relative position to a target of interest unless the target's GPS location is accurately known. For example, the GPS coordinates of a power

transmission line are generally not known exactly. Computer vision provides rich information about the environment and is potentially a good alternative to GPS as an onboard sensor for measuring the UAV's relative pose to its environment. As compared to other sensors, such as Light Detection and Ranging (LIDAR), Laser Detection and Ranging (LADAR), laser radar, and laser range finders, cameras are inexpensive, passive, relatively lightweight and can be readily mounted on board a UAV [5]. Furthermore, cameras can be alone or combined with Inertial Measurement Unitss (IMUs) to form low cost vision-based navigation systems [6].

Based on how the image is processed, integrated with other sensors, and used on an aircraft, a classification of computer vision applications for UAVs is proposed in [5]. At the most basic level computer vision uses image processing techniques to process the raw image captured onboard. This includes techniques such as object identification and tracking, camera motion estimation, e.g. [7, 8]. At the next level the information collected from image processing techniques is fused with other onboard sensors to estimate the pose of the UAV or partially aid with motion control, e.g. [9, 10]. Finally, at the highest level visual information from an onboard camera can be integrated into the closed-loop motion control of the vehicle. This is referred to as *visual servoing* [11] and is the focus of this thesis.

Visual servoing is divided into two main approaches: Position-based Visual Servoing (PBVS) and Image-based Visual Servoing (IBVS) [11]. PBVS methods reconstruct the vehicle pose from image measurements and control the motion directly in $SE(3)$. Examples of PBVS methods applied to UAVs include [12–19]. In general PBVS requires more a priori knowledge of the target scene or object. For example, to track relative pose between the UAV and target some researchers assume that a Computer-Aided Design (CAD) model of the target is known in advance. The object corresponding to this CAD model is identified in the environment and tracked. Because the CAD model Euclidean dimensions are known, UAV-to-target relative Euclidean pose can be calculated. Often a specially designed target marker (e.g. landing pad pattern) is placed in the environment to facilitate tracking. However, this can limit the application when the environment is unstructured. In addition, intrinsic parameters of the camera are required for vehicle pose estimation and the accuracy of the reconstruction depends on the accuracy of the camera calibration procedure. Other examples of PBVS rely on Simultaneous Localization and Mapping (SLAM) methods (e.g. monoSLAM, visual SLAM, Parallel Tracking and Mapping (PTAM)) that construct a map of scene points and track the UAV pose relative to it. Examples of visual SLAM include [20, 21]. A challenge of SLAM-based motion control for flying UAVs in unknown environments is that the coordinate frame recovered by SLAM is arbitrary and needs to be aligned with the world frame. The precision of the executed trajectory will depend both on this external alignment and

the internal accuracy of the SLAM model. A benefit of PBVS is that non-vision based motion controllers can be used directly. IBVS methods control the motion of the vehicle indirectly by minimizing the image feature errors in the 2D image space [11]. This thesis focuses on IBVS for a number of reasons. Since in IBVS the task function is computed directly from image plane measurements, the need for 3D position reconstruction used in PBVS methods is avoided. Hence, in general, IBVS is relatively insensitive to camera calibration even though the control law can depend on intrinsic camera parameters [22]. Furthermore, for a given camera sensor resolution IBVS has the potential to provide more accurate positioning [23, 24]. Another benefit of IBVS is the target object tends not to leave the field of view since the design minimizes image feature error which directly depends on image plane measurements. Although IBVS has many merits, the nonlinear perspective projection image kinematics complicates the control law design and it is difficult to design a globally asymptotically stable controller.

The control structure of a visual servoing law is often divided into two loops [11]. An inner loop accepts a reference velocity and determines force and torque inputs to control the robot's velocity at a relatively high bandwidth (e.g., 200 Hz). An outer loop uses video camera feedback to generate reference velocities for the inner loop. It operates at a relatively low frequency because of the low rate at which images are acquired and processed (e.g., 30 Hz). A conventional visual servoing design usually assumes that the inner loop tracks its reference velocity perfectly. Analysis and design are based on a purely kinematic model. However, it is pointed out in [25, 26] that the dynamics of a robot should be considered for high speed tasks or when the system is underactuated. An IBVS law that directly accounts for vehicle dynamics is referred as DIBVS. The UAV considered in this thesis has six Degrees of Freedom (DoF) and four actuator inputs, which make the system underactuated. The importance of treating the underactuated dynamics of a UAV is underlined in [27]. Although dynamic IBVS has clear practical significance, there is relatively sparse literature on the topic for any type of robotic application. This is likely due in part to the difficulty in rigorously accounting for the nonlinearity of the camera's perspective projection and the robot's dynamics.

The objective of this work is to explore dynamic IBVS techniques which can be applied to automate power line inspections using rotary wing UAVs. This work also aims to develop an indoor UAV experimental platform to simulate the basic aspects of motion control in inspection outdoors. Working on an outdoor platform introduces many practical challenges such as wind disturbances and difficult weather conditions. Hence, the approach adopted in this thesis is to first validate the proposed DIBVS control laws in a relatively controlled indoor environment. This decision was made after significant outdoor field work was performed by our

research team on a gas-powered helicopter UAV [28]. Our experience has been that flying small quadrotor vehicles indoors can lead to large increases in work efficiency including reduced operating costs and increased researcher safety. Working indoors with computer vision has the advantage of better image processing performance. For example, feature tracking can be achieved with higher accuracy on simpler visual targets.

1.2 Literature Survey

A number of DIBVS approaches are identified for UAVs that have appeared in the literature and classify them into four categories: the spherical image moment-based design, the homography-based method, the virtual spring approach, and the virtual camera approach.

The seminal work in [29] shows that a perspective projection camera model destroys the triangular structure of the system dynamics. Hence, a spherical projection camera model is introduced to give the image feature kinematics a passivity property, i.e., the time derivative of the norm of the image feature error function is independent of angular velocity. By introducing first-order moment point features a backstepping controller is proposed. The origin of the closed-loop system is proven to be Globally Exponentially Stable (GES). This approach requires a UAV attitude measurement to define a desired image feature. This attitude information is commonly estimated using an Attitude and Heading Reference System (AHRS) which is integrated into most UAV autopilots. Further work in [30] removes the requirement of an orientation sensor in [29] which makes the stabilization problem more challenging. A Local Exponentially Stable (LES) result is proven and a region of attraction is provided. A similar approach to the work in [29] is in [31] where first-order line image moment features with bi-normalized Plücker coordinates are proposed for visual servoing of targets with linear structure, e.g. electrical power lines. A GES result is achieved using backstepping. However, the proposed interaction matrices in [29, 30] are ill-conditioned as the image feature is insensitive to change in altitude. Hence, the performance suffers from a low rate of convergence in altitude. Various modified spherical image moment features are proposed to address this drawback. In [32] a rescaled spherical image moment is proposed. Although the interaction matrix is no longer ill-conditioned, a global asymptotic stability results can not be achieved. In [33] a partitioned image moment feature is introduced and the resulting controller is implemented experimentally. Although this approach achieves a GES result as in [29], the interaction matrix is still ill-conditioned when the camera is not close to its desired pose. The work in [34] gives a survey of the spherical projection approach using rescaled and partitioned image features. It is remarked that it is

difficult to achieve yaw control of the vehicle due to the complicated kinematics of the feature for yaw motion proposed in [29].

Another useful tool for IBVS is the homography matrix which embeds information about the relative pose of a camera given two images of the same planar object [35]. In [36] a homography matrix is used to extract the normal vector of the target plane and a ratio of distances from the origins of the camera image plane to target plane. Based on this extracted information, a scaled distance image feature is defined and its kinematics has a passivity property. An adaptive law to the unknown depth which appears in the feature kinematics, and a tracking law for the pre-recorded images are developed. A Globally Asymptotically Stable (GAS) result is achieved. In this approach only the position of the UAV is controlled, and the control of yaw motion is not addressed. Motivated by [37, 38], work in [39] proposes a six dimensional feature vector that is diffeomorphically related to camera pose. A local stabilizing control law is developed that only needs angular velocity and visual measurement of a planar target. Unfortunately, this method only works for a small range of image error and this limits its application to small variation in yaw. Compared to approaches using spherical moment features, this approach only works for planar targets.

Work in [40, 41] introduce a virtual spring approach where image feature moments lead to an interaction matrix with an identity matrix in its translational component. The work assumes knowledge of the desired height, a planar target of point features which remains parallel to the image plane. This approach is motivated by the UAV's underactuation where increasing image feature error reduces lateral position error. The control law is independent of linear velocity and a Locally Asymptotically Stable (LAS) result is proven. The virtual spring method is limited by the assumption that the image plane is parallel to the target plane. Lateral motion of a quadrotor UAV requires roll and pitch motion which means the image and target planes are no longer parallel. Even a small change in roll or pitch may cause a large change in the interaction matrix.

A different approach is based on a so-called virtual image plane which has zero roll and pitch angles and has the same position and yaw angle of the real camera's image plane. This work is first introduced in [42, 43]. This virtual plane facilitates the estimation of depth of image points. A classical IBVS method is used to generate a reference velocity screw, and an adaptive sliding mode control is used in the inner loop. This work gives a GAS result. In [27], image moments in the same virtual plane are used and the feature kinematics has a passivity property. An adaptive backstepping controller is developed to stabilize the feature error. The image kinematics with the virtual camera approach is simpler since it is independent of roll and pitch rates. A GAS result is also rigorously proven. However, similar to

spherical image moment-based approaches in [29, 31, 34], the virtual camera requires the UAV attitude measurement.

Other than the virtual spring approach which is based on an output feedback design, the work mentioned above assumes measurement of the linear velocity. To avoid the requirement of the velocity measurement, some IBVS designs use optical flow instead. For example, in [44] the same image moment features proposed in [29] are used and the optical flow is used to estimate a scaled linear velocity. A fully nonlinear adaptive control design is proposed to ensure GAS of the closed-loop. Similar work is also shown in [45]. In [46], the spherical optical flow and attitude measurement from IMU are combined to estimate a scaled translational velocity of the vehicle in the navigation frame. An adaptive output tracking controller is applied to landing a vehicle on a ship deck. A GAS result is also achieved. Instead of using the optical flow feedback, some approaches adopt an output feedback method. Work in [47, 48] uses spherical moment features and an observer to estimate the translational velocity of the vehicle. The desired value of depth is used as a nominal value for actual depth which appears in the observer. Both simulation and experimental work show the observer's robustness to unknown depth information. However, this work lacks an analysis to prove this robustness. In [49] spherical image moment features are also used for IBVS. An adaptive output feedback backstepping approach from [50] is used. Although a GAS result is provided, this work is not experimentally validated. This is possibly due to the complex controller structure which results. Output feedback methods with moment features in a virtual camera is receiving recent attention. Work in [51] uses a first-order image moment of points in a virtual camera with zero roll and pitch. An output feedback control based on the image moment feature in the virtual camera is proposed. In [52] a virtual camera with no rotational motion is proposed, and an adaptive output feedback controller from [53] is modified to track a moving target. The above two approaches were validated only in simulation.

Due to the limited Field of View (FoV) of a camera, in IBVS the output of the image should be kept in a compact set. Hence, in [54] an output constraint method from [55] is used for landing a UAV with parallel line image features. Although the work in [54] provides a new perspective in the field of visual servoing, a number of unsolved issues remain. For example, the effect of inner loop tracking should be analysed. An alternative and generally simpler approach is to constrain the input. For example, in [56, 57] input saturation controls are adopted. However, these approaches have no rigorous proof to show that the visual targets can be kept in the FoV of the camera.

1.3 Thesis Outline

Chapter 2 describes an indoor UAV platform, and the modelling of the UAV and camera. To perform practical research on dynamic visual servoing of UAVs, the indoor UAV platform has to be built. The developed platform is termed as Applied Nonlinear Control Lab (ANCL) quadrotor platform. In Chapter 2, the details of the hardware components of the UAV: a quadrotor frame, an onboard autopilot and its peripherals, an onboard computer vision system, are covered. Next, the software structure and data flow of the quadrotor autopilot firmware are briefly presented. To design dynamic visual servoing law for the ANCL platform, a tractable model is required. In Chapter 2, a pinhole camera model is considered. The kinematics of image point coordinates of a single 3D point and parameters parametrizing a single 3D line are also given based on the chosen camera model. Lastly, the UAV is modelled as a rigid body. The thrust created by a propeller is simplified as the product of a thrust gain and the square of Pulse Width Modulation (PWM) signal which is fed into the Electronic Speed Controller (ESC) that drives the motor. The countertorque created by propeller is also approximated by the product of an aerodynamic constant and the square of PWM signal. Based on the simplification of the aerodynamics above and the single rigid body dynamics, a tractable dynamic model for the quadrotor is built. The combination of the image feature kinematics and quadrotor dynamics forms the dynamic visual servoing model, which provides a basis for developing dynamic visual servoing laws through the rest of the thesis.

In Chapter 3 a novel state transformation-based dynamic visual servoing approach for underacutated UAVs is presented. In the dynamic visual servoing model built in Chapter 2, the image feature kinematics using perspective projection destroy the passivity property possessed by the rigid body dynamics [29], which complicates the dynamic visual servoing law design. Chapter 3 first chooses a cascade controller structure: an outer loop regulates the image feature error with thrust and attitude as the input, an inner loop tracks the reference attitude from the outer loop. Next, a nonlinear control technique, which is based on state transformation, is used to remove the time derivative of attitude in the outer loop. This state transformation recovers the passivity property in the dynamics with the new state. A constructive approach is given to define the state transformation by solving a system of first-order linear homogeneous Partial Differential Equations (PDEs). To illustrate the philosophy, this approach is applied to a 2D Planar Vertical Take-off and Landing (PVTOL) aircraft. Using the same philosophy and under constant height assumption, a dynamic visual servoing law with image coordinates of a single 3D point is presented to regulate the lateral motion of the quadrotor. Both simulation and experimental work are conducted to validate the performance of the proposed approach.

Chapter 4 continues to explore the cascade controller structure and the state transformation-based approach proposed in Chapter 3 to regulate the 3D translational motion and yaw motion of the quadrotor. To control this four DoF motion of the quadrotor, the feature vector should consist of at least four independent variables. In Chapter 4 a target that consists of multiple points or multiple parallel lines is considered. The image moment features in the transformed state are presented. Next, under the assumption that all points or lines are located in a horizontal plane, the image moment feature kinematics is derived. From the experimental results in Chapter 3, it can be observed that there is non-zero steady image error. It is concluded that the attitude estimate bias leads to this non-zero steady error. Lastly, by treating the attitude estimate bias as constant and assuming unknown thrust gain, a dynamic Proportional-Integral-Derivative (PID) visual servoing law is developed. Experimental validation of the proposed visual servoing law for regulating four DoF motion of the quadrotor is presented. Results for both point and line targets are given.

Chapter 5 covers adaptive dynamic visual servoing approaches with the image moment features proposed in Chapter 4. Although the controller proposed in Chapter 4 is proven to be robust to the system's unknown parameters, it requires extensive controller tuning. In Chapter 5 adaptive control design techniques are applied to dynamic visual servoing. These adaptive laws generally reduce the need for controller tuning. An adaptive backstepping design technique is first applied to develop an visual servoing law with point moment features. The same cascade structure as in Chapter 3 and Chapter 4 is adopted. The performance of this adaptive visual servoing law is validated both numerically and experimentally. Next, another adaptive DIBVS law based on backstepping for line moment features is presented. This adaptive law is based on a different cascade structure: the outer loop regulates the image feature error with the time derivative of thrust and angular velocity as the input variables, an inner loop tracks the reference angular velocity from the outer loop. Numerical simulation work is given to validate the performance of this adaptive scheme.

Chapter 6 presents a dynamic visual servoing scheme with input saturation. To keep the visual target in the camera's FoV, the roll and pitch references are constrained to small compact sets containing the origin. During typical motion of the quadrotor in visual servoing applications, the thrust on the vehicle should be kept positive. Using the same cascade control structure proposed in Chapter 3, the attitude and thrust constraint leads to an input saturation control design problem. Chapter 6 first introduces the background for saturation controller design. Next, a less conservative nested input saturation controller is proposed for the height subsystem, which ensures thrust remains positive. Then, another nested saturation

control for the lateral subsystem is developed which limits the reference roll and pitch angles to the inner attitude loop. Finally, both simulation and experimental results are used to demonstrate the effectiveness of the proposed approach to keep the visual target in the camera's FoV.

Chapter 7 summarizes the thesis and discusses possible future research.

1.4 Contributions

The contributions of this thesis are summarized as follows:

- State transformation-based dynamic image-based visual servoing approach [58, 59]. In DIBVS with a perspective projection model, the image feature kinematics replaces the UAV's 3D translational motion kinematics. This destroys the passivity property of rigid body dynamics. A state transformation-based approach is proposed to recover the passivity property. In the new state, traditional motion control design techniques can be applied to obtain a DIBVS. This approach generalizes the virtual camera approach proposed in [42, 43].

- New image moment features for visual servoing of linear structure [60] [61]. Inspired by the moment features for targets of multiple coplanar points in [62, 63], a set of image moment features for parallel lines is proposed and their kinematics derived. The kinematics of the moment feature has a simple decoupled structure which facilitates controller design.

- Dynamic PID IBVS law [59, 64]. The attitude measurement bias is unavoidable with a strap-down IMU. In addition, from experiments it is observed that the thrust gain for the propellers decreases as the voltage battery drops. The above two factors lead to non-zero steady image feature error. By treating the attitude bias and thrust gain as unknown constants, a PID controller is proposed and shown to be robust to uncertainties in gain and attitude bias.

- Adaptive DIBVS [61, 65]. Using adaptive backstepping techniques, adaptive DIBVS controllers are proposed which account for parametric system uncertainty including attitude bias, thrust gain, mass, and image feature depth.

- DIBVS with input saturation [66]. To keep the visual target in the camera's FoV and the thrust positive, nested saturation controllers are proposed for height and lateral subsystems.

- Experimental validation of controllers [58, 59, 64–66]. It is important that the controllers proposed in [58, 59, 64–66] are experimentally validated given the general lack of experimental results for DIBVS of UAVs.

Chapter 2

Experimental Platform and Modelling

Ultimately the goal is to develop a UAV for outdoor inspection of power lines. In order to achieve this goal efficiently initial indoor testing is performed. An indoor quadrotor-based platform with onboard vision system has been developed at ANCL for experimental research on visual servoing. A standard-sized quadrotor UAV was chosen for the vehicle which falls into the "mini rotary-wing" category given in [67]. Quadrotors have well-known benefits over other rotary wing UAVs such as traditional helicopters, tandem rotors, and tricopters. They are relatively inexpensive, mechanically simple, have high payload capacity, and reduced gyroscopic effects [68, Chp. 3]. The simplicity of a quadrotor's design make them reliable and easy to maintain. Their numerous positive attributes have led to their popularity with hobbyists and academic researchers in particular. Quadrotors share two main features of rotary-wing UAVs: they are underactuated and dynamically unstable. This makes their motion control a theoretical and practical challenge.

The details of the ANCL quadrotor platform are described in Section 2.1. In Section 2.2 the necessary modelling for dynamic visual servoing is presented. These models allow us to design visual servoing controls which are model-based. This allows for a rigorous analysis of performance and robustness. Camera modelling is a mature topic in the computer vision community. Normally a camera is modelled as a pinhole camera [35, 69], which is provided in Section 2.2.1. Section 2.2.1 also gives the image kinematics of some basic point and line features using the pinhole camera model. In Section 2.2.2 the quadrotor UAV dynamics is presented. The presented quadrotor dynamics is simple enough to be used for visual servoing control design, but still captures the main aspects of the vehicle's behaviour.

2.1 Experimental Platform

2.1.1 Hardware

The ANCL quadrotor vehicle is shown in Figure 2.1. It consists of a 3D Robotics (3DR) vehicle frame, four Turnigy 1100 KV brushless outrunner motors with four APC 12" multi-rotor propellers, two 3 cell 2600 mAh LiPo batteries, a Pixhawk PX4 autopilot, and an onboard CMUcam5 Pixy (Pixy) computer vision system [70]. The

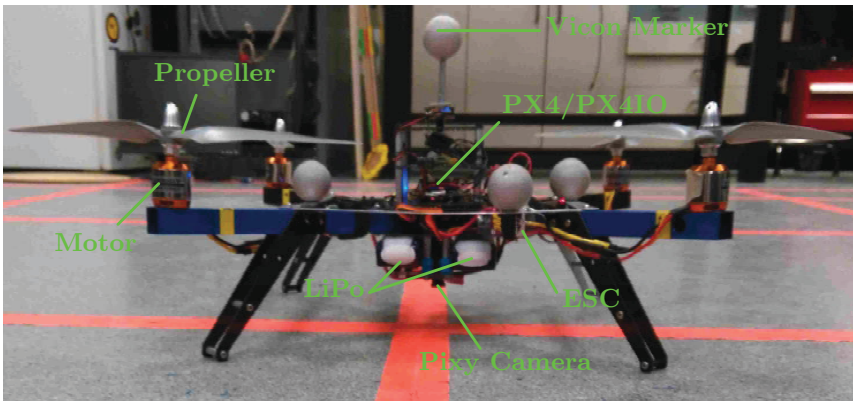


Figure 2.1: The ANCL quadrotor UAV

quadrotor frame dimensions are about 0.4 m × 0.4 m × 0.3 m, and flies in an indoor volume of 4 m × 5 m × 2 m. The autopilot is based on the PX4 hardware which is an open-hardware and open-source project [71]. Having source code publically available reduces development time and makes it possible to customize the code according to project requirements. Presently the PX4 Flight Management Unit (PX4FMU) and PX4 Input Output (PX4IO) interface board as the core autopilot hardware are used. The PX4FMU contains a 168 MHz ARM proccessor, a 3D accelerometer, a 3D gyroscope, a 3D magnetometer, and a pressure sensor. The PX4IO attaches directly to the PX4FMU and expands its interfacing capability. The PX4IO provides extra input and output interfaces, such as Universal Asynchronous Receiver/Transmitter (UART), I2C, and PWM outputs.

The peripherals connected to PX4FMU/PX4IO include four PWM outputs to four Afro 30 A ESCs, three UARTs which communicate with a 3DR 915 MHz radio, a Laird Technology (LairdTech) 2.4 Ghz radio, a Pixy vision sensor, and a Pulse Position Modulation (PPM) input connected to Specktrum satellite receiver. The pulse width of the four PWM outputs from the PX4FMU/PX4IO are the desired velocities of the four motors. Two 3 cell 2600 mAh Libo batteries power the system. The flight endurance is a maximum of about 10 minutes. The 3DR radio communicates with a ground station PC running QGroundControl (QGC) [72]. This allows for real-time monitoring and visualization of the vehicle on the ground.

QgroundControl can also send commands to the PX4FMU. For example, system parameters can be changed and sensor calibration performed. The LairdTech 2.4 Ghz radio communicates with a PC dedicated to running the Vicon Tracker software which calculates the vehicle's pose based on the visual information from an eight camera Bonita Vicon motion capture system (Vicon) [73]. The Vicon system can provide the vehicle's position with millimetre accuracy at up to 200 Hz. The pose information from Vicon can either provide a GPS-like feedback for the autopilot or ground truth for validating the controller's performance. It is possible to configure the LairdTech to send the vehicle's pose information to the PX4FMU if required. The Pixy is a low cost onboard computer vision system which can detect hundreds of point objects or blobs in each frame at up to 50 frames per seconds with a resolution of 640×480 pixels. The Pixy transmits the centre, height and width of each blob to PX4FMU. Based on this basic blob information, various image features can be defined for visual servoing. These features will be described in the following chapters. The camera used in Pixy has a focal length of 2.8 mm, an image sensor size of 1/4", a maximum resolution of 1280×800 pixels, and an M12 lens with a 75° horizontal FoV and 47° vertical FoV. It is noted that the Pixy can only detect salient feature points due to its limited computation power. The Pixy is a temporary solution and a more powerful onboard computer vision system is under development. Since the focus of this thesis is to validate visual servoing schemes instead of developing new computer vision algorithms, this temporary solution is sufficient. For safety and for manual takeoff/landing, the PX4FMU/PX4IO is connected with a 8 channel Spektrum satellite receiver paired with a Spektrum DX8 radio for manual control. Figure 2.2 gives the overview of interconnection between various components.

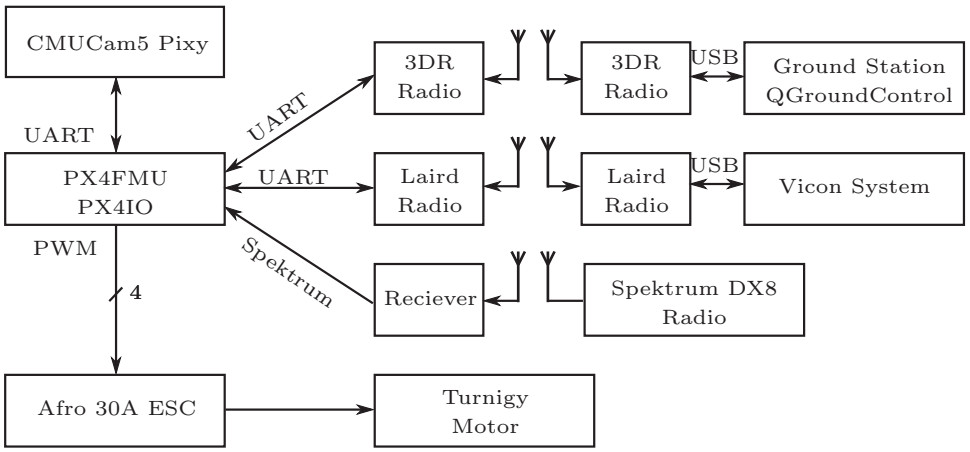


Figure 2.2: Overview of hardware interconnection for the ANCL quadrotor platform.

2.1.2 Software

The software is composed of two main parts: the PX4 autopilot software which runs on the PX4FMU and the computer vision software which runs on the Pixy. Both of them are open-source. The PX4 autopilot software collects information from the gyroscope, accelerometer, magnetometer, pressure sensor, Vicon, and Pixy. From these measurements it outputs the PWM control signals to the ESCs. The software in Pixy mainly focuses on capturing and processing images. As stated in Section 2.1.1, the focus of this thesis is to develop a visual servoing law instead of the computer vision algorithms. Hence, the Pixy is effectively used as a black box which outputs information about the salient blobs in the image. The PX4 autopilot runs on the Real-Time Operating System (RTOS) NuttX [74], which complies with both POSIX and ANSI standards. This compliance provides a familiar Unix environment which reduces development time. The autopilot performs a range of computations such as attitude estimation, control, communication, and data logging. To organize these tasks, a modular system is a practical choice [75]. Each module is implemented with one thread or process and middleware is used to support interprocess communication. A publish-subscribe micro Object Request Broker (uORB) interface design is developed to achieve interprocess communication [75].

On top of NuttX, the PX4 open-source software provides libraries and driver modules for hardware components, i.e., IMU, GPS, and PWM output. In addition, the PX4 autopilot software provides various useful modules including an *Inner* PID attitude controller, an *Outer* PID position controller, *Data logging*, an Extended Kalman Filter (EKF) for attitude estimation, a flight safety *state machine*, control signal *mixer* module which maps from desired body torques to the physical inputs of the system (which are the PWMs signal for the ESCs in the ANCL platform), a *PX4IO* module which provides drivers for various input and output interfaces, and various *sensors* modules which collect sensor information. To set up the PX4 autopilot for the project, the flight state-machine module was heavily modified to allow for switching between various controllers via a physical switch on the DX8 radio. The outer position loop module was modified to use Vicon signals as global position and velocity feedback instead of a GPS signal which is normally used for outdoor flight. This module also implements a switch for sending reference attitudes from the Vicon position module or the visual servoing module, which is introduced later, to the inner loop. The introduction of this switch provides a safety feature when the visual target leaves the FoV of the camera or the quadrotor is close to the boundary of the flight volume. In addition, following the same modular design pattern several modules are added to meet the project requirements. A *vicon* module is created to receive global position and velocity from the Vicon system through the LairdTech radio. The communication with QGroundControl is handled by a *Mavlink*

module which uses the Micro Air Vehicle Communication Protocol (MAVLink) [76]. This protocol provides all of the packet structure, sequencing, identification, and a checksum. MAVLink has proven its reliability in the field. A *Pixy* module is built for receiving the salient image point coordinates via a UART interface, and publishes a topic with this information. A *Pixy FE* module subscribes to this topic and extracts advanced image features such as image moments, line features and then publishes topics containing those features. This topic is subscribed to by various visual servoing laws modules (*VSs*) and those modules calculate the reference attitude and publish it to *Inner* module mentioned above. The abovementioned data flow is summarized Figure 2.3. In this figure rectangular boxes denote data sources such as sensors or devices. Ovals denote software modules. A blue color represents modules that were added or modified. Items shown in black are original modules from the open-source PX4 autopilot software.

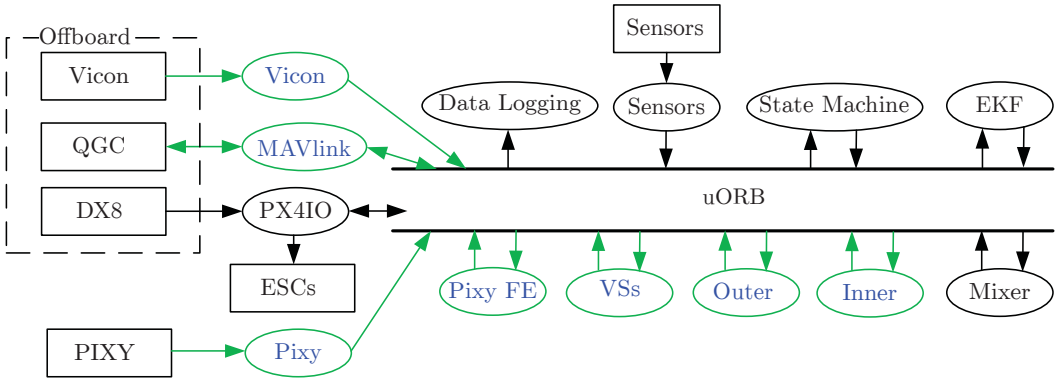


Figure 2.3: Data flow in the PX4 autopilot software. The rectangular boxes are data sources such as sensors or devices. The ovals are software modules. uORB is the object broker for the interprocess communication based on a publish-subscribe design pattern. The blue color represents modules that were added or modified. Items shown in black are the original modules from the open-source PX4 autopilot.

2.2 Modelling

2.2.1 Image Kinematics for Basic Features

Before presenting the image feature kinematics, a pinhole camera model is introduced as shown in Figure 2.4. The pinhole camera model is commonly used in computer vision community [35, 69]. To describe this model, a camera frame $\mathcal{C}$ with basis $\{c_1, c_2, c_3\}$ is introduced. The origin of $\mathcal{C}$ is fixed to the aperture of the camera. The c_3 axis is pointing in the viewing direction of the camera and is referred as the *optical axis*. The image plane is perpendicular to c_3 and intersects c_3 at coordinate λ, which is referred as the focal length of the camera. (It is noted that in reality

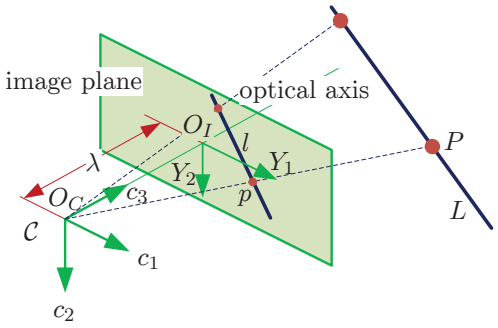


Figure 2.4: Reference frame definitions and the pinhole camera model.

the coordinates is $-\lambda$.) The intersecting point O_I is the *image centre*. In the image plane, a 2D image frame $\mathcal{Y}$ is introduced with basis $\{Y_1, Y_2\}$ parallel to c_1 and c_2, respectively.

Visual features in an image can be decomposed into points and lines. Hence, here two basic image feature kinematics, i.e., point feature kinematics and line feature kinematics, are introduce. A 3D point with coordinates $P = [X_1, X_2, X_3]^T$ in frame $\mathcal{C}$ is projected into the image plane with image coordinates $p = [y_1, y_2]^T$ in $\mathcal{Y}$. The expression of p is

$$p = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \lambda \begin{bmatrix} \frac{X_1}{X_3} \\ \frac{X_2}{X_3} \end{bmatrix} \tag{2.1}$$

As in [11] when the 3D point is static, the kinematics of p is

$$\dot{p} = L_{vp} v^c + L_{\omega p} \omega^c \tag{2.2}$$

where v^c, ω^c are the linear and angular velocity of the camera expressed in $\mathcal{C}$, respectively, and

$$L_{vp} = \begin{bmatrix} -\frac{\lambda}{X_3} & 0 & \frac{y_1}{X_3} \\ 0 & -\frac{\lambda}{X_3} & \frac{y_2}{X_3} \end{bmatrix}, \; L_{\omega p} = \begin{bmatrix} \frac{y_1 y_2}{\lambda} & -\frac{(y_1^2+\lambda^2)}{\lambda} & y_2 \\ \frac{(y_2^2+\lambda^2)}{\lambda} & -\frac{y_1 y_2}{\lambda} & -y_1 \end{bmatrix}$$

A 3D straight line L can be expressed as the intersection of two planes

$$h_i^T P + D_i = A_i X_1 + B_i X_2 + C_i X_3 + D_i = 0, \quad i = 1, 2 \tag{2.3}$$

where $h_i = [A_i, B_i, C_i]^T$ is the normal vector of ith plane expressed in $\mathcal{C}$. If the degenerate case ($D_1 = D_2 = 0$) is excluded, which means that the origin of $\mathcal{C}$ does not lie on L, as in [77] the projected line of L can be parametrized by l_I with following equation

$$l_I^T \bar{p} = 0$$

with $l_I = \begin{bmatrix} h_1 & h_2 \end{bmatrix} \begin{bmatrix} D_2 \\ -D_1 \end{bmatrix}$, $\bar{p} = p/\lambda$. It can be shown that the parametrization of the projection of L in the image plane is

$$l_\lambda^T p = 0 \tag{2.4}$$

with $l_\lambda = B(\lambda) l_I$, and $B(\lambda) = \text{diag}([1, 1, \lambda])$. Since the parametrization l is not minimal (e.g. l and σl represent the same line where σ is some nonzero constant), a new pair of variables ρ and α is used to parametrize the projected line l as

$$l_\lambda^T p = y_1 s_\alpha + y_2 c_\alpha - \rho = 0 \tag{2.5}$$

where $l_\lambda = [s_\alpha, c_\alpha, -\rho]^T$, $s_\xi = \sin(\xi)$, $c_\xi = \cos(\xi)$. This representation is ambiguous since the same line can be parametrized by $(\rho, \alpha + 2k\pi)$ and $(-\rho, \alpha + (2k+1)\pi)$. Hence, without loss of generality α is restricted to lie in $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$. The parameter α represents the angle that the line makes with the positive y_1-axis with positive α taken in the clockwise direction and restricted to the range $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$. The parameter ρ represents a perpendicular distance of the line to the origin of the image. The sign of ρ can be determined from (2.5). Figure 2.5 shows the graphical representation α

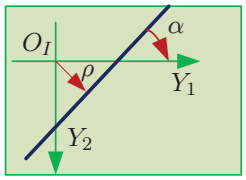


Figure 2.5: A graph of a line in the image plane and its dependence on feature parameters α and ρ.

and ρ. Differentiating (2.5) with respect to time yields

$$\dot{\rho} - (y_1 c_\alpha - y_2 s_\alpha)\dot{\alpha} = \dot{y}_1 s_\alpha + \dot{y}_2 c_\alpha \tag{2.6}$$

From (2.3), the following equation can be obtained

$$\frac{1}{Z} = -\frac{A_i y_1 + B_i y_2 + \lambda C_i}{\lambda D_i} \tag{2.7}$$

From (2.5), y_2 is expressed as a function of y_1 when $\cos\alpha$ is not equal to zero. Alternatively, y_1 can be expressed as a function of y_2 when $\sin\alpha$ is not equal to zero. Assuming L is static and using (2.2) and (2.7), (2.6) is written as

$$-\dot{\alpha} y_1 + c_\alpha(\dot{\rho} + \dot{\alpha}\rho\tan\alpha) = y_1 K_1 c_\alpha \mathrm{v}^c + K_2 c_\alpha \mathrm{v}^c$$

where

$$K_1 = \begin{bmatrix} l_\alpha \lambda s_\alpha & l_\alpha \lambda c_\alpha & -l_\alpha \rho & -\frac{\rho \tan\alpha}{\lambda} & -\frac{\rho}{\lambda} & -\frac{1}{c_\alpha} \end{bmatrix}$$
$$K_2 = \begin{bmatrix} l_\rho \lambda \sin\alpha & l_\rho \lambda \cos\alpha & -l_\rho \rho & \lambda \cos\alpha + \frac{\rho^2}{\lambda \cos\alpha} & -\lambda \sin\alpha & \rho \tan\alpha \end{bmatrix}$$

with $l_\alpha = (A_i - B_i \tan\alpha)/(D_i\lambda)$, $l_\rho = (B_i\rho/c_\alpha + \lambda C_i)/(D_i\lambda)$, and $\mathrm{v}^c = [(v^c)^T\ (\omega^c)^T]^T$. This leads to

$$\dot{l} = L_{vl} v^c + L_{\omega l} \omega^c \tag{2.8}$$

where $l = [\alpha, \rho]$,

$$L_{vl} = \begin{bmatrix} \mu_\alpha \lambda s_\alpha & \mu_\alpha \lambda c_\alpha & -\mu_\alpha \rho \\ \mu_\rho \lambda s_\alpha & \mu_\rho \lambda c_\alpha & -\mu_\rho \rho \end{bmatrix},\ L_{\omega l} = \begin{bmatrix} \frac{\rho s_\alpha}{\lambda} & \frac{\rho c_\alpha}{\lambda} & 1 \\ (\lambda + \frac{\rho^2}{\lambda}) c_\alpha & -(\lambda + \frac{\rho^2}{\lambda}) s_\alpha & 0 \end{bmatrix}$$

and $\mu_\alpha = -(A_i c_\alpha - B_i s_\alpha)/(D_i\lambda), \mu_\rho = (A_i \rho s_\alpha + B_i \rho c_\alpha + C_i \lambda)/(D_i\lambda)$.

2.2.2 Quadrotor Modelling

As shown in Figure 2.6, a navigation frame $\mathcal{N}$ is introduced to describe the dynamics of the quadrotor. This frame is fixed to a point on the Earth with a basis $\{n_1, n_2, n_3\}$,

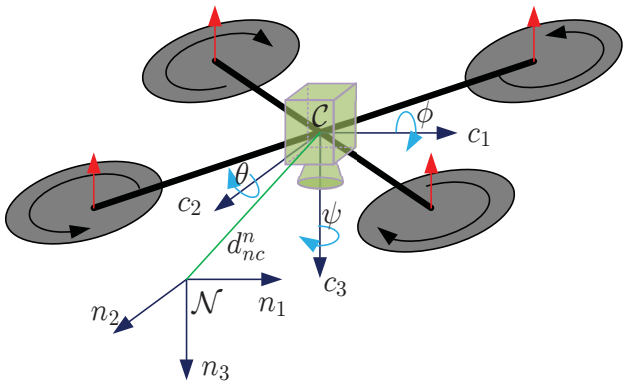


Figure 2.6: Frame definitions used for quadrotor modelling. The Euler angles ϕ-θ-ψ and displacement d^n_{nc} are shown.

which is oriented north, east, and down, respectively. The frame $\mathcal{N}$ is approximately inertial as the earth rotates slowly. The Pixy is rigidly fixed on the quadrotor as shown in Figure 2.6. The origin of the camera frame $\mathcal{C}$ is assumed to be fixed at the centre of mass of the quadrotor with its basis $\{c_1, c_2, c_3\}$ oriented forward, right, down, respectively, and the quadrotor in this thesis adopts the "×" configuration. The vector $d^n_{nc} = [d_1, d_2, d_3]^T$ denotes the displacement of the origin of $\mathcal{C}$ in $\mathcal{N}$, which measures the translational motion of the vehicle. The rotation of $\mathcal{C}$ related to $\mathcal{N}$ is represented by a matrix $R : \mathcal{C} \to \mathcal{N}$ given by $X^n = RX^c$, where X^n is a vector expressed in $\mathcal{N}$, and X^c its the same vector expressed in $\mathcal{C}$. This rotation matrix

belongs to the special orthogonal group $SO(3)$ with the properties $R^T R = I$ and $\det(R) = 1$. The kinematics of R is

$$\dot{R} = R\left[\omega^c\right]_\times \tag{2.9}$$

where for any $x \in \mathbb{R}^3$

$$\left[x\right]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

Although $SO(3)$ is a three-dimensional manifold, there is no globally defined three parameter coordinate chart of $SO(3)$ [78]. Unit quaternions and Euler angles are two popular choices for the parametrization of $SO(3)$ [79]. Unit quaternions provide a nonsingular parametrization of $SO(3)$ and lie in a non-Euclidean 3D sphere $\mathbb{S}^3$. However, this parametrization is not unique which can lead to *unwinding* behaviour [80]. Here, the so called ZYX Euler angles are used and the rotation matrix R is expressed as

$$\begin{aligned} R = R_{n_3}(\psi)R_{n_2}(\theta)R_{n_1}(\phi) &= \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \\ &= \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ c_\theta s_\psi & s_\psi s_\theta s_\phi + c_\psi c_\phi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \end{aligned}$$

where R_{n_k} denotes an elementary rotation about the n_k axis, and ϕ denotes roll, θ denotes pitch, and ψ denotes yaw. The kinematics expressed in the Euler angles is

$$\omega^c = W\left(\eta\right)\dot{\eta} \tag{2.10}$$

where $\eta = \left[\phi, \theta, \psi\right]^T$, and

$$W\left(\eta\right) = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix}$$

The Euler angle parametrization suffers from singularities which occur when $\theta = (2k+1)\pi/2,\ k \in \mathbb{Z}$. These singularities are sometimes called *gimbal lock* and are normally only encountered in acrobatic flight which is not typical in visual servoing. The Euler angle parametrization has the advantage of being an intuitive representation for attitude and is a common choice in field-tested autopilots, e.g. [28, 71, 81].

The quadrotor is treated as a rigid body, using the Newton-Euler approach [78]

the dynamics of the rigid body is written as

$$\dot{v}^c = -\left[\omega^c\right]_\times v^c + gR^T n_3 + \frac{F^c}{m} \tag{2.11a}$$

$$\dot{\omega}^c = -J^{-1}\left[\omega^c\right]_\times J\omega^c + J^{-1}\tau^c \tag{2.11b}$$

where g is the acceleration due to gravity, m is the mass of the vehicle, J is the inertia of the vehicle, F^c is the applied force expressed in $\mathcal{C}$, τ^c is the applied torque expressed in $\mathcal{C}$. The quadrotor's mass is approximately distributed symmetrically about its camera frame axes, we take $J \approx \mathrm{diag}([J_1, J_2, J_3])$. As shown in Figure 2.7,

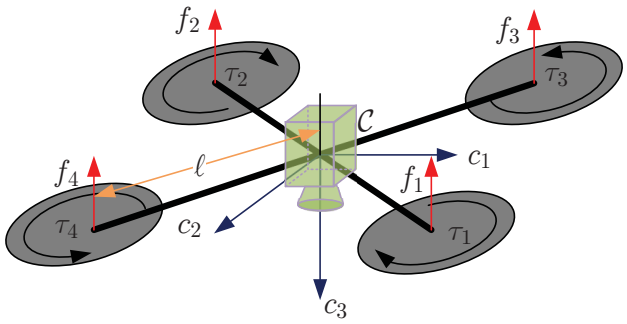


Figure 2.7: The input force and torque model for a quadrotor with cross configuration.

the quadrotor has four fixed-pitch propellers in a cross configuration with two pairs of propellers rotating in an opposite directions. Each propellers will create a thrust f_i and drag moment τ_i. Here the four motors are assumed to be identical, and the input models for f_i and τ_i are assumed to be proportional to the square of the physical input as

$$f_i = K_T \tilde{W}_i^2 \tag{2.12a}$$

$$\tau_i = K_\tau \tilde{W}_i^2 \tag{2.12b}$$

where K_T, K_τ are aerodynamic constants, $\tilde{W} = W_i - W_{min}$, W_i is the PWM width fed into the ith ESC driving the ith motor, and W_{min} is the minimum pulse width to rotate the propeller. The applied force F^c and torque τ^c are obtained as

$$F^c = T_M c_3 \tag{2.13a}$$

$$\tau^c = \begin{bmatrix} \frac{\ell}{\sqrt{2}}(f_2 + f_3 - f_1 - f_4) \\ \frac{\ell}{\sqrt{2}}(f_1 + f_3 - f_2 - f_4) \\ \sum_{i=1}^4 (-1)^i \tau_i \end{bmatrix} \tag{2.13b}$$

where $T_M = \sum_{i=1}^4 f_i$ is the thrust created by four propellers. In visual servoing of UAVs, the speed of the vehicle is slow and aggressive motion is avoided. The vehicle

usually flies near a hover configuration. As pointed in [68, 82–84], in hover small body forces and drag forces are negligible, and gyroscopic effects of propellers tend to cancel. Hence, small body forces, drag forces, and gyroscopic forces are ignored in the model using this thesis.

2.3 DIBVS Model and Problem Formulation

With basic image measurements, i.e., the image coordinates of points or the parameters of image lines, a more complicated feature s can be formed. Assuming the target is static, a dynamic IBVS model is formed by combining the image kinematics of s, rotational kinematics (2.10), and the dynamic model of the quadrotor (2.11) as

$$\dot{s} = L_v v^c + L_\omega \omega^c \tag{2.14a}$$

$$\dot{v}^c = -\left[\omega^c\right]_\times v^c + gR^T n_3 + \frac{F^c}{m} \tag{2.14b}$$

$$\dot{\eta} = W^{-1}(\eta)\omega^c \tag{2.14c}$$

$$\dot{\omega}^c = -J^{-1}\left[\omega^c\right]_\times J\omega^c + J^{-1}\tau^c \tag{2.14d}$$

where L_v and L_ω are the image Jacobians or interaction matrices. The control objective of DIBVS is to minimize the image feature error $e_s = s - s^*$ through considering the quadrotor's dynamics, where s^* is the desired image feature value. Since the quadrotor is symmetrical around its axes, The inertial matrix J is approximated as $J = \mathrm{diag}([J_1, J_2, J_3])$. Table 2.1 summarizes the quadrotor and camera model parameters.

m	1.6 kg
J_1	0.03 kg·m^2
J_2	0.03 kg·m^2
J_3	0.05 kg·m^2
λ	2.8 mm
Image size	1/4 ” (3.2×2.4) mm
FoV	75° horizontal and 47° vertical
ℓ	0.25 m

Table 2.1: Quadrotor and camera model parameters.

2.4 Conclusion

The indoor ANCL quadrotor platform has been developed to perform experimental research on visual servoing. It provides a flexible environment for research which

is a step towards developing an outdoor platform for power line inspection. The platform benefits from low cost open-source systems: the PX4 autopilot, the Pixy onboard computer vision system, and the QgroundControl ground station software. To achieve visual servoing tasks the open-source autopilot software is customized and integrated into a generic quadrotor vehicle. A pinhole camera model is used to describe image formation and the image kinematics of two basic features: 2D point image coordinate feature and 2D line parametrization image feature. A rigid body model is used to model the quadrotor. A simple and commonly used input model results by ignoring effects such as gyroscopic effects, small body forces, and drag force. Lastly, the dynamic IBVS model is presented, and the IBVS problem is formulated.

Chapter 3

State Transformation-Based DIBVS

As discussed in Chapter 1, classical IBVS can not be applied directly to an underactuated UAV where the velocities v^c and ω^c are not independently controlled. In the case of UAVs, vehicle dynamics need to be considered and a scheme which includes their effect is termed *dynamic IBVS (DIBVS)*. As mentioned in [29], the image kinematics of s in (2.14a) has dependence on angular velocity ω^c and linear velocity v^c. The angular velocity dependence destroys the triangular cascade structure which the rigid body dynamics possesses when expressed in the inertial frame. The lack of structure makes DIBVS a challenging global stabilization problem. Hence, in [29] a spherical projection model is used instead of a perspective projection model (2.1). This approach is referred to as spherical projection IBVS and is described in [29–31, 33]. Typically, spherical projection IBVS is characterized by an ill-conditioned interaction matrix. Partitioning and rescaling approaches were proposed in [32] and [33] to address the issue of ill-conditioning. This work only removes the problem locally in the partitioning approach and no rigorous proof is given in the rescaling approach.

In this chapter an approach using a state transformation is considered. The proposed approach requires the attitude information as in spherical projection IBVS, but leads to a simpler interaction matrix. The approach adopts a two-loop cascade structure: outer IBVS loop and inner attitude loop. This structure is described in Section 3.1. Section 3.2 presents the proposed approach and a simple Planar Vertical Take-off and Landing (PVTOL) aircraft example is used to illustrate the proposed scheme. Lastly, in Section 3.3 a DIBVS law using image coordinates of a single point as the image feature is developed to show how the state transformation-based approach is applied to control the motion of the quadrotor. Since only a single point is used, only lateral position of the quadrotor is regulated to zero.

3.1 Cascade Structure of DIBVS Model and Controller

The DIBVS model (2.14) is divided into two subsystems in cascade: an outer-loop which outputs the image feature error e_s with applied force F^c and attitude R as inputs, and an inner loop which outputs attitude R for an input τ^c. The controller itself has inner-outer loop structure: the outer loop input is the image feature error e_s and it outputs F^c and desired attitude R^* to the inner loop which outputs τ^c. Figure 3.1 shows a block diagram of the closed-loop system and the cascade structure of the controller. The main reason for adopting this control structure is

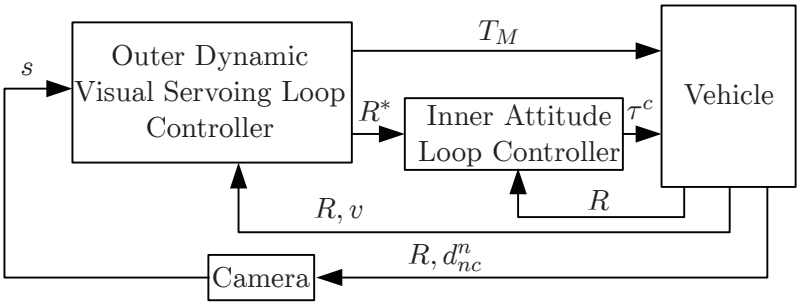


Figure 3.1: Inner-outer loop control structure.

that it provides a reliable and simple solution for motion control. In particular, each loop becomes fully actuated and this simplifies the control design and gain tuning to optimize performance. A similar two-loop structure for conventional position control is found in many autopilots, e.g. [28], [71]. Stability of the closed-loop system requires the inner loop to have a relatively high bandwidth. which is achieved with high-gain. The visual servoing problem naturally fits the inner-outer loop structure since AHRS measurements are relatively fast compared to image feedback. One alternative is to use ω^c as the control for the outer loop, e.g. [33]. However, a benefit of the proposed structure is its direct control of attitude which ensure attitude remains in a safe region. It also makes it easier to keep the target in the camera's FoV. A theoretical analysis of this structure can be found in [85].

3.2 General State Transformation-based Approach

This section begins with the general philosophy of the state transformation-based approach. In Section 3.2.2 an example is provided to illustrate the theory. Section 3.2.3 and 3.2.4 present the state transformation formula for a 3D point and a 3D line, respectively.

3.2.1 Philosophy of the State Transformation-based Approach

From(2.10), (2.14b), and (2.14b), the outer loop is written as

$$\dot{s} = L_v v^c + L_{\eta_1}\dot{\eta}_1 + L_{\eta_2}\dot{\psi} \tag{3.1a}$$

$$\dot{v}^c = H_u + H_{\eta_1}\dot{\eta}_1 + H_{\eta_2}\dot{\psi} \tag{3.1b}$$

where $[L_{\eta_1}, L_{\eta_2}] = L_\omega W$, $H_u = gR^T E_3 + F^c/m$, $[H_{\eta_1}, H_{\eta_2}] = [v^c]_\times W$, and $\eta_1 = [\phi, \theta]^T$. The control variables for (3.1) are F^c, η_1, and ψ. The yaw angle ψ can be controlled independently. It can be seen that the outer-loop contains the attitude input and its derivative. The appearance of the input derivative makes the control design challenging. One way to solve this problem is to lower the input derivative by a state transformation. A state transformation is defined as $s^a(s, \eta_1)$ and $v^a(v^c, \eta_1)$ with dependence on input η_1. The time derivative of the state transformation is

$$\dot{s}^a = \frac{\partial s^a}{\partial s} L_v v^c + \left(\frac{\partial s^a}{\partial s} L_{\eta_1} + \frac{\partial s^a}{\partial \eta_1}\right)\dot{\eta}_1 + \frac{\partial s^a}{\partial s} L_{\eta_2}\dot{\psi}$$

$$\dot{v}^a = \frac{\partial v^a}{\partial v^c} H_u + \left(\frac{\partial v^a}{\partial v^c} H_{\eta_1} + \frac{\partial v^a}{\partial \eta_1}\right)\dot{\eta}_1 + \frac{\partial v^a}{\partial v^c} H_{\eta_2}\dot{\psi}$$

If s^a and v^a satisfy the system of first-order linear homogeneous PDEs

$$\frac{\partial s^a}{\partial s} L_{\eta_1} + \frac{\partial s^a}{\partial \eta_1} = 0 \tag{3.2a}$$

$$\frac{\partial v^a}{\partial v^c} H_{\eta_1} + \frac{\partial v^a}{\partial \eta_1} = 0 \tag{3.2b}$$

the input derivative in (3.1) can be eliminated and the dynamics (3.1) expressed in the new state is

$$\dot{s}^a = \frac{\partial s^a}{\partial s} L_v v^c + \frac{\partial s^a}{\partial s} L_{\eta_2}\dot{\psi}$$

$$\dot{v}^a = \frac{\partial v^a}{\partial v^c} H_u + \frac{\partial v^a}{\partial v^c} H_{\eta_2}\dot{\psi}$$

Hence, to choose a state transformation, the PDEs (3.2) have to be solved first. Although it is usually difficult to solve PDEs explicitly, it is shown in following sections that this PDEs are solvable and a general explicit solution is also given.

3.2.2 Example: PVTOL Aircraft

A Planar Vertical Taking-Off and Landing (PVTOL) aircraft is considered to illustrate the state transformation approach with a point measurement. By setting the first and the third components of ω^c, and the second component of v^c, roll and yaw angle, i.e., ω_1^c, ω_3^c, v_2^c, ϕ, ψ, and y_2 in (2.2) and (2.14b) to zeros, the PVTOL aircraft

model is obtained as

$$\dot{y}_1 = \begin{bmatrix} -\frac{\lambda}{X_3} & \frac{y_1}{X_3} \end{bmatrix} \begin{bmatrix} v_1^c \\ v_3^c \end{bmatrix} - \frac{y_1^2 + \lambda^2}{\lambda} \omega_2 \tag{3.3a}$$

$$\begin{bmatrix} \dot{v}_1^c \\ \dot{v}_3^c \end{bmatrix} = -\omega_2 \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} v_1^c \\ v_3^c \end{bmatrix} + g \begin{bmatrix} -s_\theta \\ c_\theta \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{T_M}{m} \end{bmatrix} \tag{3.3b}$$

Relative to the 3D UAV model, (3.3) has fewer states but keeps the essential structure of (2.2) and (2.14b). The PDEs (3.2) become

$$\frac{\partial y_1^a}{\partial y_1} \frac{y_1^2 + \lambda^2}{\lambda} - \frac{\partial y_1^a}{\partial \theta} = 0 \tag{3.4a}$$

$$\frac{\partial v_i^a}{\partial v_1^c} v_3^c - \frac{\partial v_i^a}{\partial v_3^c} v_1^c - \frac{\partial v_i^a}{\partial \theta} = 0 \tag{3.4b}$$

where $i = 1, 3$. The general solutions of (3.4b) are

$$y_1^a = \kappa_1 \left(\arctan\left(\frac{y_1}{\lambda} \right) + \theta \right) \tag{3.5}$$

$$v_i^a = \kappa_{2i}((v_1^c)^2 + (v_3^c)^2, \arctan(\frac{v_1^c}{v_3^c}) + \theta) \tag{3.6}$$

where κ_1 and κ_2 are smooth functions in their arguments. If the particular solutions $\kappa_1(\zeta) = \tan\zeta$, $\kappa_{21}(\zeta_1, \zeta_2) = \zeta_1 s_{\zeta_2}$, $\kappa_{22}(\zeta_1, \zeta_2) = \zeta_1 c_{\zeta_2}$ are chosen, the state transformation is

$$y_1^a = \lambda \frac{c_\theta y_1 + s_\theta \lambda}{-s_\theta y_1 + c_\theta \lambda}$$

$$\begin{bmatrix} v_1^a \\ v_3^a \end{bmatrix} = \begin{bmatrix} c_\theta & s_\theta \\ -s_\theta & c_\theta \end{bmatrix} \begin{bmatrix} v_1^c \\ v_3^c \end{bmatrix}$$

and the outer loop expressed in the new state is

$$\dot{y}_1^a = \frac{\lambda}{X_3 \left(-s_\theta y_1 + c_\theta \lambda \right)} \begin{bmatrix} -\lambda & y_1^a \end{bmatrix} \begin{bmatrix} v_1^a \\ v_3^a \end{bmatrix} \tag{3.7a}$$

$$\begin{bmatrix} \dot{v}_1^a \\ \dot{v}_3^a \end{bmatrix} = g \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \frac{T_M}{m} \begin{bmatrix} s_\theta \\ c_\theta \end{bmatrix} \tag{3.7b}$$

As shown in Figure 3.2, a virtual camera frame $\mathcal{A}$ with basis $\{c_1^a, c_3^a\}$ is defined, which has the same translational motion as $\mathcal{C}$ but with zero pitch. Any point in $\mathcal{A}$ has coordinates

$$P^a = \begin{bmatrix} X_1^a \\ X_3^a \end{bmatrix} = \begin{bmatrix} c_\theta & s_\theta \\ -s_\theta & c_\theta \end{bmatrix} P$$

The term $\lambda/\left(X_3\left(-s_\theta y_1 + c_\theta \lambda\right)\right)$ in (3.7a) can be written as $1/(X_3 c_\theta - X_1 s_\theta)$ whose

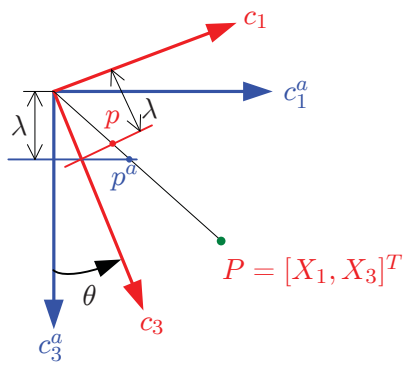


Figure 3.2: Virtual camera for the PVTOL aircraft.

denominator is the depth X_3^a of the target in the virtual camera frame. Since the vehicle always hovers above the target $X_3^a > 0$. It can be also observed that

$$y_1^a = \lambda \frac{X_3 s_\theta + X_1 c_\theta}{X_3 c_\theta - X_1 s_\theta} = \lambda \frac{X_1^a}{X_3^a}$$

This is the image point in the virtual camera plane. Hence, the state transformation approach includes the virtual camera approach in [27] as a special case.

3.2.3 State Transformation for 3D Point Target

Since points and lines are basic features in a image, this subsection and the following describe the state transformation for a single point and a single line with 3D outer loop (2.14a) and (2.14b). This state transformation will be used in the following chapters.

The state transformations for the point coordinate feature case are denoted as $y_i^a\left(y_1, y_2, \phi, \theta\right)$, $v_i^a\left(v_1, v_2, \phi, \theta\right)$, $i = 1, 2$. From (2.2), (3.1), (3.2a), the new state y_i^a has to satisfy following PDEs

$$G_{jp}(y_i^a) = 0, \quad j, i = 1, 2 \tag{3.8}$$

where

$$G_{1p} = \frac{y_1 y_2}{\lambda}\frac{\partial}{\partial y_1} + \frac{y_2^2 + \lambda^2}{\lambda}\frac{\partial}{\partial y_2} + \frac{\partial}{\partial \phi}$$

$$G_{2p} = \left(\frac{y_1^2 + \lambda^2}{\lambda} c_\phi + y_2 s_\phi\right)\frac{\partial}{\partial y_1} + \left(\frac{y_1 y_2 c_\phi}{\lambda} - y_1 s_\phi\right)\frac{\partial}{\partial y_2} - \frac{\partial}{\partial \theta}$$

As discussed in Appendix A, a solution to (3.8) exists and following the method

presented there a general solution can be computed as

$$y_i^a = \kappa_{3i}\left(\frac{s_\phi c_\theta y_2 + \lambda c_\phi c_\theta - y_1 s_\theta}{\lambda s_\phi - y_2 c_\phi}, \frac{s_\phi s_\theta y_2 + c_\phi s_\theta \lambda + y_1 c_\theta}{\lambda s_\phi - y_2 c_\phi}\right) \tag{3.9}$$

where $i = 1, 2$, and κ_{3i} is an arbitrary smooth function in its arguments. The PDEs (3.2b) are

$$G_{jv}(v_i^a) = 0, \quad j = 1, 2, i = 1, 2, 3 \tag{3.10}$$

where

$$G_{1v} = v_3^c \frac{\partial}{\partial v_2^c} - v_2^c \frac{\partial}{\partial v_3^c} + \frac{\partial}{\partial \phi}$$

$$G_{2v} = -\left(v_3^c c_\phi + v_2^c s_\phi\right)\frac{\partial}{\partial v_1^c} + v_1^c s_\phi \frac{\partial}{\partial v_2^c} + v_1^c c_\phi \frac{\partial}{\partial v_3^c} + \frac{\partial}{\partial \theta}$$

As with (3.8) the general solution is obtained as

$$v_i^a = \kappa_{4i}\big(v_2 c_\phi - v_3 s_\phi, v_1 c_\theta + v_2 s_\phi s_\theta + v_3 c_\phi s_\theta, -v_1 s_\theta + v_2 s_\phi c_\theta + v_3 c_\phi c_\theta\big) \tag{3.11}$$

where $i = 1, 2, 3$, and κ_{4i} is an arbitrary smooth function. Taking

$$\kappa_{31}(\zeta_1, \zeta_2) = \lambda \frac{E_1^T R_c q}{E_3^T R_c q}, \quad \text{and} \quad \kappa_{32}(\zeta_1, \zeta_2) = \lambda \frac{E_2^T R_c q}{E_3^T R_c q} \tag{3.12}$$

where $q = [\zeta_2, -1, \zeta_1]^T$, R_c is a constant rotation matrix, $E_1 = [1, 0, 0]^T$, $E_2 = [0, 1, 0]^T$, $E_3 = [0, 0, 1]^T$, and

$$\kappa_{4i}(\zeta_1, \zeta_2, \zeta_3) = E_i^T R_c [\zeta_1, \zeta_2, \zeta_3]^T, \; i = 1, 2, 3 \tag{3.13}$$

the outer loop transformed in the new coordinates is

$$\begin{bmatrix} \dot{y}_1^a \\ \dot{y}_2^a \end{bmatrix} = L_{vp}^a v^a + L_{\eta_2}^a R_c^T E_3 \dot{\psi} \tag{3.14a}$$

$$\dot{v}^a = \left[\dot{\psi} R_c^T E_3\right]_\times v^a + F^a \tag{3.14b}$$

where

$$L_{vp}^a = \begin{bmatrix} -\frac{\lambda}{X_3^a} & 0 & \frac{y_1^a}{X_3^a} \\ 0 & -\frac{\lambda}{X_3^a} & \frac{y_2^a}{X_3^a} \end{bmatrix}, \; L_{\eta_2}^a = \begin{bmatrix} \frac{y_1^a y_2^a}{\lambda} & -\frac{(y_1^a)^2 + \lambda^2}{\lambda} & y_2^a \\ \frac{(y_2^a)^2 + \lambda^2}{\lambda} & -\frac{y_1^a y_2^a}{\lambda} & -y_1^a \end{bmatrix}$$

and $F^a = R_c R_{\theta\phi} g_1(F^c, \eta)$, $g_1(F^c, \eta) = gR^T n_3 + F^c/m$, $X_3^a = E_3^T R_c R_{\theta\phi} P$, $R_{\theta\phi} = R_{n_2}(\theta) R_{n_1}(\phi)$. The state transformations (3.12) and (3.13) generalize the work in [27] to convert the points coordinate in the real camera image to the coordinates of a point projected to a virtual camera with zero roll and pitch. It can readily be shown that if R_c in (3.12) and (3.13) is replaced with $R_c R_{n_3}(\psi)$, the yaw rate $\dot{\psi}$

in (3.14) will disappear and this is the virtual camera used in [52].

3.2.4 State Transformation for 3D Line Target

The state transformation for a line feature is $l_i^a(\alpha, \rho, \phi, \psi)$. Using (2.8) and (3.1), (3.2a) becomes

$$G_{jl}(l_i^a) = 0, \quad j, i = 1, 2 \tag{3.15}$$

where

$$G_{1l} = \frac{\rho s_\alpha}{\lambda}\frac{\partial}{\partial \alpha} + \left(\lambda + \frac{\rho^2}{\lambda}\right) c_\alpha \frac{\partial}{\partial \rho} + \frac{\partial}{\partial \phi}$$

$$G_{2l} = \left(\frac{\rho c_\alpha}{\lambda} c_\phi - s_\phi\right)\frac{\partial}{\partial \alpha} - \left(\lambda + \frac{\rho^2}{\lambda}\right) s_\alpha c_\phi \frac{\partial}{\partial \rho} + \frac{\partial}{\partial \theta}$$

Following the same procedure used to solve (3.8) and (3.10) a general solution to (3.15) can be obtained . Similar to the 3D point case, the particular solution similar to (3.12) is given by

$$l^a = \begin{bmatrix} \alpha^a \\ \rho^a \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{l_{e1}}{l_{e2}}\right) \\ -\frac{l_{e3}}{\sqrt{l_{e1}^2 + l_{e2}^2}} \end{bmatrix} \tag{3.16}$$

where

$$l_e = \begin{bmatrix} l_{e1} \\ l_{e2} \\ l_{e3} \end{bmatrix} = B R_c R_{\theta\phi} B^{-1} \begin{bmatrix} s_\alpha \\ c_\alpha \\ -\rho \end{bmatrix} \tag{3.17}$$

and $B = \mathrm{diag}([1, 1, \lambda])$, which is also defined in (2.4). According to (3.2), the state transformation for velocity v^a should satisfy the same PDEs (3.10). If the same solution in (3.13) is chosen. The new outer loop with the state l^a and v^a is

$$\dot{l}^a = L_{vl}^a v^a + L_{\eta 2}^a R_c^T E_3 \dot{\psi} \tag{3.18a}$$

$$\dot{v}^a = \left[\dot{\psi} R_c E_3\right]_\times v^a + R_c R_{\theta\phi} g_1(F^c, \eta) \tag{3.18b}$$

where

$$L_{vl}^a = \begin{bmatrix} \mu_\alpha^a \lambda s_{\alpha^a} & \mu_\alpha^a \lambda c_{\alpha^a} & -\mu_\alpha^a \rho^a \\ \mu_\rho^a \lambda s_{\alpha^a} & \mu_\rho^a \lambda c_{\alpha^a} & -\mu_{\rho^a} \rho^a \end{bmatrix},\ L_{\eta_2}^a = \begin{bmatrix} \frac{\rho^a s_{\alpha^a}}{\lambda} & \frac{\rho^a c_{\alpha^a}}{\lambda} & 1 \\ \left(\lambda + \frac{(\rho^a)^2}{\lambda}\right) c_{\alpha^a} & -\left(\lambda + \frac{(\rho^a)^2}{\lambda}\right) s_{\alpha^a} & 0 \end{bmatrix}$$

$$\mu_\alpha^a = -\frac{(A_i^a c_{\alpha^a} - B_i^a s_{\alpha^a})}{(D_i^a \lambda)},\ \mu_\rho^a = \frac{(A_i^a \rho s_\alpha + B_i^a \rho c_{\alpha^a} + C_i^a \lambda)}{(D_i^a \lambda)}$$

and $[A_i^a, B_i^a, C_i^a]^T = R_c R_{\theta\phi}[A_i, B_i, C_i]^T, D_i^a = D_i$. It can be shown that if R_c is chosen as a function of ψ that makes $A_i^a = B_i^a = 0$, The equation (3.18a) can be simplified by setting $\mu_\alpha^a = 0$, $\mu_\rho^a = -1/X_3^a$, where X_3^a is the depth of any point on the line.

3.3 State Transformation-based DIBVS for Lateral Motion Control

3.3.1 Controller Design

In this section the quadrotor is assumed to fly at a constant height and have zero yaw motion, i.e., $v_3^c = \omega_3^c = 0$. As well, a small η assumption is considered, which implies $\dot{\eta} = \omega^c$. With the image coordinates of one point as the image feature, the outer-loop becomes

$$\dot{y}_1 = -\frac{\lambda}{X_3} v_1^c - \frac{\lambda^2 + y_1^2}{\lambda}\dot{\theta} + \frac{y_1 y_2}{\lambda}\dot{\phi} \tag{3.19a}$$

$$\dot{v}_1^c = -g\theta \tag{3.19b}$$

$$\dot{y}_1 = -\frac{\lambda}{X_3} v_2 + \frac{\lambda^2 + y_2^2}{\lambda}\dot{\phi} - \frac{y_1 y_2}{\lambda}\dot{\theta} \tag{3.19c}$$

$$\dot{v}_2^c = g\phi \tag{3.19d}$$

By ignoring the two coupling terms $y_1 y_2 \dot{\phi}/\lambda$ and $y_1 y_2 \dot{\theta}/\lambda$, the outer-loop (3.19) can be written as two independent nominal outer-loop as

$$\dot{y}_1 = -\frac{\lambda}{X_3} v_1^c - \frac{\lambda^2 + y_1^2}{\lambda}\dot{\theta} \tag{3.20a}$$

$$\dot{v}_1^c = -g\theta \tag{3.20b}$$

$$\dot{y}_2 = -\frac{\lambda}{X_3} v_2 + \frac{\lambda^2 + y_2^2}{\lambda}\dot{\phi} \tag{3.20c}$$

$$\dot{v}_2^c = g\phi \tag{3.20d}$$

The (y_1, v_1^c) and (y_2, v_2^c) subsystems of dynamics (3.20) can each be interpreted as a PVTOL aircraft model with constant altitude. The inputs for (3.20) are the Euler angles θ and ϕ, respectively. It can be seen that the input derivative also appears in (3.20). The state transformation approach is applied to design a DIBVS controller for (3.20). If κ_1 in (3.5) is chosen as $\kappa_1(\zeta) = \zeta$, a particular solution

$$y_1^a = \arctan\left(\frac{y_1}{\lambda}\right) + \theta$$

can be obtained, and the transformed (y_1, v_1^c) subsystem is

$$\dot{y}_1^a = -\frac{v_1^c \cos^2(y_1^a - \theta)}{X_3} \tag{3.21a}$$

$$\dot{v}_1^c = -g\theta \tag{3.21b}$$

A similar approach can be applied to obtain a new state for the (y_2, v_2^c) subsystem:

$$y_2^a(y_2, \phi) = \arctan\left(\frac{y_2}{\lambda}\right) - \phi$$

and the transfomed system is

$$\dot{y}_2^a = -\frac{v_2^c \cos^2(y_2^a - \phi)}{X_3} \tag{3.22a}$$

$$\dot{v}_2^c = g\phi \tag{3.22b}$$

Theorem 3.1. *The origin of the dynamics* (3.20)*with input*

$$\theta = -k_1 k_2 \frac{k_1 \arctan(\frac{y_1}{\lambda}) - v_1^c}{1 + k_1^2 k_2} \tag{3.23a}$$

$$\phi = k_3 k_4 \frac{k_3 \arctan(\frac{y_2}{\lambda}) - v_2^c}{1 + k_3^2 k_4} \tag{3.23b}$$

is locally exponentially stable if $k_1, k_3 > 0$*, and* $k_2, k_4 > 1/(gX_3)$*.*

Proof. For convenience a new variable $q = y_1^a - v_1^c/k_1$ is introduced. From (3.21), the following kinematics can be obtained

$$\dot{y}_1^a = -\frac{k_1(y_1^a - q)\cos^2(y_1^a - \theta)}{X_3}$$

$$\dot{q} = -\frac{k_1(y_1^a - q)\cos^2(y_1^a - \theta)}{X_3} + \frac{g\theta}{k_1}$$

The following Lyapunov function candidate is chosen as

$$V_\theta = \frac{1}{2}(y_1^a)^2 + \frac{1}{2}q^2$$

Its time derivative is

$$\dot{V}_\theta = -\frac{k_1}{X_3}\cos^2(y_1^a - \theta)(y_1^a)^2 + q\left(\frac{k_1 q \cos^2(y_1^a - \theta)}{X_3} + \frac{g\theta}{k_1}\right)$$

As in (3.23a), the control is chosen as

$$\theta = -k_1^2 k_2 q = -k_1^2 k_2 y_1^a + k_1 k_2 v_1^c$$

Then the time derivative of V_θ becomes

$$\dot{V}_\theta = -\frac{k_1}{X_3}\cos^2(y_1^a - \theta)(y_1^a)^2 + \frac{q^2 k_1}{X_3}\left(\cos^2(y_1^a - \theta) - g\theta k_2 X_3\right)$$

To make the time derivative of V_θ negative, the inequality $g\theta k_2 X_3 > 1$ is required

and the state (y_1^a, v_1^c) is restricted to the set

$$\left\{(y_1^a, v_1^c) \in \mathbb{R}^2 : \left|y_1^a + k_1^2 k_2 y_1^a - k_1 k_2 v_1^c\right| < \frac{\pi}{2} - \epsilon, 0 < \epsilon < \frac{\pi}{2}\right\}$$

Hence, the equilibrium point $y_1^a = v_1^c = 0$ is LES and because the state transformation is Lipschitz on its domain, stability is preserved for the original state (y_1, v_1^c). An analogous proof applies to the system (3.20c) and (3.20d) with

$$V_\phi = \frac{1}{2}(y_2^a)^2 + \frac{1}{2}\left(y_2^a - \frac{v_2^c}{k_3}\right)^2$$

□

The choice of controller gains k_1, k_2 determine the shape of the regions of attraction. To compute an estimate of the region of attraction for (3.20) The intersection of the ellipse

$$V_\theta = \frac{1}{2}(y_1^a)^2 + \frac{1}{2}\left(y_1^a - \frac{v_1^c}{k_3}\right)^2 = c$$

with the lines

$$y_1^a + k_1^2 k_2 y_1^a - k_1 k_2 v_1^c = \pm\frac{\pi}{2}$$

is obtained. The expression for c is

$$c = \frac{\pi^2}{8\left(k_1^4 k_2^2 + 1\right)}$$

An estimate of the region of attraction is

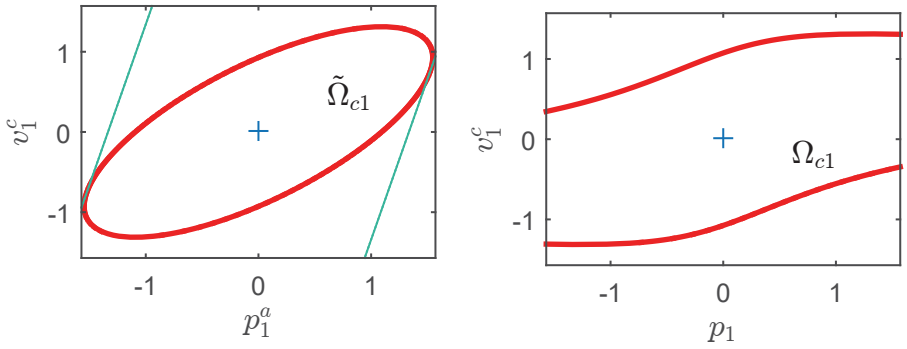
$$\tilde{\Omega}_{c1} = \left\{(y_1^a, v_1^c) \in \mathbb{R}^2 : V_\theta < c\right\}$$

This set can be expressed in the (y_1, v_1^c) coordinates using the coordinate transformation, and this set is denoted as Ω_{c1}. Examples of these estimated regions of attraction are given in Figure 3.3

To show the stability of the system (3.19) with the input (3.23) for the nominal systems (3.20), a result from [86, Lem.9.1] can be used by treating the coupling terms $y_1 y_2 \dot{\phi}/\lambda$ and $y_1 y_2 \dot{\theta}/\lambda$ as a disturbance. The details of the proof can be found in [58].

3.3.2 Simulation Results

In this section the proposed controller in (3.23) is compared with a popular and experimentally validated controller proposed in [33] and referred to as the *spherical projection IBVS*. The approach in [33] is simplified to control only lateral motion



(a) Region of attraction in new state (b) Region of attraction in original state

Figure 3.3: Estimates of region of attraction for controller gains $k_1 = 0.6$, $k_2 = 0.49$.

which results in the IBVS law

$$gR^T n_3 + \frac{F^c}{m} = -\frac{k_1^2 k_2}{m}\delta_2 \tag{3.24}$$

where

$$\delta_2 = \frac{mv}{k_1} - \delta_1 \text{ and } \delta_1 = s - R^T s^*$$

where $k_1, k_2 > 0$ are controller gains, and s is the image feature in spherical coordinates, and $s^* = [0, 0, 1]^T$ is the desired image feature. The spherical coordinates are calculated from the 2D image coordinates $y = [y_1, y_2, \lambda]^T$ using $s = y/\|y\|$. The spherical coordinates are calculated assuming no calibration error. The expressions for the spherical projection IBVS law (3.24) are more computationally complex relative to the proposed approach. Both controllers calculate reference attitude angle ϕ and θ, which provide the reference attitude to the inner loop. The inner attitude loop is controlled using

$$\tau_1^c = K_{p\phi} e_\phi + K_{d\phi} \dot{e}_\phi$$
$$\tau_2^c = K_{p\theta} e_\theta + K_{d\theta} \dot{e}_\theta$$

where $e_\phi = \phi^* - \phi$, $e_\theta = \theta^* - \theta$, and ϕ^*, θ^* are desired reference attitude from (3.23) and (3.24). It is remarked that the control laws for τ_1^c and τ_2^c determine the propeller velocities based on the simplified input model given in Section 2.2.2. Additionally, the Proportional-Integral-Derivative (PID) controlled height dynamics is included with

$$F_3^c = K_{pz} e_z + K_{iz} \int_0^t e_z(\tau) d\tau + K_{dz} \dot{e}_z$$

where $e_z = d_3^* - d_3$, and d_3^* denotes the desired height setpoint. The controller gains $K_{p\theta} = K_{p\phi} = 16$, $K_{d\theta} = K_{d\phi} = 6.4$, $K_{pz} = 16$, $K_{iz} = 0.1$ and $K_{dz} = 6.4$ are used. Both controllers use the same controller gains $k_1 = 0.6$ and $k_2 = 0.49$. The vehicle's

height is initialized to 1 m above the ground (i.e., $d_3(0) = -1$ m) and take a desired height setpoint of 1.5 m (i.e., $d_3^* = -1.5$ m). The model parameters are shown in Table 2.1. Images are captured with a 640×480 resolution. The initial value of state and control gains are summarized in Table 3.1. The system states using the

k_1	0.6
k_2	0.49
$d_3(0)$	-1.0 m
d_3^*	-1.5 m
$v(0)$	$(0, 0)$ m/s
$y(0)$	$(150, -150)$ pixels

Table 3.1: Controller gains and simulation parameters.

proposed control and the controller (3.24) are shown in Figure 3.4, where the system states using controller (3.24) are plotted in solid blue line, those corresponding to the proposed controller are plotted with a dashed blue line, and the reference angles for the inner PD loop are in red. From Figure 3.4 it is observed that the performance

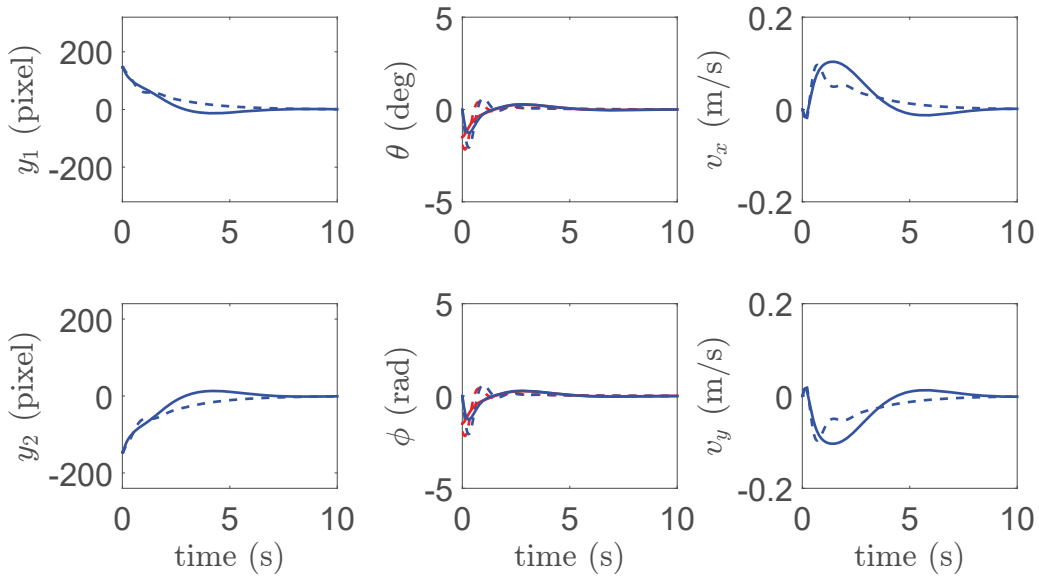


Figure 3.4: Comparison of simulated state trajectories of the proposed approach (solid blue) with spherical projection IBVS (3.24) (dashed blue). Reference angles to the inner loop are shown in red.

of the two controllers is similar. The proposed approach has the benefit of being simpler computationally relative to (3.24).

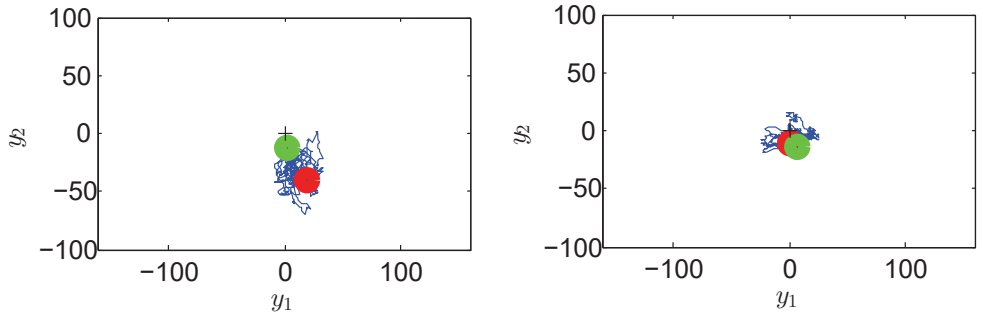
3.3.3 Experimental Results

In this section the experimental performance of the proposed control is demonstrated on the ANCL quadrotor platform. These experiments demonstrate the method's robustness to various model uncertainties and the feasibility of implementation on embedded hardware where resources are limited. As in Section 3.3.2, a comparison with spherical IBVS (3.24) is provided. In practice the proposed approach requires implementation of an inner attitude control. As is commonly done in practice, a simple PID structure is chosen for roll and pitch. An integral term was added to improve robustness. Tuning the inner loop gains required an outer loop position control due to the coupling of the rotational and translational dynamics. The Vicon system initially provided position feedback in 3 DoF to facilitate this tuning. Once a satisfactory inner-loop performance was obtained, DIBVS control one axis is implemented at time. The control gains were tuned online to obtain $k_1 = 0.6$ and $k_2 = 0.49$. The lateral velocity estimates used in the IBVS control law were obtained from the Vicon system by low pass filtering a finite difference of position. Yaw was regulated using proportional control and the on-board magnetometer. The use of a magnetometer to control yaw indoors was possible even with the existence of disturbance magnetic fields. This was because stabilization only required a constant measurement; not necessarily an accurate measurement of the earth's magnetic field. The height was controlled using PID and the Vicon estimate of altitude and vertical velocity. The experiments were performed in two phases. An initial phase consisted of using the Vicon to control the three translational DoF. This allowed us to avoid ground effects and ensure the image feature was in the camera's FoV. The second phase involved switching to IBVS.

The robustness of the controller was demonstrated for a wide range of initial conditions in d_1, d_2, d_3, and ψ. The intial values of d_1 and d_2 were varied up to 90 cm in all directions from the desired visual feature. The quadrotor was initialized at about 25 cm below the desired height in every experiment. The initial yaw was varied in a range of 60°. The height and yaw controllers would then regulate the height to $d_3 = -1.5$ m and yaw to $\psi = 0$. The experiments show that disturbances due to nonzero v_3^c and $\dot{\psi}$ do not affect closed-loop performance.

The steady state performance of the controllers is shown in Figure 3.5 and 3.6. The image trajectories of a single point using the two controllers are shown in Figure 3.5, where the initial location is shown in red, and the goal location is in green. The system states using the proposed controller are shown in Figure 3.6 in solid blue lines. The reference angles for the inner PID loop are solid red lines.

Clearly, the proposed control demonstrates good hover performance with y_1, y_2 having maximum amplitudes of 71 and 102 pixels, respectively. The Vicon data shows that this corresponds to variation in displacements of 0.18 m and 0.32 m in



(a) Proposed DIBVS (b) Spherical projection IBVS

Figure 3.5: Comparison of image feature trajectories for the experimental results in pixels. The initial location is shown in red, and the goal location is in green.

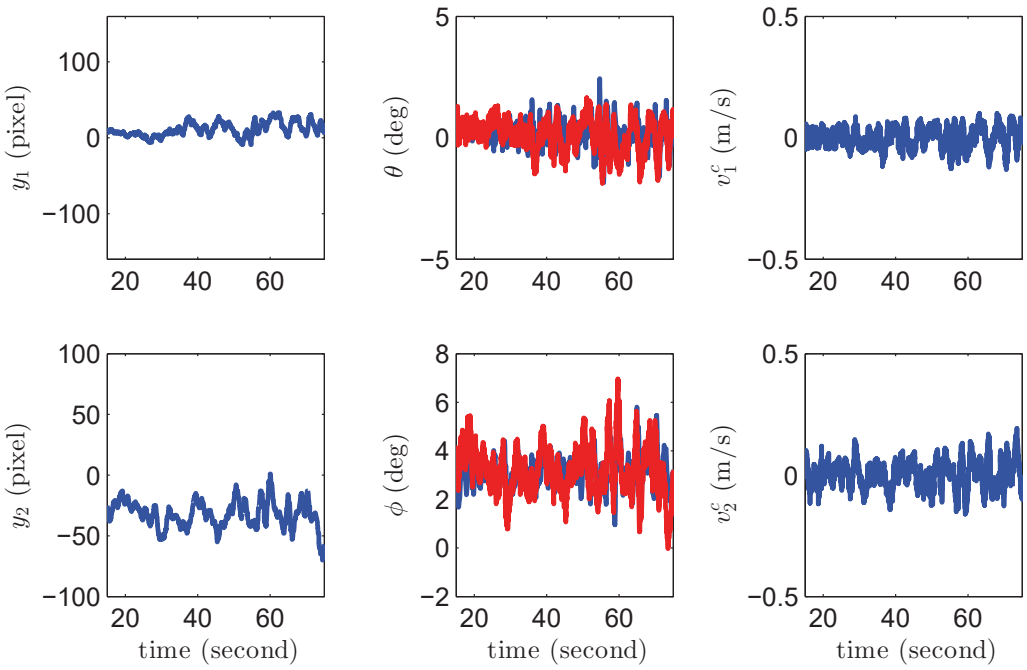


Figure 3.6: Experimental results of state trajectories for the proposed DIBVS (solid blue). Reference angles to the inner loop are shown in red.

the n_1 and n_2 directions, respectively. Roll and pitch angles have relatively small variation and exhibit non-zero average values due to non-ideal weight distribution in the vehicle. Such a weight imbalance means zero roll and pitch create large lateral velocity and causes the image feature to rapidly leave the camera's FoV. The experimental results demonstrate robustness to unmodelled dynamics neglected in the design.

To further evaluate the robustness of the proposed controller, the visual target was attached to a remote control car that was driven around the lab. Vicon markers were placed on the car to establish ground truth data. The experimental results for tracking a moving target are shown in Figure 3.7, where the quadrotor trajectory is in red, the target trajectory is in blue, and the green dots represent the time when IBVS is switched on. As seen in Figure 3.7 the tracking error is similar in magnitude to the static target experiment.

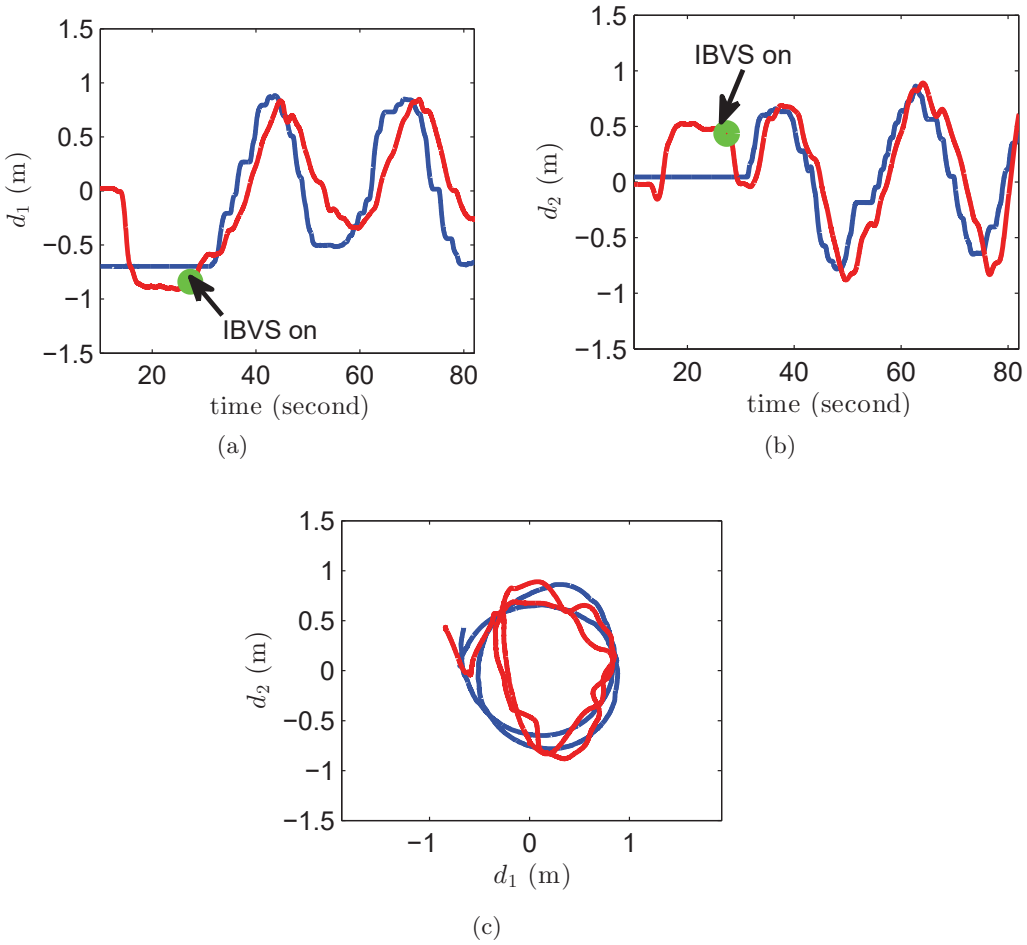


Figure 3.7: Experimental results for a moving target. The quadrotor position trajectory is in red, the target position trajectory is in blue. The green dots represent the time when IBVS is switched on.

As in Section 3.3.2 the spherical projection IBVS in [33] was implemented. As it is difficult to compare controller performance on different experimental platforms, implementing this control on the ANCL platform allows for a fair comparison with the proposed method. The experimental results of (3.24) are shown in Figure 3.8, where system states are shown in blue and the reference angles for the inner PID loop are in red. The performance is similar to the proposed controller. Figure 3.9

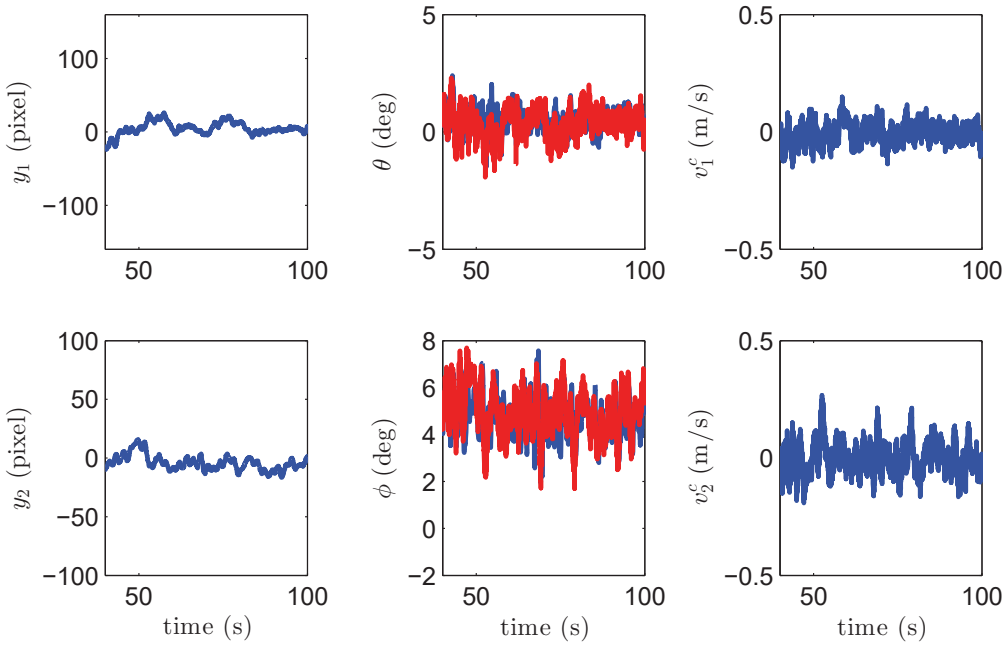
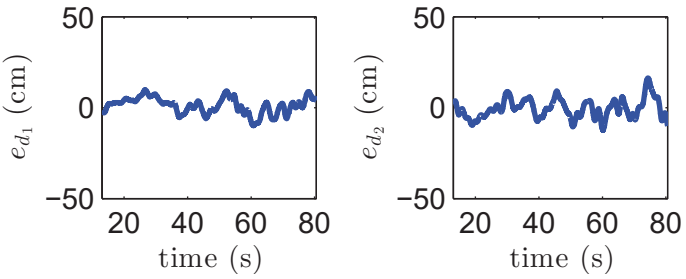


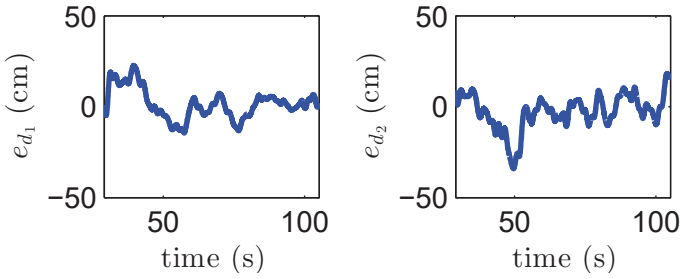
Figure 3.8: Experimental results of state trajectories (solid blue) for the spherical projection IBVS. Reference angles to the inner loop are shown in red.

shows that in both experiments the vehicle remained within a radius of about 30 cm from the origin. The lateral errors, denoted e_{d_1}, e_{d_2}, for the proposed control were 4 cm and 5 cm for the n_1 and n_2 axis, respectively.

For (3.24) the standard deviation of the lateral errors were 8 cm and 9 cm for the n_1 and n_2 axis, respectively. Other experimental quadrotor results such as [87] show similar results with a non-vision-based position controller and without compensation for advanced aerodynamic effects such as blade flapping and drag force. Their position control which compensates for these effects improved hovering performance from 40 cm to 10 cm radial error. On the other hand, the proposed controller does not rely on position estimates and hovers in a radius of about 30 cm. Another experimentally validated non-vision-based position controller is presented in [88]. This control provides hover performance in a circle of about 20 cm radius. Few experimental results for IBVS for UAVs have been published. Even fewer results exist for DIBVS laws. An exception is the work in [34] which presents a comparison



(a) Proposed DIBVS



(b) Spherical projection IBVS

Figure 3.9: Experimental results for lateral position error.

of a number of IBVS algorithms including that in Guenard et al. [33]. Some of these controllers are DIBVS and have hover performance similar to the proposed controller. However, they have increased complexity due to the spherical projection used.

3.4 Conclusion

This chapter has presented a DIBVS control with a cascade two-loop structure: an outer IBVS loop calculates reference attitude and thrust, and an inner loop tracks the given reference attitude. The outer loop of the DIBVS model contains both the input and its derivative and this complicates the controller design. Hence, a scheme using state transformation is presented to lower the order of the input derivative. The existence of the state transformation is proven and can be computed by solving a system of first-order homogeneous PDEs. The solutions of the PDEs for two basic targets, i.e., points and lines, are given explicitly. The state transformation-based IBVS is applied to control the lateral position of the the quadrotor. The design is based on two decoupled PVTOL aircraft models with constant altitude. Simulations and experiments are conducted to validate the control. A popular existing spherical projection IBVS law in [33] is compared to the proposed approach. Simulation and experimental results show similar performance of the proposed approach to the

one in [33]. However, the proposed control has the advantage of being simpler to implement.

Chapter 4

PID DIBVS using a Virtual Camera

Chapter 3 presented a state transformation-based DIBVS to control the 2D lateral motion of a quadrotor. The present chapter chooses a particular solution for the state transformation which corresponds to a virtual camera and extends the DIBVS to control 3D translational and yaw motion. In order to control the full 4 DoF motion of the vehicle with visual servoing more features are required. In Section 4.1 image features based on coplanar points and parallel lines are described and image feature kinematics are given. Section 4.2 describes important uncertainty in the DIBVS model which is compensated in the PID DIBVS proposed later in the chapter. Based on the experimental results in Figure 3.5 the image feature error converges to a non-zero constant. After a thorough investigation of the source of this error, it was concluded that roll and pitch estimate bias in the AHRS output was the cause. Even small bias (i.e., 1°) leads to significant steady state error. This is because the bias angle enters the outer-loop as an uncompensated disturbance. The bias in the roll and pitch also affects the accuracy of the state transformation proposed in Chapter 3. The analysis of the error in the state transformation due to this bias is provided. During the modelling process it is observed that the thrust gain K_T decreases slowly with battery voltage. This variation has significant influence on the vehicle's altitude and should be compensated. Along with the uncertainty in K_T, variation in the mass and image feature depth is considered. The proposed PID DIBVS is presented in Section 4.3, and a proof of closed-loop exponential stability is given. Lastly, Section 4.4 presents the experimental work to validate the proposed design.

4.1 Feature Definition

In this section image features for planar targets are presented in transformed state coordinates. The particular solutions (3.12), (3.13), and (3.16) of the PDEs (3.8), (3.10), and (3.15), respectively, are used. As discussed in Chapter 3, equations (3.12) and (3.16) transform the image features of a single point or single line to image features in a virtual camera with fixed roll and pitch. In the non-horizontal case and if the orientation of the plane in $\mathcal{N}$ is known, R_c can be taken as the product of R_{n_3} and the rotation matrix representing the orientation of the plane in $\mathcal{N}$. The kinematics of the new states for both point image coordinates p^a and line image parametrization l^a is independent of yaw motion. For a set of 3D points or parallel lines located in the target plane, useful features for visual servoing are the moment features given in [62]. For example, point image moments in [62] can be used for regulating the translational motion of the vehicle. When the target plane is not horizontal, moment features to regulate yaw motion of the vehicle is an open problem. In this thesis the focus is on horizontal target planes with normal vector $[0, 0, 1]^T$. In this case the feature to control yaw motion is available.

Using the state transformation-based approach in Chapter 3 with a horizontal target, R_c is taken as an identity matrix in (3.12), (3.13), (3.17). Figure 4.1 shows the image plane for the virtual camera, which is referred to as the virtual image plane. This plane remains parallel to the n_1-n_2 plane, i.e., the horizontal plane. To simplify the presentation of the controller design, a virtual camera frame $\mathcal{A}$ is

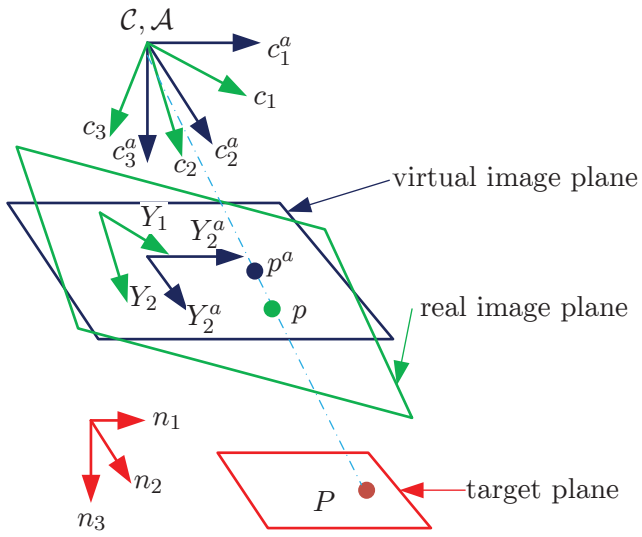


Figure 4.1: Virtual camera with zero roll and pitch angles

introduced. The origin of $\mathcal{A}$ is fixed to the origin of $\mathcal{C}$ and has the basis $\{c_1^a, c_2^a, c_3^a\}$. Since the frame $\mathcal{A}$ has zero roll and pitch, this means c_3^a is parallel to n_3. As with the image plane for the real camera, a 2D image frame $\mathcal{Y}$ is introduced for the virtual

image plane. The basis for this frame is $\{Y_1^a, Y_2^a\}$. The vector Y_k^a is parallel to c_k^a for $k = 1, 2$. The coordinates $P^c = [X_1, X_2, X_3]^T$ of a 3D point in $\mathcal{C}$ transform to $P^a = [X_1^a, X_2^a, X_3^a]^T$ in $\mathcal{A}$ with

$$P^a = R_{n_2}(\theta) R_{n_1}(\phi) P^c \tag{4.1}$$

The projection of P^a onto the virtual camera plane is p^a with coordinates

$$p^a = \begin{bmatrix} y_1^a \\ y_2^a \end{bmatrix} = \lambda \begin{bmatrix} \frac{X_1^a}{X_3^a} \\ \frac{X_2^a}{X_3^a} \end{bmatrix}$$

The parametrization of the image of a 3D line in the virtual image plane is denoted as $l^a = [\alpha^a, \rho^a]$. The following two sections present the image features for planar targets consists of only points or only parallel lines.

4.1.1 Point Moment Feature Definition and Kinematics

Assumption 4.1. *The target plane is parallel to the n_1-n_2 plane and contains $N > 1$ 3D points.*

Under Assumption 4.1, $R_c = I$ and kinematics (3.14) becomes

$$\begin{bmatrix} \dot{y}_1^a \\ \dot{y}_2^a \end{bmatrix} = L_{vp}^a v^a - S \begin{bmatrix} y_1^a \\ y_2^a \end{bmatrix} \dot{\psi} \tag{4.2a}$$

$$\dot{v}^a = \left[\dot{\psi} E_3\right]_\times v^a + F^a \tag{4.2b}$$

where

$$S = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \; L_{vp}^a = \begin{bmatrix} -\frac{\lambda}{X_3^a} & 0 & \frac{y_1^a}{X_3^a} \\ 0 & -\frac{\lambda}{X_3^a} & \frac{y_2^a}{X_3^a} \end{bmatrix}$$

and $F^a = R_{\theta\phi} g_1(F^c, \eta)$, $X_3^a = E_3^T R_{\theta\phi} P$. Hence, the kinematics of the kth point is

$$\dot{y}_{1k}^a = \frac{1}{X_{3k}^a} \begin{bmatrix} -\lambda & 0 & y_{1k}^a \end{bmatrix} v^a + y_{2k}^a \dot{\psi} \tag{4.3a}$$

$$\dot{y}_{2k}^a = \frac{1}{X_{3k}^a} \begin{bmatrix} 0 & -\lambda & y_{2k}^a \end{bmatrix} v^a - y_{1k}^a \dot{\psi} \tag{4.3b}$$

where y_{1k}^a and y_{2k}^a are the transformed coordinates of the kth image point, i.e., the image coordinates of the kth point in the virtual image plane. The depth of the kth point in $\mathcal{A}$ is denoted X_{3k}^a. From Assumption 4.1, all points in the target plane share the same depth which is denoted X_3^a. The coordinates of the centroid for the

set of points are denoted y_{1g} and y_{2g} and given by

$$y_{1g} = \frac{1}{N}\sum_{k=1}^{N} y_{1k}^{a},\ y_{2g} = \frac{1}{N}\sum_{k=1}^{N} y_{2k}^{a}$$

The kinematics of y_{1g} and y_{2g} is

$$\dot{y}_{1g} = \frac{1}{X_3^a}\begin{bmatrix}-\lambda & 0 & y_{1g}\end{bmatrix} v^a + y_{2g}\dot{\psi} \tag{4.4a}$$

$$\dot{y}_{2g} = \frac{1}{X_3^a}\begin{bmatrix}0 & -\lambda & y_{2g}\end{bmatrix} v^a - y_{1g}\dot{\psi} \tag{4.4b}$$

The centred moments are defined as

$$\mu_{ij} = \sum_{k=1}^{N} \left(y_{1k}^{a} - y_{1g}\right)^{i} \left(y_{2k}^{a} - y_{2g}\right)^{j}, \quad i, j \geq 0$$

Using (4.3) and (4.4), the kinematics of μ_{20}, μ_{02}, and μ_{11} is

$$\dot{\mu}_{20} = \frac{2}{X_3^a}\mu_{20} v_3^a + 2\mu_{11}\dot{\psi} \tag{4.5a}$$

$$\dot{\mu}_{02} = \frac{2}{X_3^a}\mu_{02} v_3^a - 2\mu_{11}\dot{\psi} \tag{4.5b}$$

$$\dot{\mu}_{11} = \frac{2}{X_3^a}\mu_{11} v_3^a - \left(\mu_{20} - \mu_{02}\right)\dot{\psi} \tag{4.5c}$$

Defining $a = \mu_{20} + \mu_{02}$ and using (4.5) gives

$$\dot{a} = \frac{2}{X_3^a} a v_3^a \tag{4.6}$$

Using $\dot{X}_3^a = -v_3^a$ and (4.6), gives

$$\frac{d}{dt}\left(X_3^a \sqrt{a}\right) = 0$$

Hence, $X_3^a\sqrt{a}$ is a constant value and

$$X_3^a \sqrt{a} = X_3^{a*}\sqrt{a^*} \tag{4.7}$$

where X_3^{a*} and a^* are desired constant values of X_3^a and a, respectively. The moment feature for height is defined as

$$s_{hp} = \sqrt{\frac{a^*}{a}} \tag{4.8}$$

and using (4.6), (4.7), the kinematics of s_{hp} is

$$\dot{s}_{hp} = -\frac{1}{2}\sqrt{a^*}\frac{\dot{a}}{a\sqrt{a}} = -\frac{1}{X_3^a}\sqrt{a^*}\frac{v_3^a}{\sqrt{a}} = -\frac{v_3^a}{X_3^{a*}} \tag{4.9}$$

Kinematics (4.9) depends only on v_3^a and this simplifies controller design. Next, the feature for controlling lateral motion is defined as

$$s_{lp} = [y_{1g}, y_{2g}]^T s_{hp} \tag{4.10}$$

Using (4.4), (4.7), (4.9), the kinematics of s_{lp} is

$$\begin{aligned}
\dot{s}_{lp} &= \begin{bmatrix} \dot{y}_{1g} \\ \dot{y}_{2g} \end{bmatrix} s_{hp} + \begin{bmatrix} y_{1g} \\ y_{2g} \end{bmatrix} \dot{s}_{hp} \\
&= \frac{s_{hp}}{X_3^a}\begin{bmatrix} -\lambda & 0 & y_{1g} \\ 0 & -\lambda & y_{2g} \end{bmatrix} v^a - S s_{hp}\begin{bmatrix} y_{1g} \\ y_{2g} \end{bmatrix} - \frac{v_3^a}{X_3^*}\begin{bmatrix} y_{1g} \\ y_{2g} \end{bmatrix} \\
&= \frac{\sqrt{a^*}}{\sqrt{a}X_3^a}\begin{bmatrix} -\lambda & 0 & y_{1g} \\ 0 & -\lambda & y_{2g} \end{bmatrix} v^a - S s_{hp}\begin{bmatrix} y_{1g} \\ y_{2g} \end{bmatrix} - \frac{v_3^a}{X_3^*}\begin{bmatrix} y_{1g} \\ y_{2g} \end{bmatrix} \\
&= \frac{1}{X_3^{a*}}\begin{bmatrix} -\lambda & 0 & y_{1g} \\ 0 & -\lambda & y_{2g} \end{bmatrix} v^a - S s_{hp}\begin{bmatrix} y_{1g} \\ y_{2g} \end{bmatrix} - \frac{v_3^a}{X_3^*}\begin{bmatrix} y_{1g} \\ y_{2g} \end{bmatrix} \\
&= -\frac{\lambda}{X_3^{a*}}\begin{bmatrix} v_1^a \\ v_2^a \end{bmatrix} - S s_{lp}\dot{\psi}
\end{aligned}$$

The feature for regulating yaw motion is taken as

$$s_{\psi p} = \frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \tag{4.11}$$

With (4.5) the kinematics of $s_{\psi p}$ is

$$\begin{aligned}
\dot{s}_{\psi p} &= \frac{1}{2}\frac{2\dot{\mu}_{11}(\mu_{20} - \mu_{02}) - 2\mu_{11}(\dot{\mu}_{20} - \dot{\mu}_{02})}{(\mu_{20} - \mu_{02})^2}\frac{1}{1 + \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)^2} \\
&= \frac{\dot{\mu}_{11}(\mu_{20} - \mu_{02}) - \mu_{11}(\dot{\mu}_{20} - \dot{\mu}_{02})}{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \\
&= \frac{\left(\frac{2}{X_3^a}\mu_{11}v_3^a - (\mu_{20} - \mu_{02})\dot{\psi}\right)(\mu_{20} - \mu_{02})}{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \\
&\quad - \frac{\mu_{11}\left(\frac{2}{X_3^a}\mu_{20}v_3^a + 2\mu_{11}\dot{\psi} - \frac{2}{X_3^a}\mu_{02}v_3^a + 2\mu_{11}\dot{\psi}\right)}{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \\
&= -\dot{\psi}
\end{aligned}$$

The kinematics of features (4.8), (4.10), and (4.11) is summarized as

$$\dot{s}_{lp} = -\frac{\lambda}{X_3^{a*}} \begin{bmatrix} v_1^a \\ v_2^a \end{bmatrix} - S s_{lp} \dot{\psi} \tag{4.12a}$$

$$\dot{s}_{hp} = -\frac{v_3^a}{X_3^{a*}} \tag{4.12b}$$

$$\dot{s}_{\psi p} = -\dot{\psi} \tag{4.12c}$$

4.1.2 Line Moment Feature Definition and Kinematics

Assumption 4.2. *The target plane is parallel to the n_1-n_2 plane and contains a set of $N > 1$ parallel lines.*

Under Assumption 4.2, $R_c = I$ and the outer loop (3.18) becomes

$$\dot{l}^a = L_{vl}^a v^a + L_{\eta 2}^a \dot{\psi} \tag{4.13a}$$

$$\dot{v}^a = \left[\dot{\psi} E_3\right]_\times v^a + F^a \tag{4.13b}$$

where

$$F^a = R_{\theta\phi} g_1\left(F^c, \eta\right), \; L_{vl}^a = -\frac{1}{X_3^a} \begin{bmatrix} 0 & 0 & 0 \\ \lambda s_{\alpha^a} & \lambda c_{\alpha^a} & -\rho^a \end{bmatrix}, \; L_{\eta_2}^a = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and X_3^a is the depth of any point in the line expressed in $\mathcal{A}$. Hence, the kinematics of the kth line parametrization becomes

$$\dot{\alpha}_k^a = \dot{\psi} \tag{4.14a}$$

$$\dot{\rho}_k^a = -\frac{1}{X_{3k}^a} \begin{bmatrix} \lambda s_{\alpha_k^a} & \lambda c_{\alpha_k^a} & -\rho^a \end{bmatrix} v^a \tag{4.14b}$$

From Assumption 4.2 it is straightforward to show that all α_k^a are equal to the same value which is denoted α^a. It is also true that X_{3k}^a has the same value, which is denoted as X_3^a, for all lines. Inspired by the image moment based work in [63], a moment-like line features for IBVS is proposed. The line centroid ρ_g^a is defined as

$$\rho_g^a = \frac{1}{N} \sum_{k=1}^{N} \rho_k^a$$

Using (4.14) gives

$$\dot{\rho}_g^a = -\frac{1}{X_3^a} \begin{bmatrix} \lambda s_{\alpha^a} & \lambda c_{\alpha^a} & -\rho_g^a \end{bmatrix} v^a \tag{4.15}$$

Next, the central moment μ_{20}^a is defined as

$$\mu_{20}^a = \sum_{k=1}^{N} \left(\rho_k^a - \rho_g^a\right)^2 \tag{4.16}$$

Differentiating μ_{20}^a and using (4.14) and(4.15) gives

$$\dot{\mu}_{20}^a = 2\sum_{k=1}^{N}\left(\rho_k^a - \rho_g^a\right)\left(\dot{\rho}_k^a - \dot{\rho}_g^a\right) = \frac{2\mu_{20}^a}{X_3^a}v_3^a \tag{4.17}$$

Using (4.17) it can be shown that

$$\frac{d}{dt}(X_3^a\sqrt{\mu_{20}}) = 0$$

Hence, $X_3^a\sqrt{\mu_{20}}$ is a constant and

$$X_3^a\sqrt{\mu_{20}} = X_3^{a*}\sqrt{\mu_{20}^{a*}} \tag{4.18}$$

where X_3^{a*} and μ_{20}^{a*} are desired constant value of X_3^a and μ_{20}, respectively. Defining

$$s_{hl} = \sqrt{\frac{\mu_{20}^{a*}}{\mu_{20}^a}} \tag{4.19}$$

and differentiating gives

$$\begin{aligned}\dot{s}_{hl} &= -\frac{1}{2}\sqrt{\mu_{20}^{a*}}\left(\mu_{20}^a\right)^{-\frac{3}{2}}\dot{\mu}_{20}^a \\ &= -\sqrt{\mu_{20}^{a*}}\frac{\left(\mu_{20}^a\right)^{-\frac{1}{2}}}{X_3^a}v_3^a = -\frac{1}{X_3^{a*}}\frac{X_3^{a*}\sqrt{\mu_{20}^{a*}}}{X_3^a\sqrt{\mu_{20}^a}}v_3^a = -\frac{1}{X_3^{v*}}v_3^a\end{aligned}$$

The moment features for the set of parallel lines are selected as

$$s_{ll} = \rho_g s_{hl} \tag{4.20}$$

$$s_{\psi l} = \sum_{k=1}^{N}\alpha_k^a \tag{4.21}$$

and s_{hl}. The kinematics of these features is

$$\dot{s}_{ll} = -\frac{\lambda}{X_3^{a*}}\begin{bmatrix}s_\alpha^a & c_\alpha^a\end{bmatrix}\begin{bmatrix}v_1^a \\ v_2^a\end{bmatrix} \tag{4.22a}$$

$$\dot{s}_{hl} = -\frac{v_3^a}{X_3^{a*}} \tag{4.22b}$$

$$\dot{s}_{\psi l} = \dot{\psi} \tag{4.22c}$$

4.2 Uncertainty in the DIBVS Model

Before introducing the visual servoing law, uncertainty in the DIBVS model is considered. During model identification experiments a slow decrease in thrust gain

K_T was observed as the battery voltage dropped during flight. This is shown in Figure 4.2 for a single Turnigy 1100 KV motor attached to an 11×4.5" propeller. In practice the effect of decreased K_T is significant as the vehicle looses altitude

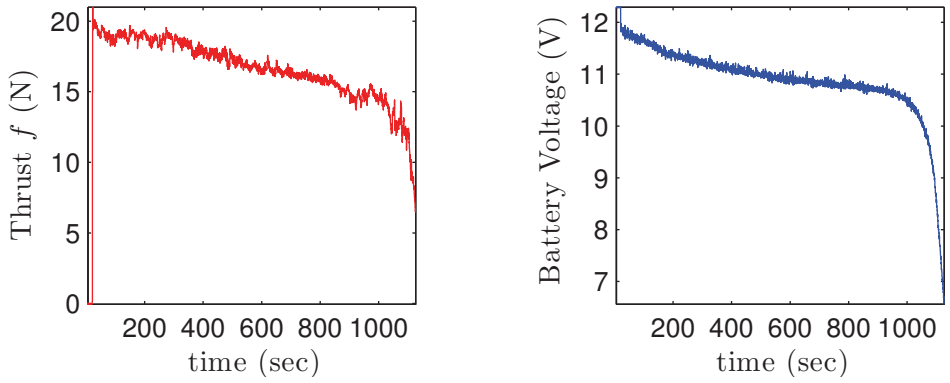


Figure 4.2: Decrease in thrust and battery voltage. Decrease in thrust corresponds to a decrease in K_T. A battery is drained with PWM pulse width set to 1.4 ms.

if no compensation is performed. To account for this variation, the parameter K_T is taken as an unknown constant. Accounting for variation in K_T is equivalent to assuming mass m is uncertain as these parameters enter as a ratio in the dynamics (2.11a).

To account for non-zero steady image feature error which results from angle estimate bias, the expression for $R_{\theta\phi}$ is taken as

$$R_{\theta\phi} = R_{n_1}^T(\phi_e) R_{n_2}^T(\theta_e) R_{n_2}(\theta_m) R_{n_1}(\phi_m)$$

where bias errors ϕ_e, θ_e are treated as constants, and ϕ_m, θ_m denote measured values. From Figure 3.6, the reference ϕ and θ must be limited to a small range to avoid the target leaving the camera's FoV. Given the small range of ϕ and θ, a small angle approximation is taken in the force model (3.14b) of F^a:

$$\begin{aligned} F^a &= R_{n_1}^T(\phi_e) R_{n_2}^T(\theta_e) R_{n_2}(\theta_m) R_{n_1}(\phi_m) \left(g R^T E_3 + \frac{F^c}{m} \right) \\ &\approx \frac{K_T u}{m} \begin{bmatrix} -\theta_m + \theta_e \\ \phi_m - \phi_e \\ -1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \end{aligned} \tag{4.23}$$

where $u = \sum_{i=1}^4 \tilde{W}_i^2$, and $\tilde{W}_i$ is defined in (2.12). It can be seen from (4.23) that the bias errors ϕ_e, θ_e enter as additive input disturbances to the outer loop. This disturbance causes the image feature errors to converge to non-zero constants in steady state.

The bias in roll and pitch also leads to an error in the state transformation ap-

proach described in Chapter 3. The relation between bias and image feature transformation error is derived. Define a 3D point expressed in $\mathcal{C}$ as $P^c = [X_1, X_2, X_3]^T$. Using (4.1) P^c can be expressed in the ideal virtual camera coordinates

$$P^{a*} = R_{n_1}^T(\phi_e)R_{n_2}^T(\theta_e)R_{n_2}(\theta_m)R_{n_1}(\phi_m)P^c \tag{4.24}$$

and in real virtual camera plane coordinates

$$P^a = R_{n_2}(\theta_m)R_{n_1}(\phi_m)P^c \tag{4.25}$$

Using (4.24) and (4.25) gives

$$P^a = R_{n_2}(\theta_e)R_{n_1}(\phi_e)P^{a*}$$

Assuming small bias implies

$$R_{n_2}(\theta_e)R_{n_1}(\phi_e) = \begin{bmatrix} c_{\theta_e} & s_{\theta_e}s_{\phi_e} & s_{\theta_e}c_{\phi_e} \\ 0 & c_{\phi_e} & -s_{\phi_e} \\ -s_{\theta_e} & c_{\theta_e}s_{\phi_e} & c_{\theta_e}c_{\phi_e} \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & \theta_e \\ 0 & 1 & -\phi_e \\ -\theta_e & \phi_e & 1 \end{bmatrix}$$

Defining $P^{a*} = [X_1^{a*}, X_2^{a*}, X_3^{a*}]^T$, the dependence of P^a on the coordinates of P^{a*} and bias is

$$P^a \approx \begin{bmatrix} 1 & 0 & \theta_e \\ 0 & 1 & -\phi_e \\ -\theta_e & \phi_e & 1 \end{bmatrix}\begin{bmatrix} X_1^{a*} \\ X_2^{a*} \\ X_3^{a*} \end{bmatrix} = \begin{bmatrix} X_1^{a*} + \theta_e X_3^{a*} \\ X_2^{a*} - \phi_e X_3^{a*} \\ -\theta_e X_1^{a*} + \phi_e X_2^{a*} + X_3^{a*} \end{bmatrix}$$

Since ϕ_e and θ_e are small and in practise $X_1^{a*} \ll X_3^{a*}$, $X_2^{a*} \ll X_3^{a*}$, the image coordinates of the point P in real camera are approximated by

$$\begin{bmatrix} y_1^a \\ y_2^a \end{bmatrix} \approx \lambda \begin{bmatrix} \frac{X_1^{a*}+\theta_e X_3^{a*}}{-\theta_e X_1^{a*}+\phi_e X_2^{a*}+X_3^{a*}} \\ \frac{X_2^{a*}-\phi_e X_3^{a*}}{-\theta_e X_1^{a*}+\phi_e X_2^{a*}+X_3^{a*}} \end{bmatrix} \approx \lambda \begin{bmatrix} \frac{X_1^{a*}+\theta_e X_3^{a*}}{X_3^{a*}} \\ \frac{X_2^{a*}-\phi_e X_3^{a*}}{X_3^{a*}} \end{bmatrix} = \begin{bmatrix} y_1^{a*} + \lambda\theta_e \\ y_2^{a*} - \lambda\phi_e \end{bmatrix}$$

Hence, $y_{1g} = y_{1g}^* + \lambda\theta_e$, $y_{2g} = p_{2g}^* - \lambda\phi_e$, $s_{lp} = s_{lp}^* + \lambda \begin{bmatrix} \theta_e \\ -\phi_e \end{bmatrix}$, $s_{hp} = s_{hp}^*$, and $s_{\phi p} = s_{\phi p}^*$. This implies there is a small and constant error between the features in the real virtual camera and the ideal virtual camera under the following assumption.

Assumption 4.3. *The bias angles θ_e, ϕ_e are small and the altitude of the vehicle is large relative to its lateral displacement.*

Since only ϕ_m, θ_m are available, $R_{\theta\phi}$ in (3.12), (3.13), (3.17) is replaced with $R_{n_2}(\theta_m)R_{n_1}(\phi_m) = R_{n_2}(\theta_e)R_{n_1}(\phi_e)R_{\theta\phi}$. Hence, R_c in (3.12), (3.13), (3.17) is

$R_{n_2}(\theta_e)R_{n_1}(\phi_e)$. Under Assumption 4.3, $R_{n_2}(\theta_e)R_{n_1}(\phi_e) \approx I$ and hence the kinematics of the real and ideal virtual camera are almost the same. For this reason the controller design presented in the following section uses a bias free model for the kinematics.

4.3 Controller Design

Section 4.1 defined feature vector $s_p = [s_{lp}^T, s_{hp}, s_{\psi p}]$ for multiple coplanar points and $s_l = [s_{ll}, s_{hl}, s_{\psi l}]^T$ for multiple coplanar parallel lines. The objective of IBVS is to minimize an image feature error $e_{sp} = s_p - s_p^*$ and $e_{sl} = s_l - s_l^*$, where s_p^* and s_l^* are the desired value of s_p and s_l, respectively. In this section a PID controller for the outer-loop based on the image moment features s_p and s_l is proposed. The control design accounts for the system uncertainty described in Section 4.2. Stability of the closed-loop is proven.

4.3.1 Yaw Motion

For both the point and line case, the kinematics of the feature for yaw motion of the vehicle is (4.12c) and (4.22c). The goal is to regulate $s_{\psi p}$ and $s_{\psi l}$ to zero. Thus, the image errors are $e_{s_{\psi p}} = s_{\psi p}$ and $e_{s_{\psi l}} = s_{\psi l}$. Evidently the controller

$$\dot{\psi} = K_\psi e_{s_{\psi p}} \tag{4.26}$$

for point features and

$$\dot{\psi} = -K_\psi e_{s_{\psi l}} \tag{4.27}$$

for line features, leads to closed-loop dynamics

$$\dot{e}_{s_{\psi p}} = -K_\psi e_{s_{\psi p}} \tag{4.28}$$

or

$$\dot{e}_{s_{\psi l}} = -K_\psi e_{s_{\psi l}} \tag{4.29}$$

Evidently the closed-loop dynamics (4.28) or (4.29) are GES.

4.3.2 Height Motion

The desired values for features s_{hp} and s_{hl} are set to 1. The image errors are $e_{s_{hp}} = s_{hp} - 1$ and $e_{s_{hl}} = s_{hl} - 1$. The subsystems for height of both points and lines are same. Thus, below s_h is taken to denote either s_{hp} or s_{hl}, and the image error is denoted $e_{s_h} = s_h - 1$. According to (4.12b), (4.22b), (3.14b), and (4.23), the

open-loop height subsystem is

$$\dot{e}_{s_h} = -\frac{v_3^a}{X_3^{a*}} \tag{4.30a}$$

$$\dot{v}_3^a = g - \frac{K_T u}{m} \tag{4.30b}$$

The PID controller

$$u = -K_{hp} e_{s_h} + K_{hd} v_3 - K_{hi} \xi_h \tag{4.31}$$

is proposed, where $\xi_h = \int_0^t e_{s_h}(\varepsilon) d\varepsilon$. Defining $\bar{\xi}_h = g/K_{hi} + \xi_h$, from (4.30) and (4.31) the closed-loop is

$$\begin{bmatrix} \dot{\bar{\xi}}_h \\ \dot{e}_{s_h} \\ \dot{v}_3^a \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{X_3^{a*}} \\ \frac{K_T K_{hi}}{m} & \frac{K_T K_{hp}}{m} & -\frac{K_T K_{hd}}{m} \end{bmatrix} \begin{bmatrix} \bar{\xi}_h \\ e_{s_h} \\ v_3^a \end{bmatrix} \tag{4.32}$$

It can be shown that when $K_{hp}, K_{hd}, K_{hi} > 0$ and satisfy

$$K_{hi} < \frac{K_{hp} K_{hd} K_T}{m} \tag{4.33}$$

the origin of closed-loop system (4.32) is GES.

4.3.3 Lateral Position Motion

Without loss of generality the desired values for s_{lp} and s_{ll} are set to zero. Hence, the image errors for lateral motion are $e_{s_{lp}} = [e_{s_{lp1}}, e_{s_{lp2}}]^T = s_{lp}$ and $e_{s_{ll}} = [e_{s_{ll1}}, e_{s_{ll2}}]^T = s_{ll}$.

Point moment feature

According to (4.12a), (3.14b), and (4.23), the open-loop lateral subsystem in the transformed state coordinates is

$$\dot{e}_{s_{lp}} = -\frac{\lambda}{X_3^{a*}} v_l^a - S e_{s_{lp}} \dot{\psi} \tag{4.34a}$$

$$\dot{v}_l^a = \frac{K_T u}{m} \begin{bmatrix} -\theta_m + \theta_e \\ \phi_m - \phi_e \end{bmatrix} - S v_l^a \dot{\psi} \tag{4.34b}$$

where $v_l^a = [v_1^a, v_2^a]^T$. As above, the PID controller

$$u \begin{bmatrix} \theta_m \\ -\phi_m \end{bmatrix} = -K_{lp} e_{s_{lp}} + K_{ld} v_l^a - u K_{li} \xi_{lp} \tag{4.35}$$

is proposed, where $\xi_l = \int_0^t e_{s_{lp}}(\varepsilon)d\varepsilon$. Defining u_g as the value of thrust which compensates gravity, and $\bar{\xi}_{lp} = \xi_{lp} + \frac{1}{K_{li}}\begin{bmatrix}\theta_e \\ -\phi_e\end{bmatrix}$, the closed-loop system becomes

$$\dot{x}_{lp} = A_{cl}x_{lp} + D(t)x_{lp} \tag{4.36}$$

where $x_{lp} = [\bar{\xi}_{lp}, e_{s_{lp}}, v_l^a]^T$

$$A_{cl} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & -\frac{\lambda}{X_3^{a*}}I \\ \frac{K_T u_g K_{li}}{m}I & \frac{K_T}{K}_{lp} mI & -\frac{K_T K_{ld}}{m}I \end{bmatrix},\ D(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\dot{\psi}S & 0 \\ \frac{K_T \tilde{u} K_{li}}{m}I & 0 & -\dot{\psi}S \end{bmatrix}$$

and $\tilde{u} = u - u_g$. If the PID control parameters satisfy

$$K_{li} < \frac{K_T K_{lp} K_{ld}}{u_g m} \tag{4.37}$$

A_{cl} is Hurwitz. Because the closed-loop for height and yaw are GES, we have $\dot{\psi} \to 0$ and $\tilde{u} \to 0$. This implies $D(t)x \to 0$. Using a converse Lyapunov theorem in [86, Thm. 4.14], we can conclude that closed-loop (4.36) is GES.

Line Moment Feature

Based on (4.22a), (3.14b), and (4.23), the lateral subsystem is

$$\dot{e}_{s_{ll}} = -\lambda\frac{v_2^a}{X_3^{a*}} - \lambda\frac{s_{\alpha^a}v_1^a + (c_{\alpha^a} - 1)v_2^a}{X_3^{a*}} \tag{4.38a}$$

$$\dot{v}_2^a = \frac{K_T u(\phi_m - \phi_e)}{m} - v_1^a\dot{\psi} \tag{4.38b}$$

A PID controller is chosen as

$$u\phi_m = K_{lp}e_{s_{ll}} - K_{ld}v_2^a + uK_{li}\xi_{ll} \tag{4.39}$$

where $\xi_{ll} = \int_0^t e_{s_{ll}}(\varepsilon)d\varepsilon$. Defining $\bar{\xi}_{ll} = -\phi_e/K_{li} + \xi_{ll}$ and using (4.38), (4.39), the closed-loop becomes

$$\dot{x}_l = A_{cl}x_l + D_l(t) \tag{4.40}$$

where $x_{ll} = [\bar{\xi}_{ll}, e_{s_{ll}}, v_2^a]^T$,

$$A_{cl} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -\frac{\lambda}{X_3^{a*}} \\ \frac{K_T u_g K_{li}}{m} & \frac{K_T K_{lp}}{m} & -\frac{K_T K_{ld}}{m} \end{bmatrix},\ D_l(t) = \begin{bmatrix} 0 \\ \frac{\lambda(s_\alpha^a v_1^a + (c_\alpha^a - 1)v_2^a)}{X_3^{a*}} \\ \frac{K_T \tilde{u} K_{li}\bar{\xi}_{ll}}{m} - v_1^a\dot{\psi} \end{bmatrix}$$

$\tilde{u} = u - u_g$. Since the closed-loop for height and yaw are GES, $\alpha \to 0$, $\tilde{u} \to 0$, $\dot{\psi} \to 0$, $D_l(t) \to 0$. Using a result in [89] if the PID control gains satisfy (4.37), then (4.40) is GES.

Inner-outer Loop Stability Analysis

If the inner loop achieves perfect tracking of the reference provided by the outer loop (4.35) or (4.39), then stability has been proven above. In this section the effect of tracking error from the inner loop on the entire closed-loop stability is analyzed. Denote $e_{\eta_1} = \eta_{m_1} - \eta^*_{m_1}$, $\eta_{m_1} = [\phi_m, \theta_m]^T$, $\eta^*_{m_1} = [\phi^*_m, \theta^*_m]^T$, and $\eta_{e_1} = [\phi_e, \theta_e]^T$, the open-loop lateral subsystem (4.34) for point features becomes

$$\dot{e}_{s_{lp}} = -\frac{\lambda}{X_3^{a*}} v_l^a - S e_{s_{lp}} \dot{\psi} \tag{4.41a}$$

$$\dot{v}_l^a = \frac{K_T u}{m} S\left(\eta^*_{m_1} - \eta_{e_1}\right) - S v_l^a \dot{\psi} + \frac{K_T u}{m} S e_{\eta_1} \tag{4.41b}$$

The desired attitude is chosen as

$$u S \eta^*_{m_1} = K_{lp} e_{s_{lp}} - K_{ld} v_l^a + u K_{li} \xi_{lp} \tag{4.42}$$

From (4.41) and (4.42), the outer loop becomes

$$\dot{x}_{lp} = A_{cl} x_{lp} + B_{lp} e_{\eta_1} + D_{lp}(t) \tag{4.43}$$

where

$$B_{lp} = \begin{bmatrix} 0 \\ 0 \\ \frac{K_T u_g}{m} S \end{bmatrix},\ D_{lp} = D(t) x_{lp} + B_u(t) e_{\eta_1},\ B_u = \begin{bmatrix} 0 \\ 0 \\ \frac{K_T \tilde{u}}{m} S \end{bmatrix}$$

Since ψ and u exponentially converge to constants, the term D_{lp} exponentially converges to zero. Using (4.35) gives

$$\dot{u} S \eta^*_{m_1} + u S \dot{\eta}^*_{m_1} = K_l \dot{x}_{lp} + \dot{u} K_{li} \xi_{lp} \tag{4.44}$$

where $K_l = [u K_{li} I_2, K_{lp} I_2, -K_{ld} I_2]$. Substituting (4.36) into (4.44) yields

$$\dot{\eta}^*_{m_1} = \frac{1}{u_g} S^{-1} K_l A_{cl} x_{lp} + \frac{1}{u_g} S^{-1} K_l B_{lp} e_{\eta_1} + D^*_{\eta_{m_1}}(t) \tag{4.45}$$

where

$$D^*_{\eta_{m_1}}(t) = \frac{1}{u} S^{-1} \left(K_l D(t) x_{lp} + K_l B_u(t) e_{\eta_1} + \dot{u}\left(K_{li}\xi_{lp} - S\eta^*_{m_1}\right) + \frac{\tilde{u}}{u u_g}\left(K_l A_{cl} x_{lp} + B_{lp} e_{\eta_1}\right) \right)$$

The vector $D^*_{\eta_{m_1}}$ exponentially converges to zero since $\dot{u}, \tilde{u}, D, B_u$ exponentially converge to zero.

Motivated by work in [68] the inner loop is approximated as

$$\dot{e}_{\eta_1} = \dot{\eta}_{m_1} - \dot{\eta}^*_{m_1} \tag{4.46}$$

$$\ddot{\eta}_{m_1} = J_{12}^{-1}\tau^c_{12} \tag{4.47}$$

where $J_{12} = \text{diag}([J_1, J_2])$, and τ^c_{12} is a vector of the first two components of τ^c. Following [85], one control design is

$$\tau^c_{12} = \ddot{\eta}^*_{m_1} - K_{\eta_1 p} e_{\eta_1} - K_{\eta_1 d}\dot{e}_{\eta_1}$$

In this case it can be shown that the control gains for the inner and outer loops can be independently chosen to ensure stability. However, computing the first and second derivative of the reference angle can introduce noise and is often avoided in practice [83]. Noise is especially of concern in visual servoing where the reference angle depends on noisy image features.

Defining the following variables

$$\delta_1 = \int_0^t e_{\eta_1} dt \tag{4.48a}$$

$$\delta_2 = \frac{e_{\eta_1}}{k_1} + \delta_1 \tag{4.48b}$$

$$\delta_3 = \frac{\dot{\eta}_{m_1}}{k_2} + \delta_2 \tag{4.48c}$$

Using (4.48a) and (4.48b) leads to

$$\dot{\delta}_1 = -k_1\delta_1 + k_1\delta_2 \tag{4.49}$$

Using (4.45), (4.46), (4.48b), (4.48c), and (4.49) gives

$$\dot{\delta}_2 = -(k_1 + \frac{K_T K_{ld}}{m})\delta_1 - \left(\frac{k_2 - k_1^2}{k_1} - \frac{K_T K_{ld}}{m}\right)\delta_2 + \frac{k_2}{k_1}\delta_3 \tag{4.50}$$

Based on (4.47) (4.48c), and (4.50), we get

$$\dot{\delta}_3 = -(k_1 + \frac{K_T K_{ld}}{m})\delta_1 - \left(\frac{k_2 - k_1^2}{k_1} - \frac{K_T K_{ld}}{m}\right)\delta_2 + \frac{k_2}{k_1}\delta_3 + \frac{1}{k_2}J_{12}^{-1}\tau^c_{12} \tag{4.51}$$

From (4.49), (4.50), (4.51), and choosing the control law as

$$\tau^c_{12} = -k_3\delta_3$$

gives

$$\dot{x}_\delta = A_\delta x_\delta + \frac{1}{k_1} B_\delta x_{lp} + D_\delta(t) \tag{4.52}$$

where $x_\delta = [\delta_1^T, \delta_2^T, \delta_3^T]^T$

$$A_\eta = \begin{bmatrix} -k_1 I_2 & k_1 I_2 & 0 \\ -(k_1 + \frac{K_T K_{ld}}{m}) I_2 & -\left(\frac{k_2 - k_1^2}{k_1} - \frac{K_T K_{ld}}{m}\right) I_2 & \frac{k_2}{k_1} I_2 \\ -(k_1 + \frac{K_T K_{ld}}{m}) I_2 & -\left(\frac{k_2 - k_1^2}{k_1} - \frac{K_T K_{ld}}{m}\right) I_2 & -\frac{k_3}{k_2} J_{12}^{-1} + \frac{k_2}{k_1} I_2 \end{bmatrix}$$

$$B_\delta = \begin{bmatrix} 0 \\ -\frac{1}{u_g} S^{-1} K_l A_{cl} \\ -\frac{1}{u_g} S^{-1} K_l A_{cl} \end{bmatrix}, \; D_\delta(t) = \frac{1}{k_1} \begin{bmatrix} 0 \\ D^*_{\eta_{m_1}}(t) \\ D^*_{\eta_{m_1}}(t) \end{bmatrix}$$

and $D_\delta(t)$ exponentially converges to zero. A nominal system for closed-loop (4.43), (4.52) is defined as

$$\dot{x}_{lp} = A_{cl} x_{lp} + B_{lp} e_{\eta_1}$$
$$\dot{x}_\delta = A_\delta x_\delta + \frac{1}{k_1} B_\delta x_{lp}$$

Considering a Lyapunov function candidate

$$V = x_{lp}^T P_l x_{lp} + \frac{1}{2} x_\delta^T x_\delta$$

where $P_l > 0$, and $A_{cl}^T P_l + P_l A_{cl} = Q_l$, $Q_l > 0$, the time derivative of V is

$$\dot{V} = -x_{lp}^T Q_l x_{lp} + 2 x_{lp}^T P_l B_{lp} e_{\eta_1} + \frac{1}{2} x_\delta^T \left(A_\eta^T + A_\eta\right) x_\delta + \frac{1}{2k_1} x_{lp}^T \left(B_\delta + B_\delta^T\right) x_\delta$$

Since P_l, B_{lp}, and B_δ are independent of gains $k_i, i = 1, 2, 3$, appropriate $k_i, i = 1, 2, 3$ can be chosen so that $A_\eta^T + A_\eta < -Q_\eta$, where $Q_\eta > 0$. If the eigenvalues of Q_η are taken sufficiently large, Young's inequality can be applied to ensure $\dot{V} < 0$. Since the terms D_δ and D_{lp} are exponentially convergent, the result [89, Lemma III.1] can be applied to show the entire closed-loop is GES. The control law is

$$\tau_{12}^c = -k_3 \delta_3 = -k_3 \left(\frac{\dot{\eta}_{m_1}}{k_2} + \frac{e_{\eta_1}}{k_1} + \delta_1\right) = -K_i \delta_1 - K_p e_{\eta_1} - K_d \dot{\eta}_{m_1} \tag{4.53}$$

where $K_i = k_3$, $K_p = k_3/k_1$, and $K_d = k_3/k_2$. Hence, the inner loop control can be implemented with a PID structure.

Similarly, the inner-outer loop stability analysis using control law (4.39) and (4.53) for line moment feature s_l can be performed using the above procedure. It can be shown that if the gain of the inner attitude loop is sufficiently large, the

entire closed-loop is GES. It is remarked that the control law is independent of desired image depth X_3^{a*} and the conditions on the gains are independent of this parameter. Hence, the proposed control does not require estimation of depth as in traditional IBVS.

4.4 Experimental Results

In this section the PID DIBVS is experimentally validated for point and line features. The control laws implemented are given in (4.26), (4.27), (4.31), (4.35), (4.39), and (4.53).

4.4.1 Point Features

The proposed controller considered in this section is (4.26), (4.31), and (4.35). Two visual target points are placed on the horizontal ground and displaced by 0.44 m. The control gains for the outer loop are given in Table 4.1.

Gain	Value
K_ψ	0.6
K_{hp}	0.09
K_{hd}	0.1
K_{hi}	0.045
K_{lp}	0.08
K_{ld}	0.22
K_{li}	0.02

Table 4.1: Outer loop gains.

The inner loop PID controller for attitude has gains K_p, K_d, and K_i. The values of these gains are in Table 4.2. An example of tracking performance for roll and

	Roll	Pitch	Yaw
K_p	0.33	0.345	0.3
K_d	0.08	0.09	0.08
K_i	0.03	0.03	0.005

Table 4.2: Inner loop gains for the point and line feature experiments.

pitch is shown in Figure 4.3. To investigate the benefit of the integral component of the PID controller to compensate the uncertainty in K_T, m, and bias ϕ_e and θ_e, two cases are considered: a PD controller with K_{pi} in (4.35) and K_{hi} in (4.31) set to zero, and a PID controller with K_{pi} and K_{hi} set to values given in Table 4.1[1]. The value of a^* in (4.8) is 0.4, which is obtained from an image when the vehicle hovers at the

[1]A video of this experiment is at www.youtube.com/watch?v=k3uyswJOJNo

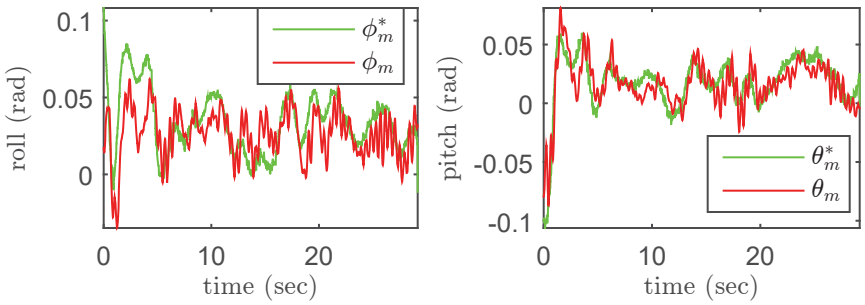


Figure 4.3: Experimental results: typical roll and pitch tracking performance.

desired height. The trajectories of $e_s = [e_{s_{lp}}^T, e_{s_{hp}}, e_{s_{\psi p}}]^T$ are shown in Figure 4.4, where Figure 4.4a and Figure 4.4b give the results with PD and PID controller, respectively. The trajectories of d_{nc}^c are shown in Figure 4.5, where the cross mark denotes initial position and circle denotes final position. Figure 4.6a and Figure 4.6b show the corresponding 3D translational position error $e_d = [e_{d_1}, e_{d_2}, e_{d_3}]^T = d_{nc}^n - d_{nc}^{n*}$ and $e_\psi = \psi - \psi^*$, where

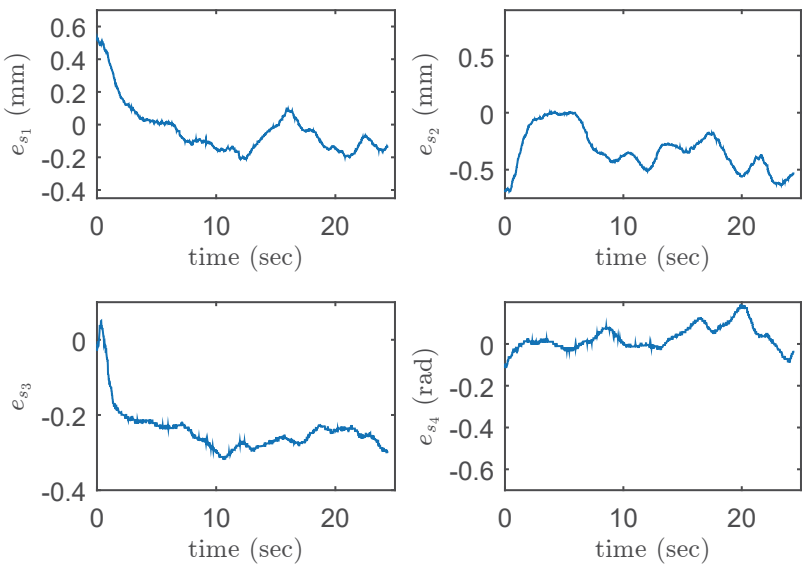
$$d_{nc}^{n*} = [0.108, -0.131, -1.393]^T \text{m}$$

and $\psi^* = -0.545$ rad correspond to the desired d_{nc}^n and ψ, respectively, when $e_{s_{lp}} = 0$, $e_{s_{hp}} = 0$, and $e_{s_{\psi p}} = 0$. It can be seen that both cases reach steady state in about 10 seconds. Table 4.3 gives the average and standard deviation of $e_{s_{lp}}$, $e_{s_{hp}}$, $e_{s_{\psi p}}$, e_d, and e_ψ after 10 seconds. It can be seen from Figure 4.4 and Table 4.3 that $e_{s_{lp}}$

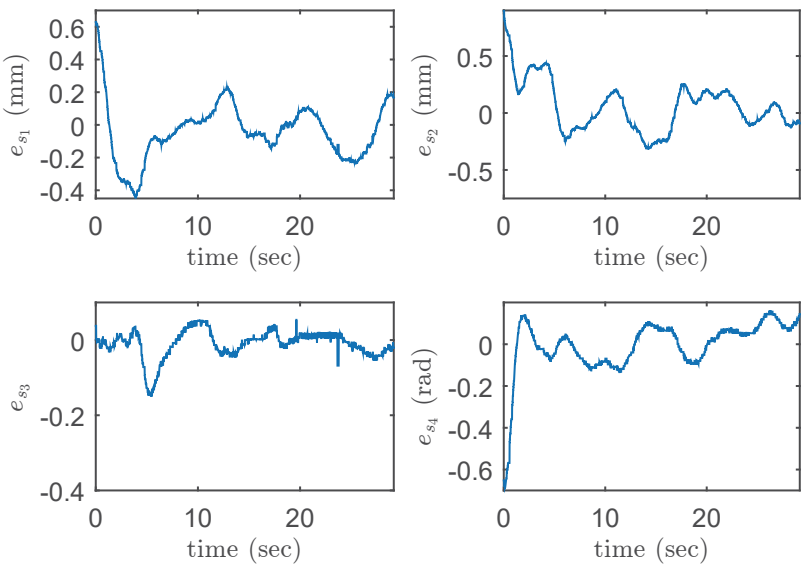
	PID controller		PD controller	
variable	μ	σ	μ	σ
$e_{s_{lp1}}$ (mm)	-0.01	0.12	-0.10	0.08
$e_{s_{lp2}}$ (mm)	0.00	0.16	-0.40	0.13
$e_{s_{hp}}$	-0.00	0.03	-0.27	0.02
$e_{s_{\psi p}}$ (rad.)	0.03	0.08	0.05	0.06
e_{d_1} (m)	0.003	0.071	0.057	0.046
e_{d_2} (m)	0.005	0.065	0.179	0.056
e_{d_3} (m)	-0.008	0.029	0.451	0.034
e_ψ (rad.)	-0.109	0.053	-0.026	0.053

Table 4.3: Statistics of the error for PID DIBVS and point features. Mean is denoted μ and standard deviation σ.

and $e_{s_{hp}}$ are significantly reduced when the integral control is added. Also, from Figure 4.4 and Table 4.3 it is evident that e_d is reduced when the integral control is added. The performance of yaw regulation for the two cases is similar since they have the same closed-loop dynamics. Standard deviations of error for the two cases are similar. Hence, the proposed PID controller effectively deals with the model

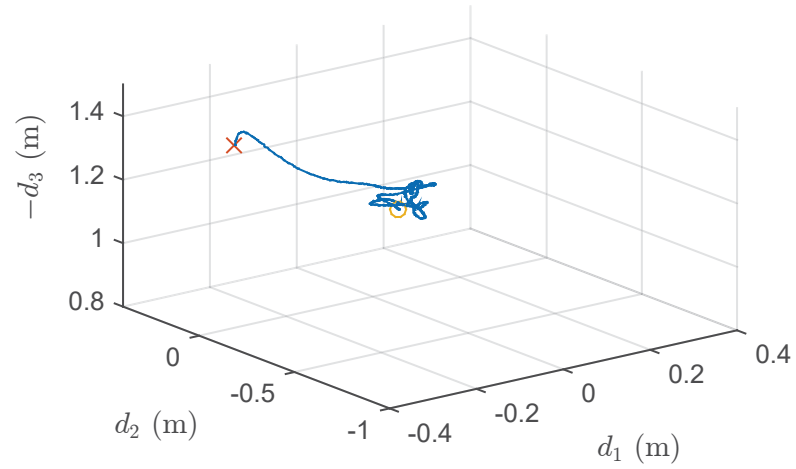


(a) PD.

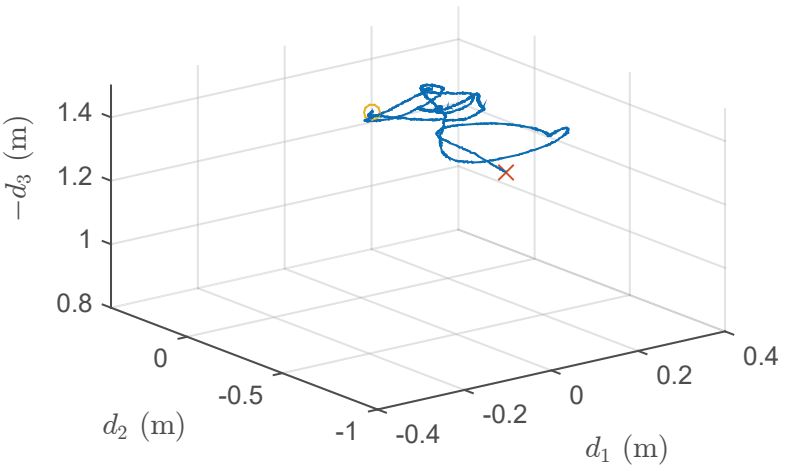


(b) PID.

Figure 4.4: Experimental results: image feature error e_s using point features.

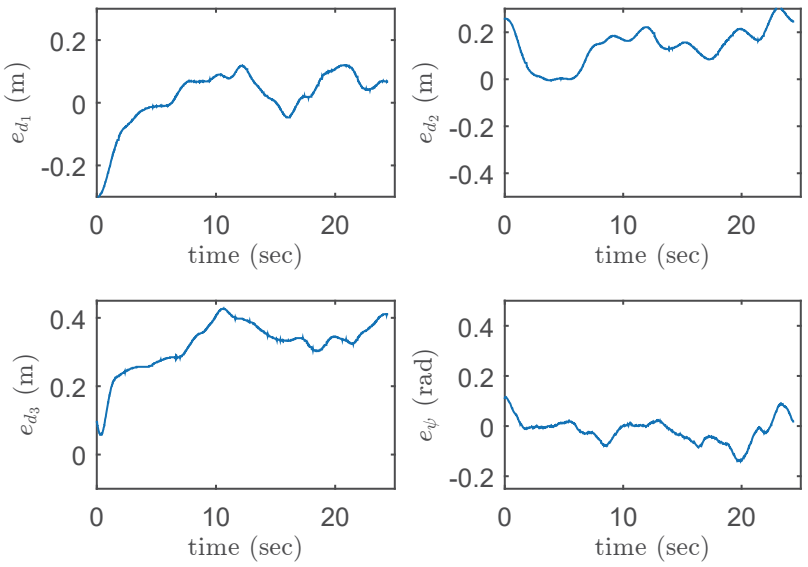


(a) PD.

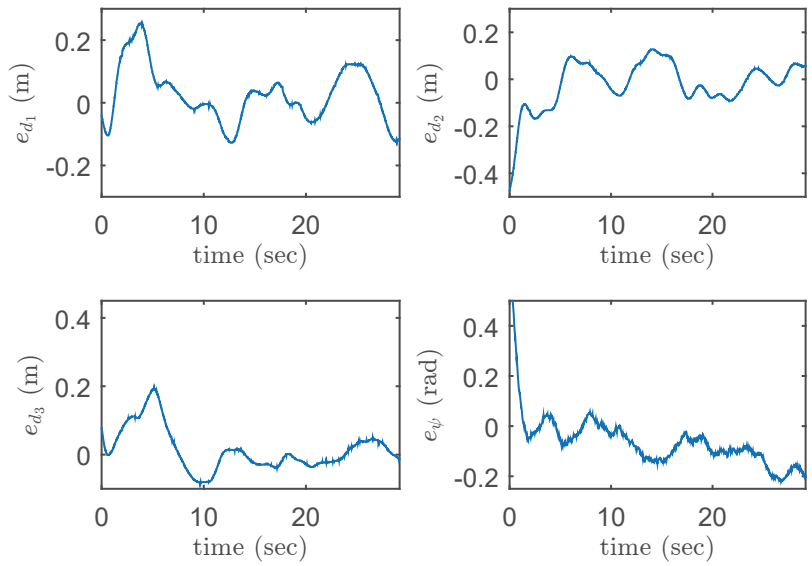


(b) PID.

Figure 4.5: Experimental results: trajectories of vehicle's 3D position d_{nc}^{c} using point features. The cross mark denotes initial position and circle denotes final position.



(a) PD.



(b) PID.

Figure 4.6: Experimental results: trajectories of vehicle's 3D position error e_d and yaw error e_ψ using point features.

uncertainties described in Section 4.2.

4.4.2 Line Features

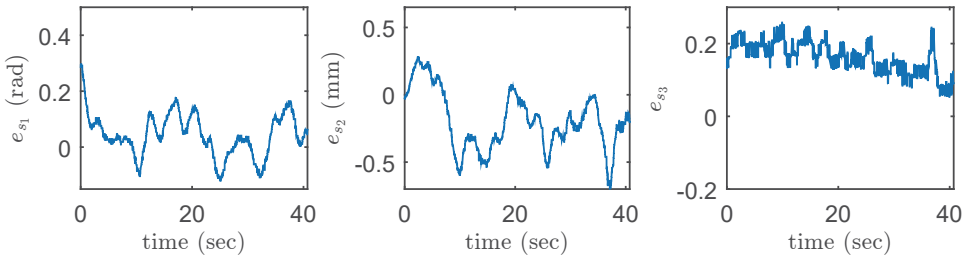
The proposed controller considered in this section is (4.29), (4.31), and (4.39). The line moment features used are (4.22). Two parallel visual target lines displaced by 0.2 m are placed on the ground. As with the point feature case, both PID and PD control are implemented to show the benefit of the integral term which compensates uncertainty[2]. The control gains for the outer loop are shown in Table 4.4. The PID control gains for the inner loop are the same as for point features and are given in Table 4.2. Hence, the inner loop attitude tracking performance is similar to that shown in Figure 4.3. Similarly, the desired value μ_{20}^{a*} in (4.18) is set to 0.082 which is

Gain	Value
K_ψ	0.4
K_{hp}	0.09
K_{hd}	0.08
K_{hi}	0.01
K_{lp}	0.08
K_{ld}	0.2
K_{li}	0.02

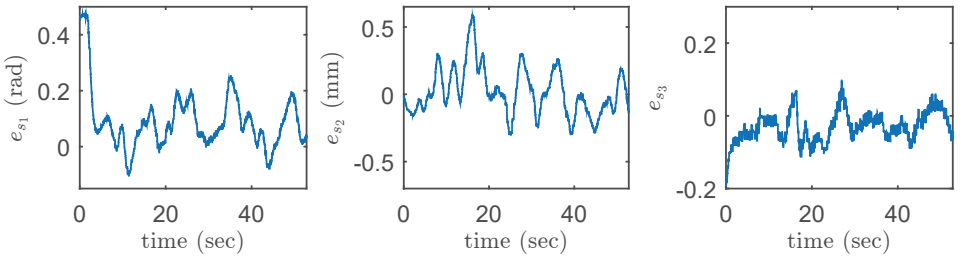
Table 4.4: Outer loop gains.

acquired from an image when the vehicle flies at the desired height. The trajectories of image feature error $e_s = [e_{s_{ll}}, e_{s_{hl}}, e_{s_{\psi l}}]^T$ are shown in Figure 4.7, where Figure 4.7a shows the performance of the PD controller and Figure 4.7b shows the performance of PID controller. The trajectories of d_{nc}^c for the two cases are shown in Figure 4.8, where the cross mark denotes initial position and the circle denotes final position. Since it is impossible to regulate the 3D position of the vehicle with a set of parallel lines, only the relevant components of 3D position errors $e_{d_2} = d_2 - d_2^*$, $e_{d_3} = d_3 - d_3^*$ and yaw error $e_\psi = \psi - \psi^*$ are shown in Figure 4.9, where $d_2^* = -0.124$ m, $d_3^* = -1.367$ m, $\psi^* = -0.753$ rad are the desired lateral position in the n_2 direction, the desired height, and desired yaw, respectively. We observe from Figure 4.7 and Figure 4.9 that the vehicle reaches its steady state after around 10 seconds. Table 4.5 gives the means and standard deviations of signals e_s, e_d, and e_ψ. It can be seen from Figure 4.7 and Table 4.5 that when the PID controller is enabled the means of image feature errors $e_{s_{ll}}$, $e_{s_{hl}}$ are significantly reduced relative to the PD controller case. The standard deviations of these signals for the two cases are similar. From Figure 4.9 and Table 4.5, the means of the corresponding 3D position errors e_{d_2}, e_{d_3} are significantly reduced. The image error $e_{s_{\psi l}}$ and the yaw error $e_\psi = \psi - \psi^*$ for the

[2]A video of the experiment is at www.youtube.com/watch?v=N9u977iVZDs



(a) PD.

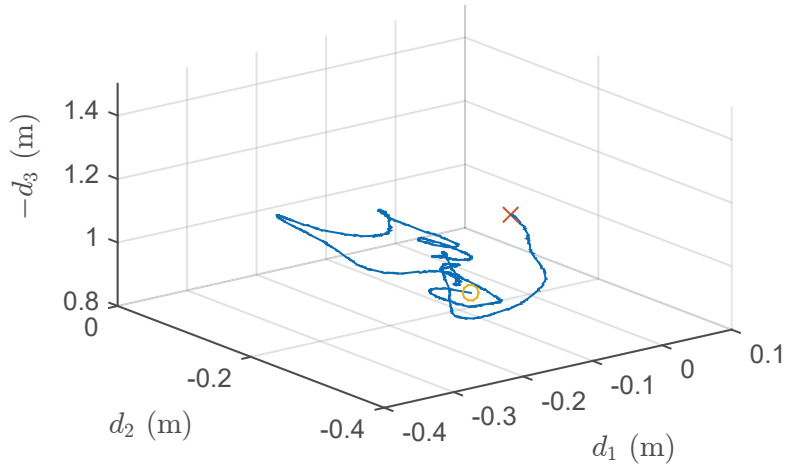


(b) PID.

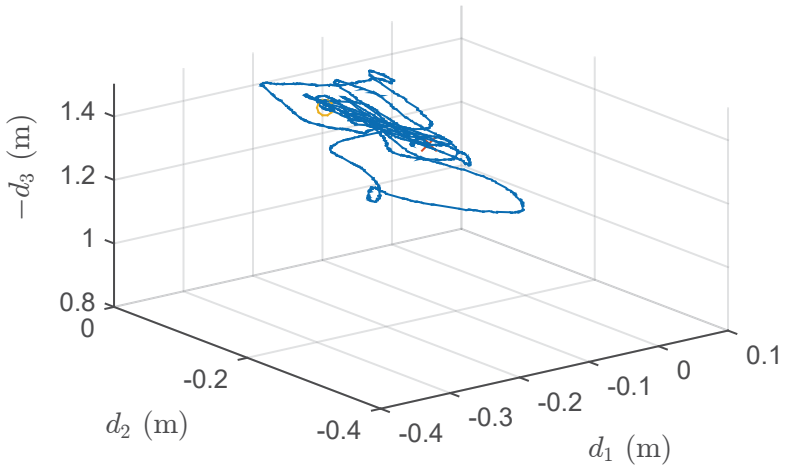
Figure 4.7: Experimental results: image feature error e_s using line features.

	PID controller		PD controller	
variable	μ	σ	μ	σ
$e_{s_{\psi l}}$ (rad.)	0.07	0.07	0.04	0.07
$e_{s_{ll}}$ (mm)	0.03	0.18	-0.26	0.17
$e_{s_{hl}}$	-0.02	0.04	0.15	0.04
e_{d_2} (m)	-0.024	0.079	0.106	0.072
e_{d_3} (m)	-0.009	0.049	-0.228	0.040
e_ψ (rad.)	0.093	0.091	0.041	0.056

Table 4.5: Statistics of the error for PID DIBVS and line features. Mean is denoted μ and standard deviation σ.

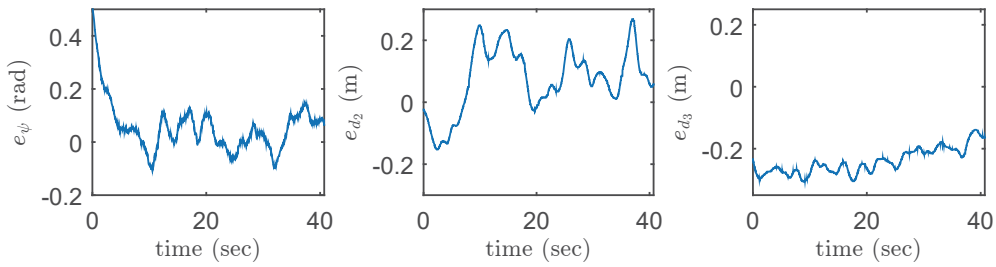


(a) PD.

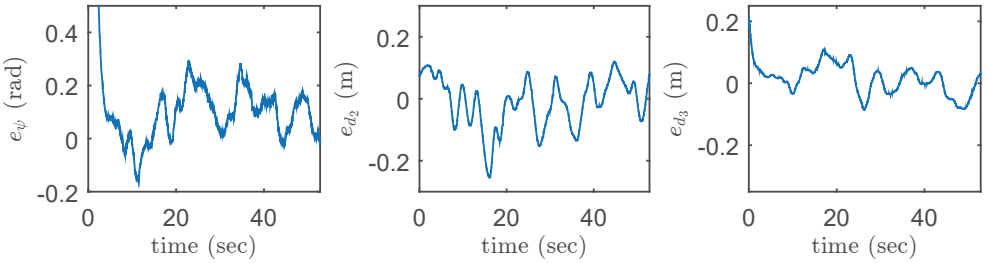


(b) PID.

Figure 4.8: Experimental results: trajectories of vehicle's 3D position d_{nc}^{c} using line features. The cross mark denotes initial position and the circle denotes final position.



(a) PD.



(b) PID.

Figure 4.9: Experimental results: trajectories of vehicle's 3D position error e_d and yaw error e_ψ using line features. Only e_{d_2}, e_{d_3} and e_ψ are shown since it is impossible to regulate the 3D position of the vehicle with a set of parallel lines.

two cases are reduced to small values starting from large initial values. The mean and standard deviation of $e_{s_{\psi l}}$ and e_ψ at steady state for the two cases are similar because both have the same closed-loop dynamics.

4.5 Conclusion

Using the virtual camera particular solution of the state transformation-based approach in Chapter 3, a set of image moment features in the virtual camera is proposed for DIBVS. The corresponding image feature kinematics are also given. Model uncertainties including thrust gain and attitude measurement bias are accounted for in the proposed PID DIBVS. An analysis of roll and pitch bias on the virtual camera transformation error is provided. In addition, the control does not require an estimate of depth, which is typically needed in classical IBVS. The ANCL quadrotor platform is used to validate the proposed control. The experimental results demonstrate improved performance with lower mean image and position error relative to PD DIBVS.

Chapter 5

Adaptive DIBVS

The PID DIBVS presented in Chapter 4 is robust to image feature depth X_3^{a*}, thrust constant K_T, vehicle mass, and angle estimate bias. To ensure outer loop stability, tuning is required to ensure control gains satisfy constraint (4.33) and (4.37). The first design in this chapter considers the application of adaptive control methods to account for system uncertainty while avoiding the need to satisfy constraints (4.33) and (4.37). Section 5.1 presents an adaptive DIBVS for planar targets consisting of points. The two loop controller structure is the same as discussed in Section 3.1. The design divides the outer loop into three subsystems: the height subsystem, the lateral motion subsystem, and the yaw motion subsystem. An adaptive backstepping method is applied to each subsystem. Simulation and experimental validations are in Section 5.1.2 and 5.1.3. Next, Section 5.2.1 presents an adaptive DIBVS for line moment features with a different cascade structure borrowed from [33]. This considers uncertainty in depth and focal length. Simulation and experimental validations of this control are in Section 5.2.3.

5.1 Adaptive DIBVS for Point Moment Features

5.1.1 Outer-loop Control

As in Chapter 4, a small angle approximation is suitable for the input force model given the small range of roll and pitch in IBVS. The approximate relation is

$$F^a \approx \frac{K_T u}{m} \begin{bmatrix} -\theta_m + \theta_e \\ \phi_m - \phi_e \\ -1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{5.1}$$

where ϕ_m, θ_m are measured roll and pitch; ϕ_e, θ_e are roll and pitch measurement bias; K_T is defined in (2.12); and u defined in (4.23). The kinematics (4.12b) and (4.22b), dynamics (3.14b), and (5.1) are combined to obtain the DIBVS height

subsystem dynamics for point moment features:

$$\dot{s}_{hp} = -\frac{1}{X_3^{a*}} v_3^a \tag{5.2a}$$

$$\dot{v}_3^a = g - \frac{K_T u}{m} \tag{5.2b}$$

where s_{hp} is defined in (4.8). In (5.2) the control input is taken as

$$u = \left(K_{h2} + \hat{C}_z K_{h1}^2\right) \delta_{hp2} + \hat{C}_g \tag{5.3}$$

with

$$\dot{\hat{C}}_g = K_{hg} \delta_{hp2} \tag{5.4}$$

where

$$\delta_{hp1} = s_{hp} - 1 \tag{5.5}$$

$$\delta_{hp2} = \frac{v_3^a}{K_{h1}} - \delta_{h1} \tag{5.6}$$

K_{h1}, K_{h2}, K_{hg} are positive control gains, $\hat{C}_g$ is the estimate of $C_g = mg/K_T$, and $\hat{C}_z$ is the estimate of $C_z = m/(X_3^{a*} K_T)$. The update laws for $\hat{C}_g$ and $\hat{C}_z$ are (5.4) and (5.12), respectively.

From (4.12a), (3.14b), and (5.1), the DIBVS lateral motion subsystem dynamics is

$$\dot{s}_{lp} = -\frac{\lambda}{X_3^{a*}} v_l^a - S s_{lp} \dot{\psi} \tag{5.7a}$$

$$\dot{v}_l^a = \frac{K_T u}{m} S(\eta_{m_1} - \eta_{e_1}) - S v_l^a \dot{\psi} \tag{5.7b}$$

where s_{lp}, v_l^a, S are defined in (4.10), (4.34), (4.2b), respectively; $\eta_{e_1} = [\phi_e, \theta_e]^T$; and $\eta_{m_1} = [\phi_m, \theta_m]^T$. In (5.7) the input is taken as η_{m_1}. Without loss of generality the desired feature value $s_{lp}^* = 0$. The control is taken as

$$u S \eta_{m_1} = u S \hat{\eta}_{e_1} - \lambda \hat{C}_z K_{l1}^2 \delta_{lp2} - K_{l2} \delta_{lp2} \tag{5.8}$$

where K_{l1}, K_{l2} are positive gains; $\hat{\eta}_{e_1} = [\hat{\phi}_e, \hat{\theta}_e]^T$; and $\hat{\theta}_e, \hat{\phi}_e$ are estimates of θ_e, ϕ_e, respectively. The update law for $\hat{\eta}_{e_1}$ is

$$\dot{\hat{\eta}}_{e_1} = -u K_{lrp} S^T \delta_{lp2} \tag{5.9}$$

where K_{lrp} is positive, and

$$\delta_{lp1} = s_{lp} \tag{5.10}$$

$$\delta_{lp2} = \frac{1}{K_{l1}} v_l^a - \delta_{lp1} \tag{5.11}$$

The update law for $\hat{C}_z$ in (5.3) and (5.8) is

$$\dot{\hat{C}}_z = K_z \left(\lambda K_{l1} \delta_{lp2}^T \delta_{lp2} + K_{h1} \delta_{hp2}^2 \right) \tag{5.12}$$

where $K_z > 0$. From Chapter 4, the kinematics of $s_{\psi p}$ is

$$\dot{s}_{\psi p} = -\dot{\psi} \tag{5.13}$$

Without loss of generality the desired value of $s_{\psi p}$ is set to zero. In order to make $s_{\psi p}$ converge to zero, the yaw reference is taken as

$$\psi = K_\psi \int_0^t s_{\psi p}(\xi) d\xi \tag{5.14}$$

where $K_\psi > 0$.

Theorem 5.1. *The equilibrium* $[\delta_{hp1}, \delta_{hp2}, \delta_{lp1}, \delta_{lp2}, s_{\psi p}]^T = 0$ *of the closed-loop system* (5.2), (5.7), (5.13) *with control law* (5.3), (5.8), (5.14) *and parameter update laws* (5.4), (5.9), (5.12) *is GAS.*

Proof. Using the definition of δ_{h1} in (5.5), the definition of δ_{h2} in (5.6), and the height subsystem dynamics (5.2) gives

$$\dot{\delta}_{hp1} = -\frac{K_{h1}}{X_3^{a*}} \left(\delta_{hp1} + \delta_{hp2} \right) \tag{5.15a}$$

$$\dot{\delta}_{hp2} = -\frac{K_T}{m K_{h1}} \left(u - C_g \right) + \frac{K_{h1}}{X_3^{a*}} \left(\delta_{hp1} + \delta_{hp2} \right) \tag{5.15b}$$

The Lyapunov function candidate for the height subsystem (5.15) is chosen as

$$V_h = \frac{1}{2} \delta_{hp1}^2 + \frac{1}{2} \delta_{hp2}^2 + \frac{K_T}{2 m K_{h1}} \frac{\tilde{C}_g^2}{K_{hg}}$$

where $\tilde{C}_g = \hat{C}_g - C_g$. Based on (5.15a) and (5.15b), the time derivative of V_h is

$$\begin{aligned}
\dot{V}_h &= -\frac{K_{h1}}{X_3^{a*}}\delta_{hp1}\left(\delta_{hp1}+\delta_{hp2}\right) + \frac{K_T}{mK_{h1}}\frac{\tilde{C}_g\dot{\hat{C}}_g}{K_{hg}} \\
&\quad + \delta_{hp2}\left(-\frac{K_T}{mK_{h1}}\left(u - C_g\right) + \frac{K_{h1}}{X_3^{a*}}\left(\delta_{hp1}+\delta_{hp2}\right)\right) \\
&= -\frac{K_{h1}}{X_3^{a*}}\delta_{hp1}^2 + \frac{K_T}{mK_{h1}}\frac{\tilde{C}_g\dot{\hat{C}}_g}{K_{hg}} - \frac{K_T}{mK_{h1}}\delta_{hp2}\left(u - C_g - C_z K_{h1}^2\delta_{hp2}\right)
\end{aligned}$$

where $\tilde{C}_z = \hat{C}_z - C_z$. Using the adaptive control (5.3), (5.4) gives

$$\begin{aligned}
\dot{V}_h &= -\frac{K_{h1}}{X_3^{a*}}\delta_{hp1}^2 + \frac{K_T}{mK_{h1}}\frac{\tilde{C}_g\left(K_{hg}\delta_{hp2}\right)}{K_{hg}} \\
&\quad - \frac{K_T}{mK_{h1}}\delta_{hp2}\left(\left(K_{h2}+\hat{C}_z K_{h1}^2\right)\delta_{hp2} + \hat{C}_g - C_g - C_z K_{h1}^2\delta_{hp2}\right) \\
&= -\frac{K_{h1}}{X_3^{a*}}\delta_{hp1}^2 + \frac{K_T}{mK_{h1}}\tilde{C}_g\delta_{hp2} - \frac{K_T K_{h2}}{mK_{h1}}\delta_{hp2}^2 - \frac{K_T}{mK_{h1}}\delta_{hp2}\left(\tilde{C}_z K_{h1}^2\delta_{hp2} + \tilde{C}_g\right) \\
&= -\frac{K_{h1}}{Z^{v*}}\delta_{h1}^2 - \frac{K_T K_{h2}}{mK_{h1}}\delta_{h2}^2 - \frac{K_T K_{h1}}{m}\delta_{h2}^2\tilde{C}_z
\end{aligned} \tag{5.16}$$

Using the definition of δ_{l1} in (5.10), the definition of δ_{l2} in (5.11), and the lateral motion subsystem dynamics (5.7) gives

$$\dot{\delta}_{lp1} = -\frac{\lambda K_{l1}}{X_3^{a*}}\left(\delta_{lp1}+\delta_{lp2}\right) - \dot{\psi}S\delta_{lp1} \tag{5.17a}$$

$$\begin{aligned}
\dot{\delta}_{lp2} &= \frac{1}{K_{l1}}\left(\frac{K_T u}{m}S(\eta_{m_1}-\eta_{e_1}) - Sv_l^a\dot{\psi}\right) + \frac{\lambda K_{l1}}{X_3^{a*}}\left(\delta_{lp1}+\delta_{lp2}\right) + \dot{\psi}S\delta_{lp1} \\
&= \frac{K_T u}{mK_{l1}}S(\eta_{m_1}-\eta_{e_1}) - \frac{SK_{l1}}{K_{l1}}\left(\delta_{lp1}+\delta_{lp2}\right)\dot{\psi} + \frac{\lambda K_{l1}}{X_3^{a*}}\left(\delta_{lp1}+\delta_{lp2}\right) + \dot{\psi}S\delta_{lp1} \\
&= \frac{K_T u}{mK_{l1}}S(\eta_{m_1}-\eta_{e_1}) - S\delta_{lp2}\dot{\psi} + \frac{\lambda K_{l1}}{X_3^{a*}}\left(\delta_{lp1}+\delta_{lp2}\right)
\end{aligned} \tag{5.17b}$$

The Lyapunov function candidate for the lateral subsystem (5.17) is

$$V_l = \frac{1}{2}\delta_{lp1}^T\delta_{lp1} + \frac{1}{2}\delta_{lp2}^T\delta_{lp2} + \frac{K_T}{2mK_{l1}K_{lrp}}\left(\tilde{\theta}_e^2 + \tilde{\phi}_e^2\right)$$

where $\tilde{\theta}_e = \hat{\theta}_e - \theta_e$ and $\tilde{\phi}_e = \hat{\phi}_e - \phi_e$. Based on (5.17a), (5.17b), and the skew-symmetry property of S

$$x^T S x = 0, \quad \text{for all } x \in \mathbb{R}^2$$

the time derivative of V_l is

$$\begin{aligned}
\dot{V}_l &= \delta_{lp1}^T \left(-\frac{\lambda K_{l1}}{X_3^{a*}} \left(\delta_{lp1} + \delta_{lp2} \right) - \dot{\psi} S \delta_{lp1} \right) + \frac{K_T}{m K_{l1} K_{lrp}} \left(\tilde{\theta}_e \dot{\hat{\theta}}_e + \tilde{\phi}_e \dot{\hat{\phi}}_e \right) \\
&\quad + \delta_{lp2}^T \left(\frac{K_T u}{m K_{l1}} S(\eta_{m_1} - \eta_{e_1}) - S \delta_{lp2} \dot{\psi} + \frac{\lambda K_{l1}}{X_3^{a*}} \left(\delta_{lp1} + \delta_{lp2} \right) \right) \\
&= -\frac{\lambda K_{l1}}{X_3^{a*}} \delta_{lp1}^T \delta_{lp1} + \frac{K_T}{m K_{l1} K_{lrp}} \left(\tilde{\theta}_e \dot{\hat{\theta}}_e + \tilde{\phi}_e \dot{\hat{\phi}}_e \right) + \delta_{lp2}^T \left(\frac{K_T u}{m K_{l1}} S(\eta_{m_1} - \eta_{e_1}) + \frac{\lambda K_{l1}}{X_3^{a*}} \delta_{lp2} \right) \\
&= -\frac{\lambda K_{l1}}{X_3^{a*}} \delta_{lp1}^T \delta_{lp1} + \frac{K_T}{m K_{l1}} \left(\frac{\tilde{\eta}_{e_1}^T \dot{\hat{\eta}}_{e_1}}{K_{lrp}} + \delta_{lp2}^T \left(u S(\eta_{m_1} - \eta_{e_1}) + \lambda C_z K_{l1}^2 \delta_{lp2} \right) \right) \qquad (5.18)
\end{aligned}$$

where $\tilde{\eta}_e = \hat{\eta}_{e_1} - \eta_{e_1} = [\tilde{\phi}_e, \tilde{\theta}_e]^T$. Substituting (5.8) and (5.9) into (5.18) gives

$$\begin{aligned}
\dot{V}_l &= -\frac{\lambda K_{l1}}{X_3^{a*}} \delta_{lp1}^T \delta_{lp1} - \frac{K_T}{m K_{l1}} \tilde{\eta}_{e_1}^T u S^T \delta_{lp2} \\
&\quad + \frac{K_T}{m K_{l1}} \delta_{lp2}^T \left(u S \hat{\eta}_{e_1} - \lambda \hat{C}_z K_{l1}^2 \delta_{lp2} - K_{l2} \delta_{lp2} - u S \eta_{e_1} + \lambda C_z K_{l1}^2 \delta_{lp2} \right) \\
&= -\frac{\lambda K_{l1}}{X_3^{a*}} \delta_{lp1}^T \delta_{lp1} - \frac{K_T}{m K_{l1}} \tilde{\eta}_{e_1}^T u S^T \delta_{lp2} + \frac{K_T}{m K_{l1}} \delta_{lp2}^T \left(u S \tilde{\eta}_{e_1} - \lambda \tilde{C}_z K_{l1}^2 \delta_{lp2} - K_{l2} \delta_{lp2} \right) \\
&= -\frac{\lambda K_{l1}}{X_3^{a*}} \delta_{lp1}^T \delta_{lp1} - \frac{K_T K_{l2}}{m K_{l1}} \delta_{lp2}^T \delta_{lp2} + \frac{K_T}{m K_{l1}} \left(-\tilde{\eta}_{e_1}^T u S^T \delta_{lp2} + \delta_{lp2}^T \left(u S \tilde{\eta}_{e_1} - \lambda \tilde{C}_z K_{l1}^2 \delta_{lp2} \right) \right) \\
&= -\frac{\lambda K_{l1}}{X_3^{a*}} \delta_{lp1}^T \delta_{lp1} - \frac{K_T K_{l2}}{m K_{l1}} \delta_{lp2}^T \delta_{lp2} - \frac{\lambda K_T K_{l1}}{m} \tilde{C}_z \delta_{lp2}^T \delta_{lp2} \qquad (5.19)
\end{aligned}$$

The Lyapunov function candidate for the closed-loop translational motion is chosen as

$$V = V_h + V_l + \frac{K_T}{m K_z} \tilde{C}_z^2$$

From the update law (5.12) and (5.16), (5.19), the time derivative of V is

$$\dot{V} = -\frac{K_{h1}}{Z^{v*}} \delta_{h1}^2 - \frac{K_T K_{h2}}{m K_{h1}} \delta_{h2}^2 - \frac{\lambda K_{l1}}{Z^{v*}} \delta_{l1}^T \delta_{l1} - \frac{K_T K_{l2}}{m K_{l1}} \delta_{l2}^T \delta_{l2}$$

According to the LaSalle-Yoshizawa theorem [50, Thm. 2.1] the equilibrium point $[\delta_{h1}, \delta_{h2}, \delta_{l1}, \delta_{l2}]^T = 0$ is GAS.

From (5.13) and (5.14) the closed-loop yaw subsystem is

$$\dot{s}_4 = -K_\psi s_4$$

which is GES. Hence, the equilibrium $[\delta_{hp1}, \delta_{hp2}, \delta_{lp1}, \delta_{lp2}, s_{\psi p}]^T = 0$ of the closed-loop is GES. □

It is remarked that in the stability analysis above, the convergence of the translational motion subsystem does not require exponential convergence of $\dot{\psi}$. This should

be compared to the stability analysis of the PID DIBVS in Chapter 4 where exponential convergence of $\dot{\psi}$ was needed. This fact makes the adaptive DIBVS more suitable for applications where the vehicle tracks a time varying yaw trajectory.

5.1.2 Simulation Results

The visual servoing objective is to hover the vehicle at a constant position and yaw by regulating image feature error. Two cases are considered in simulation. The first case is the proposed control law with parameter update laws (5.4), (5.9), and (5.12) turned off. The second case simulates the entire proposed adaptive DIBVS. Comparing the two cases shows the benefits of adding the parameter adaptation. The model parameters used are given in Table 2.1. Image resolution was set to 320×200, the thrust coefficient $K_T = 4 \times 10^7$ N/s^2, and desired depth $X_3^{a*} = 1.335$ m. The initial displacement of the vehicle in the navigation frame is $[0.15, -0.15, -1]^T$ m with zero roll and pitch, and the initial value of yaw is $\pi/3$ rad. The initial values of v^a and ω^c are zero. Both ϕ_e and θ_e are set to 2°. The initial estimates of C_g, ϕ_e, θ_e used in the control law are 3.8×10^{-4} s, 0 rad, 0 rad, respectively. Two target points are located at coordinates $[-0.18, 0, 0]^T$ m and $[0.18, 0, 0]^T$ m in $\mathcal{N}$. The control gains are given in Table 5.1. A high gain PID controller is used to track reference attitude.

Gain	Value
K_{h1}	0.9
K_{h2}	0.09
K_{hg}	0.04
K_{l1}	0.3182
K_{l2}	0.07
K_{lrp}	0.02
K_ψ	0.5
K_z	0.01

Table 5.1: Outer loop control gains.

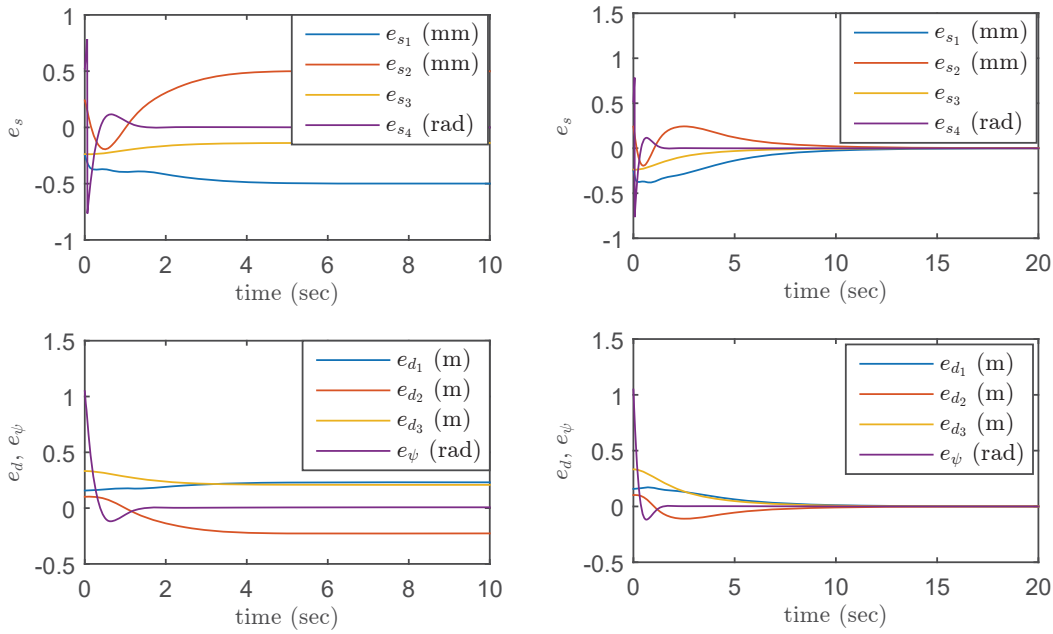
The image feature error is denoted

$$e_s = [e_{s_1}, e_{s_2}, e_{s_3}, e_{s_4}]^T = [s_{lp}^T, s_{hp}, s_{\psi p}]^T - [0, 0, 1, 0]^T$$

The desired translational displacement of the UAV is denoted d_{nc}^{n*} and translational error is

$$e_d = [e_{d_1}, e_{d_2}, e_{d_3}]^T = d_{nc}^n - d_{nc}^{n*}$$

where $d_{nc}^{n*} = [0.048, -0.047, -1.335]^T$ m. This value of d_{nc}^{n*} corresponds to $e_s = 0$. The desired yaw of the UAV is $\psi^* = -0.221$ rad, and yaw tracking error is $e_\psi = \psi - \psi^*$. The trajectories of e_s, e_d, and e_ψ are shown in Figure 5.1a. This figure shows that



(a) Non-adaptive case.

(b) Adaptive case.

Figure 5.1: Simulated trajectories of image feature error e_s, vehicle's 3D position error e_d, and e_ψ.

when the parameter update law is off, the image features e_{s_1}, e_{s_2}, and e_{s_3} converge to −0.499 mm, 0.499 mm, and −0.138, respectively. The vehicle position error converges to $e_d = [0.231, -0.227, 0.208]^T$ m. This significant steady state position error is due to the additive input disturbance ϕ_e, θ_e, and the inaccurate estimate of K_T/m.

Next, the parameter update laws are turned on to ensure e_s and e_d converge to zero. This convergence is shown in Figure 5.1b. The estimates $\hat{\phi}_e$, $\hat{\theta}_e$, $\hat{C}_g$ and $\hat{C}_z$ are shown in Figure 5.2. It can be seen that $\hat{\phi}_e$, and $\hat{\theta}_e$ converge to their actual value 2° = 0.0349 rad. The estimate $\hat{C}_g$ also converges to 0.3924 which corresponds to the amount of thrust needed to compensate gravity. The value of $\hat{C}_z$ converges to 7.5×10^{-9} kg·s^2/(m·N). Since there is no uncertainty in the yaw kinematics, in both simulation cases e_{s_4} and e_ψ converge to zero. Hence, the simulations show that the proposed adaptive control compensates for constant additive input disturbances (i.e., measurement bias) and uncertainty in K_T and X_3^{a*}.

5.1.3 Experimental Results

In practice the vehicle's state measurements contain noise which can lead to divergence of the adaptive law (5.12). A typical solution to this problem is to use the projection algorithm as in [90]. This method requires a range of C_z is known *a*

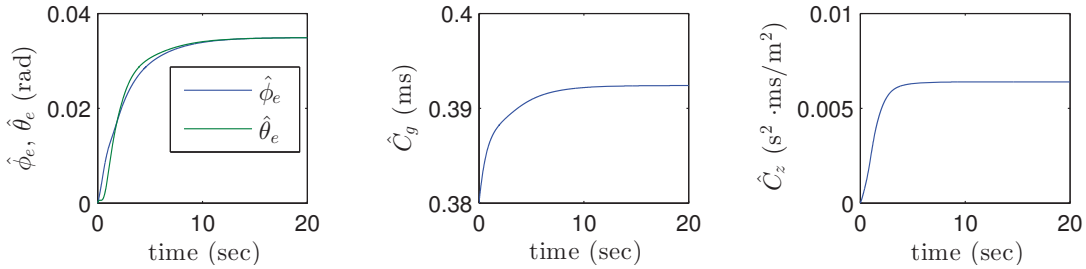


Figure 5.2: Simulated trajectories of estimated parameters $\hat{\phi}_e, \hat{\theta}_e, \hat{C}_g$, and $\hat{C}_z$. In the units of $\hat{C}_g$ and $\hat{C}_z$, "ms" denotes milliseconds.

priori. Another approach is to remove the term involving $\hat{C}_z$ in (5.3) and (5.8), and to choose controller gains satisfying $K_{l2} > \lambda C_z K_{l1}^2$ and $K_{h2} > C_z K_{h1}^2$. Such an approach ensures the closed-loop is asymptotically stable in the presence of measurement noise. Since both abovementioned approaches assume a known range for C_z, the method which eliminates $\hat{C}_z$ is chosen due to its relative simplicity. The values of K_{h1}, K_{h2}, K_{l1}, and K_{l2} still require tuning so they satisfy their inequality constraints. Hence, the control law (5.3) and parameter update (5.4) are implemented as

$$u = K_{h2}\delta_{hp2} + K_{hg}\int_0^t \delta_{hp2}(\xi)d\xi \tag{5.20}$$

It is remarked that when the vehicle hovers, u is approximately constant and equal to a value which compensates gravity. Thus, u is taken as a constant in the control (5.8). Consequently, the control (5.8) is combined with the parameter update law (5.9) to obtain

$$\eta_{m_1} = -K_{lrp}\int_0^t S^T \delta_{lp2}(\xi)d\xi - K_{l2}S^{-1}\delta_{lp2} \tag{5.21}$$

The values of the control gains in (5.20), (5.21), and (5.14) are the same as in the simulation and given in Table 5.3. The PID inner loop attitude controller used in experiment is the same as that presented in Section 4.4.1. A block diagram of the implementation is shown in Figure 5.3, where the mixer block distributes PWM to the individual ESCs based on u and τ^c.

As in the simulation we consider two cases in experiment [1],i.e., with the adaptive law (5.4) and (5.9) on and off. In the adaptive case, the initial values of $\hat{\phi}_e$, $\hat{\theta}_e$ are set to zero. The nominal value of C_g in the non-adaptive case is 0.435, which is obtained from a manual flight test, and this value is the initial value of $\hat{C}_g$ in the adaptive case. The trajectory of the image feature error e_s is given in Figure 5.4 for both cases. The trajectories of the two image points in the image plane are shown in Figure 5.5, where the initial coordinates are shown as a square and the circle

[1] A video of the adaptive DIBVS experiment is at www.youtube.com/watch?v=O1HvoiQXg7k

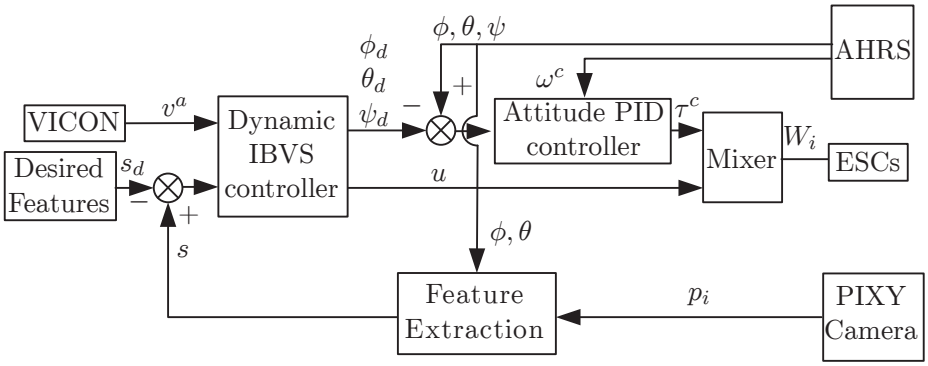
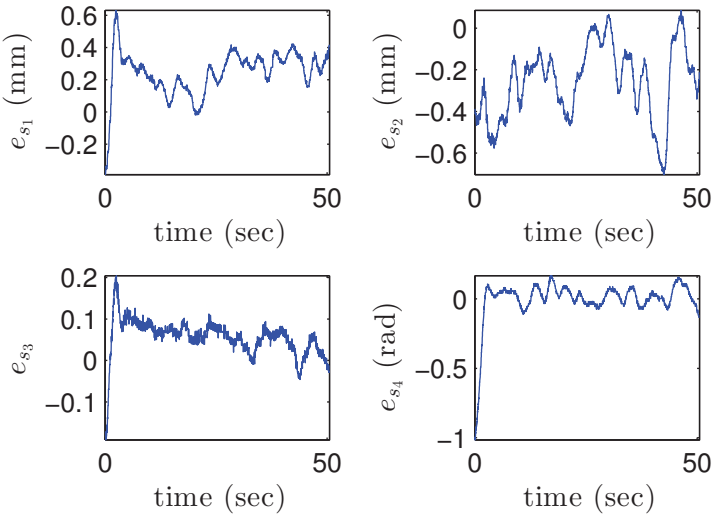


Figure 5.3: Block diagram of Adaptive DIBVS control with inner-outer loop structure.

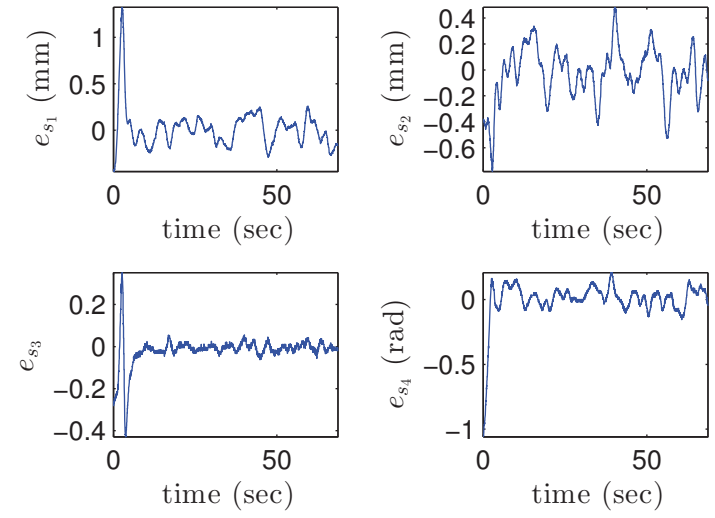
denotes the final point. Figure 5.6 shows the trajectories of vehicle position d_{nc}^n for two cases. Figure 5.7a and Figure 5.7b show the corresponding 3D translational error e_d and e_ψ. Figure 5.8 illustrates the estimates $\hat{\phi}_e$, $\hat{\theta}_e$, and $\hat{C}_g$. From the above figures, both cases reach steady state in about 15 seconds. Table 5.2 gives the mean and standard deviation of e_s, e_d and e_ψ after 15 seconds. It can be seen from Figure 5.4 that in the non-adaptive case e_{s_3} decreases slowly due to the slowly decreasing battery voltage. On the other hand, in the adaptive case e_{s_3} stays near 0 for most of the experiment. Table 5.2 shows that in the adaptive case e_{s_1} and e_{s_2} have significantly reduced mean values and similar standard deviations relative to the non-adaptive case. The trajectories of e_{s_4} converge to zero for both cases, and the performance for regulating yaw motion is similar in the two cases. This is to be expected since the same control law (5.14) is used. The trajectories in the image plane shown in Figure 5.5 are consistent with Table 5.2. That is, the lower mean values of e_{s_1} and e_{s_2} correspond to a more centered steady state image coordinates. Figure 5.7 and Table 5.2 show the 3D position errors in the adaptive case are much smaller than the non-adaptive case. Hence, the adaptive nature of the proposed control helps to improve visual servoing performance.

	Adaptive control		Non-adaptive control	
Error	Mean	Standard deviation	Mean	Standard deviation
e_{s_1} (mm)	0.00	0.13	0.25	0.1042
e_{s_2} (mm)	-0.01	0.18	-0.23	0.1694
e_{s_3}	-0.00	0.02	0.05	0.03
e_{s_4} (rad)	0.02	0.07	0.02	0.06
e_{d_1} m	-0.008	0.072	-0.128	0.060
e_{d_2} (m)	-0.002	0.077	0.116	0.070
e_{d_3} (m)	0.004	0.021	-0.074	0.044
e_ψ (rad)	0.070	0.061	-0.108	0.082

Table 5.2: Statistics of the error signals e_s, e_d , and e_ψ.

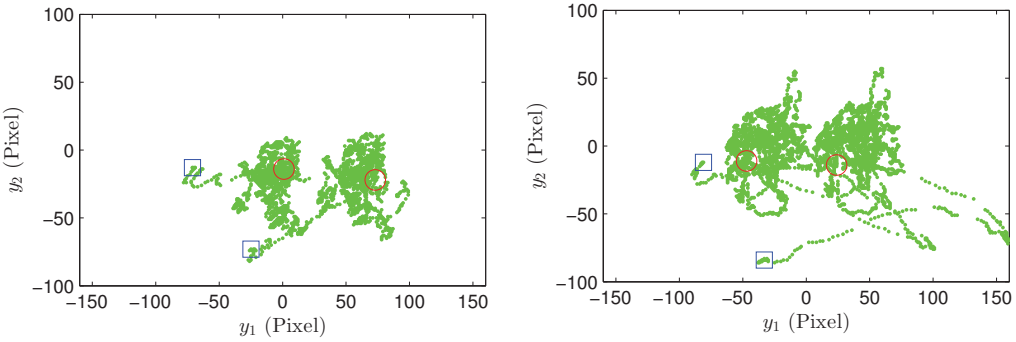


(a) Non-adaptive case.



(b) Adaptive case

Figure 5.4: Experimental results of the trajectories of image feature error e_s.



(a) Non-adaptive case

(b) Adaptive case

Figure 5.5: Experimental results of the image point trajectories shown in the image plane. The initial coordinates are shown as a square, and a circle denotes the final point.

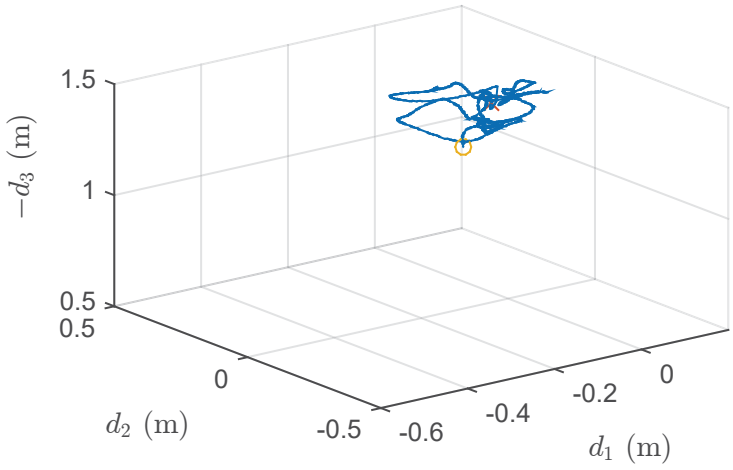
5.2 Adaptive DIBVS for Line Moment Features

This section presents a different adaptive DIBVS for a set of coplanar parallel lines. As compared to the adaptive DIBVS proposed in Section 5.1, the control system has a different cascade structure. The outer loop still regulates the image feature error, but has angular velocity ω^c instead of Euler angle η as an output. The inner loop tracks reference angular velocity from the outer loop. The applied force model parameters and mass are assumed known. The proposed DIBVS is adaptive to depth, which is typically required in conventional IBVS.

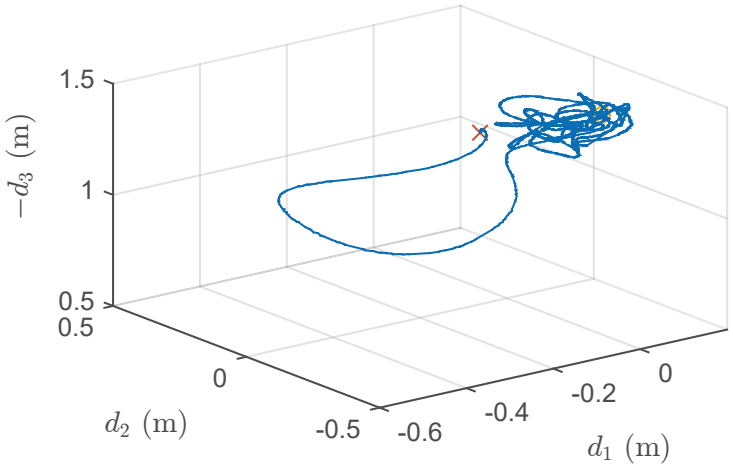
5.2.1 Translational Motion

As in Chapter 4, the feature vector for a set of horizontal parallel lines is taken as $s = [s_{ll}, s_{hl}, s_{\psi l}]^T$ where the components are defined in (4.19), (4.20), and (4.21). The motion control objective is to stabilize the UAV above parallel lines at a desired height. This height is determined by μ_{20}^{a*} defined in (4.16), and when $\mu_{20}^{a} = \mu_{20}^{a*}$ the desired value $s_{hl}^* = 1$. The control objective is also to position the UAV laterally relative to the parallel lines. This means the desired value of $s_{ll}^* = 0$. The value of $s_{\psi l}^*$ is 0 which implies translational motion can only occur along the "forward" direction of the lines. The "forward" direction of the vehicle is definied by a vector obtained by projecting the c_1^a-axis onto the line. Clearly, it is assumed the singular case initial conditions where c_1^a is perpendicular to the line are avoided, i.e., $\alpha(0)^a = \pi/2$. Finally, the vehicle velocity in the direction of the parallel lines is regulated to v_d. That is, as shown in Figure 5.9, the desired UAV velocity v^{a*} projected onto the direction of the line should be v_d. This means

$$(v^a)^T h_{\alpha^a} - v_d = 0 \tag{5.22}$$

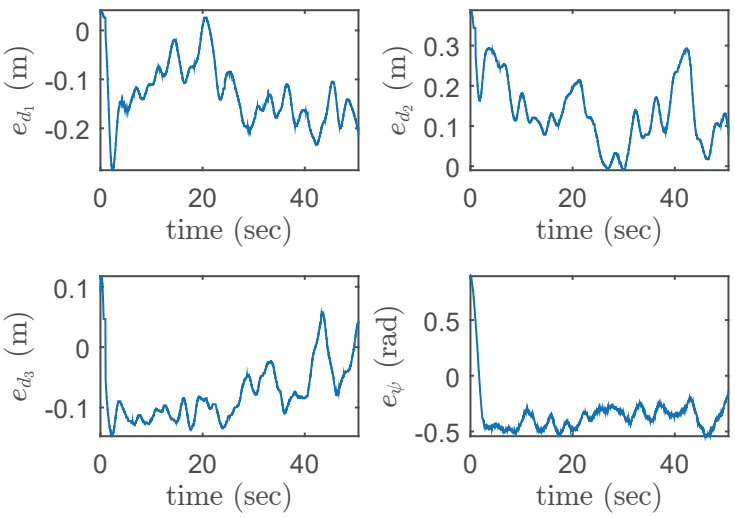


(a) Non-adaptive case

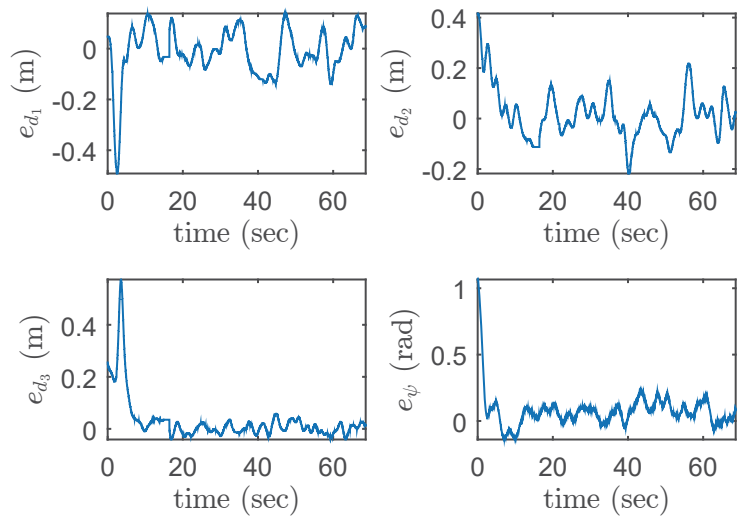


(b)

Figure 5.6: Experimental results: trajectory of the vehicle's 3D position d_{nc}^n. The initial and final values are denoted by a cross and circle, respectively.



(a) Non-adaptive case



(b)

Figure 5.7: Experimental results: the vehicle's 3D position error e_d and e_ψ.

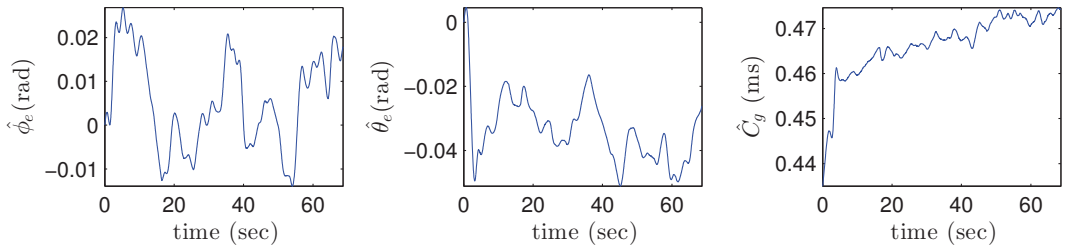


Figure 5.8: Experimental results: trajectories of estimated parameters $\hat{\phi}_e$, $\hat{\theta}_e$, and $\hat{C}_g$. In the units of $\hat{C}_g$, "ms" denotes milliseconds.

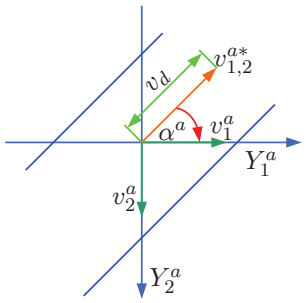


Figure 5.9: The relationship (5.22) between v^{a*}, α^a and v_d. where the vector $v_{1,2}^{a*}$ is the projection of v^{a*} onto the image plane, and the image plane basis vectors for the virtual camera are denoted $\{Y_1^a, Y_2^a\}$.

where $h_{\alpha^a} = [c_{\alpha^a}, -s_{\alpha^a}, 0]^T$. The image feature error is defined as

$$e_s = \begin{bmatrix} e_{st} \\ e_{s\psi} \end{bmatrix}$$

where $e_{st} = [e_{s1}, e_{s2}]^T = [s_{hl} - 1, s_{ll}]^T$, $e_{s\psi} = s_{\psi l}$. From (2.9), (2.11b), (3.14b), (4.22), the DIBVS model is

$$\dot{e}_{st} = -\frac{1}{X_3^{a*}} D(\lambda) L_{st}(\alpha^a) v^a \tag{5.23a}$$

$$\dot{e}_{s\psi} = \dot{\psi} \tag{5.23b}$$

$$\dot{v}^a = \left[\dot{\psi} E_3\right]_\times v^a + F^a \tag{5.23c}$$

$$\dot{R} = R\left[\omega^c\right]_\times \tag{5.23d}$$

$$\dot{\omega}^c = -J^{-1}\left[\omega^c\right]_\times J\omega^c + J^{-1}\tau^c \tag{5.23e}$$

where

$$L_{st}(\alpha^a) = \begin{bmatrix} 0 & 0 & 1 \\ s_{\alpha^a} & c_{\alpha^a} & 0 \end{bmatrix}, \quad D(\lambda) = \begin{bmatrix} 1 & 0 \\ 0 & \lambda \end{bmatrix}, \quad F^a = -\frac{T_M}{m} R_{\theta\phi} E_3 + g E_3$$

Here, as compared to the Section 5.1, the input for system (5.23) is T_M and τ^c, which means K_T, m, is assumed known. The outer loop is (5.23a), (5.23c), and (5.23d) with ω^c and T_M as inputs, while the inner loop is (5.23e). The proposed control is adaptive to two parameters γ_1 and γ_2 define as

$$\gamma_1 = \frac{1}{X_3^{a*}}, \quad \gamma_2 = \frac{\lambda}{\hat{\lambda} X_3^{a*}}$$

Introducing these parameters eliminates the requirement of knowing the desired depth X_3^{a*}. Three error variables are introduced

$$\delta_{l1} = \begin{bmatrix} e_{st} \\ v_d \end{bmatrix} \tag{5.24a}$$

$$\delta_{l2} = Q^{-1} H(\hat{\lambda}) W_{\alpha^a} v^a - \delta_{l1} \tag{5.24b}$$

$$\delta_{l3} = F' + k_{l2}\delta_{l2} + \hat{M} Q \delta_{l2} \tag{5.24c}$$

where

$$H(\hat{\lambda}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \hat{\lambda} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} k_{l1} I_2 & 0 \\ 0 & 1 \end{bmatrix}, \quad W_{\alpha^a} = \begin{bmatrix} L_{st}(\alpha^v) \\ h_{\alpha^a}^T \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ s_{\alpha^a} & c_{\alpha^a} & 0 \\ c_{\alpha^a} & -s_{\alpha^a} & 0 \end{bmatrix}$$

$$F' = \Delta_1 W_{\alpha^a} F^a, \quad \Delta_1 = Q^{-1} H(\hat{\lambda}), \quad \hat{M} = \begin{bmatrix} \hat{\gamma}_1 & 0 & 0 \\ 0 & \hat{\gamma}_2 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

k_{l1}, k_{l2} are constant gains, and $\hat{\gamma}_1, \hat{\gamma}_2$ denote estimated values of γ_1, γ_2, respectively.

Theorem 5.2. *If the control T_M and ω^c satisfy*

$$\begin{aligned} &\Delta_1 W_{\alpha^a} R_{\theta\phi} \left([\omega^c]_\times T_M + \dot{T}_M I_3 \right) E_3 = \\ &\left(k_2 I_3 + \hat{M} Q \right) \left(\hat{M} Q \delta_{l1} - k_{l2}\delta_{l2} + \delta_{l3} \right) + \dot{\hat{M}} Q \delta_{l2} + \delta_{l2} + k_{l3}\delta_{l3} \end{aligned} \tag{5.25}$$

where k_{l1}, k_{l2}, $k_{l3} > 0$, and the update law for $\hat{\gamma}_1$ and $\hat{\gamma}_2$ is

$$\dot{\hat{\gamma}}_1 = k_\gamma \left(k_{l1}\delta_{21}^2 + \left(k_{l2} + \hat{\gamma}_1 k_{l1} \right) k_{l1} \delta_{31} \left(\delta_{11} + \delta_{21} \right) \right) \tag{5.26a}$$

$$\dot{\hat{\gamma}}_2 = k_\gamma \left(k_{l1}\delta_{22}^2 + \left(k_{l2} + \hat{\gamma}_2 k_{l1} \right) k_{l1} \delta_{32} \left(\delta_{12} + \delta_{22} \right) \right) \tag{5.26b}$$

where δ_{ij} is the jth component of vector δ_{li}, and $k_\gamma > 0$, then the error signals e_{st}, δ_{l2}, δ_{l3} globally asymptotically converge to zero.

Proof. The controller in (5.25) is obtained via a backstepping approach. The backstepping procedure starts with the image error kinematics (5.23). A Lyapunov function candidate V_1 is chosen as

$$V_1 = \frac{1}{2} e_{st}^T e_{st} \tag{5.27}$$

Then the time derivative of (5.27) is

$$\dot{V}_1 = -\frac{1}{X_3^{a*}} e_{st}^T D(\lambda) L_{st}(\alpha^a) v^a$$

Since X_3^{a*} is positive, if the following equations holds

$$D(\lambda)L_{st}(\alpha^a)v^a = k_1 e_{st}$$

it can be seen that system (5.23) is GES. In order to make e_{st} converge to zero and the UAV fly at a desired velocity along the power lines, the input velocity v^{a*} should satisfy

$$H(\lambda)W_{\alpha^a}v^{a*} = Q\delta_{l1}$$

where $H(\lambda) = \text{diag}([1, \lambda, 1])$. With the term δ_{l2} defined in (5.24b) describing the difference between the desired and the actual velocity, the time derivative of δ_{l1} and L_1 can be obtained as

$$\dot{\delta}_{l1} = -\frac{1}{X_3^{a*}}\begin{bmatrix} D(\lambda)L_{st}(\alpha^a) \\ 0 \end{bmatrix} W_{\alpha^a}^{-1}H^{-1}(\lambda)Q(\delta_{l2} + \delta_{l1}) = -MQ\left(\delta_{l1} + \delta_{l2}\right) \tag{5.28}$$

where $M = \text{diag}([\gamma_1, \gamma_2, 1])$, and

$$\begin{aligned} \dot{V}_1 &= -\frac{1}{X_3^{a*}}e_{st}^T D(\lambda)L_{st}(\alpha^a)W_{\alpha^a}^{-1}H^{-1}(\hat{\lambda})Q(\delta_{l2} + \delta_{l1}) \\ &= -\frac{1}{X_3^{a*}}e_{st}^T D(\lambda)D^{-1}(\hat{\lambda})k_{l1}e_{st} - \delta_{l1}^T MQ\delta_{l2} \end{aligned} \tag{5.29}$$

Differentiating W_{α^a} in (5.24b) yields

$$\begin{aligned} \dot{W}_{\alpha^a} &= \begin{bmatrix} 0 & 0 & 0 \\ c_{\alpha^a} & -s_{\alpha^a} & 0 \\ -s_{\alpha^a} & -c_{\alpha^a} & 0 \end{bmatrix}\dot{\alpha}^a \\ &= \begin{bmatrix} 0 & 0 & 1 \\ s_{\alpha^a} & c_{\alpha^a} & 0 \\ c_{\alpha^a} & -s_{\alpha^a} & 0 \end{bmatrix}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\dot{\psi} = W_{\alpha^a}\left[\dot{\psi}E_3\right]_\times \end{aligned} \tag{5.30}$$

Using (5.24b), (5.28), (5.30), and the expression of F' in (5.24c) yields

$$\begin{aligned} \dot{\delta}_{l2} &= Q^{-1}H(\hat{\lambda})\left(\dot{W}_{\alpha^a}v^a + W_{\alpha^a}\dot{v}^a\right) - \dot{\delta}_{l1} \\ &= Q^{-1}H(\hat{\lambda})\left(W_{\alpha^a}\left[\dot{\psi}E_3\right]_\times v^a + W_{\alpha^a}\left(-\left[\dot{\psi}E_3\right]_\times v^a + F^a\right)\right) + MQ\left(\delta_{l1} + \delta_{l2}\right) \\ &= F' + MQ\left(\delta_{l1} + \delta_{l2}\right) \end{aligned} \tag{5.31}$$

Continuing the backstepping method, the second Lyapunov function candidate is defined as

$$V_2 = V_1 + \frac{1}{2}\delta_{l2}^T\delta_{l2} + \frac{1}{2k_\gamma}\left(\tilde{\gamma}_1^2 + \tilde{\gamma}_2^2\right) \tag{5.32}$$

where $\tilde{\gamma}_i = \gamma_i - \hat{\gamma}_i$, $i = 1, 2$. From (5.29) and (5.31), the time derivative of (5.32) is

$$\dot{V}_2 = -\frac{1}{X_3^{a*}} e_{st}^T D(\lambda) D^{-1}(\hat{\lambda}) k_{l1} e_{st} + \delta_{l2}^T \left(F' + MQ\delta_{l2} \right) - \frac{1}{k_\gamma} \left(\tilde{\gamma}_1 \dot{\hat{\gamma}}_1 + \tilde{\gamma}_2 \dot{\hat{\gamma}}_2 \right) \tag{5.33}$$

Choosing

$$F' = -k_{l2}\delta_{l2} - \hat{M}Q\delta_{l2} \tag{5.34}$$

then (5.33) can be written as

$$\dot{V}_2 = -\frac{1}{X_3^{a*}} e_{st}^T D(\lambda) D^{-1}(\hat{\lambda}) k_{l1} e_{st} - k_{l2}\delta_{l2}^T\delta_{l2} + \delta_{l2}^T \tilde{M} Q \delta_{l2} - \frac{1}{k_\gamma} \left(\tilde{\gamma}_1 \dot{\hat{\gamma}}_1 + \tilde{\gamma}_2 \dot{\hat{\gamma}}_2 \right)$$

where $\tilde{M} = M - \hat{M}$. Based on the definition of δ_{l3} in (5.24c) an expression for F' is solved and substituted into (5.31) to obtain

$$\dot{\delta}_{l2} = MQ\delta_{l1} - k_{l2}\delta_{l2} + \tilde{M}Q\delta_{l2} + \delta_{l3}$$

and the time derivative of V_2 is

$$\dot{V}_2 = -\frac{1}{X_3^{a*}} e_{st}^T D(\lambda) D^{-1}(\hat{\lambda}) k_{l1} e_{st} - k_{l2}\delta_{l2}^T\delta_{l2} + \delta_2^T \tilde{M} Q \delta_{l2} - \frac{1}{k_\gamma} \left(\tilde{\gamma}_1 \dot{\hat{\gamma}}_1 + \tilde{\gamma}_2 \dot{\hat{\gamma}}_2 \right) + \delta_{l2}^T\delta_{l3} \tag{5.35}$$

Using (5.34) the time derivative of δ_{l3} is

$$\begin{aligned}
\dot{\delta}_{l3} &= \dot{F}' + \left(k_{l2} I_3 + \hat{M}Q \right) \left(MQ\delta_{l1} - k_{l2}\delta_{l2} + \tilde{M}Q\delta_{l2} + \delta_{l3} \right) + \dot{\hat{M}}Q\delta_{l2} \\
&= \dot{F}' + \left(k_{l2} I_3 + \hat{M}Q \right) \left(\hat{M}Q\delta_{l1} - k_{l2}\delta_{l2} + \delta_{l3} \right) + \dot{\hat{M}}Q\delta_{l2} + \left(k_{l2} I_3 + \hat{M}Q \right) \tilde{M}Q \left(\delta_{l2} + \delta_{l1} \right)
\end{aligned} \tag{5.36}$$

Considering the Lyapunov function candidate

$$V_3 = V_2 + \frac{1}{2}\delta_{l3}^T\delta_{l3}$$

and using (5.35) and (5.36), gives

$$\begin{aligned}
\dot{V}_3 = &-\frac{1}{X_3^{a*}} e_{st}^T D(\lambda) D^{-1}(\hat{\lambda}) k_{l1} e_{st} - k_{l2}\delta_{l2}^T\delta_{l2} \\
&+ \delta_{l3}^T \left(\dot{F}' + \left(k_2 I_3 + \hat{M}Q \right) \left(\hat{M}Q\delta_{l1} - k_{l2}\delta_{l2} + \delta_{l3} \right) + \dot{\hat{M}}Q\delta_{l2} + \delta_{l2} \right) \\
&+ \delta_{l2}^T \tilde{M} Q \delta_{l2} - \frac{1}{k_\gamma} \left(\tilde{\gamma}_1 \dot{\hat{\gamma}}_1 + \tilde{\gamma}_2 \dot{\hat{\gamma}}_2 \right) + \delta_{l3}^T \left(k_{l2} I_3 + \hat{M}Q \right) \tilde{M}Q \left(\delta_{l2} + \delta_{l1} \right)
\end{aligned}$$

From the definition of $R_{\theta\phi}$, an expression for its time derivative is

$$\dot{R}_{\theta\phi} = R_{\theta\phi} \left[\omega^c \right]_\times - \left[\dot{\psi} E_3 \right]_\times R_{\theta\phi} \tag{5.37}$$

Then from the expression of F', (5.30), and (5.37) gives

$$\dot{F}' = -\Delta_1 W_{\alpha^a} R_{\theta\phi} \left([\omega^c]_\times T_M E_3 + \dot{T}_M E_3 \right)$$

It can be seen that choosing the control law (5.25), and the parameter update law (5.26), the time derivative of L_3 can be written as

$$\dot{V}_3 = -\frac{1}{X^{a*}} e_{st}^T D(\lambda) D^{-1}(\lambda) k_{l1} e_{st} - k_{l2} \delta_{l2}^T \delta_{l2} - k_{l3} \delta_{l3}^T \delta_{l3}$$

Hence, the system is GES. □

It is remarked that (5.25) appears to define the four inputs, i.e., three angular velocity inputs and thrust. However, expanding (5.25) reveals only expressions for the first and second components of ω^c are determined. The third component is given by the yaw control in next subsection.

5.2.2 Yaw Rate

In order to control e_α, the yaw rate of the UAV is taken as

$$\dot{\psi} = -k_\alpha e_{s\psi} = -k_\alpha s_{\psi l}$$

with $k_\alpha > 0$. Since the Euler rates and ω^c are related by (2.10), the third component of the angular velocity fed to the inner loop is

$$\omega_3^c = \frac{c_\theta}{c_\phi} \left(-k_\alpha s_{\psi l} - \omega_2^c \frac{s_\phi}{c_\theta} \right) \tag{5.38}$$

The DIBVS law for T_M and ω^c is given by (5.25) and (5.38). As mentioned above, (5.25) defines the control for the first two components of ω^c and (5.38) provides the third component ω_3^c.

5.2.3 Simulation Results

In this section the proposed DIBVS (5.25), (5.26), and (5.38) is tested in simulation. The control law update frequency is 30 Hz. The inner loop PID control for angular velocity is updated at the frequency of 200 Hz which is a typical IMU measurement rate. Four lines (i.e., $N = 4$) are defined in the navigation frame $\mathcal{N}$ as

$$\begin{aligned}
L_1 &: \quad 2X_1 + X_2 = 1,\ X_3 = 4 \\
L_2 &: \quad 2X_1 + X_2 = -1,\ X_3 = 4 \\
L_3 &: \quad 2X_1 + X_2 = 2,\ X_3 = 4 \\
L_4 &: \quad 2X_1 + X_2 = 0,\ X_3 = 4
\end{aligned}$$

The control gains and scaling coefficient β are given in Table 5.3. The initial states

Gain	Value
k_{l1}	10
k_{l2}	5
k_{l3}	2
k_α	0.6
k_γ	1

Table 5.3: Control gains

of the vehicle $d_{nc}^n(0), v^a(0), \omega^c(0)$ are taken as zero. Also, the initial conditions for the parameter estimates are $\hat{\gamma}_i = 0$, $i = 1, 2$, and $R(0) = I_3$. The reference roll and pitch are bounded by $\pm 10°$. The desired velocity along the parallel lines is set as $v_d = 0.2$ m/s and the $\mu_{20}^{v\star}$ is chosen to correspond to a UAV position with $d_3 = -1$. That is, the UAV should move down by 1 m to achieve the desired s_{hl}^*.

First, focal length is assumed perfectly calibrated. Figure 5.10 shows the resulting trajectory of the UAV in the navigation frame $\mathcal{N}$, and the translational velocity expressed the camera frame $\mathcal{C}$. The trajectories of the proposed line feature and its image error are shown in Figure 5.11. It can been seen that the altitude of the UAV decreases to its desired value of -1 m in a few seconds. This makes s_{h1} converges to s_{hl}^*. Also, the desired velocity of $v_d = 0.2$ m/s is achieved. As expected, Figure 5.11 shows that the image errors converge to zero. Figure 5.12 shows the estimated value $\hat{\gamma}_i$, $i = 1, 2$ converges to a constant value.

In order to check the proposed approach's robustness to calibration error of intrinsic camera parameters and measurement noise in attitude, the estimated focal length is set to $\hat{\lambda} = 1.5\lambda$ and attitude noise is added with a spectral power of 2.5×10^{-6}, i.e., roughly corresponding to a magnitude of $2°$. Before using the measured attitude to convert the real image feature to the virtual image feature, a second-order low pass filter is used to filter the attitude signal. The simulation results with noise corrupted attitude measurement are shown in Figure 5.13, 5.14, and 5.15. In spite of the presence of noise and calibration error the performance of the proposed controller is satisfactory with trajectories for d_{nc}^n and v^c being similar to the ideal case in Figure 5.10.

Assumption 4.2 restricts the 3D target line to a horizontal plane. In some scenarios this assumption may not hold, and the case where the lines lie in a plane rotated $20°$ about n_1 axis is simulated. The same noise level and error in estimated focal length was considered. The results are shown in Figure 5.16 and 5.17. As shown in Figure 5.16, unlike previous cases the UAV keeps moving upwards because it controls its height above the inclined plane to a constant to achieve the desired value s_1^*. It can also be seen from Figure 5.17 that the image feature errors converge to zero even though the 3D target lines lie in a non-horizontal plane.

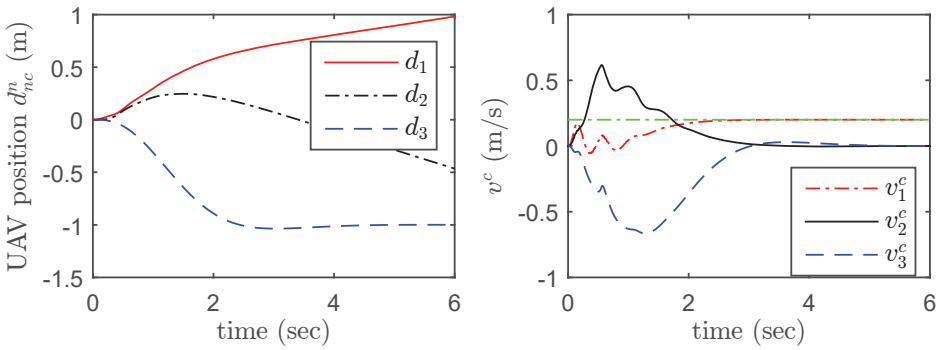


Figure 5.10: Simulated trajectories of UAV's position d_{nc}^n and velocity v^c.

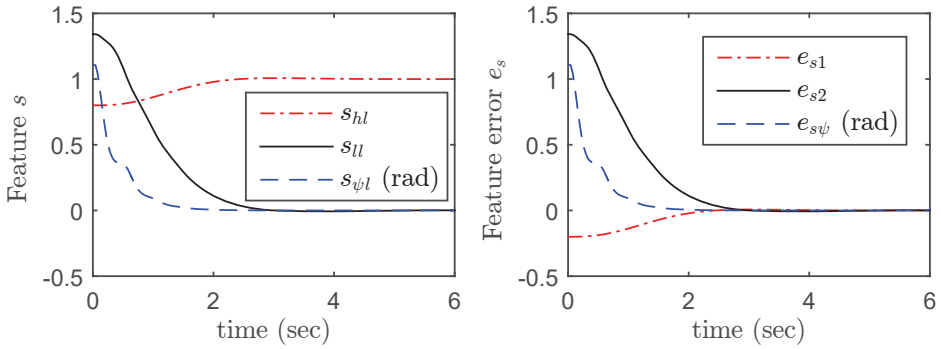


Figure 5.11: Simulated trajectories of line feature s and error e_s.

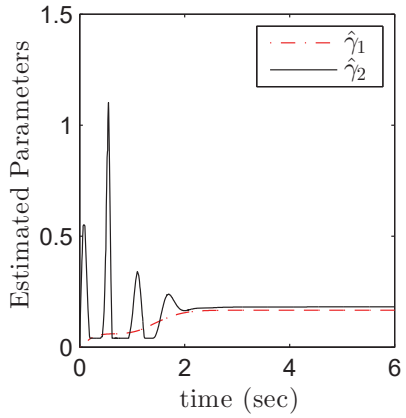


Figure 5.12: Simulated trajectories of parameter estimates $\hat{\gamma}_i$, $i = 1, 2$.

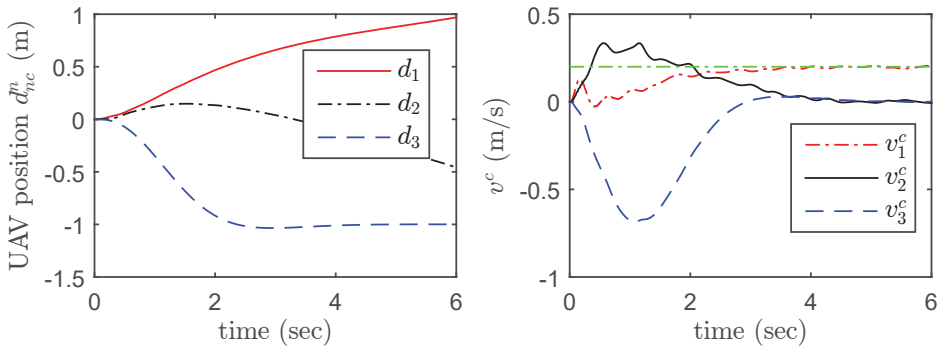


Figure 5.13: Simulated trajectories of UAV's position d^n_{nc} and velocity v^c with attitude measurement noise and $\hat{\lambda} = 1.5\lambda$.

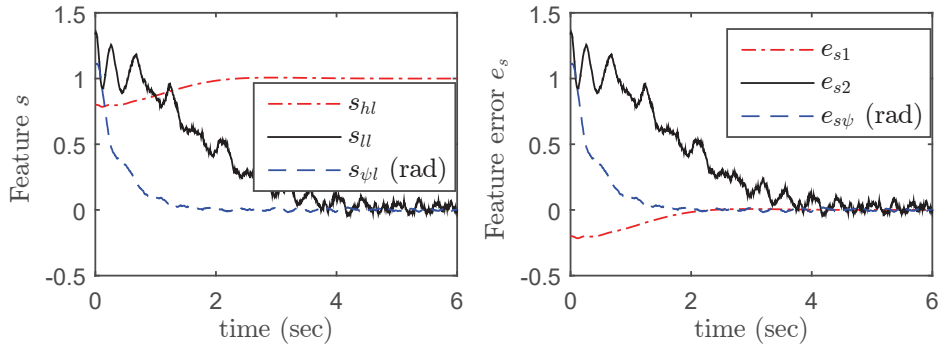


Figure 5.14: Simulated trajectories of line feature s and its error e_s with attitude measurement noise and $\hat{\lambda} = 1.5\lambda$.

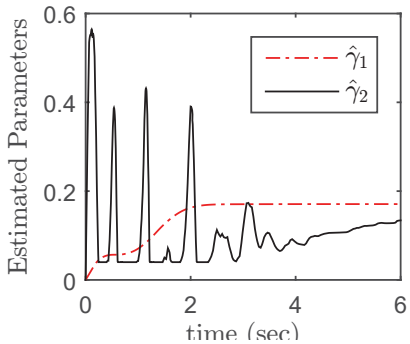


Figure 5.15: Simulated trajectories of parameter estimates $\hat{\gamma}_i$, $i = 1, 2$ with attitude measurement noise and $\hat{\lambda} = 1.5\lambda$.

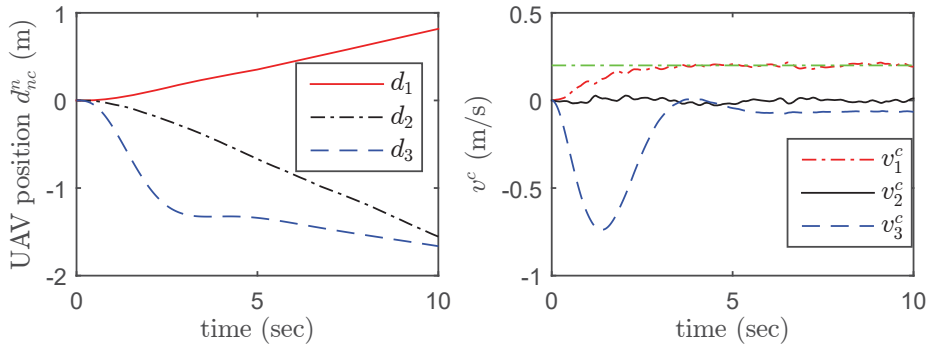


Figure 5.16: Simulated trajectories of UAV's position d_{nc}^n and velocity v^c with attitude measurement noise, $\hat{\lambda} = 1.5\lambda$ and non-horizontal target plane.

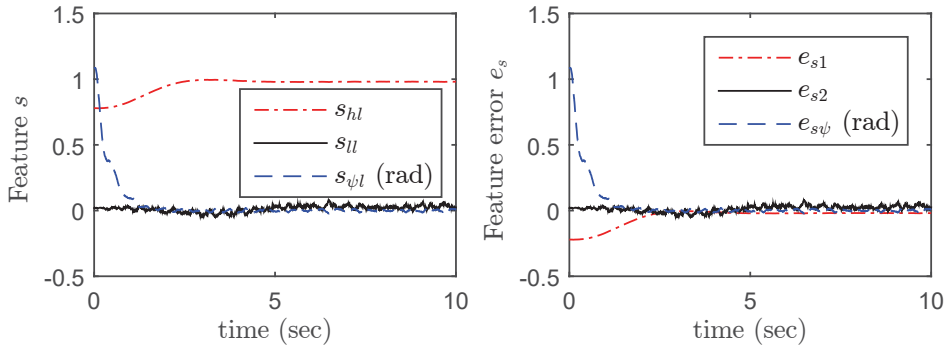


Figure 5.17: Simulated trajectories of line feature s and error e_s with attitude measurement noise, $\hat{\lambda} = 1.5\lambda$ and non-horizontal target plane.

5.3 Conclusion

This chapter presents two adaptive DIBVS using the moment features for points and parallel lines given in Chapter 4. This first control is for point moment features. It does not assume known depth, mass, or thrust coefficient. The method is robust to measurement bias in roll and pitch. As compared to the PID DIBVS in Chapter 4, the conditions on the control gains can be readily satisfied. The second control employs a different cascade structure where the inner loop tracks angular velocity. This control is adaptive to the image feature depth. Although the simulation results presented in Section 5.2.3 demonstrate that the proposed approach still works with lines in a non-horizontal plane, no rigorous proof of this robustness is provided.

Chapter 6

DIBVS with Input Saturation

As mentioned in Chapter 1, a camera's limited FoV means that during visual servoing the vehicle motion should be appropriately constrained to keep the target in view. For the designs proposed in Chapters 3–5 this is normally achieved by reducing controller gains and restricting the initial image feature error to a sufficiently small neighbourhood. In this chapter a practical solution is proposed where reference attitude is constrained in order to keep the target in the FoV. A similar approach is taken in [56, 57]. Constraining attitude corresponds to an input saturation design for the outer loop of the DIBVS model described in Chapter 4. Another motivation for considering an input saturation design is that thrust needs to be positive for quadrotors [52]. Compared to output saturation control proposed in [55], input saturation control is simpler and easier to implement. In this chapter, thrust gain, mass, and desired depth are treated as unknown parameters as in Chapter 4 and 5. The attitude measurement is assumed accurate, i.e., the attitude bias is not considered. This chapter is organized as follows. Section 6.1 describes several saturation functions used in the input saturation control. The details of the control design and the proof of stability for the closed-loop system are given in 6.2. Lastly, Section 6.3 provides simulation and experimental results.

6.1 Background

The proposed input saturation control requires a number of saturation function which are defined in this section. The first saturation function is a continuous function $\sigma_i : \mathbb{R} \to \mathbb{R}$ with following properties:

(a) $\xi\sigma_i(\xi) > 0,\ \xi \neq 0$

(b) $0 \leq \frac{d\sigma_i}{d\xi} \leq k_i,\ \forall \xi \in \mathbb{R}$

(c) $\sigma_i(\xi) = k_i\xi, |\xi| \leq l_i$

(d) $|\sigma_i(\xi)| \le M_i, |\xi| \ge l_i$

The second saturation function $\Sigma_i : \mathbb{R}^2 \to \mathbb{R}^2$ is defined as

$$\Sigma_i(\zeta) = [\sigma_i(\zeta_1), \sigma_i(\zeta_2)]^T$$

where σ_i is the first saturation function with properties defined above, $\zeta \in \mathbb{R}^2$, and $\zeta = [\zeta_1, \zeta_2]^T$.

Theorem 6.1. *Given two vector $\delta, \zeta \in \mathbb{R}^2$, if $\|\delta\| < l_i/2$ and $\|\zeta\| > l_i/2$, where $\|\cdot\|$ is the Euclidean norm, then $\zeta^T \Sigma_i(\zeta + \delta) > 0$.*

Proof. The vector ζ is parametrized by θ and α_ζ as

$$\zeta = \frac{\alpha_\zeta l_i}{2} \begin{bmatrix} c_\theta \\ s_\theta \end{bmatrix}$$

where $|\alpha_\zeta| > 1$. As shown in Figure 6.1, the angle between δ and ζ is denoted χ, and

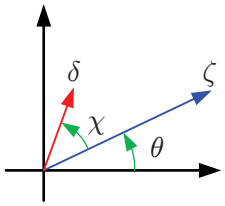


Figure 6.1: Vectors δ and ζ.

vector δ can be written as

$$\delta = \frac{\alpha_\delta l_i}{2} \left(c_\chi \begin{bmatrix} c_\theta \\ s_\theta \end{bmatrix} + s_\chi \begin{bmatrix} s_\theta \\ -c_\theta \end{bmatrix} \right)$$

where $|\alpha_\delta| < 1$. Then the expression $\zeta^T \Sigma_i(\zeta + \delta)$ is

$$\zeta^T \Sigma_i(\zeta + \delta) = \frac{\alpha_\zeta l_i}{2} \varphi(\theta, \alpha_\zeta, \chi, \alpha_\delta)$$

where

$$\varphi(\theta, \alpha_\zeta, \chi, \alpha_\delta) = \begin{bmatrix} c_\theta & s_\theta \end{bmatrix} \Sigma_i \left(\frac{\alpha_\zeta l_i}{2} \begin{bmatrix} c_\theta \\ s_\theta \end{bmatrix} + \frac{\alpha_\delta l_i}{2} \left(c_\chi \begin{bmatrix} c_\theta \\ s_\theta \end{bmatrix} + s_\chi \begin{bmatrix} s_\theta \\ -c_\theta \end{bmatrix} \right) \right)$$

To prove $\zeta^T \Sigma_i(\zeta + \delta) > 0$ is equivalent to proving $\varphi(\theta, \alpha_\zeta, \chi, \alpha_\delta) > 0$. For the case $0 \le \theta < \pi/2$ and the non-decreasing property (b), the following inequality can be

obtained

$$\varphi(\theta,\alpha_\zeta,\chi,\alpha_\delta) \geq \begin{bmatrix} c_\theta & s_\theta \end{bmatrix} \Sigma_i \left(\frac{l_i}{2} \begin{bmatrix} c_\theta \\ s_\theta \end{bmatrix} + \frac{\alpha_\delta l_i}{2} \left(c_\chi \begin{bmatrix} c_\theta \\ s_\theta \end{bmatrix} + s_\chi \begin{bmatrix} s_\theta \\ -c_\theta \end{bmatrix} \right) \right)$$
$$= \begin{bmatrix} c_\theta & s_\theta \end{bmatrix} \Sigma_i \left(\frac{l_i}{2} \begin{bmatrix} c_\theta + \alpha_\delta \left(c_\chi c_\theta + s_\chi s_\theta \right) \\ s_\theta + \alpha_\delta \left(c_\chi s_\theta - s_\chi c_\theta \right) \end{bmatrix} \right)$$

Since $|\alpha_\delta| < 1$, it follows that $c_\theta + \alpha_\delta \left(c_\chi c_\theta + s_\chi s_\theta \right) \leq 2$ and $s_\theta + \alpha_\delta \left(c_\chi s_\theta - s_\chi c_\theta \right) \leq 2$. It can be concluded that each element of Σ_i lies in its linear region and

$$\varphi(\theta,\alpha_\zeta,\chi,\alpha_\delta) \geq \frac{l_i}{2} \begin{bmatrix} c_\theta & s_\theta \end{bmatrix} \begin{bmatrix} c_\theta + \alpha_\delta \left(c_\chi c_\theta + s_\chi s_\theta \right) \\ s_\theta + \alpha_\delta \left(c_\chi s_\theta - s_\chi c_\theta \right) \end{bmatrix} = \frac{l_i \left(1 + \alpha_\delta c_\chi \right)}{2}$$

Since $|\alpha_\delta| < 1$, it can be shown that $\varphi(\theta,\alpha_\zeta,\chi,\alpha_\delta) > 0$. For the case $\pi/2 \leq \theta < 2\pi$, the procedure presented above can be used to show $\varphi(\theta,\alpha_\zeta,\chi,\alpha_\delta) > 0$. This proves $\zeta^T \Sigma_i \left(\zeta + \delta \right) > 0$. □

It is remarked that the above theorem can be extended to the general case where $\Sigma_i : \mathbb{R}^n \to \mathbb{R}^n$ and

$$\Sigma_i(\zeta) = \left[\sigma_i(\zeta_1), \ldots, \sigma_i(\zeta_n) \right]^T$$

where $\zeta = \left[\zeta_1, \ldots, \zeta_n \right]^T \in \mathbb{R}^n$. The last saturation function introduced is $h : \mathbb{R}^2 \to \mathbb{R}^2$ which is defined as

$$h(u) = \frac{u}{1 + u^T u} \tag{6.1}$$

It can be easily seen that $\| h(\cdot) \| \leq 1$.

6.2 Controller Design

6.2.1 Height Control

From Chapter 4, the height subsystem is

$$\dot{s}_{hp} = -\frac{1}{X_3^{a*}} v_3^a \tag{6.2a}$$
$$\dot{v}_3^a = g - \frac{K_T u}{m} \tag{6.2b}$$

where u is defined in (4.23). Defining $b_1 = 1/X_3^{a*}, b_2 = K_T/m$, and $d_g = mg/K_T$, the height subsystem (6.2) becomes

$$\dot{s}_{hp} = -b_1 v_3^a \tag{6.3a}$$
$$\dot{v}_3^a = b_2 (d_g - u) \tag{6.3b}$$

Here b_1, b_2, and d_g are taken as unknown parameters, and d_g is assumed to lie in a known compact set

$$\Omega_{dg} = \{d_g \in \mathbb{R} : |d_g - d_{g0}| \leq d_{gm}\}$$

where d_{g0} is any initial estimate of d_g which ensures every element of Ω_{dg} is greater than zero. Defining

$$\delta_{h1} = s_{hp} - 1 \tag{6.4a}$$

$$\delta_{h2} = v_3^a - \sigma_1(\delta_{h1}) \tag{6.4b}$$

and using (6.3), the time derivatives of δ_{h1} and δ_{h2} are

$$\dot{\delta}_{h1} = -b_1(\sigma_1(\delta_{h1}) + \delta_{h2}) \tag{6.5a}$$

$$\dot{\delta}_{h2} = b_2(d_g - u) + b_1 \frac{d\sigma_1}{d\delta_{h1}}(\sigma_1(\delta_{h1}) + \delta_{h2}) \tag{6.5b}$$

The control law for subsystem (6.3) is taken as

$$u = \sigma_2(\delta_{h2}) + \hat{d}_g + d_{g0} \tag{6.6}$$

with $\hat{d}_g$ providing an estimate of $d_g - d_{g0}$ and updated from

$$\dot{\hat{d}}_g = \text{Proj}(\mu, \hat{d}_g) = \mu - \frac{\nu_1 \nu_2}{4\left(\epsilon^2 + 2\epsilon\left(d_{gm} - \epsilon\right)\right)^{n+1}\left(d_{gm} - \epsilon\right)^2} \bigtriangledown p_d(\hat{d}_g) \tag{6.7}$$

where $\mu = k_{dg}\delta_{h2}$, $p_d(\hat{d}_g) = \hat{d}_g^2 - \left(d_{gm} - \epsilon\right)^2$, $\bigtriangledown p_d(\hat{d}_g) = 2\hat{d}_g$, n is an non-negative integer,

$$\nu_1 = \begin{cases} (\hat{d}_g^2 - (d_{gm} - \epsilon)^2)^{n+1}, & \text{if } p_d(\hat{d}_g) > 0 \\ 0, & \text{otherwise} \end{cases}$$

$\nu_2 = \hat{d}_g\mu + \sqrt{\left(\hat{d}_g\mu\right)^2 + \delta^2}$, and $\epsilon, \delta > 0$ are positive constants. The function Proj$(\cdot)$ is a projection operator from [91] which is n times continuously differentiable and guarantees $\hat{d}_g$ falls in the compact set

$$\bar{\Omega}_{dg} = \left\{\hat{d}_g \in \mathbb{R} : |\hat{d}_g| \leq d_{gm}\right\} \tag{6.8}$$

Theorem 6.2. *The equilibrium* $[\delta_{h1}, \delta_{h2}]^T = 0$ *of the closed-loop system* (6.3) *with the input* (6.6) *and* (6.7) *is GAS if the control parameters* k_1, k_2, l_1, l_2, *and* M_1 *satisfy*

$$-\left(b_2k_2 - b_1k_1\right)l_2 + 2b_2d_{gm} + b_1k_1M_1 < 0 \tag{6.9a}$$

$$k_1l_1 > l_2 \tag{6.9b}$$

$$k_2l_2/2 > d_{gm} \tag{6.9c}$$

and the control signal u lies in the following compact set

$$\Omega_u = \{u \in \mathbb{R} : d_{g0} - d_{gm} - M_2 \le u \le d_{g0} + d_{gm} + M_2\}$$

Proof. Using the law (6.7) it can be shown that $|\tilde{d}_g| < 2d_{gm}$, where $\tilde{d}_g = d_g - \hat{d}_g - d_{g0}$. With the control law (6.6), dynamics (6.3b) becomes

$$\dot{v}_3^a = -b_2(\sigma_2(\delta_{h2}) - \tilde{d}_g)$$

When $\delta_{h2} \ge l_2$ and using (6.9c) it can be shown that $\sigma_2(\delta_{h2}) - \tilde{d}_g \ge k_2l_2 - 2d_{gm} > 0$. Hence, it follows that

$$v_3^a(t) \le v_3^a(0) - b_2(k_2l_2 - 2d_{gm})t \tag{6.10}$$

From the definition of δ_{h2} in (6.4b), inequality (6.10) can be written as

$$\delta_{h2} + \sigma_1(\delta_{h1}) \le \delta_{h2}(0) + \sigma_1(0) - b_2(k_2l_2 - d_m)t$$

Using property (d) of σ_i gives

$$\delta_{h2} \le \delta_{h2}(0) + 2M_1 - b_2(k_2l_2 - 2d_{gm})t$$

Hence, if

$$t > T_2 = \frac{\delta_{h2}(0) + 2M_1 - l_2}{b_2(k_2l_2 - 2d_{gm})}$$

then δ_{h2} enters the linear of σ_2. For the initial value $\delta_{h2}(0) = l_2$, the time derivative of δ_{h2} is

$$\dot{\delta}_{2h} = -b_2(k_2l_2 - \tilde{d}_g) + b_1\frac{\partial\sigma_1}{\partial\delta_1}(\sigma_1(\delta_1) + l_2)$$

If inequality (6.9a) is satisfied, it follows that $\dot{\delta}_{2h} < 0$. Similarly, it can be shown that when $\delta_{h2} < -l_2$, δ_{h2} will enter the linear region of σ_2 in finite time, and $\dot{\delta}_{h2} > 0$ when $\delta_{h2} = -l_2$ if inequality (6.9a) is satisfied. Hence, it follows that δ_{h2} will enter the linear region in a finite time T_{h2} and stay in this region for $t > T_{h2}$. Using (6.5a) and the fact that $\delta_{h2}(t) \le l_2, t > T_{h2}$, it is also easy to show that when inequality (6.9b) is satisfied, δ_{h1} will enter the linear region of σ_1 at a finite time T_{h1}. Hence, after a

finite time, the system (6.5) becomes

$$\dot{\delta}_{h1} = -b_1(k_1\delta_{h1} + \delta_{h2}) \tag{6.11a}$$

$$\dot{\delta}_{h2} = -b_2(k_2\delta_{h2} - \tilde{d}_g) + b_1k_1(k_1\delta_{h1} + \delta_{h2}) \tag{6.11b}$$

To show the stability of closed-loop system (6.11), the Lyapunov function candidate is chosen as

$$V_h = \frac{k_1^2}{2}\delta_{h1}^2 + \frac{1}{2}\delta_{h2}^2 + \frac{b_2}{2k_{dg}}\tilde{d}_g^2 \tag{6.12}$$

Using (6.11), the time derivative of (6.12) is

$$\begin{aligned}
\dot{V}_h &= k_1^2\delta_{h1}\left(-b_1\left(k_1\delta_{h1} + \delta_{h2}\right)\right) - \frac{b_2}{k_{dg}}\tilde{d}_g\dot{\tilde{d}}_g + \delta_{h2}\left(-b_2\left(k_2\delta_{h2} - \tilde{d}_g\right) + b_1k_1\left(k_1\delta_{h1} + \delta_{h2}\right)\right) \\
&= -b_1k_1^3\delta_{h1}^2 - b_1k_1^2\delta_{h1}\delta_{h2} - \frac{b_2}{k_{dg}}\tilde{d}_g\dot{\tilde{d}}_g + \left(-b_2k_2 + b_1k_1\right)\delta_{h2}^2 + b_1k_1^2\delta_{h1}\delta_{h2} + b_2\delta_{h2}\tilde{d}_g \\
&= -b_1k_1^3\delta_{h1}^2 - \left(b_2k_2 - b_1k_1\right)\delta_{h2}^2 - b_2\tilde{d}_g\left(\frac{1}{k_{dg}}\dot{\tilde{d}}_g - \delta_{h2}\right)
\end{aligned}$$

From [91] the projector operator has the property

$$\tilde{d}_g\text{Proj}(\mu, \hat{d}_g) \geq \tilde{d}_g\mu$$

Hence, it can be shown that

$$\begin{aligned}
\dot{V}_h &\leq -b_1k_1^3\delta_{h1}^2 - \left(b_2k_2 - b_1k_1\right)\delta_{h2}^2 - b_2\tilde{d}_g\left(\frac{1}{k_{dg}}k_{dg}\delta_{h2} - \delta_{h2}\right) \\
&= -b_1k_1^3\delta_{h1}^2 - \left(b_2k_2 - b_1k_1\right)\delta_{h2}^2
\end{aligned}$$

From the inequality (6.9a), it can be shown that

$$\left(b_2k_2 - b_1k_1\right) > \frac{2b_2d_{gm} + b_1k_1M_1}{l_2} > 0$$

Using the LaSalle-Yoshizawa theorem [50, Thm. 2.1] it has been shown the equilibrium $[\delta_{h1}, \delta_{h2}]^T = 0$ is GAS. □

6.2.2 Lateral and Yaw Motion Control

With the assumption of zero attitude measurement bias and Assumption 4.1, the lateral subsystem with point image moment feature is

$$\dot{s}_{lp} = -\frac{\lambda}{X_3^{a*}}v_l^a - Ss_{lp}\dot{\psi} \tag{6.13a}$$

$$\dot{v}_l^a = \frac{K_Tu}{m}S\eta_1 - Sv_l^a\dot{\psi} \tag{6.13b}$$

where S is defined in (4.2), $v_l^a = [v_1^a, v_2^a]^T$, and $\eta_1 = [\phi, \theta]^T$. Choosing $\bar{u} = uS\eta_1$ as the input, the lateral system (6.13) becomes

$$\dot{s}_{lp} = -\lambda b_1 v_l^a - S s_{lp} \dot{\psi} \tag{6.14a}$$

$$\dot{v}_l^a = b_2 \bar{u} - S v_l^a \dot{\psi} \tag{6.14b}$$

For the lateral subsystem (6.14) the control law is chosen as

$$\bar{u} = -\Sigma_3 \left(v_l^a - k_4 h(\beta s_{lp}) \right) \tag{6.15}$$

where k_4 and β are positive constants. The derivative of $h(\beta s_{lp})$ at zero is proportional to β. The parameter β is used to tune the rate of closed-loop convergence. From Chapter 4, the kinematics of image feature for yaw is

$$\dot{s}_{\psi p} = -\dot{\psi} \tag{6.16}$$

The controller in (4.26), i.e.,

$$\dot{\psi} = K_\psi s_{\psi p} \tag{6.17}$$

is adopted, and the closed-loop becomes

$$\dot{s}_{\psi p} = -K_\psi s_{\psi p} \tag{6.18}$$

Theorem 6.3. *The equilibrium* $[s_{lp}, v_l^a, s_{\psi p}]^T = 0$ *of the closed-loop system* (6.13), (6.16) *with the input* (6.15) *and* (6.17) *is GAS if the control gains satisfies*

$$k_4 < l_3/2 \tag{6.19}$$

K_ψ, $k_3 > 0$, *and the magnitude of each element of the control input is less than* M_3.

Proof. The first step is to show the saturation function Σ_3 will enter its linear region in finite time. To prove this, a energy storage function is introduced as

$$V_{l2} = \frac{1}{2} \left(v_l^a \right)^T v_l^a$$

Using the control law (6.15), the time derivative of V_{l2} is

$$\begin{aligned} \dot{V}_{l2} &= \left(v_l^a \right)^T \left(-b_2 \Sigma_3 \left(v_l^a - k_4 h(\beta s_{lp}) \right) - S v_l^a \dot{\psi} \right) \\ &= -b_2 \left(v_l^a \right)^T \Sigma_3 \left(v_l^a - k_4 h(\beta s_{lp}) \right) \end{aligned}$$

From Theorem 6.1, it can be concluded that $\dot{V}_{l2} < 0$ when $\| v_l^a \| > l_2/2$ and (6.19) is satisfied. Hence, the saturation function Σ_3 enters its linear region in a finite time.

The closed-loop system becomes

$$\dot{s}_{lp} = -\lambda b_1 v_l^a - S s_{lp} \dot{\psi} \tag{6.20a}$$

$$\dot{v}_l^a = b_2 \left(-k_3 v_l^a + k_3 k_4 h \left(\beta s_{lp} \right) \right) - S v_l^a \dot{\psi} \tag{6.20b}$$

The Lypunov function candidate is chosen as

$$V_l = \frac{1}{\lambda b_1 \beta} \left(\sqrt{1 + \beta^2 s_{lp}^T s_{lp}} - 1 \right) + \frac{1}{2 b_2 k_3 k_4} (v_l^a)^T v_l^a$$

The time derivative of V_l is

$$\begin{aligned}
\dot{V}_l &= \frac{\beta^2}{\lambda b_1 \beta} \frac{s_{lp}^T \dot{s}_{lp}}{\sqrt{1 + \beta^2 s_{lp}^T s_{lp}}} + \frac{1}{b_2 k_3} (v_l^a)^T \dot{v}_l^a \\
&= \frac{\beta}{\lambda b_1} \frac{s_{lp}^T \left(-\lambda b_1 v_l^a - S s_{lp} \dot{\psi} \right)}{\sqrt{1 + \beta^2 s_{lp}^T s_{lp}}} \\
&\quad + \frac{1}{b_2 k_3 k_4} (v_l^a)^T \left(b_2 \left(-k_3 v_l^a + k_3 k_4 h \left(\beta s_{lp} \right) \right) - S v_l^a \dot{\psi} \right) \\
&= -\frac{\beta s_{lp}^T v_l^a}{\sqrt{1 + \beta^2 s_{lp}^T s_{lp}}} - \frac{1}{k_4} (v_l^a)^T v_l^a + (v_l^a)^T h(\beta s_{lp}) \\
&= -\frac{1}{k_4} (v_l^a)^T v_l^a
\end{aligned} \tag{6.21}$$

The Lyapunov function candidate for both lateral motion and yaw motion is chosen as

$$V_s = V_l + \frac{1}{2} s_{\psi p}^2 \tag{6.22}$$

Using (6.18) and (6.21), the time derivative of V_s is written as

$$\dot{V}_s = -\frac{1}{k_4} (v_l^a)^T v_l^a - K_\psi s_{\psi p}^2$$

Hence, using LaSalle's Invariance Theorem the equilibrium $\left[s_{lp}, v_l^a, s_{\psi p} \right]^T = 0$ of the closed-loop system is GAS. □

6.3 Simulation and Experimental Results

In this section simulation and experimental results are provided to validate the proposed DIBVS with input saturation.

6.3.1 Simulation Results

As in Section 5.1.2, the system model parameters used in the simulation are in Table 2.1, the image resolution is 320×200, and the thrust coefficient K_T is chosen as $K_T = 4 \times 10^7$ N/s^2. The initial displacement of the vehicle in the navigation frame is $[0.55, -0.55, -2.3]^T$ m with zero roll and pitch, and the initial value of yaw is $\pi/3$ rad. The initial velocities are set as $v^2(0) = [-0.4, -0.4, 0.2]^T$ and $\omega^c(0) = 0$. The visual target consists of two points located at coordinates $[-0.18, 0, 0]^T$ m and $[0.18, 0, 0]^T$ m in $\mathcal{N}$. The desired value of the image feature $s = [s_{lp}, s_{hp}, s_{\psi p}]^T$ is chosen as $s^* = [0, 0, 1, 0]^T$. The image feature error is $e_s = [e_{s1}, e_{s2}, e_{s3}, e_{s4}]^T = s - s^*$. The desired 3D translational displacement of the UAV is $d_{nc}^{n*} = [0, 0, -1.5]^T$ m which corresponds to $e_s = 0$. A high gain PID controller is used to track reference attitude.

The saturation function σ_i is chosen as

$$\sigma_i(x) = \begin{cases} k_i x, & \text{if } |x| < l_i \\ \text{sign}(x)\left(k_i l_i + c_{1i} \arctan\left(c_{2i}\left(|x| - l_i\right)\right)\right), & \text{otherwise} \end{cases} \tag{6.23}$$

where $c_{1i} = 2(M_i - k_i l_i)/\pi$, $c_{2i} = k_i / c_{1i}$. Figure 6.2 shows an example of the graph for σ_i. The controller parameters for σ_i, $i = 1, 2, 3$ are given in Table 6.1. The

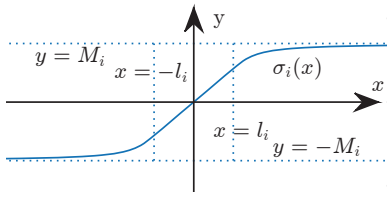


Figure 6.2: Saturation function σ_i.

Gains	k_i	l_i	M_i
$i = 1$	1	0.5	0.55
$i = 2$	0.4	0.42	0.17
$i = 3$	0.5	0.16	0.1

Table 6.1: Parameters for saturation functions σ_i, $i = 1, 2, 3$.

other control parameters are selected as $k_4 = 0.06$, $k_{dg} = 0.04$, $K_\psi = 0.1$, and the parameters in (6.7) are chosen as $n = 1$, $\epsilon = 0.001$, $\delta = 0.01$. From the value of d_{nc}^{n*} it follows that $X_3^{a*} = 1.5$ m. Hence, the values of $b_1 = 1/X_3^{a*} = 0.5$ and $b_2 = K_T/m = 25$ can be computed. From the values of K_T and m it follows that $d_g = 0.3924$. The initial estimate of $\hat{d}_g$ is $d_{g0} = 0.38$, and the estimated value $\hat{d}_g$ is limited in the compact set $\bar{\Omega}_{dg}$ defined in (6.8) with $d_{gm} = 0.04$. It can be easily verified that the control gains chosen above satisfy the constraints (6.9) and (6.19).

To verify the input saturation helps to keep the visual target in the FoV of the camera, two cases are considered in simulation. The first case the input saturation controllers (6.6) and (6.15) are applied. In the second case, all saturation functions in (6.6) and (6.15) are removed. This leads to the control being the linear region expression of (6.6):

$$u = k_2\delta_{h2l} + d_{g0} + \hat{d}_{gl} \tag{6.24}$$

where $\delta_{h2l} = v_3^a - k_1\delta_{h1}$, and $\hat{d}_{gl}$ is updated by

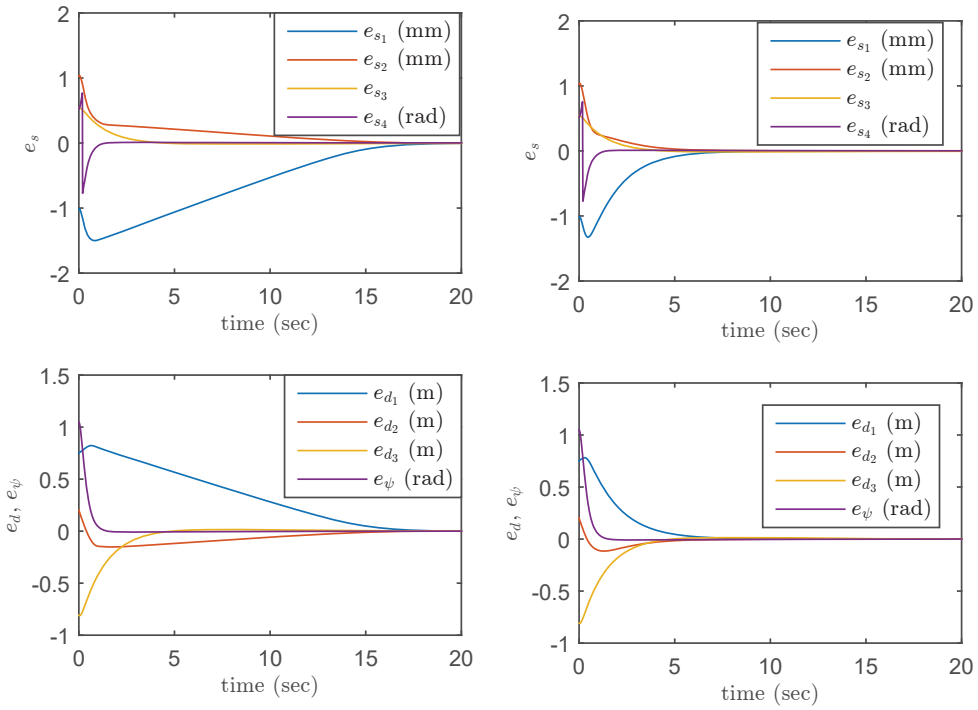
$$\dot{\hat{d}}_{gl} = k_{dg}\delta_{h2l} \tag{6.25}$$

It can be easily verified that the origin of the closed-loop system (6.3) with input (6.24) and (6.25) is GES. To remove the saturation function from control (6.15), the expression is linearised at the origin to obtain

$$\bar{u} = -k_3(v_l^a - k_4\beta s_{lp}) \tag{6.26}$$

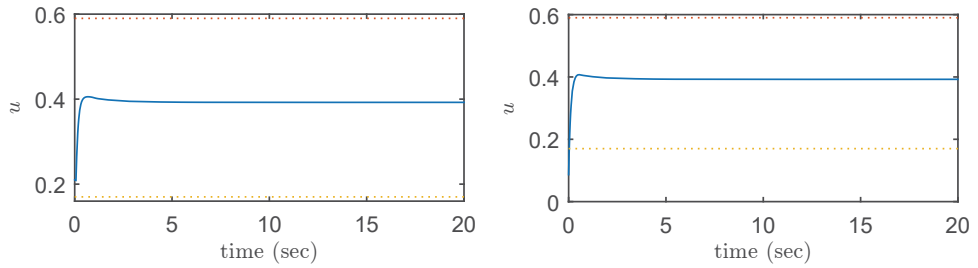
It can be easily proven that the origin of the closed-loop (6.13) with input (6.26) is GES.

Figure 6.3 shows the trajectories of e_s, and $e_d = d_{nc}^n - d_{nc}^{n*}$ for both cases. In both cases the image feature error and position error converge to zero in finite time. The difference between the two cases is that e_s and e_d asymptotically converge to zero with controller (6.6) and(6.15), whereas they exponentially converge to zero using controller (6.24) and (6.26). As expected, the control signal u lies in $\Omega_u = \{u \in \mathbb{R} : 0.17 \leq u \leq 0.59\}$ using the controllers (6.6) and (6.15). Figure 6.4a shows the trajectory of the control signal u where its values lie in Ω_u. The dotted line in Figure 6.4a denotes the boundary of Ω_u. Figure 6.4b shows the trajectory of u with controller (6.24) and (6.26). Here it can be seen u leaves Ω_u. Figure 6.5 shows the trajectories of the reference input η_1 for the inner loop for both cases. It can be seen that the input signal, i.e., the roll and pitch fall in the compact sets $\Omega_\phi = \{\phi \in \mathbb{R} : |\phi| \leq 0.1\}$ and $\Omega_\theta = \{\theta \in \mathbb{R} : |\theta| \leq 0.1\}$ when the proposed input saturation law is used. The reference roll and pitch signal leave the compact set Ω_ϕ and Ω_θ when (6.24) and (6.26) are applied. Figure 6.6 shows the image trajectory for the two cases. The rectangle in Figure 6.6 represents the boundary of the camera's image plane. From the figure, when input saturation is applied the visual target is kept in the FoV of the camera. The image of one of the two points leaves the FoV of the camera when the input saturation is off.



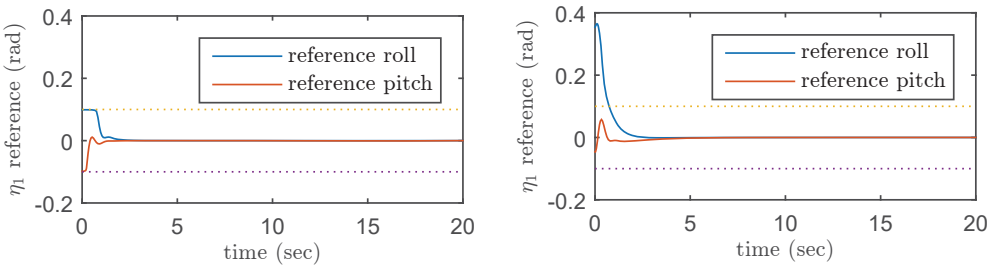
(a) Using control law (6.6) and (6.15) (b) Using control law (6.24) and (6.26)

Figure 6.3: Simulated trajectories of image feature error e_s, vehicle's 3D position error e_d, and e_ψ.



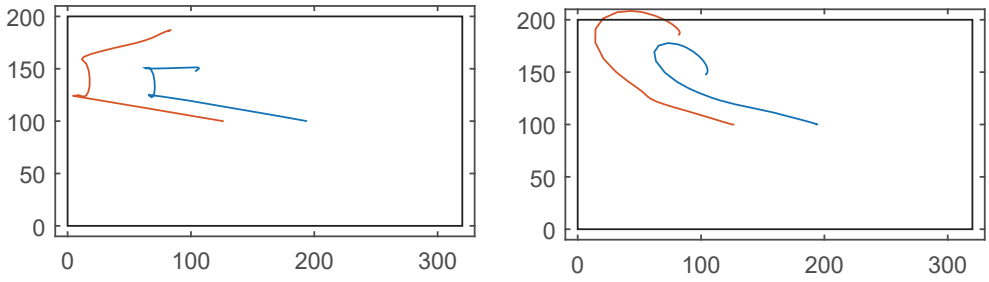
(a) Using control law (6.6) and (6.15) (b) Using control law (6.24) and (6.26)

Figure 6.4: Simulated trajectories of input u.



(a) Using control law (6.6) and (6.15) (b) Using control law (6.24) and (6.26)

Figure 6.5: Simulated trajectories of the input signals for the inner loop.



(a) Using control law (6.6) and (6.15) (b) Using control law (6.24) and (6.26)

Figure 6.6: Simulated image plane trajectories of the two points.

6.3.2 Experimental Results

In this experiment two visual points are put on the horizontal ground with a displacement of 0.4 m[1]. The saturation function σ_i used in the experiment is (6.23). The control parameters k_i, l_i, M_i, $i = 1, 2, 3$, k_{dg}, k_4, K_ψ, ϵ, δ, d_{gm} are chosen as in Section 6.3.1. The inner loop attitude control used is the same PID controller in Section 4.4.1.

Figure 6.7 shows the trajectory of image feature error e_s. Figure 6.9 gives the 3D trajectory of the vehicle's CoM. The corresponding 3D trajectories of translational error $e_d = [e_{d_1}, e_{d_2}, e_{d_3}]^T = d_{nc}^n - d_{nc}^{n*}$ and yaw error $e_\psi = \psi - \psi^*$ are shown in Figure 6.8. The definitions of d_{nc}^{n*} and ψ^* are in Section 4.4.1, and their values are

$$d_{nc}^{n*} = [0.077, -0.078, -1.140]^T \text{m}$$

and $\psi^* = 1.157$ rad. It can be seen that all error signals reach steady state after 10 seconds. The average and standard deviation of e_s, e_d, and e_ψ in steady state are in Table 6.2. With the saturation controller proposed in Section 6.2, the regulation errors are significantly reduced. Errors e_{s_i}, e_{d_i}, $i = 1, 2$ converge to non-zero constants because the attitude measurement bias is not compensated. This bias was accounted for in Chapter 4 and 5. However, steady state performance is significantly improved relative to the PD control in Section 4.4 and the non-adaptive controller in Section 5.1.3. This is because the saturation controller proposed in this chapter allows for high gains when the error approaches zero. This high gain reduces the average steady state error. The control output trajectories for the outer loop are shown in Figure 6.10. As expected the control u remains in Ω_u and reference roll and pitch lie in Ω_ϕ and Ω_θ, respectively. Figure 6.11 shows the trajectories of the points in the image plane. The square denotes initial image point and the circle denotes the final point. Evidently the trajectories remains in the camera's FoV.

[1] A video of the DIBVS with input saturation is at www.youtube.com/watch?v=X8ws1oHJdcA

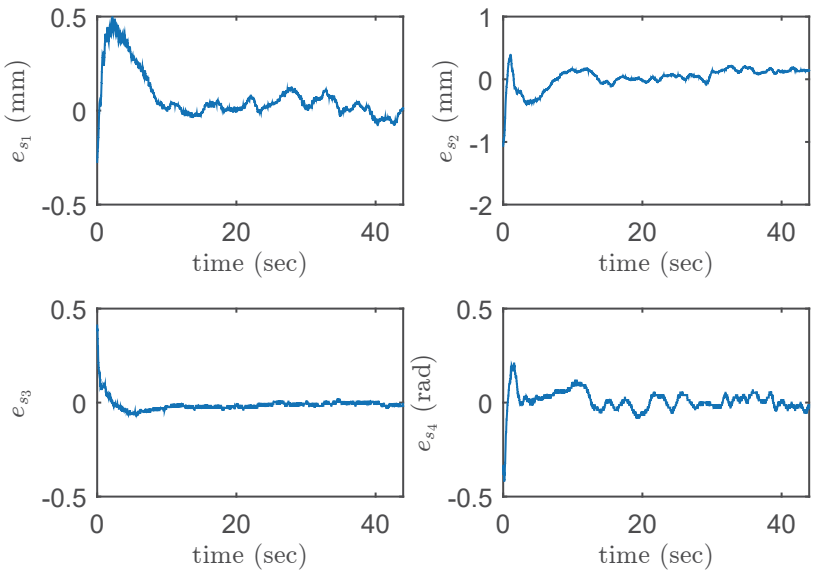


Figure 6.7: Experimental results: trajectories of image feature error e_s.

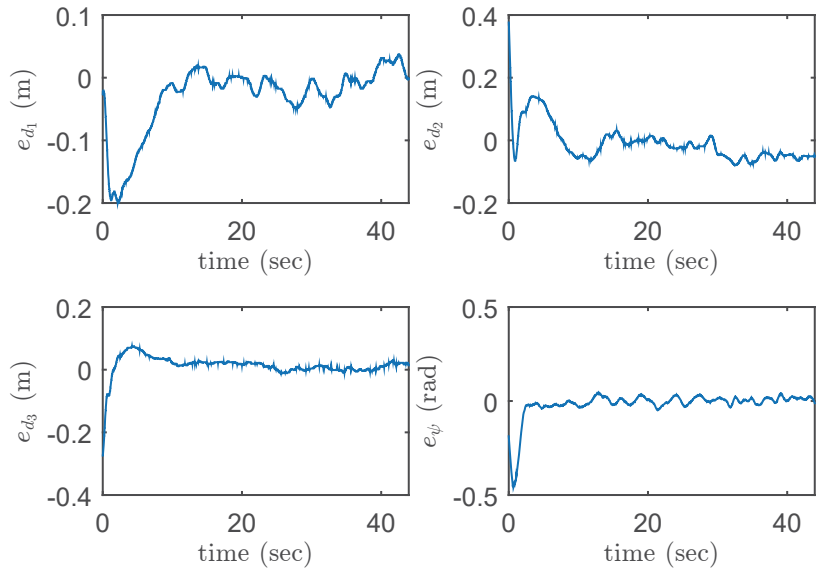


Figure 6.8: Experimental results: trajectories of vehicle's 3D position error e_d and yaw error e_ψ.

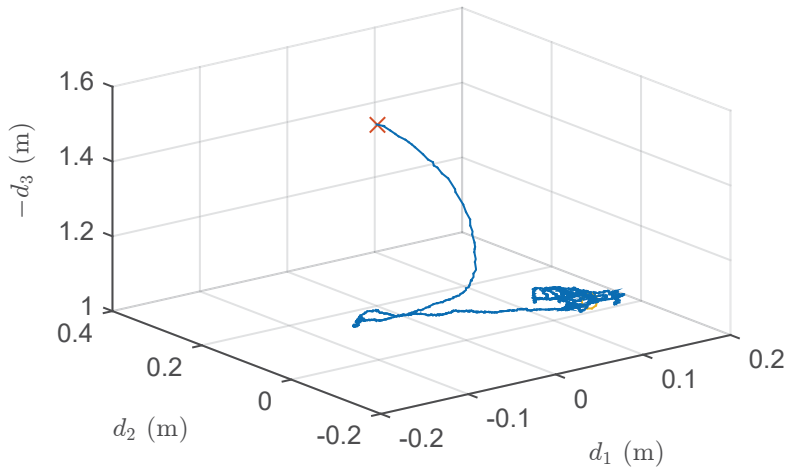


Figure 6.9: Experimental results: trajectories of vehicle's 3D position d_{nc}^{c}.

Image Feature	Mean	Standard deviation
e_{s_1} (mm)	0.02	0.04
e_{s_2} (mm)	0.07	0.08
e_{s_3}	-0.01	0.02
e_{s_4} (rad)	0.00	0.03
e_{d_1} (m)	-0.008	0.020
e_{d_2} (m)	-0.028	0.028
e_{d_3} (m)	0.010	0.011
e_{ψ} (rad)	0.003	0.017

Table 6.2: Statistics of e_s, e_d , and e_{ψ}.

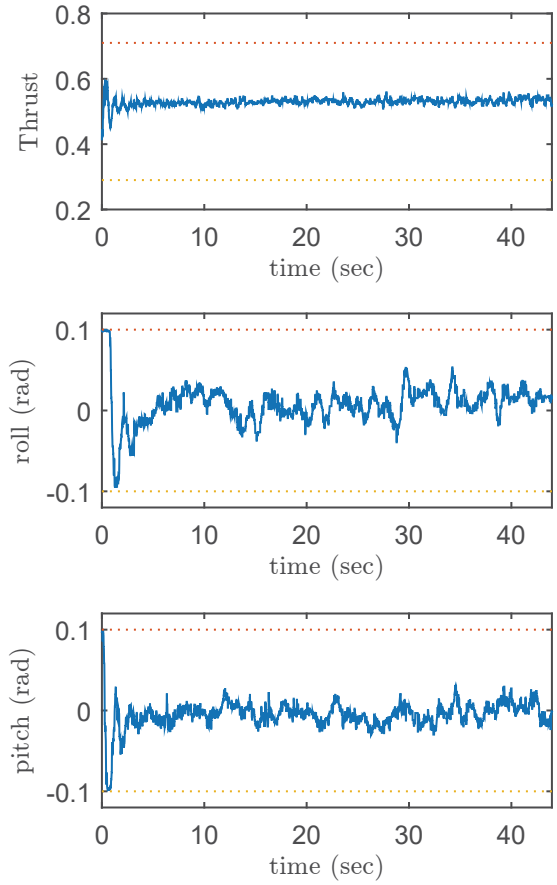


Figure 6.10: Experimental results: trajectories of outer loop controller outputs.

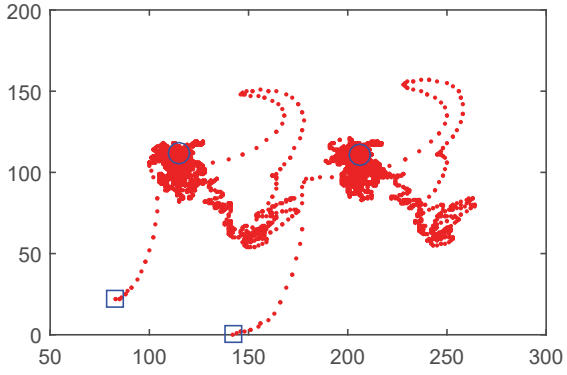


Figure 6.11: Experimental results: Trajectories of the points in the image plane. The square denotes initial image point and the circle denotes the final point.

6.4 Conclusion

Two DIBVS with input saturation were proposed to keep the visual target in the camera's FoV and keep thrust positive. This objective is met indirectly by designing an input saturation control for the lateral subsystem. This ensures reference roll and pitch are bounded. By avoiding large roll and pitch when the image feature error is large, the visual target can be kept in the camera's FoV. A unique feature of the input saturation controllers proposed is that they allow high gains when the image feature error approaches zero. This helps to reduce the steady state error. Simulation and experimental results demonstrated the performance of the proposed control.

Chapter 7

Summary and Future work

7.1 Summary of Results

This thesis has proposed new IBVS theory for rotary wing UAVs. IBVS is a challenging control problem given the nonlinearity of the system model, the underactuated UAV dynamics, various system uncertainties, and a complex embedded experimental platform. An important aspect of the work involved experimental validation of the controller designs on the ANCL quadrotor platform.

This thesis contributed to the ANCL quadrotor platform itself. A description of this platform and the contributions made are in Chapter 2. The platform development is a large collaborative effort together with other researchers at ANCL and the communities which support the PX4 autopilot, CMUcam5 Pixy computer vision system, and QGroundControl ground station software. These components are open-source (hardware and/or software) projects. Their open nature makes them particularly suitable for the development of research projects in general and the development of IBVS as described in this thesis in particular. The work in this thesis involved the modification of the position and attitude control modules so they could be integrated into an indoor flight environment. Additional modules were added to the PX4 autopilot. These include a communication module between Pixy and PX4, an image moment feature extraction module, and IBVS controller modules. The controller designs in Chapter 3–6 are model-based. This has the benefit of allowing their performance and robustness to be analyzed rigorously. Chapter 2 presents a DIBVS model whose inputs are physical signals, i.e., PWM inputs to the ESCs. Certain assumptions have be chosen in the modelling whose validity is based on our experimental testing. The models presented are sufficiently simplified so they can be used for controller design, and yet still capture important dominant effects. For example, they retain important nonlinear dependence on state.

A state transformation approach is proposed in Chapter 3. This state transformation removes the time derivative of the input, i.e., the Euler angle rate in the outer

loop. The reason for removing this dependence is to simplify the system structure for control design. A constructive method based on solving first-order homogeneous PDEs is provided to compute the state transformation. The state transformation approach generalizes the virtual camera approach which has recently appeared in the literature. A particular choice of state transformation which does not coincide with the virtual camera method is provided in Section 3.3 to regulate the lateral motion of a quadrotor.

Using another particular solution of the state-transformation approach in Chapter 4, i.e., the image coordinates of points or parameters of lines in a virtual camera which remains parallel to the target plane, image moment features for planar targets with a set of points or parallel lines are proposed. The image moment feature kinematics is given and used to derive a PID DIBVS. This control accounts for uncertainties including attitude estimate bias, unknown thrust gain, mass, and image depth. Experimental results demonstrate the controller's robustness properties.

An alternative way to account for parametric uncertainty is to use an adaptive control design. Chapter 5 first applies the adaptive backstepping design to visual servoing of planar targets with multiple points. Both simulation and experimental results show its robustness to unknown attitude estimate biases, thrust gain, mass, and image depth. Using the same adaptive backstepping technique, another DIBVS law is proposed for planar targets with parallel lines. This control adopts a different cascade structure where the input of the outer loop is angular velocity, and the inner loop tracks a reference angular velocity provided by the outer loop. The proposed IBVS control for parallel lines does not require a depth estimate and is robust to camera calibration errors.

The control gains for the DIBVS laws mentioned above have to be chosen small enough to avoid large values of attitude. Normally this keeps the visual target in the camera's FoV. Alternatively, an input saturation DIBVS law is proposed in Chapter 6. This IBVS law consists of two nested saturation controllers. The first input saturation controller is developed for the height subsystem and guarantees positive thrust. The second controller constrains the reference roll and pitch in a compact set. This ensures bounded reference roll and pitch which helps to keep the visual target in the FoV of the camera. The design treats thrust gain, mass, and image depth as unknowns. Simulation and experimental results show the controller can keep the visual target in the FoV, provide positive thrust, and is robust to parametric uncertainty.

To compare the performance of three approaches, i.e., PID, adaptive, and input saturated DIBVS; the mean values of steady state errors in experiment are presented. As well, the performance of a PD DIBVS is included. A target consisting of multiple points is considered, and the results are given in Table 7.1. As can be

Image Feature	Input saturation	Adaptive	PID	PD
e_{s_1} (mm)	0.02	0.00	-0.01	-0.10
e_{s_2} (mm)	0.07	-0.01	0.00	-0.40
e_{s_3}	-0.01	0.00	0.00	-0.27
e_{s_4} (rad)	0.00	0.02	0.03	0.05
e_{d_1} (m)	-0.008	-0.008	0.003	0.057
e_{d_2} (m)	-0.028	-0.002	0.005	0.179
e_{d_3} (m)	0.010	0.004	-0.008	0.451
e_{ψ} (rad)	0.003	0.070	-0.109	-0.026

Table 7.1: Experimental values of mean steady state error for four IBVS laws.

seen, the adaptive IBVS and PID IBVS have the best performance as they directly account for attitude bias. The steady state error using DIBVS with input saturation approach is significantly reduced as compared to PD IBVS and slightly worse than PID IBVS and adaptive IBVS. This is because the input saturation control provides high control gains near the origin. These high gains lead to reduced steady state error. Among the proposed approaches, only in the adaptive IBVS design is the stability of the lateral closed-loop subsystem independent of yaw motion. This allows us to track time-varying yaw trajectories.

7.2 Future Work

In this section possible future extensions and applications of the research in this thesis are provided.

- Output feedback DIBVS. The IBVS law developed in this thesis relies on the measurement of the vehicle's linear velocity. These measurements can be obtained from GPS or a Vicon system. In some applications GPS or Vicon measurements are not available. Hence, an interesting future topic of research is to apply output feedback methods to DIBVS. Two output feedback methods show particular promise: the high-gain observer-based output feedback in [51], and the adaptive output feedback in [50, Chp. 7]. Preliminary work using a high-gain observer design computes a scaled linear velocity estimate by low pass filtering the derivative of the image feature measurement. From experimental results it can confirmed that the rate of change in the image feature is proportional to linear velocity. The high-gain observer-based approach is a practical choice because of its simple structure. The adaptive output backstepping method provides a GAS result with a complex controller structure [49]. A comparison of the performance of these output feedback methods would be useful.

- Although the state transformation approach generalizes the virtual camera

method, only in the PVTOL vehicle case was it possible to find a particular solution for the transformation which differed from the virtual camera. Future work would involve searching for other particular solutions of the PDEs given in Chapter 3. The solutions considered should transform the kinematics into a form which is simple enough for controller design.

- In order to simplify the kinematics of the image moment features and the time derivative of the norm of the image feature to decouple from the yaw motion, the target is assumed horizontal. Although simulation results shows the proposed approaches are robust to non-horizontal target planes, a proof of robustness is missing. Future work involves extending the theory to non-horizontal planar targets.

- The saturation control proposed in Chapter 6 does not consider attitude measurement bias, and this leads to non-zero steady state error. Although the input saturation controller can reduce the non-zero steady state error with high gains near the origin, it is impossible to completely remove it. Future work should extend the input saturation controller to compensate steady state tracking error for attitude measurement bias.

Bibliography

[1] 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, 2012.

[2] C. Whitworth, A. Duller, , D. Jones, and G. Earp, "Aerial video inspection of overhead power lines," *Power Engineering Journal*, vol. 15, no. 1, pp. 25–32, 2001.

[3] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*. New York, NY: McGraw-Hill, 2008.

[4] 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.

[5] I. F. M. Bernal, "On-board visual control algorithms for unmanned aerial vehicles," Ph.D. dissertation, Universidad Politécnica de Madrid, Madrid, Spain, 2011.

[6] S. M. Weiss, "Vision based navigation for micro helicopters," Ph.D. dissertation, Department of Mechanical and Process Engineering, Swiss Federal Institute of Technology, Zurich, Switzerland, 2012.

[7] 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. Boston, MA: Springer US, 2008, vol. 258, pp. 95–102.

[8] 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.

[9] J. Artieda, J. Sebastian, P. Campoy, J. Correa, I. Mondragn, C. Martnez, and M. Olivares, "Visual 3-D SLAM from UAVs," *Journal of Intelligent & Robotic Systems*, vol. 55, no. 4-5, pp. 299–321, 2009.

[10] F. Kendoul, K. Nonami, I. Fantoni, and R. Lozano, "An adaptive vision-based autopilot for mini flying machines guidance, navigation and control," *Autonomous Robots*, vol. 27, no. 3, pp. 165–188, 2009.

[11] F. Chaumette and S. Hutchinson, "Visual servo control part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[12] O. Shakernia, Y. Ma, T. J. Koo, and S. Sastry, "Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control," *Asian Journal of Control*, vol. 1, no. 3, pp. 128–145, 1999.

[13] E. Altug, J. Ostrowski, and C. Taylor, "Quadrotor control using dual camera visual feedback," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, vol. 3, Taipei, Taiwan, September 2003, pp. 4294–4299.

[14] A. D. Wu, E. N. Johnson, and A. A. Proctor, "Vision-aided inertial navigation for flight control," *Journal of Aerospace Computing, Information, and Communication*, vol. 2, no. 9, pp. 348–360, 2005.

[15] L. Mejias, S. Saripalli, P. Campoy, and G. S. Sukhatme, "Visual servoing of an autonomous helicopter in urban areas using feature tracking," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 185–199, 2006.

[16] S. Azrad, F. Kendoul, and K. Nonami, "Visual servoing of quadrotor micro-air vehicle using color-based tracking algorithm," *Journal of System and Dynamics*, vol. 4, no. 2, pp. 255–268, 2010.

[17] L. García Carrillo, E. Rondon, A. Sanchez, A. Dzul, and R. Lozano, "Stabilization and trajectory tracking of a quad-rotor using vision," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 103–118, 2011.

[18] 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*, Taipei, Taiwan, October 2010, pp. 628–633.

[19] 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, 2014.

[20] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular slam," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[21] F. Fraundorfer, L. Heng, D. Honegger, G. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor MAV," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, October 2012, pp. 4557–4564.

[22] B. Espiau, "Effect of camera calibration errors on visual servoing in robotics," in *Experimental Robotics III*, ser. Lecture Notes in Control and Information Sciences, T. Yoshikawa and F. Miyazaki, Eds. Berlin, Germany: Springer, 1994, vol. 200, pp. 182–192.

[23] S. Lieberknecht, S. Benhimane, P. Meier, and N. Navab, "A dataset and evaluation methodology for template-based tracking algorithms," in *Proceedings of the 8th IEEE International Symposium on Mixed and Augmented Reality*, Orlando, FL, October 2009, pp. 145–151.

[24] T. Dick, C. Perez, M. Jägersand, and A. Shademan, "Realtime registration-based tracking via approximate nearest neighbour search." in *Proceedings of Robotics Science and Systems*, Berlin, Germany, June 2013.

[25] P. Corke and M. Good, "Dynamic effects in high-performance visual servoing," in *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, May 1992, pp. 1838–1843.

[26] F. Chaumette and S. Hutchinson, "Visual servo control part II: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[27] H. Jabbari, 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.

[28] B. Godbolt, "Experimental nonlinear control of a helicopter unmanned aerial vehicle (UAV)," Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, 2013.

[29] 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.

[30] ——, "Image based visual servo control for a class of aerial robotic systems," *Automatica*, vol. 43, no. 11, pp. 1975–1983, 2007.

[31] R. Mahony and T. Hamel, "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.

[32] 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, October 2006, pp. 4304–4309.

[33] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 331–340, 2008.

[34] 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, 2009.

[35] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3D Vision: From Images to Geometric Models*. New York, NY: Springer-Verlag, 2003.

[36] 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 and the European Control Conference*, Seville, Spain, December 2005, pp. 6078–6084.

[37] S. Benhimane and E. Malis, "Homography-based 2D visual tracking and servoing," *The International Journal of Robotics Research*, vol. 26, no. 7, pp. 661–676, 2007.

[38] 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.

[39] ——, "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, 2013.

[40] R. Ozawa and F. Chaumette, "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.

[41] ——, "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.

[42] 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, May 2012, pp. 971–976.

[43] D. Lee, H. Lim, K. H., 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.

[44] R. Mahony, P. Corke, and T. Hamel, "Dynamic image-based visual servo control using centroid and optic flow features," *Journal of Dynamic systems, measurement and control*, vol. 130, p. 011005, 2007.

[45] F. L. Bras, R. Mahony, T. Hamel, and P. Binetti, "Dynamic image-based visual servo control for an aerial robot: Theory and experiments," *International Journal of Optomechatronics*, vol. 2, no. 3, pp. 296–325, 2008.

[46] B. Herissé, T. Hamel, R. Mahony, and F.-X. Russotto, "Landing a VTOL unmanned aerial vehicle on a moving platform using optical flow," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 77–89, 2012.

[47] R. Mebarki and B. Siciliano, "Velocity-free image-based control of unmanned aerial vehicles," in *Proceedings of the 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, July 2013, pp. 1522–1527.

[48] R. Mebarki, V. Lippiello, and B. Siciliano, "Nonlinear visual control of unmanned aerial vehicles in GPS-denied environments," *IEEE Transactions on Robotics*, 2015, accepted.

[49] F. Le Bras, T. Hamel, R. Mahony, and A. Treil, "Output feedback observation and control for visual servoing of VTOL UAVs," *International Journal of Robust and Nonlinear Control*, vol. 21, no. 9, pp. 1008–1030, 2011.

[50] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos, *Nonlinear and Adaptive Control Design*, 1st ed. New York, NY: John Wiley & Sons, 1995.

[51] H. Jabbari Asl, G. Oriolo, and H. Bolandi, "Output feedback image-based visual servoing control of an underactuated unmanned aerial vehicle," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 228, no. 7, pp. 435–448, 2014.

[52] A. Abdessameud and F. Janabi-Sharifi, "Image-based tracking control of VTOL unmanned aerial vehicles," *Automatica*, vol. 53, pp. 111–119, 2015.

[53] A. Abdessameud and A. Tayebi, "Global trajectory tracking control of VTOL-UAVs without linear velocity measurements," *Automatica*, vol. 46, no. 6, pp. 1053–1059, 2010.

[54] L. Burlion and H. de Plinval, "Keeping a ground point in the camera field of view of a landing UAV," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 5763–5768.

[55] L. Burlion, "A new saturation function to convert an output constraint into an input constraint," in *Proceedings of the 20th Mediterranean Conference on Control Automation*, July 2012, pp. 1217–1222.

[56] N. Metni and T. Hamel, "A UAV for bridge inspection: Visual servoing control law with orientation limits," *Automation in Construction*, vol. 17, no. 1, pp. 3–10, 2007.

[57] H. de Plinval and L. Burlion, "Nonlinear visual servoing control for VTOL UAVs with field of view constraint," in *Advances in Aerospace Guidance, Navigation and Control*, J. Bordeneuve-Guibé, A. Drouin, and C. Roos, Eds. Cham, Switzerland: Springer, 2015, pp. 531–548.

[58] G. Fink, H. Xie, A. F. Lynch, and M. Jagersand, "Nonlinear dynamic image-based visual servoing of a quadrotor," *Journal of Unmanned Vehicle Systems*, vol. 3, no. 1, pp. 1–21, 2015.

[59] H. Xie and A. Lynch, "State transformation based dynamic visual servoing for an unmanned aerial vehicle," *International Journal of Control*, 2015, to appear.

[60] H. Xie, A. Lynch, and M. Jagersand, "IBVS of a rotary wing UAV using line features," in *Proceedings of the 27th IEEE Canadian Conference on Electrical and Computer Engineering, 2014*, Toronto, ON, May 2014, pp. 1–6.

[61] H. Xie, A. F. Lynch, and M. Jagersand, "Dynamic IBVS of a rotary wing UAV using line features," *Robotica*, 2014, 18 pages, available on FirstView.

[62] 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, 2004.

[63] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1116–1127, 2005.

[64] G. Fink, H. Xie, A. F. Lynch, and M. Jagersand, "Experimental validation of dynamic visual servoing for a quadrotor using a virtual camera," in *Proceedings of the 2015 International Conference on Unmanned Aircraft Systems*, Denver, CO, June 2015, pp. 1231–1240.

[65] H. Xie, G. Fink, A. F. Lynch, and M. Jagersand, "Adaptive dynamic visual servoing of a UAV," *IEEE Transactions on Aerospace and Electronic Systems*, 2015, accepted.

[66] H. Xie and A. F. Lynch, "Dynamic image-based visual servoing with input saturation," *IEEE-ASME Transactions on Mechatronics*, 2015, submitted.

[67] G. Cai, J. Dias, and L. Seneviratne, "A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends," *Unmanned Systems*, vol. 2, no. 2, pp. 175–199, 2014.

[68] P. Castillo, R. Lozano, and A. E. Dzul, *Modelling and control of mini-flying machines*, M. J. Grimble and M. A. Johnson, Eds. London, UK: Springer-Verlag, 2005.

[69] R. Hartley and A. Zisserman, *Multiple View Geometry in computer vison*, 2nd ed. Cambridge, UK: Cambridge University Press, 2003.

[70] Pixy CMUcam5. Accessed 5 June 2015. [Online]. Available: http://www.cmucam.org/projects/cmucam5/wiki

[71] PX4 autopilot. Institute for Visual Computing, Swiss Federal Institute of Technology Zurich. Accessed 1 July 2014. [Online]. Available: https://pixhawk.org/choice

[72] L. Meier. QGroundControl. Institute for Visual Computing, Swiss Federal Institute of Technology Zurich. Accessed 1 May 2015. [Online]. Available: http://www.qgroundcontrol.org/

[73] Vicon Motoin Capture System. Accessed 5 June 2015. [Online]. Available: http://www.vicon.com/

[74] G. Nutt. NuttX real-time operating system. Accessed 4 November 2014. [Online]. Available: http://www.nuttx.org/

[75] 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.

[76] L. Meier. (2015) Mavlink: Micro air vehicle communication protocol. Accessed 5 May 2015. [Online]. Available: http://qgroundcontrol.org/mavlink/start

[77] 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, 1992.

[78] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL: CRC Press, 1994.

[79] M. D. Shuster, "A survey of attitude representations," *The Journal of the Astronautical Sciences*, vol. 41, pp. 439–517, 1993.

[80] S. P. Bhat and D. S. Bernstein, "A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon," *Systems & Control Letters*, vol. 39, no. 1, pp. 63–70, 2000.

[81] H. Lim, J. Park, D. Lee, and H. J. Kim, "Build your own quadrotor: Open-source projects on unmanned aerial vehicles," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 33–45, 2012.

[82] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, SC, August 2007.

[83] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, vol. 5, New Orleans, LA, April 2004, pp. 4393–4398.

[84] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," Ph.D. dissertation, Ecole Polytechnique Federale de Luasanne, Lausanne, Switzerland, 2007.

[85] F. Kendoul, I. Fantoni, and R. Lozano, "Asymptotic stability of hierarchical inner-outer loop-based flight controllers," in *Proceedings of the 17th World Congress The International Federation of Automatic Control*, Seoul, Korea, July 2008, pp. 1741–1746.

[86] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2001.

[87] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed," *Control Engineering Practice*, vol. 19, no. 9, pp. 1023–1036, 2011.

[88] J. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *IEEE Control Systems Magazine*, vol. 28, no. 2, pp. 51–64, 2008.

[89] R. Marino and P. Tomei, "Adaptive observers with arbitrary exponential rate of convergence for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 7, pp. 1300–1304, 1995.

[90] H. Khalil, "Adaptive output feedback control of nonlinear systems represented by input-output models," *IEEE Transactions on Automatic Control*, vol. 41, no. 2, pp. 177–188, 1996.

[91] Z. Cai, M. de Queiroz, and D. Dawson, "A sufficiently smooth projection operator," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 135–139, 2006.

[92] J. M. Lee, *Introduction to Smooth Manifolds*. New York, NY: Springer-Verlag, 2006.

[93] R. Marino and P. Tomei, *Nonlinear Control Design: Geometric, Adaptive, and Robust*. Hertfordshire, UK: Prentice Hall, 1995.

[94] E. Goursat, *A Course in Mathematical Analysis: Differential Equations: Volume Two, Part Two*. New York, NY: Dover, 1959.

Appendix A

Solving a System of first-order linear homogeneous PDEs

A system of $q < n$ linear homogeneous PDEs in an unknown function $f(x_1, x_2, ..., x_n)$ is defined as

$$G_i(f) = \sum_{j=0}^{n} a_{ji} \frac{\partial f}{\partial x_j} = 0, \quad 1 \leq i \leq q \tag{A.1}$$

where $G_i = (a_{1i}, \ldots, a_{ni})^T$ is a vector field with a_{ji} smooth functions of n independent variables $x_i, 1 \leq i \leq n$. According to the Frobenius Theorem, e.g. [92], there exists $n - q$ independent solutions of (A.1) if the q-dimensional distribution

$$\mathcal{D} = \text{span}\{G_1, G_2, ..., G_q\}$$

is involutive. A distribution $\mathcal{D}$ is said involutive if for any two vector fields $G_i, G_j \in \mathcal{D}$ their Lie bracket $[G_i, G_j] \in \mathcal{D}$. In the case where for any $G_i, G_j \in \mathcal{D}$ the Lie bracket $[G_i, G_j] = 0$, the system of PDEs is said complete. In this case there exist local coordinates $z_1, \ldots, z_n$ where vector fields are simultaneously rectified, i.e.,

$$G_i = \frac{\partial}{\partial z_i}, \quad 1 \leq i \leq q$$

This result is stated in [93, Thm. A.4.5].

It can be shown that the Lie brackets $[G_{1p}, G_{2p}]$ in (3.8) , $[G_{1v}, G_{2v}]$ in (3.10), and $[G_{1l}, G_{2l}]$ in (3.16) are equal to zero. Hence, distributions $\mathcal{D}_p = \text{span}\{G_{1p}, G_{2p}\}$, $\mathcal{D}_v = \text{span}\{G_{1v}, G_{2v}\}$, and $\mathcal{D}_l = \text{span}\{G_{1l}, G_{2l}\}$ are involutive. Hence, PDEs (3.8), (3.10), and (3.16) are solvable and a method for computing the general solution is given next. The approach is based off [94, p. 271].

It is assumed distribution $\mathcal{D}$ is q-dimensional and therefore $G_i, 1 \leq i \leq q$ in (A.1) are independent. Further, the system of PDEs (A.1) is assumed complete and so the Lie brackets $[G_i, G_j] = 0, 1 \leq i, j \leq q$. If $y_2, \ldots, y_n$ are independent integrals of

$G_k(f) = 0$, for some k, and a function y_1 is chosen such that $y_1, y_2, \ldots, y_n$ are new independent variables, $G_k(f) = 0$ for some k reduces to $\partial f/\partial y_1 = 0$. The system of PDEs expressed in the y-coordinates becomes

$$L_i(f) = \begin{cases} \dfrac{\partial f}{\partial y_1} = 0, & i = k \\ \displaystyle\sum_{j=2}^{n} b_{ji} \frac{\partial f}{\partial y_j} = 0, & i \neq k \end{cases} \tag{A.2}$$

where b_{ji} is function of $n-1$ independent variables $y_2, y_3, \ldots, y_n$. Hence, $L_i(f) = 0, i \neq k$, forms a complete system of $q-1$ PDEs in $n-1$ independent variables $y_2, y_3, \ldots, y_n$. By repeating this procedure, the system can be reduced to one PDE in $n-q+1$ independent variables. The general solution of this PDE will be an arbitrary smooth function of $n-q$ independent solutions.

The procedure is illustrated by computing a solution to (3.8) where $q = 2$ and $n = 4$ and

$$G_{jp}(y_i^a) = 0, \quad j = 1, 2 \tag{A.3}$$

where

$$G_{1p} = \frac{y_1 y_2}{\lambda} \frac{\partial}{\partial y_1} + \frac{y_2^2 + \lambda^2}{\lambda} \frac{\partial}{\partial y_2} + \frac{\partial}{\partial \phi}$$

$$G_{2p} = \left(\frac{y_1^2 + \lambda^2}{\lambda} c_\phi + y_2 s_\phi \right) \frac{\partial}{\partial y_1} + \left(\frac{y_1 y_2 c_\phi}{\lambda} - y_1 s_\phi \right) \frac{\partial}{\partial y_2} - \frac{\partial}{\partial \theta}$$

It is straightforward to show that the integral of $G_{1p}(y_i^a) = 0$ is an arbitrary smooth function of $\left(y_2 s_\phi + \lambda c_\phi\right)/y_1$, $\left(y_2 c_\phi - \lambda s_\phi\right)/y_1$, and θ. Taking new independent variables $y_{11} = \left(y_2 s_\phi + \lambda c_\phi\right)/y_1$, $y_{21} = \left(y_2 c_\phi - \lambda s_\phi\right)/y_1$, ϕ, θ. The function $y_i^a(y_1, y_2, \phi, \theta)$ is expressed in the new independent variables and denoted $y_{i1}^a(y_{11}, y_{21}, \phi, \theta)$, and (A.3) becomes

$$\frac{\partial y_{i1}^a}{\partial \phi} = 0 \tag{A.4a}$$

$$\frac{\partial y_{i1}^a}{\partial y_{11}} \left(y_{11}^2 + 1\right) + \frac{\partial y_{i1}^a}{\partial y_{21}} y_{11} y_{21} + \frac{\partial y_{i1}^a}{\partial \theta} = 0 \tag{A.4b}$$

Hence, it can be concluded that y_{i1}^a is not a function of ϕ. The PDE (A.4b) has three independent variables y_{11}, y_{21}, and θ. The general integral of (A.4b) is an arbitrary smooth function of $(s_\theta - y_{11} c_\theta)/y_{21}$ and $(c_\theta + y_{11} s_\theta)/y_{21}$. Hence, transformed to the original state y_1, y_2, ϕ, and θ, the general integral of (A.3) is

$$\kappa_2 \left(\frac{s_\phi c_\theta y_2 + \lambda c_\phi c_\theta - y_1 s_\theta}{\lambda s_\phi - y_2 c_\phi}, -\frac{s_\phi s_\theta y_2 + c_\phi s_\theta \lambda + y_1 c_\theta}{\lambda s_\phi - y_2 c_\phi} \right)$$