Precision Landing of a Quadrotor UAV on a Moving Target Using Low-cost Sensors

by

Kevin Ling

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

With the use of unmanned aerial vehicles (UAVs) becoming more widespread, a need for precise autonomous landings has arisen. In the maritime setting, precise autonomous landings will help to provide a safe way to recover UAVs deployed from a ship. On land, numerous applications have been proposed for UAV and unmanned ground vehicle (UGV) teams where autonomous docking is required so that the UGVs can either recover or service a UAV in the field. Current state of the art approaches to solving the problem rely on expensive inertial measurement sensors and RTK or differential GPS systems. However, such a solution is not practical for many UAV systems.

A framework to perform precision landings on a moving target using low-cost sensors is proposed in this thesis. Vision from a downward facing camera is used to track a target on the landing platform and generate high quality relative pose estimates. The landing procedure consists of three stages. First, a rendezvous stage commands the quadrotor on a path to intercept the target. A target acquisition stage then ensures that the quadrotor is tracking the landing target. Finally, visual measurements of the relative pose to the landing target are used in the target tracking stage where control and estimation are performed in a body-planar frame, without the use of GPS or magnetometer measurements. A comprehensive overview of the control and estimation required to realize the three stage landing approach is presented.

Critical parts of the landing framework were implemented on an AscTec Pelican testbed. The AprilTag visual fiducial system is chosen for use as the landing target. Implementation details to improve the AprilTag detection pipeline are presented. Simulated and experimental results validate key portions of the landing framework. The novel relative estimation scheme is evaluated in an indoor positioning system. Tracking and landing on a moving target is demonstrated in an indoor environment. Outdoor tests also validate the target tracking performance in the presence of wind.

Acknowledgements

Many thanks to my supervisor, Professor Steven Waslander, for providing me with the opportunity to study an interesting field. His insight and guidance have been invaluable over the past two years.

I would also like to thank my fellow lab mates at the Waterloo Autonomous Vehicles Lab: Arun Das, Neil Mathew, James Servos, Sirui Song, Adeel Ahktar, Nima Mohajerin, Dr. Michael Tribou, Michael Smart, Sid Ahuja, and Abdel El Bably. They have been great minds to bounce ideas off of and to share experience with.

Lastly, I would like to thank the coop students that have worked in our lab for all their assistance with field testing. In particular, I greatly appreciate the help of Derek Chow who took the research to heart and was instrumental to the early development of our flight control code.

Dedication

I dedicate this to my family and friends who have supported me and given me the opportunities to learn and grow as an individual.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

As quadrotor unmanned aerial vehicles (UAVs) have become more accessible to commercial, research, and military users, many applications have been pushing the operating limits of the vehicles and using them in increasingly difficult settings. For many scenarios, a large and stationary landing site is no longer guaranteed, which greatly increases the difficulty for autonomous landings. These scenarios can be generalized to two main areas: maritime landings and unmanned ground vehicle (UGV) docking. The former includes applications where UAVs are retrieved on a ship over water. In the latter, UAVs are teamed up with UGVs to perform more elaborate tasks that require a longer range or longer flight time than can be achieved by UAVs alone. This cooperation often requires the UAV to be able to land on the UGV autonomously while in the field. In both cases, a safe and reliable system for autonomous precision landing would be highly beneficial.

In 2012, researchers from the University of Waterloo deployed a hexacopter from a ship in order to place a GPS beacon on an iceberg, shown in Figure 1.1. The conditions throughout this deployment were near ideal for the ship but proved to be very challenging for the hexacopter. The waves were 1-2 m in height, winds were around 10-20 knots, and the boat was moving at a significant speed. In the process of recovering the hexacopter, the human pilot encountered difficulties with wind and as a result the hexacopter crashed and was no longer operational. This example serves to demonstrate why falling back to human controlled flight for these difficult landings is not sufficient. In this case, a highly skilled pilot was still unable to safely recover the UAV.

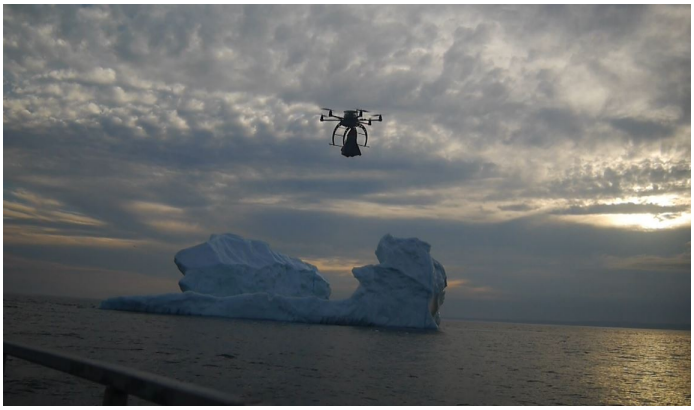


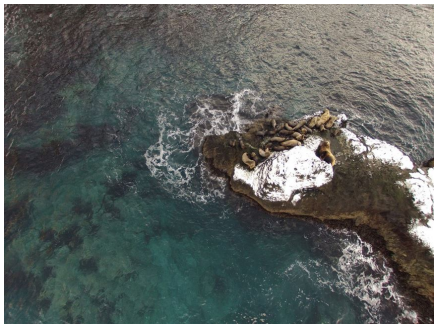
Figure 1.1: A hexacopter returns to a ship after dropping a GPS beacon on an iceberg.

A more successful example of maritime UAV deployment comes from researchers at the University of Alaska Fairbanks. An Aeryon Scout quadrotor system was flown from a ship to perform wildlife monitoring [28]. The Scout and a photo taken by it are shown in Figures 1.2a and 1.2b. After the surveying was complete, the UAV recovery was performed in a semi-autonomous manner. A human operator commanded the quadrotor to descend to about head-level on the ship while a second operator walked beneath the vehicle to grab the legs. The first operator then shuts off the motors. A number of other maritime applications have also been proposed including coastal surveillance [21] and search and rescue tasks.

While the semi-autonomous UAV recovery method has proven to be effective, that approach requires multiple skilled operators and greatly increases the likelihood of human injury. The requirement for skilled operators makes the use of UAVs prohibitively difficult for many groups. It also eliminates the possibility of more efficient robotic teams that do not require human operators. Maritime based UAV applications can benefit greatly from autonomous UAV landings for retrieval.

On land, researchers have proposed coordinated robotic teams comprised of UAVs and unmanned ground vehicles (UGVs). By teaming up UAVs with UGVs, long range tasks that go beyond the flight range of a UAV become feasible. These long range tasks include large-scale infrastructure inspection and search and rescue missions. Since UAVs have restrictive weight payloads, they often carry lower capacity batteries which limit their flight time. UGVs have been proposed as mobile charging stations to recharge UAVs without the need to return to the original launch site. This approach can also be applied to enable persistent monitoring tasks [14]. While works on coordinated UAV and UGV teams are

(a) An Aeryon Scout quadrotor in flight.



(b) Sea lions surveyed from a Scout quadrotor.

Figure 1.2: Researchers from the University of Alaska Fairbanks used an Aeryon Scout quadrotor deployed from a ship to monitor sea lion populations in the Aleutian islands [8].

still in their infancy, it is clear that precision landings of the UAVs on the moving UGVs will be necessary to facilitate recharging during long range missions.

Autonomous landing of a quadrotor UAV on a small moving area poses many challenges [26, 2]. Low-cost GPS units found on most quadrotor UAVs have errors as high as 5 m. Combined with the GPS errors on the landing target, the relative position estimate could be off by as much as 10 m if calculated using GPS alone. Attempting a landing with such low-quality estimates will have a very high probability of failure.

Magnetometers on UAVs are also commonly problematic. They are subject to a barrage of electrical noise from the motors and computers of the vehicle, resulting in large heading errors which are further aggravated by operation in polar regions of the Earth. Poor knowledge of the vehicle's heading prohibits the generation of purposeful control commands required for accurate inertial positioning and landing.

In the maritime deployment scenario, the movements of the ship due to the waves adds an additional layer of difficulty to the precision landing problem. The movement of the ship is difficult to predict due to waves, ocean currents, and other disturbances. Unexpected ship movements can cause a UAV to touch down on the ship deck at an inopportune time where the ship is rising quickly or the attitude of the ship is not suitable for a landing. Low-quality pose measurements and a moving landing surface make autonomous precision landing a difficult task to perform.

Fortunately, many popular applications for UAVs, such as surveillance, search, and mapping missions, inherently require the inclusion of a camera on the UAV platform. The camera can be used to generate accurate relative pose estimates between the UAV and the landing area. The relative positioning error can be further improved through the use of

visual targets mounted on the landing site, which are easily identified and tracked using the camera. In this work, a downward facing camera mounted on a UAV is leveraged to track a visual target and produce high quality position estimates. The high-quality vision based estimates are then used to perform a precision landing on a moving target.

1.2 Related Work

Vision Based Landing

Early on, researchers were interested in the idea of extracting accurate positioning information from vision data. The method through which visual data are used has evolved over time from simple colour tracking to a vision based simultaneous localization and mapping approach. Shakernia *et al.* propose a vision based approach extending the ego motion problem of tracking features in an unstructured environment [23]. The authors extended existing camera pose estimation algorithms to work with coplanar points located on the landing pad. Saripalli *et al.* demonstrated successful landings on a target helipad using vision to augment the position estimates from DGPS [22]. The authors also present an image processing scheme to extract the position of the target helipad from a binary image, a high level state machine to drive the helicopter functions, and a behaviour based control scheme used to detect the landing target and maintain stability. They do not, however, consider the effects of wind in the inertial frame, and rely on a precise vehicle model for nonlinear control. A more recent work by Tribou *et al.* proposes a method through which a quadrotor could follow an arbitrary relative trajectory through position based servoing by tracking features on an object and generating a sparse map. Again, a precise vehicle model is required for the feedback linearizing controller, and the method was not demonstrated experimentally [25].

Ground Vehicle Docking

A large focus of precision landings is to facilitate docking with an UGV. Voos and Bou-Ammar proposed a 2D non-linear tracking controller to accomplish the task [27]. Their approach comes from classical missile guidance problems. They describe the dynamics of the two vehicle system with differential equations describing the distance between the two vehicles and the line-of-sight angle. Then they present a control law which is demonstrated to be asymptotically stable. As their focus is solely on the control aspect of the problem,

Voos and Bou-Ammar did not present any experimental results for their proposed controller. They suggest using a differential GPS for the majority of the approach and to improve the estimation quality with ultrasonic or vision based measurements for the final landing.

In some scenarios, it is feasible to have a UAV and UGV work in cooperation to facilitate the docking. A cooperative landing scheme was presented by Daly *et al.* [2]. In their approach, the two vehicles use a joint decentralized controller in order to reach a rendezvous point for landing. The authors also analyse stability of the joint controller in the presence of time delays and determine an upper bound on acceptable communication delays. Subsequent experiments [12] have successfully demonstrated their approach in both indoor and outdoor environments. A real-time kinematic (RTK) GPS provided sub centimetre level positioning accuracy for the outdoor experiment.

Kim *et al.* perform a landing on a moving target using visual colour detection and an adaptive path following controller [10]. The quadrotor testbed they used was equipped with a smartphone with a fish-eye lens attachment. All image processing and target detection was performed on the smartphone. An unscented Kalman filter takes the target position in the camera frame as inputs and estimates the horizontal distance from the quadrotor to the tag in the world frame. The solution by Kim *et al.* was developed around the same time as the solution that is presented in this work. While both approaches are similar, the solution by Kim *et al.* differs in many design choices including: target detection method, estimation framework, and controller design. Their estimation method focuses on the camera dynamics and the fish-eye lens model rather than quadrotor dynamics.

Maritime Landing

Moving to the maritime setting, the problem becomes a bit more difficult. Ships are subject to the movements of the waves and can move unpredictably. Researchers in the maritime setting have a larger focus on estimating the orientation of the ship deck in order to determine a safe time to land.

Boeing's Unmanned Little Bird (ULB) is perhaps the best demonstration of a successful ship landing by a rotary-wing UAV. They have demonstrated over 100 landings on a moving trailer platform and 16 landings on a ship in the ocean [6]. The system relies on high accuracy relative position and inertial orientation measurements from a NovAtel RTK GPS with an integrated IMU, a radar altimeter, and a ship fitted with a Landing Period Designator (LPD). The LPD measures the wave induced motion and predicts times where the ship's movement will be at a minimum. Additionally, a grate and anchor system was

implemented to ensure that the helicopter would latch onto the landing pad and eliminate the risk of tip over. Although effective, the solution exceeds the payload capacity and intended cost of the small UAVs under consideration. Furthermore, the requirements for an LPD and specialized landing grate are restrictive as they limit the types of ships a UAV can land on.

Garratt *et al.* take a different approach and tackle the issue of estimating deck attitude on a ship. In their work, they look at landing an autonomous helicopter on a large ship using a narrowband light beacon on the ship and a custom made lidar which scans a circle on the ship's deck [5]. The light beacon is detected by an optical camera on the helicopter and provides a bearing to the landing platform. The lidar, which forms their deck attitude sensing system (DASS), determines the relative orientation and range of the ship deck. The DASS performed very well in their experiments, however the added bulk of a custom lidar would be restrictive given the payload constraints of small UAVs.

Venugopalan *et al.* present a solution to autonomously land a Parrot AR Drone on a kayak [26]. The authors approached the problem by using the hover mode of the vehicle to stabilize the vehicle against disturbances. Images captured by a downward facing camera on the Parrot were streamed to an offboard computer to calculate the relative position of the landing pad and send the appropriate commands to close the gap between the quadrotor and kayak. Also presented is a novel manoeuvre where the quadrotor rolls and pitches on the spot in order to search beyond the camera's field of view. Although the authors successfully demonstrate autonomous landing using a low cost vehicle, the approach is limited by requiring offboard computation and is demonstrated only in benign conditions due to the limited flight envelope of the vehicle employed. Venugopalan *et al.* ignore the issue of attitude changes on the landing platform.

1.3 Contributions and Approach

This work presents a framework to perform a precise autonomous landing of a vertical take-off and landing (VTOL) aircraft onto a moving target. The landing is achieved through three stages. The first stage is a rendezvous, where the quadrotor calculates a rendezvous point to intercept the landing target. Rendezvous requires low relative pose accuracy and can rely on GPS and inertial measurement unit (IMU) data. Then a target acquisition stage ensures that the quadrotor is locked onto the visual target for the final descent. If necessary, a search pattern will be flown in order to acquire the target. In the final descent, visual measurements of a target on the landing pad provide high quality pose estimates to guide the quadrotor to a safe landing.

During the final descent, a novel relative estimation scheme is used in order to track the position and velocity of the aircraft with respect to the moving landing target. These estimates occur in what is defined as the body-planar frame, which eliminates the need to rely on inaccurate data from both GPS and magnetometers.

Aside from a downward facing camera, this work relies on sensors that are common on UAV systems. No assumptions are made on the quality of the sensors beyond the minimum quality required for stable flight. These considerations make the approach presented in this work easily implementable on a wide variety of VTOL UAVs.

The following chapters will document the design and theory behind this precision landing framework. Chapter 2 describes the dynamics and sensor models involved with a typical quadrotor system. Chapter 3 presents the control and estimation framework to accomplish the precision landing. Chapter 4 discusses the experimental setup and results from simulation and flight testing. Finally, Chapter 5 concludes the findings and results of this work and proposes future improvements.

Chapter 2

System Modelling

In this chapter, the system models relevant to the precision boat landing are presented. Section 2.1 outlines the major coordinate frames used within this work. Section 2.2.1 describes the dynamics related to the quadrotor vehicle as well as the relative dynamics between the quadrotor and the moving landing target. The relative dynamics are used for estimation and control during the target tracking stage of the landing. Finally, sensor models are presented in Section 2.3.

2.1 Coordinate Conventions

Five main coordinate frames are required in this work. The inertial or east-north-up (ENU) frame, denoted E, is the basis for GPS measurements and absolute positioning. East, north, and up will also be referred to as the inertial x_E, y_E, and z_E, respectively. The quadrotor body frame Q, shown in Figure 2.1 is defined with the x_Q-axis along the line from rotor 2 to rotor 4 emanating from the centre of gravity, y_Q along the line from rotor 1 to rotor 3, and z_Q upward. The quadrotor body frame is used to define vehicle motion and forces when convenient. A camera frame is defined at the optic centre of the camera with z_C axis perpendicular to the image plane. The body-planar frame, denoted by Q_0, is defined as a coordinate frame fixed to the centre of rotation of the quadrotor and rotated to the same yaw as the quadrotor. The horizontal plane of the body-planar frame is coplanar with the horizontal plane of the inertial frame, E. Finally, a target or marker frame, M, is defined as the body frame of the fiducial marker with x_M pointing forward, y_M to marker-right, and z_M up. The five coordinate frames and the transformations between them are illustrated in Figure 2.1.

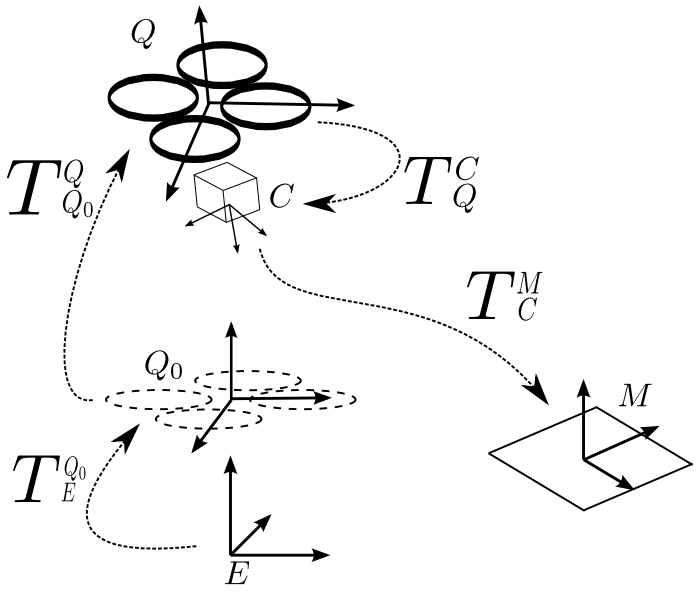


Figure 2.1: Coordinate frames and transformations. Relative position measurements of a fiducial marker, M, are taken in the camera frame, C, and then rotated into the body-planar frame of the quadrotor, Q_0, where estimation and control occur for the final sequence of the landing.

Let $R_A^B \in SO(3)$ represent the rotation matrix which rotates a point in frame A into frame B. Let P_A^B represent the position of the origin in frame B with respect to frame A. The homogeneous transformation between frame A and frame B, T_A^B, can be expressed as

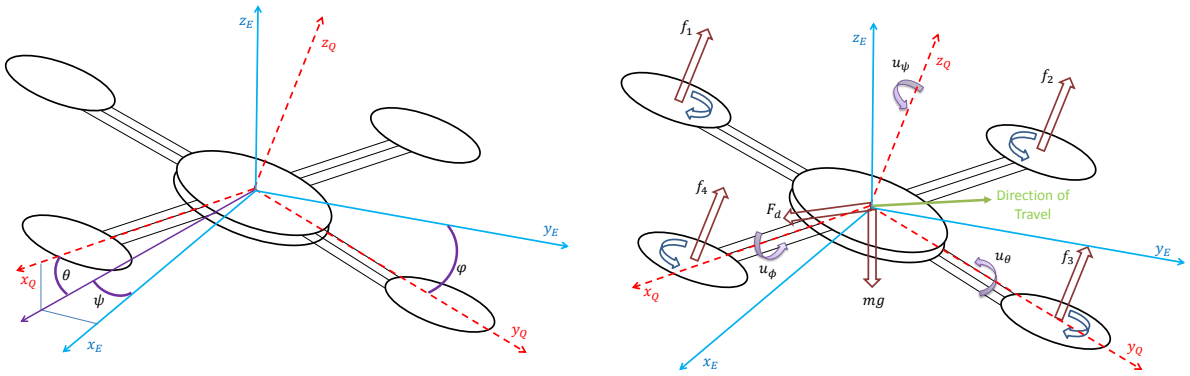
$$T_A^B = \begin{bmatrix} R_A^B & P_A^B \\ 0 & 1 \end{bmatrix}. \tag{2.1}$$

Since a large focus of this work focuses on x and y axis translation in the inertial or body-planar frames, an overbar is used to denote the horizontal components of a vector $\mathbb{R}^3$. For instance,

$$P_E = [x_E, y_E, z_E]^T \qquad\qquad \bar{P}_E = [x_E, y_E]^T.$$

Rotations in $\mathbb{R}^2$ about the yaw-axis alone will also be denoted with an overbar,

$$\bar{R}_\psi = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix}.$$



(a) Quadrotor coordinate frames.

(b) Quadrotor free-body diagram.

Figure 2.2: **(a)** 3-2-1 Euler angles are used to describe the rotations of the quadrotor system. The angles ϕ, θ, and ψ illustrate the axis of rotation for roll, pitch, and yaw, respectively. **(b)** The primary forces acting on a quadrotor are gravity, thrust, and drag. Force f_i is produced by i^{th} rotor [18].

2.2 System Dynamics

2.2.1 Quadrotor Dynamics

A free-body diagram of quadrotor system is shown in Figure 2.2b. The rotors along the y_Q axis, rotors 1 and 3, spin in the clockwise direction while the rotors along the x_Q axis, rotors 2 and 4, spin in the counter-clockwise direction. Each rotor generates a thrust in the z_Q direction and a torque about the z_Q axis, which are used to actuate the quadrotor.

The net thrust generated by all four rotors accelerates the system in the z_Q direction. If either the roll, ϕ, or pitch, θ, of the vehicle is non-zero, a component of the net thrust will accelerate the quadrotor in the x_E-y_E plane. Net torques about the body x_Q and y_Q axes can be achieved by varying the amount of thrust produced by opposing rotors. A net torque about the body z_Q is created by differences in the net speeds of counter-rotating rotors. That is, if the net speed of rotors 1 and 3 is different than the net speed of rotors 2 and 4, there will be a net torque about the z_Q axis.

Euler angles are used to describe the attitude of the vehicle through a series of roll, pitch, and yaw rotations. To avoid singularities with the Euler angles, the rotation angles

must be restricted to $-\frac{\pi}{2} < \phi < \frac{\pi}{2}$, $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$, and $-\pi < \psi < \pi$.

The quadrotor dynamic models used in this work are well established [7, 17], and rely on mixed-frame rigid body dynamic equations for translation and rotation. The mixed frame is used since measurements are taken in both the inertial and the body frame. In particular, the GPS receiver measures the absolute position in the inertial frame while the accelerometers, gyroscopes, and magnetometers measure accelerations, rotation rates, and the magnetic field vector in the body frame.

The quadrotor inertial position is denoted

$$P_E = [x_E, y_E, z_E]^T \in \mathbb{R}^3$$

and the body rotation rate is

$$\omega_Q = [p, q, r]^T \in \mathbb{R}^3.$$

From rigid body dynamics, the resulting equations of motion are

$$m\ddot{P}_E = F_E \tag{2.2}$$

$$J\dot{\omega}_Q + \omega_Q \times J\omega_Q = M_Q, \tag{2.3}$$

where m is the mass of the vehicle, $F_E \in \mathbb{R}^3$ denotes the applied forces in the inertial frame, and $M_Q \in \mathbb{R}^3$ denotes the applied moments in the body frame. The inertia of the vehicle, J, is the diagonal matrix

$$J = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix}.$$

The relationship between the body rotational rates and Euler rates [4, p. 40] is

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \dfrac{\sin(\phi)}{\cos(\theta)} & \dfrac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{2.4}$$

In simulation, additional disturbance terms are added to both the translational and rotational dynamics to represent unmodelled dynamics. The resulting models are then

$$m\ddot{P}_E = F_E + D_E \tag{2.5}$$

$$J\dot{\omega}_Q + \omega_Q \times J\omega_Q = M_Q + D_Q, \tag{2.6}$$

where $D_E \in \mathbb{R}^3$ denotes disturbance forces drawn from a zero-mean Gaussian distribution, $\mathcal{N}(0, \Sigma_F)$, in the inertial frame and $D_Q \in \mathbb{R}^3$ denotes disturbance torques drawn from a zero-mean Gaussian distribution, $\mathcal{N}(0, \Sigma_M)$, in the body frame.

Translational Dynamics

Forces acting on the vehicle can be modelled as thrust, F_t, drag, F_d, and gravity, mg, giving the equation

$$F_E = F_t + F_d - mg. \tag{2.7}$$

The drag term, F_d, is detailed further in Section 2.2.1. Given that the i^{th} rotor produces a thrust f_i in the body z_Q direction, the collective thrust is

$$u_z = \sum_{i=1}^{4} f_i. \tag{2.8}$$

The collective thrust, u_z, is used as an input to the system. Then the resulting thrust in the inertial frame can be written as

$$F_t = R_Q^E \begin{bmatrix} 0 \\ 0 \\ u_z \end{bmatrix}, \tag{2.9}$$

which is purely vertical in the body frame.

Rotational Dynamics

The moments applied to the vehicle are most easily expressed in the quadrotor body frame and include moments generated through differences in rotor thrusts, M_t, and rotational drag, M_d. The rotational dynamics are

$$M_Q = M_t + M_d. \tag{2.10}$$

The torques generated by the rotors are used as the rotational control inputs and are related to the individual rotor thrusts by

$$M_t = \begin{bmatrix} u_\phi \\ u_\theta \\ u_\psi \end{bmatrix} = \begin{bmatrix} d(f_4 - f_2) \\ d(f_1 - f_3) \\ \tau(f_1 - f_2 + f_3 - f_4) \end{bmatrix}, \tag{2.11}$$

where d denotes the moment arm of the rotor hub about the centre of gravity and τ denotes the static thrust to torque ratio. The rotational drag M_d is described in Section 2.2.1.

Drag Model

For simulation and estimation purposes, parasitic drag is the only source of translational drag modelled in this work. Parasitic drag is the resistive force caused by an object moving through air. A common definition for parasitic drag [17] is

$$F_{di} = \frac{1}{2}\rho_{air} C_i v_i^2, \tag{2.12}$$

where F_{di} is the parasitic drag for axis i_E, ρ_{air} is the air density, C_i is the aerodynamic coefficient for axis i, and v_i is the relative airspeed of the quadrotor for axis i. The parasitic drag is in the direction that opposes the relative airspeed of the vehicle. A simplifying assumption that the parasitic drag acts on the centre of gravity of the quadrotor is made. Since the vehicle is symmetric and most of the area is in line with the rotor plane, this assumption is reasonable. The implication of this assumption is that the translational parasitic drag only applies a translational force on the quadrotor and does not produce a moment about the centre of gravity of the quadrotor.

Grouping the two constant terms and taking into account the direction of the relative airspeed, the drag model can be written in vector form as

$$F_d = \begin{bmatrix} F_{dx} \\ F_{dy} \\ F_{dz} \end{bmatrix} = \operatorname{sgn}(v_w - \dot{P}_E) * \mu_E^t * \left((v_w - \dot{P}_E) * (v_w - \dot{P}_E) \right), \tag{2.13}$$

where $*$ denotes element-wise multiplication, $v_w \in \mathbb{R}^3$ is the inertial free stream velocity of the air (i.e. the wind field in which the quadrotor finds itself), and $\mu_E^t = R_Q^E[\mu_x^t, \mu_y^t, \mu_z^t]^T$ is a rotated vector of body frame translational aerodynamic drag coefficient.

Rotational drag is modelled similarly to the translational case. It is treated as a parasitic drag that is proportional to the square of the angular velocity and in the direction opposing the angular velocity. This is expressed as

$$M_d = -\operatorname{sgn}(\omega_Q) * \mu_Q^r * (\omega_Q * \omega_Q), \tag{2.14}$$

where $\mu_Q^r = [\mu_x^r, \mu_y^r, \mu_z^r]$ is a vector of rotational aerodynamic drag coefficient in the body frame.

With the drag terms defined, the quadrotor model is complete. The translational

dynamics are fully described by

$$m\ddot{P}_E = R_Q^E \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{4} f_i \end{bmatrix} + \text{sgn}(v_w - \dot{P}_E) * \mu_E^t \\ * \Big((v_w - \dot{P}_E) * (v_w - \dot{P}_E) \Big) - mg. \quad (2.15)$$

The rotational dynamics are fully described by

$$J\dot{\omega}_Q + \omega_Q \times J\omega_Q = \begin{bmatrix} d(f_4 - f_2) \\ d(f_1 - f_3) \\ \tau(f_1 - f_2 + f_3 - f_4) \end{bmatrix} - \text{sgn}(\omega_Q) * \mu_Q^r * (\omega_Q * \omega_Q)\,. \quad (2.16)$$

Omitted Dynamics

Within the research community, there is still no general consensus on which aerodynamic effects are the most significant and should be included in the dynamic model. Mokhtari and Benallegue model only parasitic drag [17]. Hoffman *et al.* suggest that, at moderate speeds, drag forces are less significant than the forces produced by rotor thrust and blade flapping [7]. On the contrary, other sources neglect blade flapping and suggest that rotor drag has a significant effect, even at low speeds [13, 11]. Both Martin *et al.* and Leishman *et al.* demonstrate that estimation techniques that take into account rotor drag have improved performance. Both blade flapping and rotor drag were left out of this work and could be added as future improvements.

2.2.2 Relative Dynamics

The target tracking stage of the landing process is done using relative positioning between the quadrotor and the target. The inertial position of the moving landing target is denoted as $P_M = [x_M, y_M, z_M]$. The states that are tracked are the position and velocity of the target, M, in the body-planar frame, Q_0. Since the Q_0 frame is centred at P_E in the inertial frame, the relative position is defined as

$$P_r = R_E^{Q_0}(P_M - P_E) = \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix}.$$

If the landing target is mounted on a vehicle with known dynamics and inputs, the landing site vehicle's dynamic model can be used to propagate the estimated target position. However, knowledge of the target vehicle's model and inputs are not guaranteed. In the maritime landing scenario, the effect of waves is often unpredictable and could be the dominant dynamics for a slow moving ship. Since this work focuses on developing control for the quadrotor, independent of the vehicle on which the quadrotor is to land on, no assumptions are made about the dynamics of the target. The position of the target determines the commands that are given to the quadrotor, but the position of the target is not directly controlled and is assumed to operate independently. Thus, a constant velocity model is used to propagate the states of the landing target. The target's dynamics can be written in the continuous time as

$$\begin{bmatrix} \dot{P}_r \\ \ddot{P}_r \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_r \\ \dot{P}_r \end{bmatrix} + D_r, \tag{2.17}$$

where $D_r \in \mathbb{R}^6$ is a vector of disturbances drawn from appropriate zero-mean Gaussian distributions.

Since the moving target is modelled with a constant velocity, the relative dynamics in the horizontal plane are the negated quadrotor dynamics in the body-planar frame. The resulting dynamics are similar to the dynamics in Equation (2.7), and are given as

$$m\ddot{P}_r = -F_t - F_d - mg. \tag{2.18}$$

2.3 Sensors

In this section, the sensors assumed available on the quadrotor system and their models are discussed. Studying the sensor models serves to further justify the choice to disregard GPS and magnetometer measurements in the final target tracking stage of the landing.

The sensors required for the precision landing framework presented in this work are: 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer, GPS receiver, barometer, sonar, and a downward facing camera. The accelerometers, gyroscopes, and barometer are used in both the inertial and relative tracking stages of flight. In the rendezvous and target acquisition stages, control and estimation are performed in the inertial frame. These two stages rely heavily on the GPS and magnetometer measurements. Then in the target tracking stage the GPS and magnetometer are ignored as they have unobservable biases which corrupt their measurements. Instead, pose measurements are obtained through the use of a downward facing camera to track a fiducial marker.

2.3.1 Inertial Estimation Sensors

The sensors for inertial frame estimation are used during the rendezvous and target acquisition stages of the landing. These two stages are both discussed in Chapter 3. For the inertial frame estimation, the sensors used are: accelerometers, gyroscopes, magnetometers, GPS receiver, barometer, and sonar.

Accelerometers

Accelerometers measure the specific acceleration of the quadrotor [11]. They are subject to noise from high-frequency vibrations while in flight. Accelerometers are also typically biased, however the bias is considered slow-changing and can effectively be calibrated out prior to takeoff. Thus the accelerometer model is

$$y_a = a_Q + \epsilon_a, \tag{2.19}$$

where $y_a \in \mathbb{R}^3$ is vector of measured accelerations, $a_Q \in \mathbb{R}^3$ is the vector of actual specific accelerations, and $\epsilon_a \in \mathbb{R}^3$ is a noise vector drawn from a zero-mean Gaussian distribution, $\mathcal{N}(0, \Sigma_a)$.

Gyroscopes

Gyroscopes measure the body rotational rates: p, q, and r. MEMS gyroscopes are typically subject to both noise from quadrotor vibrations in flight as well as a time varying bias. The model is given as

$$y_{gyro} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \epsilon_{gyro} + b_{gyro} \tag{2.20}$$

where $y_{gyro} \in \mathbb{R}^3$ is the output vector of the 3-axis gyroscope, $\epsilon_{gyro} \in \mathbb{R}^3$ is a noise vector drawn from a zero-mean Gaussian distribution, $\mathcal{N}(0, \Sigma_{gyro})$, and b_{gyro} is a time varying bias driven by white noise. The bias is also modelled with a decay factor [16]. The resulting bias update rule is

$$\dot{b}_{gyro} = -\frac{1}{\tau} b_{gyro} + \epsilon_{b,gyro}, \tag{2.21}$$

where τ is a time constant for the bias and $\epsilon_{b,gyro}$ is a noise vector drawn from a zero-mean Gaussian distribution, $\mathcal{N}(0, \Sigma_{b,gyro})$.

Magnetometers

Magnetometers once again operate similarly to the GPS receiver and the gyroscopes. The measurement model is

$$y_m = \psi + \epsilon_m + b_m, \tag{2.22}$$

where the bias, b_m is updated similarly to Equation (2.21). Magnetometers are a large source of error on a quadrotor system since the motors that actuate the system create a frequently changing magnetic field. As well, mounting a magnetometer too close to a computer, other electronic devices, or ferromagnetic metals can also cause a significant degradation in performance. Yaw errors up to 30° have been observed in flight on a low-cost quadrotor platform.

GPS Receiver

The GPS receiver follows a similar sensor model as the gyroscopes. The measurement model is

$$y_{gps} = P_E + \epsilon_{gps} + b_{gps}. \tag{2.23}$$

The GPS measurement bias evolves similarly to that of the gyroscopes, presented in Equation (2.21). For a low-cost GPS receiver, the measurements typically fall within a 5 m radius of the true position.

Barometer

The barometer measures the absolute altitude of the quadrotor but with a significant time varying bias. The measurement model is

$$y_{barom} = z_E + \epsilon_{barom} + b_{barom}, \tag{2.24}$$

with the appropriate noise and a bias defined similarly to Equation (2.21). Barometers measure the air pressure prior to takeoff and use changes in measured air pressure to determine the current height of the quadrotor. If the local air pressure changes then the barometer's bias will change in flight. Wind gusts can affect the air pressure and cause significant bias changes for a barometer.

Sonar

The sonar also measures the absolute altitude of the quadrotor. However, it is a bias free measurement. The measurement model is

$$y_{barom} = z_E + \epsilon_{barom}. \tag{2.25}$$

Although the sonar measurements are not biased, sonar measurements are often faulty due to missed ultrasonic pings, electrical noise, vibrations, or interference from rotor downwash [15]. As such, a naive pre-filter is often used to reject erroneous sonar measurements prior to using the measurement.

Bias Estimation

Although many of the sensors are biased, some of them can be coupled together in order to eliminate biases. Gyroscope biases can be estimated and corrected through sensor fusion with accelerometers. When the quadrotor is at a steady state with no net acceleration, or reasonably close, the accelerometers can be used as an inclinometer to determine the roll and pitch of the vehicle. This allows the gyroscope bias to be eliminated while in flight so that the gyroscope measurements are still extremely useful. The yaw gyroscope about the z_Q axis can be bias corrected through fusion with magnetometer measurements.

Similarly, barometer bias can be estimated with sonar. While the quadrotor is below the maximum height that a sonar can measure, the sonar is typically much more accurate than the barometer. The reason that a barometer is still useful is in case the sonar receives a false reading that reports a lower altitude than the actual altitude. As well, barometers do not have an effective height limit.

The GPS receiver and magnetometer, on the other hand, have unobservable biases. With the sensors available, there are no non-biased absolute measurements for the inertial position, P_E, or the yaw, ψ. This results in uncorrectable bias corruption on the GPS receiver and magnetometer measurements. This motivates the decision to ignore measurements from these two sensors in the final target tracking stage of the landing where accurate measurements are a requirement.

2.3.2 Relative Estimation Sensors

For the target tracking stage of the landing, estimation and control are performed in a relative frame. The estimation and control are described further in Chapter 3. The

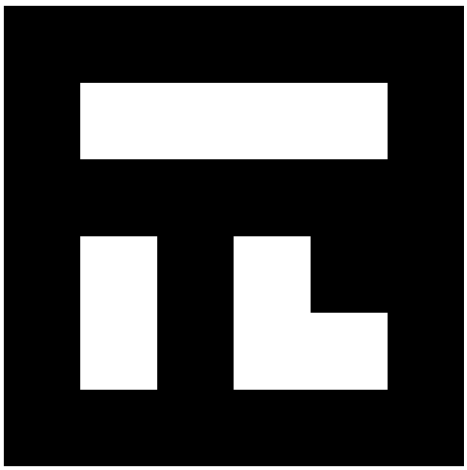


Figure 2.3: An example of an AprilTag fiducial marker.

primary source of position measurements to drive the relative estimates is vision. GPS and magnetometer measurements are not used for the relative estimation as they are sources of significant errors.

In the relative position estimation case, the gyroscopes, accelerometers, sonar, and barometer are used as they are in the inertial estimation case. The models for these sensors are identical to those presented in Section 2.3.1.

Target Detection

Since most quadrotor missions require a camera on the quadrotor, vision was selected as a viable option to obtain high quality pose estimates. Vision also provides a great cost-benefit ratio as even a low-cost camera can provide significantly improved pose measurements over fused IMU and GPS measurements.

There are many visual fiducial marker systems currently available. Many of them are compatible with the framework set out in this work. For the experimental portions of this work, the AprilTag visual fiducial system [19] was selected to measure the relative pose of the landing target. This visual fiducial system was chosen over lighter-weight alternatives for several reasons: it provides a 6-DOF pose estimate, the detection range scales with the tag size, and it generates few false positives. An example of an AprilTag fiducial marker is shown in Figure 2.3. The AprilTag is a two-dimensional barcode where data are encoded

in the black and white pixels. The encoded data correspond with an identification number that is checked to greatly reduce the likelihood of false positives.

A C++ implementation of the AprilTag system was used [9] with some modifications in order to increase the computational speed of AprilTag detection, so that a reasonable measurement update rate could be achieved onboard the quadrotor. The quadrilateral detection algorithm was replaced with a simpler Canny edge detector and Hough transform in OpenCV.

The AprilTag detection algorithm returns the relative pose of the tag in the camera frame T_M^C. This transformation is converted into the body-planar frame, Q_0, through the transformation $T_Q^{Q_0} T_C^Q$. This gives the equation

$$\begin{bmatrix} R_{Q_0}^M & P_r \\ 0 & 1 \end{bmatrix} = T_Q^{Q_0} T_C^Q T_M^C. \tag{2.26}$$

The relative position, P_r, is then used as the measurement in the relative position estimator. The rotation $R_{Q_0}^M$ also provides valuable information on the orientation of the landing target.

The simulated AprilTag detection system provides bias free relative pose estimates with additive zero-mean Gaussian white noise. This model can be written as

$$y_{target} = \begin{bmatrix} P_r \\ \psi_r \\ \theta_r \\ \phi_r \end{bmatrix} + \begin{bmatrix} \epsilon_{P,t} \\ \epsilon_{\psi,t} \\ \epsilon_{\theta,t} \\ \epsilon_{\phi,t} \end{bmatrix}. \tag{2.27}$$

Chapter 3

Quadrotor Control and Estimation

The proposed precision landing process consists of three stages: rendezvous, target acquisition, and target tracking. The three stages are illustrated in Figure 3.1. First, the quadrotor flies at a constant velocity to a rendezvous point. The rendezvous point is calculated based on the current quadrotor position, the programmed maximum quadrotor speed, and the current movement of the target as determined by GPS. Then when the quadrotor is sufficiently close to the target, the target acquisition stage begins. During the target acquisition stage, the quadrotor regulates its GPS position to that of the target. If needed, the quadrotor will fly a search pattern around the target in order to gain sight of it. Once the target is acquired, position control and estimation changes from the inertial frame to the body-planar frame and the final stage begins. The final stage is the target tracking stage. During this stage, the camera on the quadrotor is used to provide relative pose measurements to the target. GPS position measurements and magnetometer data are no longer used as those sensors have a high degree of error. When the relative position error is below a certain threshold, the quadrotor begins descending to land on the target. Throughout the descent, the quadrotor continues to track the target as long as the target can fit within the camera's field of view.

The three stages of the flight are performed with different position controllers, which are described in this chapter. Attitude stabilization and control is assumed to be performed at a lower-level. The low-level attitude control should rely only on inertial measurement data from the gyroscopes and accelerometers. There are a large number of multirotor autopilots commercially available that will perform attitude control so this assumption is reasonable.

The altitude control and simulation attitude control are implemented as described by Hoffmann *et al.* [7]. For altitude control, a PID controller tracks the desired height.

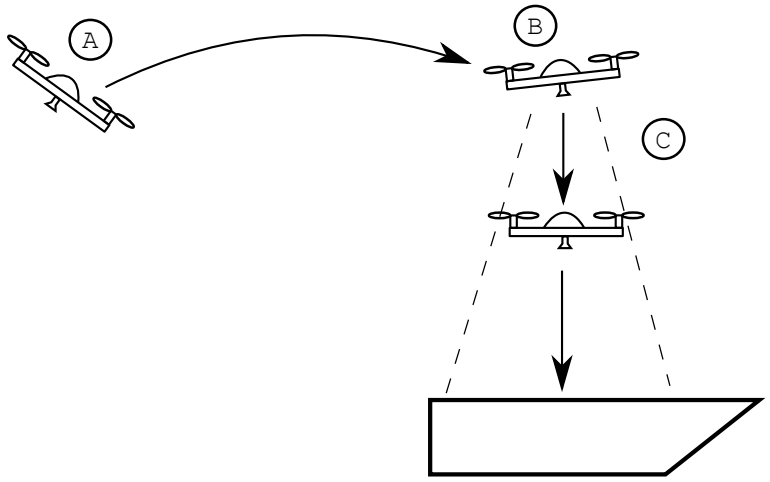


Figure 3.1: The three stages of flight for the precision landing. In the first stage, the quadrotor approaches the target on a rendezvous trajectory. The second stage is target acquisition where the quadrotor searches for the target. Then in the final stage, the quadrotor performs target tracking while it descends to land on the target.

Attitude estimation in simulation is performed using an extended Kalman filter which uses IMU measurements to estimate ϕ, θ, ψ, $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$, and gyroscope biases. Attitude control is performed using a PID controller with an additional double-derivative term which acts on the angular acceleration.

Yaw control is not discussed in this chapter, nor are yaw reference signals generated. The relative position estimation framework was designed in such a way that the estimation is robust to any choice of yaw and changes in yaw. This choice was made to increase the practical usability of this control and estimation framework. In semi-autonomous or autonomous operation of a quadrotor, the yaw is typically commanded in such a way to point an onboard sensor towards an area of interest. While in this work we only assume one camera that is in a downward facing configuration, some quadrotor platforms have both a downward facing camera and a forward facing camera. By not placing restrictions on the yaw, the user, or a high-level mission controller, is free to change the yaw so that the forward facing camera can make observations in any arbitrary direction throughout the landing process.

3.1 Roll and Pitch Inversion

In all stages of flight, the nonlinear inversion

$$\phi_d = \sin^{-1}\left(\frac{mu_y}{u_z}\right) \tag{3.1}$$

$$\theta_d = -\sin^{-1}\left(\frac{mu_x}{u_z\cos(\phi_d)}\right), \tag{3.2}$$

is applied to map desired horizontal accelerations in the body-planar frame, u_x and u_y, to commanded roll and pitch angles, ϕ_d and θ_d, respectively [17]. The inversion is obtained by ignoring the drag term in Equation (2.15) and solving for the roll and pitch that produce a desired acceleration.

The drag term was ignored in Equations (3.1) and (3.2) since it is accounted for with a feedforward term applied in all stages of flight. The feedforward term is calculated with

$$u_f = \frac{\text{sgn}(v_d) * \bar{\mu}_E^t * (v_d * v_d)}{m} \tag{3.3}$$

where $v_d \in \mathbb{R}^2$ is the desired horizontal velocity in the inertial frame and $u_f = [u_{f,x}, u_{f,y}]^T$ is the feedforward acceleration vector. This relation is obtained by finding the acceleration required to balance out the drag force defined in Equation (2.13) at the desired velocity. In Equation (3.3), the desired ground speed of the quadrotor is used as an approximation for the airspeed in the absence of proper wind estimation. For low airspeeds, this approximation should be reasonable. The addition of a proper wind estimation method is left as a future improvement to this work.

3.2 Rendezvous

When the autonomous landing sequence is initiated, the current estimated position of the quadrotor, the position of the target, and the current estimated velocity of the target are used to calculate a rendezvous point. Two similar triangles are constructed; one with the positions of the quadrotor, target, and rendezvous point and the other with the target's current velocity vector and a constant velocity vector from the quadrotor. Both velocity vectors are pointing to some rendezvous point which is still unknown, as shown in Figure 3.2. The constant speed at which the quadrotor will approach the rendezvous point is

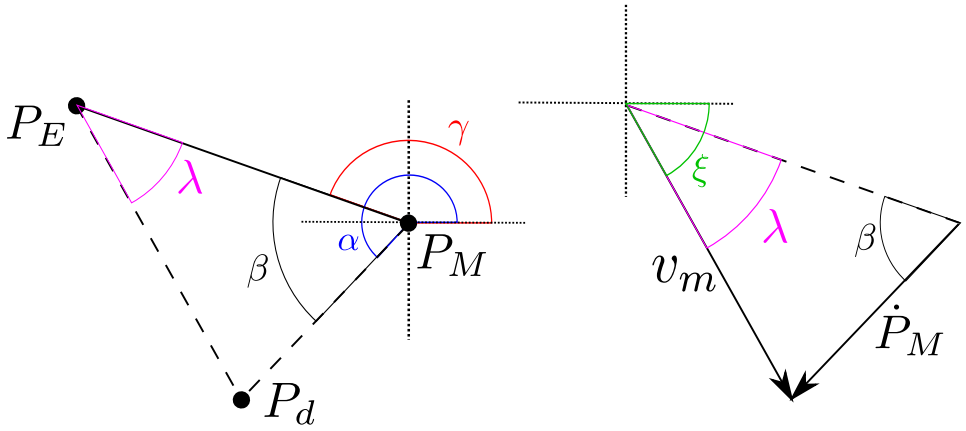


Figure 3.2: Similar triangles used to calculate the rendezvous point. The first triangle is formed by the position of the quadrotor, target, and the rendezvous point to be calculated, P_d. The second triangle is formed by the target's current velocity vector and the vector of the quadrotor moving to the rendezvous point at a fixed speed, v_m.

pre-programmed and is denoted v_m. It is assumed that $v_m > |\dot{P}_s|$ so that a rendezvous is always possible.

The angle, β, is defined as the angle between the target's velocity vector, $\dot{P}_s$, and the vector between the quadrotor and the target, $P_E - P_M$. It can be calculated as $\beta = |\alpha - \gamma|$. The angle α is the heading of the target's velocity and the angle γ is the heading of the distance vector $P_E - P_s$. Using the law of sines, the angle, λ, between the distance vector of the two vehicles and the rendezvous point can be calculated as

$$\lambda = \arcsin\left(\frac{\left|\dot{\bar{P}}_s\right| \sin\beta}{\left|\dot{\bar{P}}_m\right|}\right). \tag{3.4}$$

By inspection, the desired quadrotor heading, ξ, will be either

$$\xi = 180^\circ - \gamma + \lambda$$

or

$$\xi = 180^\circ - \gamma - \lambda.$$

Both solutions are propagated through to calculate the time to rendezvous, then the negative or non-minimum time will be discarded.

The desired speed is rotated into the correct heading to obtain the desired quadrotor velocity vector,

$$\dot{\bar{P}}_m = \bar{R}_\xi [v_m, 0]^T. \tag{3.5}$$

Using the two equations for the rendezvous point, $P_d \in \mathbb{R}^2$,

$$P_d = t_r \dot{\bar{P}}_s + \bar{P}_s = t_r \dot{P}_m + \bar{P}_I, \tag{3.6}$$

two solutions are obtained for the time to rendezvous, t_r, one for each value of ξ. Only the smallest positive value of t_r is kept as that is the shortest time to rendezvous. The ξ that corresponds to the extraneous time is also discarded. The rendezvous point is now trivially calculated with Equation (3.6). The rendezvous point can be updated as needed to track a non-constant velocity target motion.

With the rendezvous point calculated, the quadrotor is commanded on a constant velocity approach to intersect the ship's path. PID feedback controllers with feedforward input regulate the along-track and cross-track errors [7]. The along-track and cross-track errors are defined as

$$\begin{bmatrix} e_{at} \\ e_{ct} \end{bmatrix} = \bar{R}_\xi (\bar{P}_0 - \bar{P}_E), \tag{3.7}$$

where P_0 is the position of the quadrotor when the current rendezvous point was calculated, e_{at} is the along-track error which is along the vector from P_0 to the rendezvous point, and e_{ct} is the cross-track error which is perpendicular to the vector from P_0 to the rendezvous point.

The integrator in the controller is used to compensate for any wind disturbances. For the rendezvous stage of the landing, the feedback control laws are

$$u_{at} = K_{at,d} \dot{e}_{at} + K_{at,i} \int \dot{e}_{at} dt \tag{3.8}$$

$$u_{ct} = K_{ct,p} e_{ct} + K_{ct,d} \dot{e}_{ct} + K_{ct,i} \int e_{ct} dt, \tag{3.9}$$

where $K_{at,d}$ and $K_{at,i}$ are the derivative and integral gains for the along-track controller, respectively, and $K_{ct,p}$, $K_{ct,i}$, and $K_{ct,d}$ are the proportional, integral, and derivative gains for the cross-track controller, respectively. Note that in the along track controller the velocity error is integrated as the goal is to ensure that the quadrotor flies at a constant speed in the along-track direction. Combining the feedback control signals with the feedforward term and rotating the desired accelerations into the body-planar frame yields

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \bar{R}_\psi \left(\bar{R}_\xi^T \begin{bmatrix} u_{at} \\ u_{ct} \end{bmatrix} + u_f \right). \tag{3.10}$$

Finally the control signals are mapped to the roll and pitch commands as described in Section 3.1.

Transition to the target acquisition phase of flight can be commanded to occur once the quadrotor is within a prescribed radius of the ship GPS position, or the rendezvous time falls below a prescribed minimum value.

3.3 Target Acquisition

Now that the quadrotor is in the vicinity of the landing target, it can begin to search for the target. As soon as the target is detected, the quadrotor will move to the final stage of the landing and track the target.

The desired position and velocity for the quadrotor during this stage of the flight are chosen to match the target position estimates, such that error is defined as

$$e_E = \bar{P}_s - \bar{P}_E. \tag{3.11}$$

Depending on the the camera's field of view, the height at which the quadrotor approaches the target, and the current GPS measurement errors on the quadrotor and the target positions, the target may not be in sight of the camera when the GPS positions of the target and the quadrotor line up. If that is the case, the reference position can be modified to incorporate a search pattern relative to the target's current GPS position and velocity estimate in the inertial frame. The ship's current estimated velocity is set as the desired velocity when computing the feedforward command.

Two PID controllers are used for feedback control. The feedback control laws are given as

$$u_{E,x} = K_{E,p}e_{E,x} + K_{E,d}\dot{e}_{E,x} + K_{E,i}\int e_{E,x}dt \tag{3.12}$$

$$u_{E,y} = K_{E,p}e_{E,y} + K_{E,d}\dot{e}_{E,y} + K_{E,i}\int e_{E,y}dt, \tag{3.13}$$

where $K_{E,p}$, $K_{E,i}$, and $K_{E,d}$ are the proportional, integral, and derivative gains for the inertial position controller, respectively. The control signals that are mapped to commanded roll and pitch are

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \bar{R}_\psi \left(\begin{bmatrix} u_{E,x} \\ u_{E,y} \end{bmatrix} + u_f \right). \tag{3.14}$$

3.4 Target Tracking

Relative position estimation is done entirely in the body-planar frame using a Kalman filter. Target detections, transformed into the body-planar frame, provide the measurement updates for the Kalman filter. Upon the first target detection, the relative position estimator will initialize its position states to the new visual measurements. The velocity states in the estimator will be initialized to $\dot{\bar{P}}_r = \bar{R}_E^{Q_0}(\dot{\bar{P}}_s - \dot{\bar{P}}_E)$, using the current inertial estimates. Any error in the velocity initialization will be corrected quickly if visual measurements continue to be received. The quadrotor's estimated roll, pitch, and collective thrust are used as inputs to propagate the states of the estimator.

Due to the nature of quadrotor vehicles, as the quadrotor moves towards the target it points the downward facing camera away from the target. This effect coupled with a potentially low-field of view camera can result in frequent target losses, where the target is not visible in the camera image. In order to successfully track the target from any position other than directly above, it is necessary for the system to be able to deal with frequent failures in target detection. The estimated states of the relative estimator are $x = \begin{bmatrix} \bar{P}_r & \dot{\bar{P}}_r \end{bmatrix}^T$ and the state prediction equation is

$$\dot{x} = \begin{bmatrix} \bar{R}_{\Delta\psi}\dot{\bar{P}}_r \\ -2\left(\dot{\psi} \times \dot{P}_r\right) + \ddot{\bar{P}}_r \end{bmatrix}, \tag{3.15}$$

where $\ddot{\bar{P}}_r$ is calculated from the relative dynamics presented in Equation 2.18. At each time step, $\Delta\psi$ is obtained by integrating the z-axis gyroscope and it is used to rotate the current state estimates. While propagating the state estimates without visual measurements for a prolonged period will result in significant drift, it will be reasonably accurate for short periods of time between visual measurements. Thus relative position tracking is abandoned when target measurements have not been received for some prescribed period of time due to occlusion. If the quadrotor cannot regain sight of the target within the time limit, the system reverts to the target acquisition stage until the next target detection occurs.

The prediction step in the relative position estimation is dependent on wind, as the airspeed directly affects the drag term. Again, a wind estimation scheme is beyond the scope of this work, so it is expected that the estimation performance may degrade in extreme wind conditions. Adding wind estimation, such as presented by Waslander and Wang [29], is left as an area of future work.

Similar to the target acquisition stage of the landing process, the feedback control laws

are

$$u_{r,x} = K_{r,p}e_{r,x} + K_{r,d}\dot{e}_{r,x} + K_{r,i}\int e_{r,x}dt \tag{3.16}$$

$$u_{r,y} = K_{r,p}e_{r,y} + K_{r,d}\dot{e}_{r,y} + K_{r,i}\int e_{r,y}dt, \tag{3.17}$$

where $K_{r,p}$, $K_{r,i}$, and $K_{r,d}$ are the proportional, integral, and derivative gains for the inertial position controller, respectively. However, the control signals that are mapped to commanded roll and pitch are now

$$\begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} u_{I,x} \\ u_{I,y} \end{bmatrix} + \bar{R}_\psi u_f. \tag{3.18}$$

The major difference is that the feedback control signals are no longer rotated. This eliminates possible corruption from a biased yaw estimate. GPS measurements are only used in the feedforward term of the control law. As a result, the main sources of positioning error have been eliminated from the feedback loop. This allows higher gains to be used for the target tracking stage which will result in improved tracking performance.

Descent

When the relative position error is sufficiently small, the height command for the quadrotor is ramped down at a fixed rate. If a disturbance causes the relative position error to increase beyond a certain threshold during the descent, the height command will freeze and wait until the relative position error is once again sufficiently small.

As the quadrotor descends, it reaches a point where the target no longer fits in the field of view of the camera. For a 180° field of view fisheye lens and a square target with 30 cm side lengths, the minimum height to observe the target is about 0.5 m. At this point, the quadrotor continues the descent using the same control laws as the rest of the target tracking stage. However, the estimates are now based solely on the predictive model in Equation (3.15). The feedforward term is still applied to match the velocity of the target. The tracking performance here should be reasonably accurate given that the quadrotor must have been tracking reasonably up until this point. The remaining descent to land on the target will not be very long so the estimation without new visual measurements will not incur significant drift.

A possible consideration for future work would be to incorporate a second, smaller target for the descent just before landing. In such a scenario, the quadrotor would track

the original larger target until it is at a low enough height to track the smaller target properly. At that time, the quadrotor will switch to tracking the smaller target which it will be able to see until it lands. Smaller targets have the advantage of a smaller minimum detection range, but at the cost of a smaller maximum detection range. A larger target is beneficial in the target acquisition stage since it can be detected from a greater altitude, allowing the quadrotor to fly higher where the camera can capture a larger area in each image.

3.5 Controller Switching

The transition between inertial and relative control is smoothed by seeding the controller integrators with the values of the previous stage. While flying in either inertial control or relative control mode, properly tuned integrators should wind up to cancel the effects of wind. Rotating the controller integrators from the inertial to relative frame, and vice versa, provides a good starting estimate of the control effort required to overcome wind. This initial seeding of the controllers will incorporate both GPS and magnetometer errors, however the initial integrator values will be closer to what is needed to compensate for wind than if the integrators were reset when switching controllers. Consequently errors due to wind should correct more quickly.

In the rendezvous and target acquisition stages, position control integrators are tracked in the inertial frame. Moving to relative control, the integrators are rotated into the body-planar frame such that

$$\begin{bmatrix} I_{r,x} \\ I_{r,y} \end{bmatrix} = \bar{R}_{\psi} \begin{bmatrix} I_{E,x} \\ I_{E,y} \end{bmatrix}, \tag{3.19}$$

where $I_{r,x}$ and $I_{r,y}$ are the x and y error integrators in the body-planar frame, respectively, and $I_{E,x}$ and $I_{E,y}$ are the x and y error integrators in the inertial frame, respectively.

It is also guaranteed that the controllers will not switch rapidly since the relative position estimator always attempts to propagate the relative states and regain sight of the target before switching back to inertial control.

3.6 Height Control and Estimation

Height Estimation

In the final target tracking stage, the altitude is estimated relative to the visual target's altitude. Height information comes from four different sensors: barometer, z_Q-axis accelerometer, sonar, and camera. The barometer and accelerometer data are fused on the low-level platform of the experimental system and the outputs are height (z_E) and climb rate ($\dot{z}_E$) estimates. The two estimates from the low-level estimator are then treated as inputs to the height estimator that fuses in the sonar and camera measurements.

The barometer, camera, and sonar provide absolute measurements of height. However, each sensor measures from slightly different points of reference and the offsets cannot be calibrated out. The barometer has a significant time varying bias and it is largely affected by gusts of wind. If the target is on a raised platform but the quadrotor is off to the side of the platform, the sonar and camera measurements will be different. In order to consistently fuse height measurements coming from different sensors, a Kalman filter is used and biases between sensors are estimated in-flight. The estimated states are

$$x_{alt} = \begin{bmatrix} z_E \\ \dot{z}_E \\ b_{barom} \\ b_{sonar} \end{bmatrix}, \tag{3.20}$$

where b_{barom} is the height measurement bias between the barometer and the sonar and b_{sonar} is the height measurement bias between the sonar and the target. Clearly, the two bias terms can be expanded as

$$b_{barom} = z_{sonar} - z_{barom} \tag{3.21}$$

and

$$b_{sonar} = z_{target} - z_{sonar}. \tag{3.22}$$

The measurement matrices for the Kalman filter are then

$$C_{barom} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.23}$$

$$C_{sonar} = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} \tag{3.24}$$

$$C_{target} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}. \tag{3.25}$$

The z_E-component of the dynamic equation presented in Equation (2.2) is used for the prediction update step. The process noise and measurement noise covariances are hand tuned. Since the attitude is estimated separately and all of the states for height enter the dynamic equation linearly, a linear Kalman filter is sufficient.

For the stages of flight based on inertial estimates (rendezvous and target acquisition stages) a similar height estimator without the target measurements and without the b_{sonar} state is used.

Height Control

Height control is implemented in a similar fashion to the one described by Hoffmann *et al.* [7]. A feedforward term $T_{nom} = mg$ is applied as a base thrust to cancel out the force due to gravity. Then a PID feedback controller is used to track to the desired height command. Finally, compensation for the roll and pitch boosts the thrust to ensure the desired component of thrust in the z_E direction is achieved in the case of non-zero roll and pitch. The resulting control law is

$$u_z = \frac{1}{\cos\phi\cos\theta}\left(K_{z,p}e_z + K_{z,i}\int e_z dt + K_{z,d}\dot{e}_z + T_{norm}\right), \tag{3.26}$$

where $K_{z,p}$, $K_{z,i}$, and $K_{z,d}$ are the proportional, integral, and derivative gains for the height controller, respectively, and the error e_z is difference between the commanded height and the actual height.

Chapter 4

Experimental Setup and Results

The framework presented in this thesis was tested fully in simulation and partially implemented on an experimental testbed. The focus of the experimental implementation is on the relative estimation and control for the target tracking stage of the landing. GPS based quadrotor flight has already been demonstrated in many other works and we employ existing methods in this work.

Flight tests were performed on an AscTec Pelican quadrotor, detailed in Section 4.1. Relative position estimates in the target tracking stage are driven by visual measurements of an AprilTag fiducial marker. Special considerations in implementing and tuning the AprilTag detection pipeline are discussed in Section 4.2. Finally, simulated and experimental results are presented and discussed in Section 4.3. The experimental results were obtained through flight tests both indoors and outdoors. The tests demonstrate successful target tracking in both the indoor and outdoor environments as well as a successful landing on a moving UGV in an indoor setting.

4.1 Quadrotor Testbed

Flight tests were performed on an AscTec Pelican quadrotor helicopter, shown in Figure 4.1a. The Pelican features an Intel Atomboard, which is a 1.6 GHz, dual-core, single-board computer. This gives the Pelican enough processing power to perform vision based tasks in near real-time. A Ubuntu operating system is installed on the Atomboard. All of the custom software runs in a Robot Operating System (ROS) environment in Ubuntu.

(a) An AscTec Pelican quadrotor

(b) The sonar and downward facing camera mounted on the bottom of the Pelican

Figure 4.1: **(a)** An AscTec Pelican quadrotor was used as the testbed for experimental implementation of the framework described in this thesis. **(b)** A sonar and a downward facing camera, with a 180° fisheye lens, were mounted on the bottom of the vehicle.

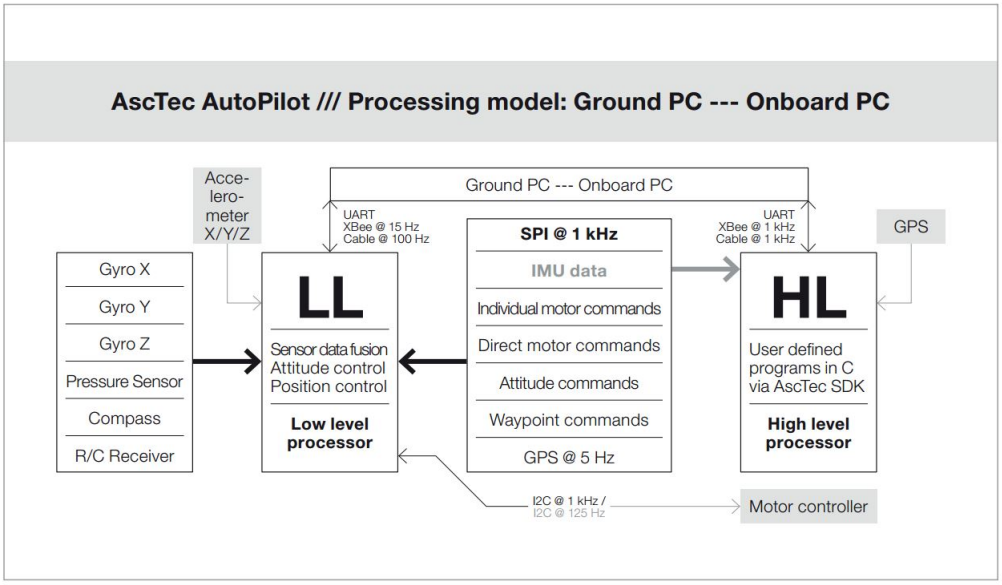


Figure 4.2: Hardware architecture overview for the Pelican quadrotor [24].

In addition to the Atomboard, the Pelican contains dual microcontrollers integrated into a flight control unit (FCU). The architecture of the Pelican is shown in Figure 4.2. The two microcontrollers are the low-level processor and the high-level processor. The low-level processor runs a closed-source firmware which performs all the tasks required for basic flight and stability. The high-level processor is user programmable to perform tasks that are not too computationally expensive.

All of the sensor fusion, estimation, and control that is required for basic flight is performed on the low-level processor. This includes the attitude control and attitude estimation which are leveraged in this work. The low-level controller also provides position level control, however it must be run in the attitude control mode in order to implement the estimation framework proposed in this work. For roll and pitch control, the closed-source firmware implements a well tuned PD controller to track desired commands quickly. The low-level processor takes care of sending motor speed commands to the motor controllers in order to achieve the desired torques and collective thrust. The interface exposed by the low-level controller requires four commands: roll, pitch, yaw rate, and collective thrust.

The high-level processor is left open to be programmed to perform custom tasks. For this work, the `asctec_mav_framework` firmware was loaded on the high-level processor.

The sensors included with the Pelican on the FCU are: 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer, and barometer. A Maxbotix MB 1240 sonar and uEye UI-1221LE camera are mounted externally on the quadrotor, shown in Figure 4.1b, and interface directly with the Atomboard through USB. Further details on the camera are provided in Section 4.2.

Asctec MAV Framework and ROS

ROS is a modular framework for building robotic software that allows for quick prototyping and development. It is a popular framework in the research community since it facilitates code reuse. In order to communicate with the Pelican through ROS, the `asctec_mav_framework`, released by researchers at ETH Zurich, was used [1].

The `asctec_mav_framework` is a ROS driver written specifically for use with the AscTec line of multirotor vehicles. The `asctec_mav_framework` includes custom firmware that must be flashed onto the high-level processor so that the high-level processor acts as a communication bridge between the low-level processor and the Atomboard. Through the `asctec_mav_framework`, sensor measurements and attitude estimates on the FCU can be read on the Atomboard at up to 1 kHz. Attitude commands can also be issued at up to 1 kHz, although such a rate is not practical.

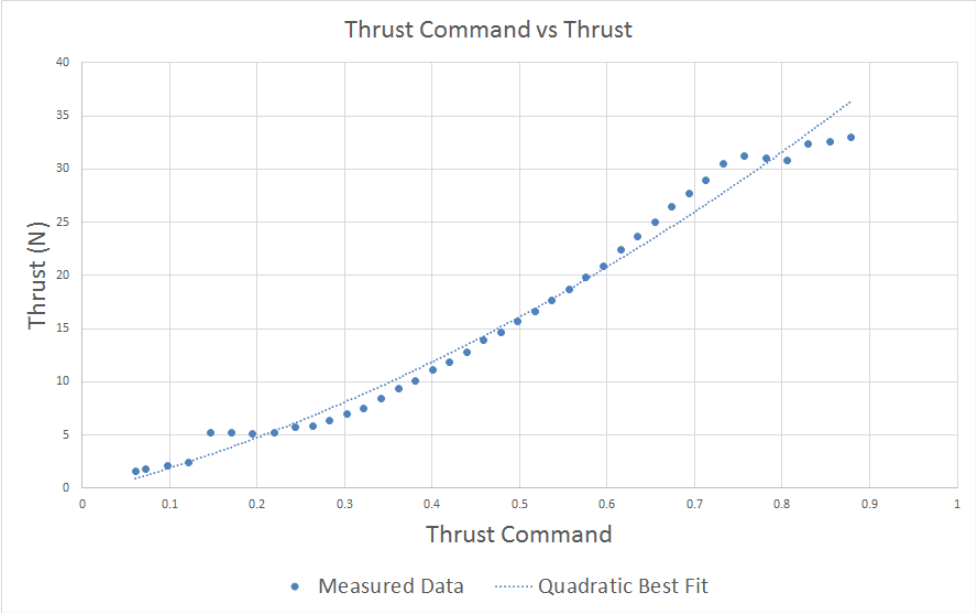


Figure 4.3: Thrust testing data showing the generated thrust vs thrust command.

Thrust Mapping

The `asctec_mav_framework` accepts thrust commands between 0 and 1. These commands are then interpreted by the low-level processor to generate a collective thrust. In order to send meaningful thrust commands to the low-level processor, thrust testing was performed to determine the mapping from thrust command to generated thrust.

The thrust testing was performed by attaching weights to the Pelican and placing it on a scale. The amount of weight added was well beyond the maximum payload of the vehicle to prevent it from moving at all thrust levels. The weight of the Pelican with all the additional weight was recorded. As various thrust commands were issued to the quadrotor, the change in the measured weight was recorded. The measured change in weight times the acceleration due to gravity gives the amount of thrust produced for each thrust command. The results of the thrust testing are shown in Figure 4.3. Excel was used in order to generate a quadratic best fit function for the measured data. The resulting thrust mapping is

$$u_z = 23.133u_{z,c}^2 + 21.646u_{z,c} - 0.492, \tag{4.1}$$

where $u_{z,c}$ is the thrust command sent to the Pelican's low-level processor. This thrust mapping is rearranged in order to calculate the thrust command, $u_{z,c}$, to correspond to a

desired thrust, u_z, produced by the height controller described in Section 3.6.

4.2 AprilTag Pose Measurement

Using a uEye UI-1221LE camera with a wide angle lens, monochrome images are captured with a 180° field of view and a 752 × 480 resolution. The UI-1221LE features a global shutter which greatly reduces the amount of image distortion due to the movement of the camera. The camera is set to capture images at 20 Hz for AprilTag detection. Three special considerations were made in order to ensure the quality of the visual measurements used for estimation,

- The camera exposure and edge detector were tuned to detect the AprilTag and ignore most other edge features.
- Images are windowed before processing in order to speed up AprilTag detection.
- Camera delays are calibrated and accounted for to match an image with the correct quadrotor orientation at the time the image was taken.

4.2.1 Camera Exposure and Canny Thresholds

The camera is manually tuned to run with the lowest exposure that still produces usable contrasts for edge detection. There are two main rationales for choosing a short camera exposure. First, longer exposure times can result in motion blur which is undesirable. Secondly, shorter exposure times result in darker images which tend to have fewer detectable features.

The camera's exposure time is the duration that the aperture is opened, allowing light to fall on the camera's sensor. Movement while the aperture is open causes the image focused on the camera sensor to change while the image is recording, resulting in a distorted image. Significant amounts of motion blur causes the AprilTag detection algorithm to fail to detect the target since the quadrilaterals of the AprilTag cannot be detected in a distorted image. Any image frame that has motion blur will result in a failed target detection. This results in a reduced measurement update rate which is detrimental to state estimation.

Reducing the exposure time also allows less light to fall on the camera's sensor, resulting in a darker image. Darker images tend to have a lower contrast ratio, making natural

(a) Image of an AprilTag taken mid flight (b) Edges detected in the AprilTag image

Figure 4.4: **(a)** A picture of an AprilTag taken mid flight. The image is very dark, but sufficient for detecting the AprilTag. **(b)** A picture of an AprilTag with correctly tuned Canny thresholds. No edges other than the AprilTag appear.

features more difficult to detect. The AprilTag is made of a black shape on a white background to give the highest possible amount of contrast so it will still be visible in an image where no natural features can be extracted. This makes it simpler to tune the Canny edge detector thresholds to detect the edges of the AprilTag and as few additional edges as possible.

Care must be taken to not set the exposure time too low or the image may be too dark for the AprilTag to be distinguishable. Indoors, an exposure of 2 ms was found to be a good balance so that there is no motion blur and the AprilTag provides the only easily distinguishable edges. Outdoors, the camera is run at its minimum exposure, 0.16 ms. The Canny edge detector thresholds were then manually tuned so that they consistently detect the AprilTag. Figure 4.4a shows an image taken indoors while in flight. The low exposure makes the image dark but the AprilTag is still noticeable. Figure 4.4b shows the result of performing edge detection on the image in Figure 4.4a. The edges of the target are the only edges detected.

4.2.2 Image Windowing

At full resolution, the Atomboard does not have enough computational power to run the AprilTag detection algorithm at 20 Hz, the frame rate at which the camera is set to capture images. Resizing the images to a resolution of 376×240, a quarter of the native resolution, allows the Atomboard to process AprilTag detections at the full 20 Hz. However, by resizing the image, the maximum detection range for the target is also reduced. Instead, image windowing was implemented so that only a region of interest would be processed for

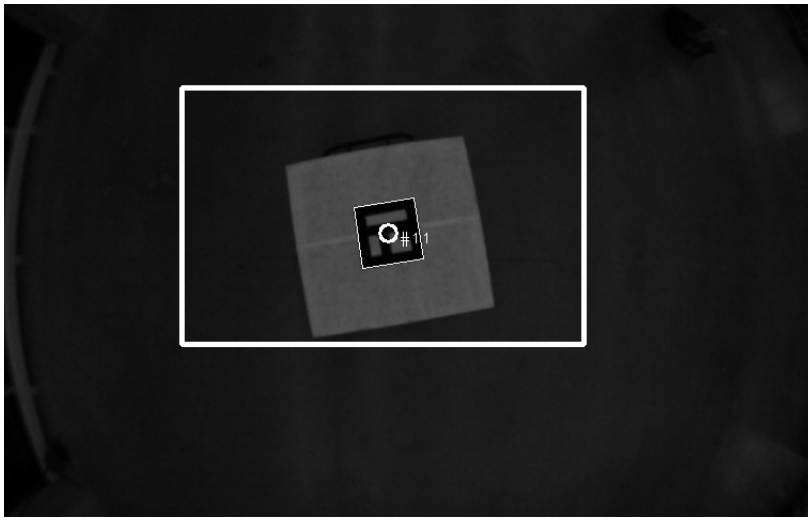


Figure 4.5: A window, or region of interest, designated by the white rectangle, is processed to find the AprilTag. The window is centred at a coordinate determined by rotating the estimated target position and reprojecting it into the image frame.

AprilTag detection.

The target's relative position estimate, described in Section 3.4, is used to determine where to place the detection window. The estimated target position, P_r, is rotated into the camera frame and reprojected into the image frame. This gives the pixel coordinate at which the detection window is centred. A window of 376×240 pixels is formed around the window centre.

Given the estimated position of the target, P_r, in the body-planar frame, Q_0, the position is rotated back into the camera frame,

$$\begin{bmatrix} P_{r,C} \\ 1 \end{bmatrix} = T_Q^C T_{Q_0}^Q \begin{bmatrix} P_r \\ 1 \end{bmatrix}. \tag{4.2}$$

The position $P_{r,C}$ is the relative position of the target in the camera frame, C. The OpenCV camera model [20] is used to reproject $P_{r,C}$ into the image frame. The equations

for reprojection are

$$x' = \frac{x_{r,C}}{z_{r,C}} \tag{4.3}$$

$$y' = \frac{y_{r,C}}{z_{r,C}} \tag{4.4}$$

$$r^2 = x'^2 + y'^2 \tag{4.5}$$

$$x'' = x' \left(1 + k_1 r^2 + k_2 r^4\right) + 2p_1 x' y' + p_2 \left(r^2 + 2x'^2\right) \tag{4.6}$$

$$y'' = y' \left(1 + k_1 r^2 + k_2 r^4\right) + p_1 \left(r^2 + 2y'^2\right) + 2p_2 x' y' \tag{4.7}$$

$$x_{image} = f_x x'' + c_x \tag{4.8}$$

$$y_{image} = f_y y'' + c_y, \tag{4.9}$$

where k_1 and k_2 are radial distortion coefficients, p_1 and p_2 are tangential distortion coefficients, f_x and f_y are the camera focal lengths, and c_x and c_y are the optical centre in pixel coordinates. All of the camera parameters were identified using the ROS `camera_calibration` tool. The reprojection equations result in the estimated position of the target in the image frame, $P_{image} = [x_{image}, y_{image}]^T$. The detection window will now be centred at P_{image}.

Two opposing corners are required to define the detection window. The upper-left, w_{UL}, and bottom-right, w_{BR}, corners of the window are chosen to be the representative corners for the rectangular detection window. They are simply

$$w_{UL} = P_{Image} - \begin{bmatrix} 188 \\ 120 \end{bmatrix} \tag{4.10}$$

$$w_{BR} = P_{Image} + \begin{bmatrix} 188 \\ 120 \end{bmatrix}. \tag{4.11}$$

Finally, the corners of the detection window are bounded to ensure that they are still within the image. In the process of bounding the corners, a check is performed to ensure that the size of the detection window is not reduced below a prescribed threshold. If either dimension of the resulting detection window is less than 85% of its original length, the detection window will not be applied and the entire image will be processed with the AprilTag detection algorithm. The result of the image windowing is shown in Figure 4.5.

4.2.3 Attitude Compensation for AprilTag Measurements

The AprilTag detection algorithm returns the 6-DOF pose of the target in the camera frame, T_M^C. This result needs to be rotated into the body-planar frame, Q_0, before it can

be used in the relative estimator presented in Section 3.4. As shown in Equation (2.26), the two required transformations are T_C^Q and $T_Q^{Q_0}$. The rotation of the first transformation can be determined by inspection and the translation from the FCU to the camera is roughly measured with a ruler. The resulting transformation is

$$T_C^Q = \begin{bmatrix} 0 & -1 & 0 & 0.06 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -0.09 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.12}$$

The second transformation, $T_Q^{Q_0}$, is a pure rotation but is more difficult to determine.

The transformation $T_Q^{Q_0}$ is obtained from the attitude estimates based on IMU measurements. Unfortunately, the camera images and attitude estimates are not synchronized. This poses a problem since the attitude dynamics are fast and can change significantly in a short period of time. Applying a transformation, $T_Q^{Q_0}$, on an visual pose measurement with a mismatched attitude estimate can produce an erroneous result. Thus, the attitude estimate that corresponds with the time at which the image was captured must be used in order to correctly determine the pose of the target in the body-planar frame.

There is a significant time delay associated with visual pose measurements. The total delay has been measured to be between 50 to 120 ms during flight. The delay comes from two main sources. First, there is a time delay from when the image is taken on the camera to the time that it is transferred to the Atomboard, through the camera driver, and accessible in a ROS based program. This delay is denoted as Δt_{camera}. Secondly, there is a delay while the computer runs the AprilTag detection algorithm, denoted as $\Delta t_{detection}$. This second delay is variable and depends on the current CPU load and the properties of the image being processed. If a larger number of edges are detected in an image, the AprilTag detection algorithm can take a longer time to run. The second time delay is easily measured through time stamps while the first is not easily measurable. If both time delays are known, the time at which the image was captured can be calculated by subtracting the delays,

$$t_{capture} = t - \Delta t_{detection} - \Delta t_{camera}, \tag{4.13}$$

where t is the current time at which the visual pose measurement is received. Then the attitude estimate closest to $t_{capture}$ can be used to construct $T_Q^{Q_0}$. Since Δt_{camera} is not easily measured, it is hand tuned through a calibration test. The delay Δt_{camera} is assumed to be constant since it is mostly determined by the hardware and camera drivers and should not change noticeably under load.

The calibration test is performed by having a human physically hold the quadrotor in their hands. The quadrotor is held directly over a target and rotated on the spot. Either a roll or a pitch angle is applied sinusoidally, while taking care to ensure that no translations are occurring. If the attitude estimate and visual pose measurement are synchronized, the relative position estimate in the horizontal plain will remain near $\bar{P}_r = [0, 0]^T$. The time delay, Δt_{camera}, is hand tuned until a reasonable synchronization is achieved. A value of 0.04 ms for Δt_{camera} was determined.

Clearly, the accuracy of the visual pose measurement relies heavily on the accuracy of the attitude estimate. Methods to improve the onboard attitude estimates [13, 11] can be added as a future extension to further improve the visual measurements.

4.3 Results

4.3.1 Simulated Results

All three segments of the proposed landing framework were simulated in MATLAB. The simulations validate the rendezvous trajectory planning and the performance of the relative position estimator in wind. Quadrotor dynamics in simulation are implemented as described in Section 2.2.1 with the addition of a first order motor model. Wind was simulated by applying a fixed base wind vector of $[3, -4, 0.2]^T$ metres/second and adding wind gusts on top. The wind gusts were generated using the Dryden wind gust model [29]. The simulation begins with the quadrotor far away from the target at an altitude of 10 m. The quadrotor moves through the three stages of the landing as it progresses towards, and ultimately lands on, the target. A 3-dimensional view of the simulated landing approach is shown in Figure 4.6.

In the first stage, the quadrotor calculates a rendezvous point to reach the target. GPS measurements for the quadrotor and the target are generated according to the sensor model presented in Section 2.3 and the noise and bias parameters are tuned to be a gradual random walk to a maximum of about 5 m error. GPS position updates are generated at 10 Hz with inertial control performed at the same rate. Throughout the simulation, the target follows an arc trajectory as an arbitrary and independent motion. A Kalman filter with a constant velocity model is used to smooth the position and estimate a velocity for the target. The rendezvous point is recalculated every 5 seconds and is only updated if it differs from the previous rendezvous point by more than 5 metres.

When the quadrotor is within a 15 m radius of the target, the target acquisition stage begins. The target is simulated as a square with 20 cm side lengths. A downward facing

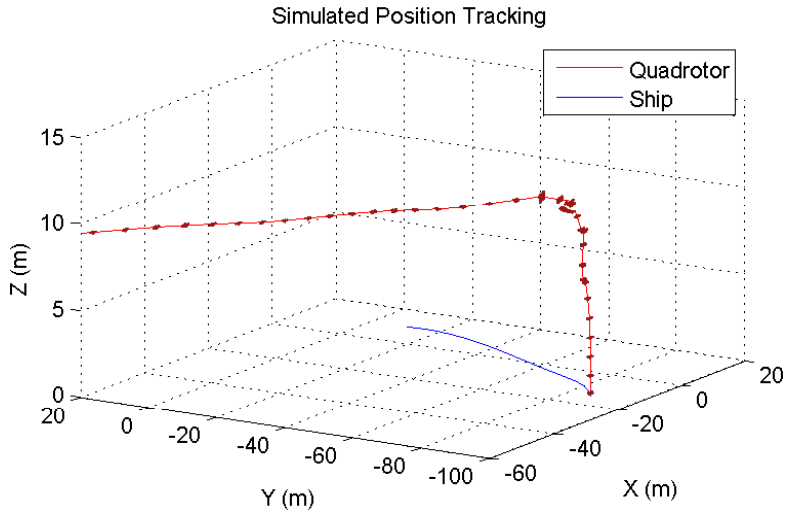


Figure 4.6: A 3D view of the simulated quadrotor and ship positions with the quadrotor orientation overlaid on top of its position.

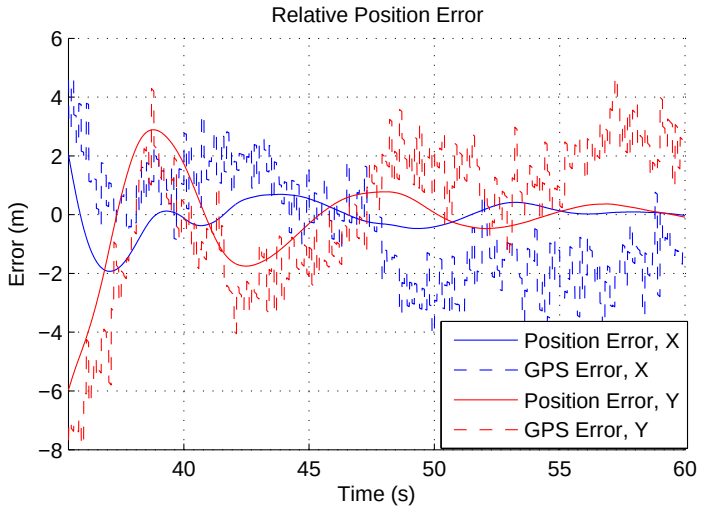


Figure 4.7: Simulated relative position errors between the quadrotor and target for the target acquisition and target tracking stages of the landing. Both the actual relative errors and the relative errors calculated from GPS are shown. The relative tracking stage begins at $t = 39.8$ seconds.

camera is simulated with a 65° field of view and a square aspect ratio. The view of the camera is rotated with the quadrotor and projected onto the ground plane to determine what the camera can see. If the four corners of the target fall within the view of the camera, the target is treated as detected and a relative pose measurement is generated. Gaussian noise with a standard deviation of 3 cm is applied to relative position measurements. At an altitude of 10 m, the target typically falls in the field of view of the camera without the need to fly a search pattern. For the flight shown, the target acquisition stage lasted 4.4 seconds.

As soon as the target is detected by the quadrotor, the target tracking stage begins. The control rate switches to 20 Hz to take advantage of the higher update rate and higher accuracy measurements. The gains, shown in Table 4.1, are also increased in the target tracking stage. Relative errors at the time that the quadrotor switches from inertial to relative control are shown in Figure 4.7. The Relative GPS errors show the position discrepancy between the GPS measurements and the actual positions.

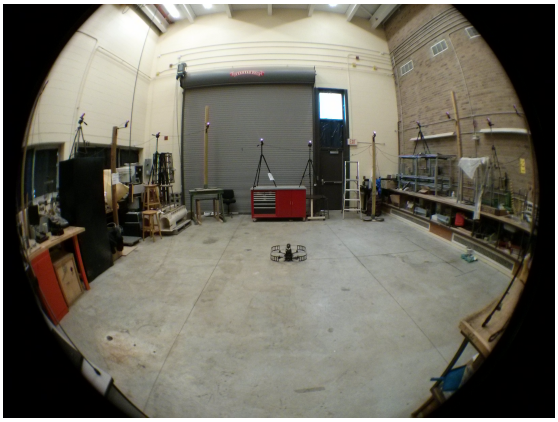
Gain	$K_{at,i}$	$K_{at,d}$	$K_{ct,p}$	$K_{ct,i}$	$K_{ct,d}$	$K_{E,p}$	$K_{E,i}$	$K_{E,d}$	$K_{r,p}$	$K_{r,i}$	$K_{r,d}$
Value	0.04	0.3	0.2	0.04	0.4	0.2	0.15	0.85	0.6	0.3	0.85

Table 4.1: Simulation controller gains

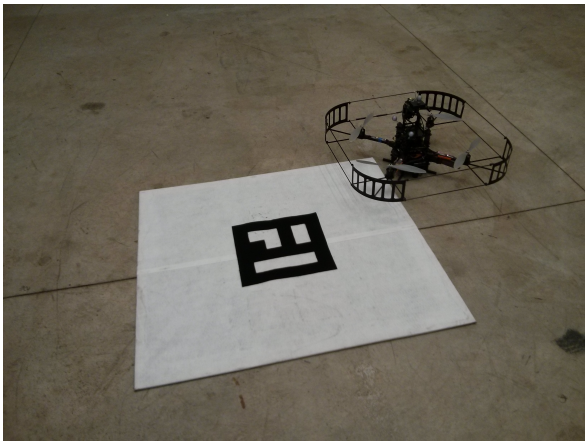
RMS Errors		
Rendezvous	Along Track (m/s)	1.1521
	Cross Track (m)	1.1845
Target Acquisition	X-Position (m)	1.1734
	Y-Position (m)	2.9955
Target Tracking	X-Position (m)	0.3382
	Y-Position (m)	0.7320
Quadrotor GPS	X-Position (m)	2.4769
	Y-Position (m)	2.0442
Magnetometer	Yaw (°)	6.8675

Table 4.2: RMS errors in simulation for the three stages of the landing, GPS measurements throughout the simulation, and magnetometer measurements throughout the simulation.

RMS errors associated with each stage of flight are presented in Table 4.2. As the system enters the relative tracking stage, the positioning error drops significantly. The results of the simulation show that the tracking performance is still good in the presence



(a) Indoor flight testing arena



(b) Target used for flight tests

Figure 4.8: **(a)** A flight arena with motion capture cameras was used for initial indoor tests to tune and validate control and estimation. **(b)** A 30 cm × 30 cm AprilTag made from felt is used for the majority of test flights.

of simulated wind. This demonstrates that the comprehensive solution presented in this thesis has a strong potential to work on a physical platform.

4.3.2 IPS Test Flights

Test flights were performed in an indoor flight arena with an indoor positioning system (IPS). The IPS was used in order to evaluate relative estimation and control schemes with proper ground truth. An OptiTrack motion capture camera system, with 14 cameras placed in a ring around the testing area, served as the IPS. The setup provided a 3 m × 3 m × 2.5 m flight volume. A photograph of the flight arena is shown in Figure 4.8a.

With the exception of the target orientation measurement tests, all tests used a 30 cm × 30 cm AprilTag shown in Figure 4.8b. The AprilTag was handmade from a felt material to eliminate glare from reflections. The centre of the AprilTag was aligned with the origin of the IPS system prior to testing and the target was left stationary so that all relative movements between the target and the quadrotor would be due to the quadrotor.

Initial tests were aimed at evaluating the quality of the relative position and velocity estimates. Tests were also performed to determine how well the AprilTag detection system could measure the orientation of a moving target in flight. Once these evaluations

were complete and the estimation and control systems were confirmed to be performing sufficiently, autonomous landings were tested.

For all tests, the IPS measurements that serve as ground truth were logged offboard. The timing between the onboard and offboard measurements were synchronized manually.

Manual Evaluation Flights

Flights were performed with manual position control to collect evaluation data. The height controller was enabled and commanded to hold an altitude at 1.8 m. The relative position estimates from one the flights are shown in Figure 4.9. Relative velocity estimates for the same flight are shown in Figure 4.10. Ground truth for the relative velocity was obtained by performing a single difference on the position ground truth and applying a central average smoothing filter.

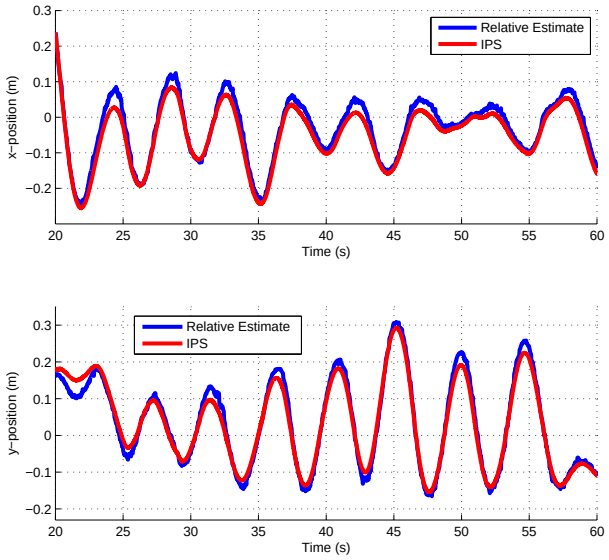


Figure 4.9: Plot showing the relative position estimates and the ground truth data for a test flight with manual position control and a stationary target.

The relative position estimates closely resemble the ground truth. There is a slight scaling issue where the amplitude of the movement as reported by the relative estimator is greater than the amplitude measured by the IPS. This error could be a result of quadrotor

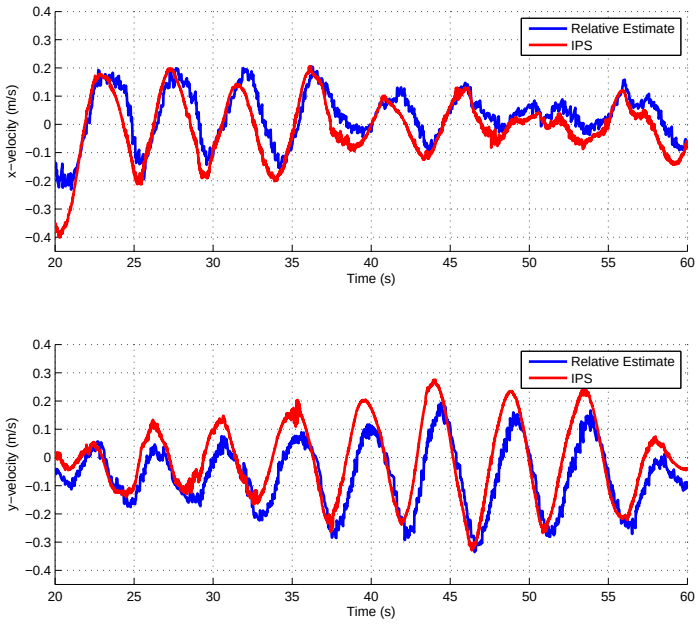


Figure 4.10: Plot showing the relative velocity estimates and the ground truth data for a test flight with manual position control and a stationary target.

attitude estimation errors when the orientation of the quadrotor is rapidly changing or the result of a calibration error. The relative velocity estimates closely resemble the trends in the ground truth, but with additional noise and considerable delay. The delay is measured to be approximately 460 ms. Efforts to tune the covariances of the relative estimator produced more noise and worse velocity estimation in terms of following the correct velocity trend. While delay will set an upper bound on the bandwidth of the position controller, subsequent tests in Section 4.3.3 demonstrate successful landings in the presence of the delayed velocity estimates.

Target Orientation Measurements

A test evaluating the AprilTag detection algorithm's ability to measure the orientation of a moving target in flight were performed with a tag printed on paper with 15 cm side lengths. The tag was mounted on a pole and moved around by a human. The tag changed both position and orientation throughout the flight and its position was tracked autonomously

by the quadrotor. The resulting measurements and ground truth for this test are presented in Figure 4.11. All of the measurements have been rotated into the body-planar frame of the quadrotor. The results of the target orientation measurement tests show that the AprilTag detection algorithm is very good at measuring the yaw of the target but has significantly more error when measuring the roll and pitch of the target. Roll and pitch tend to be more difficult to discern from a visual measurement since they cause the target to appear skewed in the image. At larger roll and pitch angles, an additional degree of roll or pitch creates a much smaller pixel change that a one degree change when the target is at a neutral roll and pitch. Changes in yaw cause a much larger change in pixels since all of the features of the tag move with it. The results in Figure 4.11 are presented without any estimation. Estimation could be applied on top of the measurements to come closer to the ground truth.

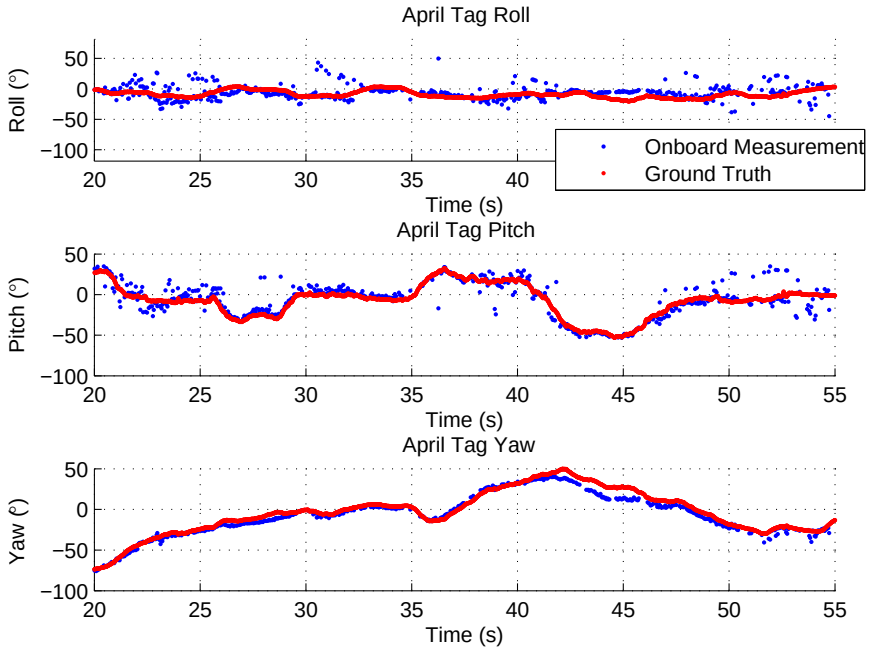


Figure 4.11: Measured AprilTag attitude in the body-planar frame and the ground truth measurements of the target's orientation rotated into the quadrotor's body-planar frame.

While the framework presented in this thesis does not make use of the visually measured target orientation, these measurements can be used to improve the framework in future works. The measurements provide information on the orientation of the landing surface. In

the maritime landing scenario, such information is useful to determine safe landing periods. Another possible extension would be to use the landing surface's orientation to plan an aggressive landing trajectory such as proposed by Dougherty *et al.* [3].

4.3.3 Indoor Landing Tests

Stationary Target Landing in IPS

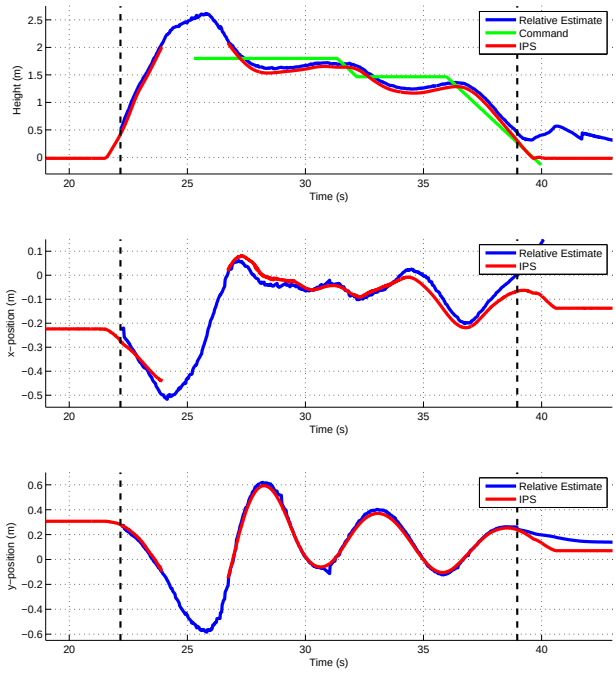


Figure 4.12: Plot showing the relative position estimates and the ground truth data for an autonomous landing on a stationary target. Dotted black lines at $t = 22$ seconds and $t = 39$ seconds indicate the times at which visual measurements started and stopped. The gap in IPS data is due to the quadrotor temporarily leaving the volume of the motion capture setup.

Landing tests were first performed in the flight arena shown in Figure 4.8a with the motion capture IPS system. The target was left stationary to verify that the descent part

of the final target tracking stage worked without wind disturbances or target motion. The error threshold to begin the descent was set to 30 cm. When the relative distance between the quadrotor and the target exceeds 30 cm, the quadrotor's height command is frozen. Once a height command is frozen, the quadrotor continues to track the target until the relative error is once again below the threshold, at which point the height command will continue to decrease. A descent rate of -0.4 m/s was chosen as a quick and safe descent rate.

The estimated relative position and ground truth for the stationary target landing are shown in Figure 4.12. Takeoff and the initial positioning near the target are performed manually before engaging autonomous flight. The green `Command` line on the height plot of Figure 4.12 indicates when the quadrotor is under autonomous control. The dotted black lines indicate where visual measurements begin and end, due to the quadrotor being too low view the complete target. After the second dotted black line the relative estimates are from propagating the predictive model, which will incur significant drift. However, they are still displayed in Figure 4.12 since those estimates are used for position control until the quadrotor has come to a stop on the ground. The results show that the quadrotor lands well within the 30 cm error threshold for descent. The landing is complete when the green `Command` line ends.

As the quadrotor approaches the ground around the 39 second mark, the relative height estimate begins to falter since visual height measurements are no longer available. This can likely be corrected through tuning the covariances on the relative height estimator or switching to an inertial height estimate for control for the final few seconds of the landing.

Indoor Moving Target Landing

Once successful landings were demonstrated on a stationary target, testing was moved to a larger area to perform testing with a moving target. A photograph of the test area is displayed in Figure 4.13. A manually driven Clearpath Husky UGV was used to carry and move the landing target. The UGV was driven in an arc and then a straight line for the landing. A video of the experiment is available at http://youtu.be/oycwswSWEB8.

The takeoff and initial positioning of the quadrotor over the tag were performed manually. The relative position measurements throughout the flight and autonomous landing are presented in Figure 4.14. The green `Command` line on the height plot of Figure 4.14 indicates when the quadrotor is under autonomous control and the black dotted lines indicate when visual measurements are available. The mean of the absolute error was calculated for the time interval between 29 and 43 seconds, which is the tracking segment of the flight

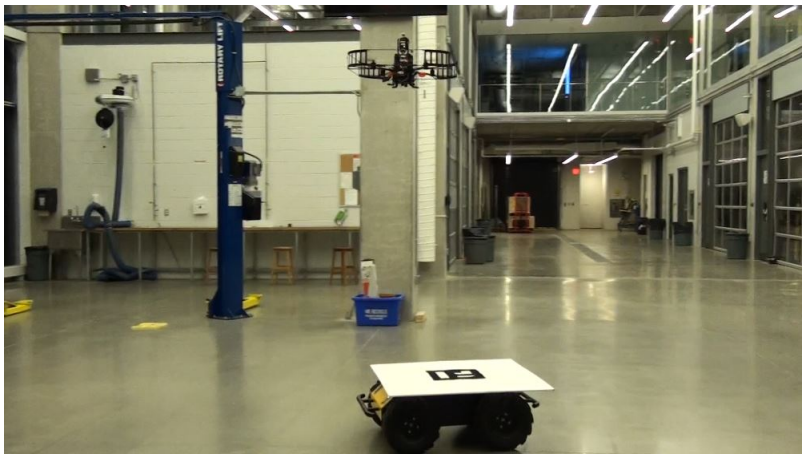


Figure 4.13: Indoor flight test with a moving target.

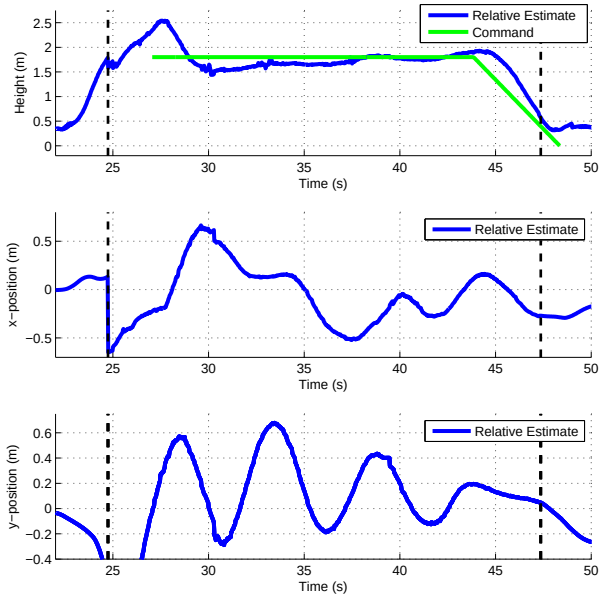


Figure 4.14: Plot showing the relative position estimates for an autonomous landing on a moving target. Dotted black lines at $t = 24.7$ seconds and $t = 47.4$ seconds indicate the times at which visual measurements started and stopped.

prior to the descent. The average error in the x direction is 0.2812 m and the average error in the y direction is 0.2411 m. The norm of the average errors is 0.3705 m.

The plots show that the quadrotor was able to maintain within a reasonable following distance of the target throughout the flight. The large sways in the y-position plot are due to the arc movement of the UGV. While moving through an arc, the UGV has a constant acceleration, which breaks the constant velocity assumption made about the target dynamics. This makes it more difficult for the estimator to accurately track the changing velocity. The quadrotor does not know the movements the UGV will make and must accumulate error before it can react to changes in the UGV motion. Once the UGV begins to move in a straight line, the quadrotor tracks the tag more accurately and it begins the landing.

4.3.4 Outdoor Moving Target Tracking

Figure 4.15: An open field where outdoor tests were performed.

The final set of tests were performed in an outdoor environment shown in Figure 4.15. Attempts at an autonomous landing outdoors were hindered by wind disturbances when the quadrotor was at a low altitude on the descent. At lower altitudes, the area that the camera observes is much smaller, making it difficult for the quadrotor to regain sight of the target after a disturbance forces the target out of view. On several test flights, the wind consistently pushed the quadrotor away from the target while the quadrotor was

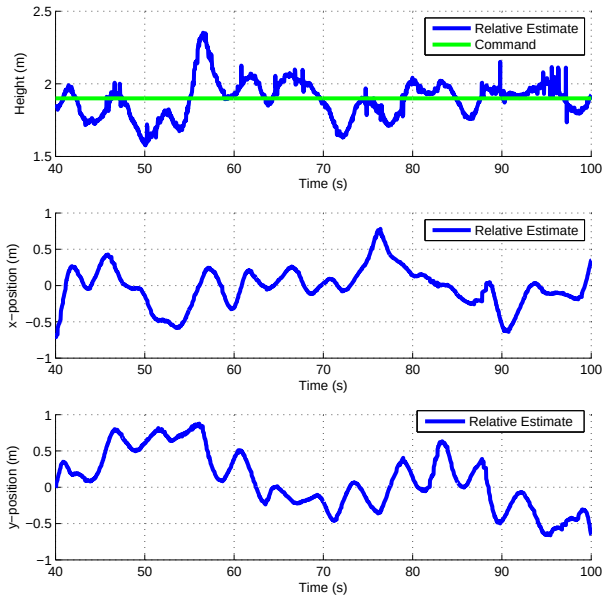


Figure 4.16: Plot showing the relative position estimates during an autonomous target tracking flight.

at a low altitude, causing the target to be lost. The quadrotor was unable to recover tracking and was forced to return to the starting altitude and begin the descent over again. A more representative dynamic model that accounts for wind would improve the quadrotor's ability to recover in such situations. Although the wind caused problems for the landing sequence, the autonomous position tracking of a moving target, at a higher altitude, remained successful in the presence of wind. Winds were in the range of 5 to 10 km/h during the experiment. A video of the outdoor test is available at http://youtu.be/oycwswSWEB8?t=44s.

Takeoff and landing were performed manually for the outdoor test. During autonomous flight, the quadrotor was commanded to maintain a relative height of 1.9 m. Figure 4.16 shows the relative position estimates during the autonomous tracking segment of the flight. The mean of the absolute error was calculated for the entire flight segment shown in Figure 4.16. The average error in the x direction is 0.1991 m while the average error in the y direction is 0.3339 m. The norm of the average errors is 0.3888. Thus, the errors in target tracking outdoors are comparable to the errors in target tracking indoors. The height control, on the other hand, has much more difficulty remaining near the set point. This

is potentially due to the changing air pressure causing the barometer to report inaccurate changes in height. Performing covariance tuning in the height estimator to reduce reliance on the barometer readings may be able to improve the height control performance.

Unfortunately, there is no ground truth in the outdoor environment to evaluate the relative velocity estimates against. The extent to which the relative velocity estimates are affected by wind is unknown. Work is currently underway to integrate an RTK GPS system onto the quadrotor testbed. This would allow for a proper evaluation of the relative estimator in the presence of wind as a future endeavour.

Chapter 5

Conclusion

With unmanned air vehicle (UAV) use becoming widespread, a number of current and proposed applications have raised a demand for precision landing on moving targets. In the case of maritime landings, autonomous landings will make for easier UAV recovery when deploying from a ship. This will be beneficial to help guarantee safe recovery of equipment and to eliminate risk of injury posed to humans from current maritime UAV recovery techniques. On land, UAV and unmanned ground vehicle (UGV) teams can benefit from the increased landing autonomy to perform autonomous docking.

Performing a precision landing is difficult for several reasons. Low-cost sensors common to many UAV systems have significant errors. In particular, GPS receivers are subject to a large random walk bias and magnetometers receive interference from many of the electrical components on UAVs. The unpredictable movement of a non-stationary landing target can also exacerbate the problem. Current state of the art systems rely on high quality measurements from expensive inertial measurement units and RTK GPS, however it is not practical to use expensive sensors on all UAV systems. Since many UAVs carry a camera as their payload, vision is a natural alternative to provide higher quality pose estimates.

This thesis presents a framework for performing precision autonomous UAV landings by using a downward facing camera for target detection. The benefits of this framework are: no additional or expensive sensors are required on the UAV, the autonomous landing framework is independent of the yaw control leaving the possibility for a forward facing camera to be used during the landing, and the landing target can be placed on any flat surface as no assumptions are made about the target's dynamics. The landing is performed in three stages. The first stage, rendezvous, calculates a course for the UAV to intercept the target. The second stage, target acquisition, ensures that the UAV is locked onto

the target. The final stage, target tracking, positions the UAV above the landing target and commands it to land when the relative error is sufficiently small. Control laws are presented for all stages of flight and a novel relative estimation scheme is given for the target tracking stage.

Experimental data collected from a series of test flights has validated the autonomous landing approach presented. The AprilTag visual fiducial system was selected for use as the moving target in the experiments. Successful target tracking and landing was documented in an indoor environment. When tested outdoors, the target tracking was successful while the autonomous landings encountered problems with unpredictable disturbances. Wind gusts cause disturbance forces for the quadrotor and make the position error too high to recover from while in the critical final descent. While the effects of wind on the vehicle must be addressed prior to full validation of the methodology, this work presents both simulation results for the whole framework and individual experimental validation of key components.

Several strategies can be pursued to improve the performance in wind. A proper wind estimation technique coupled with a control law that actively rejects wind disturbance forces will help minimize the effect of wind on the quadrotor. The wind estimation opens the door to using higher order dynamic models that take into account additional non-linear effects such as bladeflapping and rotor drag. A more representative dynamic model will improve the ability of the relative estimator to predict motions over short periods of time. This in turn allows the quadrotor to make better predictions of its position when visual measurements are lost due to wind. With decent position predictions, thee quadrotor can navigate back to and regain sight of the target, and then finish the landing. While the indoor landings presented did not have any problems due to the use of a single target, the addition of a second, smaller target will also help keep the target in sight during the descent since it will fit in the camera's field of view at low altitudes.

The novel relative estimation scheme was presented and validated in this work. The state estimates produced were adequate for tracking and, in indoor flights, landing. However, further improvements can be made to the relative estimation that will result in improved tracking and landing performance. Improved attitude estimates will ensure that visual target measurements are correctly rotated into the body-planar frame for estimation. Works in the literature suggest that this can be accomplished through the use of accelerometers and a drag-force model [11, 13]. The addition of a more powerful computer for vision processing would also result in tangible gains. More computational power will allow target detections to occur at a greater rate and with less computational delays. This could help to reduce the delay in the relative velocity estimates.

For some time, researchers have been proposing applications for UAV and UGV teams. Many of these applications have autonomous docking as a prerequisite. It is required in order to autonomously deploy and retrieve a UAV at a long range or to perform autonomous charging in the field by using a UGV as a mobile charging station. Both of these tasks are required for large-scale autonomous robotic missions. The framework presented in this thesis has the potential to be a viable solution to the autonomous docking problem. With the improvements discussed, the framework will be more robust. That will take it another step closer to deployment and provide another stepping stone to reach the goal of fully autonomous large-scale UAV and UGV teams.

References

[1] M. Achtelik and ETH Zurich. `asctec_mav_framework`. http://wiki.ros.org/asctec_mav_framework, 2014.

[2] J. M. Daly, Y. Ma, and S. L. Waslander. Coordinated landing of a quadrotor on a skid-steered ground vehicle in the presence of time delays. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4961–4966, San Francisco, CA, Sept 2011.

[3] John Dougherty, Daewon Lee, and Taeyoung Lee. Laser-based guidance of a quadrotor UAV for precise landing on an inclined surface. In *American Control Conference*, Portland, OR, 2014.

[4] M. Dreier. *Introduction to Helicopter and Tiltrotor Flight Simulation*. American Institute of Aeronautics and Astronautics, 2007.

[5] M. Garratt, H. Pota, A. Lambert, S. Eckersley-Maslin, and C. Farabet. Visual tracking and LIDAR relative positioning for automated launch and recovery of an unmanned rotorcraft from ships at sea. *Naval Engineers Journal*, 121(2):99–110, 2009.

[6] M. Hardesty, S. Kennedy, S. Dixon, T. Berka, J. Graham, and D. Caldwell. Development of navigation and automated flight control system solutions for maritime VTOL UAS operations. In *Association for Unmanned Vehicle Systems International's Unmanned Systems North America 2012*, Las Vegas, NV, 2012.

[7] 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*, 19(9):1023–1036, 2011.

[8] Aeryon Labs Inc. Aeryon labs: Commercial & industrial applications. http://www.aeryon.com/applications/commercial.html, 2014.

[9] Michael Kaess. Apriltags c++ library. http://people.csail.mit.edu/kaess/apriltags/, 2013.

[10] J. Kim, Y Jung, D. Lee, and D. Shim. Outdoor autnomous landing on a moving platform for quadrotors using an omnidirectional camera. In *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, Orlando, FL, 2014.

[11] Robert C Leishman, John Macdonald, Randal W Beard, and Timothy W McLain. Quadrotors and accelerometers: State estimation with an improved dynamic model. *IEEE Control Systems*, 34(1):28–41, 2014.

[12] Y. Ma. Coordinated landing and mapping with aerial and ground vehicle teams. Master's thesis, University of Waterloo, 2012.

[13] Philippe Martin and Erwan Salaun. The true role of accelerometer feedback in quadrotor control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1623–1629. IEEE, 2010.

[14] Neil Mathew, Stephen L Smith, and Steven L Waslander. A graph-based approach to multi-robot rendezvous for recharging in persistent tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3497–3502, Karlsruhe, Germany, 2013. IEEE.

[15] Maxbotix. Maxsonar operation on a multi-copter. http://www.maxbotix.com/articles/067.htm, 2013.

[16] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk. Comprehensive simulation of quadrotor UAVs using ROS and Gazebo. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 400–411. Springer, 2012.

[17] A. Mokhtari and A. Benallegue. Dynamic feedback controller of Euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2359–2366, New Orleans, LA, Apr 2004.

[18] P. Mukherjee and S. Waslander. Direct adaptive feedback linearization for quadrotor control. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Minneapolis, MN, 2012.

[19] Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407, Shanghai, China, May 2011.

[20] OpenCV. Camera calibration with opencv. http://docs.opencv.org/trunk/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html, 2014.

[21] E. Pinto, P. Santana, and J. Barata. On collaborative aerial and surface robots for environmental monitoring of water bodies. In *Technological Innovation for the Internet of Things*, pages 183–191. Springer, 2013.

[22] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme. Vision-based autonomous landing of an unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2799–2804, Washington, DC, May 2002.

[23] O. Shakernia, Y. Ma, T.J. Koo, J. Hespanha, and S. Sastry. Vision guided landing of an unmanned air vehicle. In *IEEE Conference on Decision and Control (CDC)*, pages 4143–4148, Pheonix, AZ, Dec 1999.

[24] Ascending Technologies. Catalogue 2014: Asctec research line. http://www.asctec.de/downloads/flyer/AscTec-Research-Line_Catalogue_2014.pdf, 2014.

[25] M. J. Tribou, A. Akhtar, and S. L. Waslander. Relative position-based visual servoing control for quadrotors. In *AIAA Guidance, Navigation and Control Conference (GNC)*, Minneapolis, MN, Aug 2012.

[26] T. K. Venugopalan, T. Taher, and G. Barbastathis. Autonomous landing of an unmanned aerial vehicle on an autonomous marine vehicle. In *IEEE/MTS Oceans*, pages 1–9, Hampton Roads, VA, Oct 2012.

[27] H. Voos and H. Bou-Ammar. Nonlinear tracking and landing controller for quadrotor aerial robots. In *IEEE International Conference on Control Applications (CCA)*, pages 2136–2141, Anchorage, AK, May 2010.

[28] G. Walker. Augmenting steller sea lion surveys in the Western Aleutians with unmanned aircraft. Technical Report North Pacific Research Board Project Number 1120, 2013.

[29] S. L. Waslander and C. Wang. Wind disturbance estimation and rejection for quadrotor position control. In *AIAA Infotech@Aerospace Conference and AIAA Unmanned Unlimited Conference*, Seattle, WA, Apr 2009.