University of Alberta

Experimental Nonlinear Control of a Helicopter Unmanned Aerial Vehicle (UAV)

by

Bryan Godbolt

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Control Systems

Department of Electrical and Computer Engineering

Fall 2013
Edmonton, Alberta

To Kristen

Abstract

Helicopter Unmanned Aerial Vehicles (UAVs) present a challenging control problem since their dynamics are nonlinear, underactuated and non-minimum phase. Although it is inherently an applied research field, due to the difficulty of building and maintaining an experimental platform relatively few experimental results exist in the literature. The approach followed in this thesis is to combine rigorous analysis with thorough experimental testing. This testing ensures validity of the designs. We present our experimental platform which is designed to be flexible so that it can accommodate nonlinear control research. Existing accounts of helicopter testbeds focus on hardware details. Since autopilot software design and development also requires a significant investment, we describe our implementation which has been released as open source for the benefit of the community. Due to the intractability of existing helicopter models, many control designs use non-physical inputs. We propose simple, invertible expressions relating the non-physical inputs to the physical inputs. In particular, modelling of the main rotor typically results in complicated expressions with extensive state dependence. While it is unlikely that angular velocity has a significant influence on the thrust, we show using experimental results that previous attempts to simplify this expression using a hover assumption are invalid during vertical flight. The platform is validated using a model-based PID control law. This control is derived using passivity to ignore nonlinear terms which do not affect stability. Among the class of vehicles with similar flight capabilities, helicopters possess a coupling between the rotational inputs and translational dynamics which is unique. This coupling is sometimes referred to as the Small Body Force (SBF) and is ignored in the literature for controller synthesis. We derive an

experimentally-validated control design which accounts for the effect of the tail rotor in the SBF. In addition, we show why the contribution of the main rotor flapping in the SBF cannot be compensated using the same approach, and give a robustness analysis of their effect on the closed-loop. Finally, based on recent results we propose a control design which accounts for state constraints by enforcing bounds on translational velocity and roll-pitch travel.

Table of Contents

List of Tables

List of Figures

List of Symbols

$(\cdot)^\vee$	Inverse of $\hat{\cdot}$	
$\mathcal{B}$	Body-fixed coordinate frame, spanned by $\{e_1', e_2', e_3'\}$	
δ_M	Main rotor collective servo pulse width	[μs]
δ_p	Normalized longitudinal swashplate tilt angle	
δ_r	Normalized lateral swashplate tilt angle	
δ_T	Tail rotor collective servo pulse width	[μs]
η	Euler angle (ϕ, θ, ψ) parametrization of R	[rad]
$\hat{\cdot}$	Isomorphism of $\mathbb{R}^3$ onto $\mathfrak{so}(3)$	
$\mathcal{N}$	Inertial navigation frame, spanned by $\{e_1, e_2, e_3\}$	
ω	Angular velocity of the helicopter	[rad/s]
Ω_M	Main rotor speed	[rad/s]
Ω_T	Tail rotor speed	[rad/s]
ϕ	Roll angle	[rad]
ψ	Yaw or heading angle	[rad]
θ	Pitch angle	[rad]
Θ_M	Main rotor collective pitch	[rad]
Θ_T	Tail rotor collective pitch	[rad]
a	Longitudinal main rotor flapping angle	[rad]
b	Lateral main rotor flapping angle	[rad]
J	Inertia matrix	[kg m^2]
K_d^r	Rotational control derivative gain matrix	[s^{-1}]
K_d^t	Translational control derivative gain matrix	[kg/s]
K_i^r	Rotational control integral gain matrix	[s^{-3}]

K_i^t	Translational control integral gain matrix	[kg/s^3]
K_p^r	Rotational control proportional gain matrix	[s^{-2}]
K_p^t	Translational control proportional gain matrix	[kg/s^2]
m	Mass	[kg]
p	Angular velocity in e_1' direction	[rad/s]
p^b	Position of the helicopter's center of mass expressed in $\mathcal{B}$	[m]
p^n	Position of the helicopter's center of mass expressed in $\mathcal{N}$	[m]
q	Angular velocity in e_2' direction	[rad/s]
Q_M	Main rotor countertorque	[N m]
Q_T	Tail rotor countertorque	[N m]
R	Orientation of $\mathcal{B}$ with respect to $\mathcal{N}$	
r	Angular velocity in e_3' direction	[rad/s]
T_M	Main rotor thrust	[N]
T_T	Tail rotor thrust	[N]
u	Velocity in e_1' direction	[m/s]
v	Velocity in e_2' direction	[m/s]
v^b	Velocity of the helicopter's center of mass expressed in $\mathcal{B}$	[m/s]
v^n	Velocity of the helicopter's center of mass expressed in $\mathcal{B}$	[m/s]
v_i	Induced velocity	[m/s]
w	Velocity in e_3' direction	[m/s]

Chapter 1

Introduction

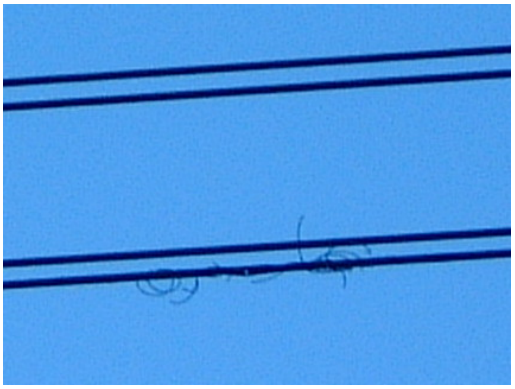
A helicopter is modelled as a nonlinear, underactuated mechanical system with four independent inputs and six degrees of freedom. Therefore, it poses a challenging control problem which is of wide interest to researchers due to the helicopter's flight capabilities. In this thesis, the term helicopter refers to a small-scale version of the aircraft [1]. Small-scale helicopters are commercially available for hobbyist R/C pilots to fly recreationally. These helicopters are characterized by a large thrust-to-weight ratio which allows them to perform flight manoeuvres not possible for full-scale helicopters [2]. When flown autonomously, it is common to use the term Unmanned Aerial Vehicle (UAV), or more recently, Unmanned Aircraft System (UAS) [3]. In order to use a commercial R/C helicopter as a research apparatus, it must be outfitted with navigation sensors and an autopilot computer. Our experimental platform is the Applied Nonlinear Controls Lab (ANCL) Helicopter which is shown in Figure 1.1. This platform has been used to conduct navigation research [4, 5] as well as the control validation described in this thesis.

Helicopters are among a class of aircraft which possess flight capabilities such as vertical takeoff and landing, and hover. Other such aircraft include multirotors and ducted fans. See [6] for a recent survey. These flight capabilities make helicopters ideal for surveillance or inspection tasks. For instance, high voltage transmission line networks require regular inspection. This inspection is currently conducted using manned aircraft, however, it is well-suited for a helicopter UAV. Transmission towers are often located in remote areas where the terrain is rugged, as shown in Figure 1.2a. Since a helicopter does not require a runway, it can be deployed remotely regardless of terrain. Additionally, as shown in Figure 1.2b defects can be subtle. Thus, the ability to hover near a defect to allow close-up images to be taken from multiple angles is beneficial.

Many control methods have been applied to helicopters in the past. Designs based on linear approximations can be found in [2, 3, 7–10]. Our primary focus is nonlinear control, and some examples of existing work are given in the follow-

Figure 1.1: ANCL Helicopter in flight at Bergen R/C Helicopters in Cassopolis, MI, October 12, 2012.

(a) Transmission tower located in a remote region of British Columbia, Canada.



(b) Example of a conductor defect on a high voltage transmission line.

Figure 1.2: Example pictures taken during inspection of the transmission line network in British Columbia, Canada. Images courtesy of BC Hydro, Vancouver, Canada.

ing. However, rather than attempt an exhaustive list we will focus on work which has been particularly influential in this thesis. Input-output feedback linearization is performed in [11]. The complete dynamics are shown to have critically stable tracking dynamics. Therefore an approximate model is used to define the control. A backstepping control is designed in [12, 13] using a model similar to [11]. As compared to the linearization approach, the backstepping design avoids parameterizing the rotation matrix. Another example of a backstepping approach where singularities arising from a parameterization of the orientation are avoided is [14]. However, the orientation of a rigid body cannot be globally stabilized by a continuous control [15]. The backstepping control design in [9] uses a high order model which includes actuator dynamics as well as exact thrust modelling. However, due to the complexity of the input expressions the control requires a numerical solution to be applied. Experimental results are provided for this control. In [16] a nonlinear control is designed which addresses the problem of input saturation, and experimental results are given. Input saturation is also addressed in [17] where the design uses columns of the rotation matrix to avoid a parameterization of the orientation. Other nonlinear approaches to helicopter control include [18–20].

In general, a mathematical model of the helicopter can be very complicated [21]. Therefore, it is necessary to determine which aspects of the model will be considered for the purpose of control design. This choice cannot be made absolutely. Instead, simplifications should be made in the context of the control objective. For example, many high-order effects exist such as flapping dynamics [1] and gyroscopic moments [22] may become relevant during aggressive 3D manoeuvres. However, practical trajectories used for surveillance, inspection or waypoint navigation can be flown accurately without compensating for such effects. Alternatively, helicopters are often operated in variable and uncertain environments. In particular, the effect of wind gusts can be considered as in [23, 24] and more generally in the context of parameter and other uncertainty [25]. Although feedback such as PID will generally react to wind with a small tracking error, this type of work can further improve the response to these disturbances especially during non steady-state conditions. For the purpose of the work herein, we will choose a model which allows for the analysis of the principal manner in which the helicopter is actuated yet retains tractability for control design.

The designs proposed in this thesis make use of the hierarchical nature of helicopter dynamics. In this respect the helicopter dynamics are similar to multirotors where the horizontal motion is forced by reorienting the aircraft to direct the thrust. Since the orientation acts as an input to the control it is natural to treat the translational and rotational dynamics as two subsystems in cascade. In fact, this approach has become standard [3, 7, 16, 26, 27]. There is also practical benefit to dividing the

control design into subsystems. The implementation and testing can be performed independently. For example, we have used a PID inner-loop which is tuned and known to work to test a nonlinear outer-loop control. Furthermore, the rotational state measurements can be performed independently of position measurements. This independence allows us to handle a loss of GPS by regulating the trim attitude.

1.1 Thesis Overview

Although helicopter control research is an applied field, relatively few experimental results exist in the literature. One possible explanation for the lack of experimental work is the challenging nature of platform development and maintenance. Most outdoor testbeds use GPS and inertial sensors for state measurement, an on-board computer for an autopilot, and a modem for communication with a ground station. Early accounts of experimental helicopter platforms are [2, 7]. More recent descriptions include [10, 28]. In general platform descriptions focus on the hardware. Indeed, hardware selection is an important aspect of platform development and detailed knowledge of working hardware configurations is valuable [29]. In Chapter 2 we describe the ANCL Helicopter. Over the lifetime of the project we have made several changes to the hardware components and configuration. These changes often occur as the result of a crash or other problem, and we therefore describe the reasoning in order to provide more insight into the platform than is commonly available in the literature. Software implementation also poses a significant implementation challenge. Therefore, we provide a detailed description of our software design, focusing on key aspects such as modularity and error-handling. The source for our autopilot software is available for download [30]. This contribution to the helicopter research community is expected to provide future benefits from interaction with other groups.

In this thesis we focus on model-based control. A control design based on a model allows for a rigorous statement of performance. Developing an accurate model for a helicopter is the subject of many references. An encyclopedic treatment of full-scale helicopter modelling is [21]. Small-scale helicopter modelling can be found in e.g., [1, 3, 5, 9, 28]. Due to complexity, general aerodynamic models are intractable for control design using exact methods. A linear approximation can be used to simplify these general expressions. However, this approach is likely to result in a control which compensates insignificant effects. For example, the main rotor disk can be reoriented with respect to the airframe in order to cause its thrust to act as a lever-arm about the center of mass. The dynamics which govern this reorientation are fast and unlikely to be important in the main rotor thrust model. On an R/C helicopter the physical inputs are servo motors. Servos also possess dynamics which

are included in e.g., [9]. However, their dynamics are unlikely to have a significant effect for applying a control law. Due to the complexity of modelling the actuation, it has become common practice to design controls using the main and tail rotor thrusts, and the main rotor flapping angles as inputs [13]. These inputs simplify control design by separating the structural nonlinearities which are fundamental to the helicopter dynamics from the complex aerodynamic modelling of the applied forces and torques. In fact, when actuator dynamics are neglected the inputs appear in the same expression as the nonlinearieties arising from the aerodynamics. Thus, if the input expressions are invertible the dynamics can easily be accounted for exactly. The approach followed for the presentation of the helicopter model in Chapter 3 is to separate the dynamics and the input modelling. In particular, Section 3.1 provides an overview of the helicopter dynamics with the inputs used for control design. Rather than focusing on absolute accuracy, in Section 3.2 we present a simplified mapping between the inputs used for control and the physical inputs which is experimentally justified. The goal of this simplified model is to improve on e.g., [16] where the physical inputs are related by gains. The control design in Chapter 4 follows this same philosophy. When the inputs are the thrusts and flapping angles, the linear approximation about hover becomes double integrators for the translational and rotational dynamics. Therefore, PID control is justified by the model and not applied heuristically as suggested in some references (see for instance [31] among many others). Furthermore, we use Lyapunov analysis to show the nonlinear terms which arise in the translational dynamics when the states are considered in the body-fixed frame are passive and therefore do not affect stability. Results are given for simulations and experimental testing. The flight tests include hover as well as time varying trajectory tracking.

The relative orientation of the main rotor disk with respect to the airframe is used to force the rotational dynamics. This orientation is parameterized by the flapping angles. Additionally, the tail rotor is used to control the heading. Since all three of these inputs are linear forces they also influence the translational dynamics. This coupling between the translational and rotational dynamics is often referred to as the Small Body Forces (SBF) [12]. In the literature the SBF are neglected for control design. See for instance [11, 13]. However, an analysis of their effect on the closed-loop is performed in [11]. A more detailed robustness analysis showing ultimate boundedness is performed in [12, 14]. In Chapter 5 we derive a control which compensates the contribution of the tail rotor to the SBF. Additionally, we show how including the SBF due to the flapping angles results in unstable dynamics for the reference roll and pitch. While these dynamics preclude direct compensation of the effect of the flapping angles a robustness analysis is conducted which shows uniform ultimate boundedness of the closed-loop tracking error in their presence.

As mentioned above, helicopters are an ideal platform for inspection tasks. However, depending on sensor type it may be necessary to limit translational velocity in order to allow adequate sampling coverage. Furthermore, a rotational control which satisfies bounds on roll and pitch would be useful in a situation where a camera is fixed to the aircraft or mounted with limited pan and tilt. The envisioned mission is to track waypoints over a target area (e.g., along a transmission line). For this type of flight, translational velocity bounds could be roughly satisfied by using a reference trajectory which is compatible with these limits. However, based on experience gained performing flight tests for the controls described above we have found that it is desirable to implement waypoint navigation using setpoint regulation. In particular, setpoint regulation is simpler for implementation, and simpler designs are preferable in practice. With this motivation, we propose a design in Chapter 6 which is based on recent work [32, 33] and satisfies translational velocity constraints as well as bounds on roll and pitch travel.

1.2 Contributions

The contributions of this thesis may be summarized as follows:

- Platform development and software framework [34, 35]. A complete experimental platform requires careful design. Documentation of hardware configurations including the relevant design decisions is provided. In addition to hardware details, we have released the source for our autopilot software, and provide a detailed description of its design which focuses on modularity and error-handling.

- Experimentally justified model simplification [36, 37]. General aerodynamic models are intractable for control design using exact methods. We propose simple expressions to relate the physical inputs to the inputs commonly used in the control literature.

- Model-based PID control [38, 39]. This simple linear control design is derived based on the model using an approximation as well as Lyapunov analysis to neglect some nonlinear terms. Experimental results are provided for hover, line, and circle trajectories.

- Nonlinear control which compensates the contribution of the tail rotor to the SBF [40, 41]. The inputs to the rotational dynamics also force the translational dynamics. In the literature this effect is ignored for control design. We derive a control which compensates for the effect due to the tail rotor and provide a robustness analysis which addresses the remaining terms.

- Nonlinear control design which satisfies state constraints. Specifically, the satisfaction of translational velocity and orientation bounds is addressed. The motivation for this design is based on experience gained from field tests and it is expected to be a valuable control technique to use in practice.

Chapter 2

Experimental Platform

The ANCL Helicopter experimental platform consists of the airframe and electronic hardware described in Section 2.1, and the autopilot software described in Section 2.2. The purpose of this platform is to facilitate applied helicopter control research. Therefore, component selection and design decisions are based on reconfigurability as well as reliability. The hardware configuration of this platform has undergone an evolution. As problems are discovered and addressed, components are replaced, relocated, or modified. A preliminary configuration of the hardware was presented in [4]. Updates made in order to obtain initial control results along with a detailed description of the software were described in [35]. Most recently, hardware modifications were described in [39]. In this chapter we will present the current hardware configuration, however, we will describe the more significant differences from these references.

Software design and implementation also requires an extensive effort. We present our autopilot software framework which is designed to be modular with an emphasis on error-handling. As mentioned above hardware changes are necessary as the platform evolves. By designing the autopilot using a modular architecture, new code can easily be inserted to accommodate new hardware. Error-handling is also an important aspect of an autopilot. Examples of errors include unreliable state estimates and controller malfunctions such as division by zero. If state measurements are used which are unreliable, the helicopter is likely to crash. In fact, during the early stages of controller testing our helicopter crashed when the controller used an attitude measurement which had an error of over 60°. Therefore, even for a research platform it is important to test for error conditions and take appropriate action. For example, if the translational state estimate is known to be invalid the control could be switched to attitude regulation only. Alternatively, if the entire state estimate is invalid the control could be deactivated and the servos could be commanded with neutral positions [10]. We have made our autopilot software available publicly [30] in order to allow other groups to benefit from the implementation work we have

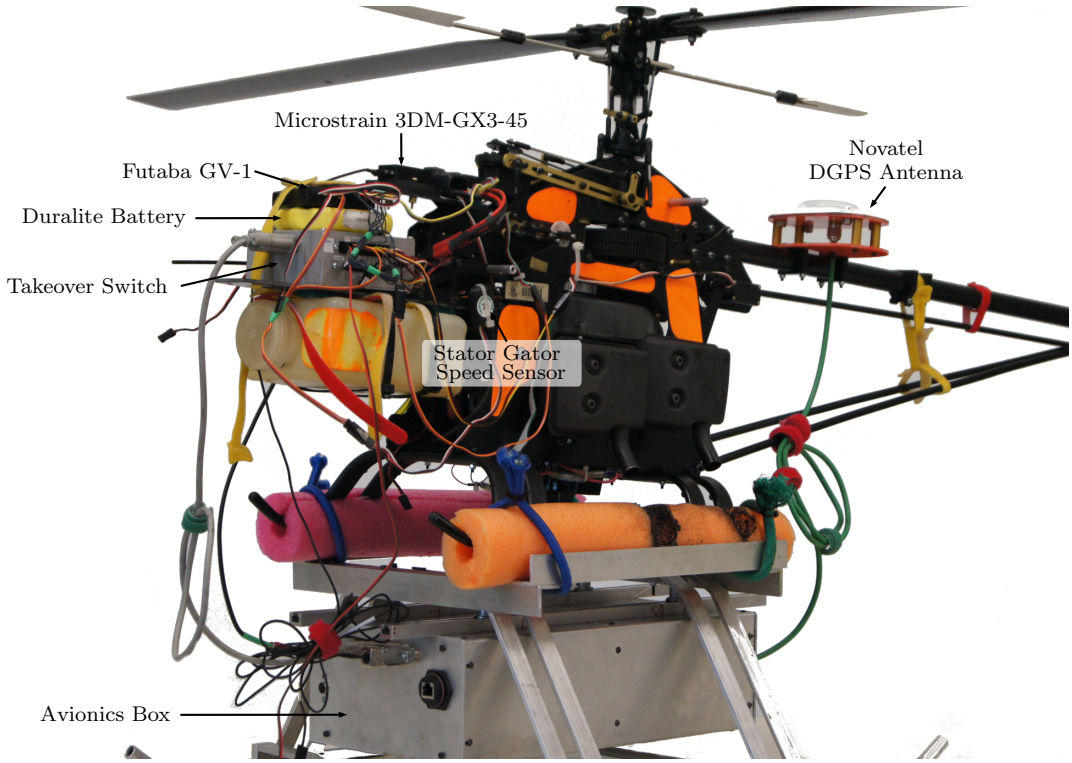


Figure 2.1: Platform diagram showing the mounting location of the system components. The cables are loose in order to avoid occluding the connections. When the helicopter is prepared for flight these cables are secured using velcro straps.

done. Additionally, we envision future benefit from communication with these other groups.

2.1 Airframe and Electronic Hardware

The ANCL Helicopter includes the Bergen R/C Industrial Twin helicopter, the avionics box, navigation sensors, the Takeover Switch (TS), and a ground station laptop. A diagram of the complete platform is shown in Figure 2.1.

The Industrial Twin is powered by a two cylinder 52 cc gasoline engine with a flight endurance of 30-45 minutes depending on payload, it has a 1.8 m main rotor diameter, and is capable of lifting a 10 kg payload. It has five control inputs which are actuated by five servos: main rotor lateral and longitudinal cyclic pitch, main and tail rotor collective pitch, and engine throttle. We remark that these five servos correspond to actuation in only four independent degrees of freedom since the collective pitch and the throttle together influence the thrust. When the helicopter is in its stock configuration the tail rotor collective pitch is controlled by a Futaba GY-401 gyro which stabilizes the yaw angle by tracking a pilot reference.

In order to circumvent the need to identify the gyro, it can be bypassed allowing direct computer control of the tail collective. For manual control a Futaba T9CHP radio is used with a Futaba R149DP receiver.

The avionics include an embedded computer, a GPS-aided inertial navigation system, a DGPS receiver, and a radio modem. The embedded computer is an Ampro ReadyBoard 800 single board computer with a Pentium M 1.4 GHz CPU, 1 GB RAM, and an 8 GB CF card for storage. The Microstrain 3DM-GX3-45 (referred to as the GX3) provides a Kalman filter for state estimation which is referred to as the navigation filter and is a GPS-aided INS. The GX3 additionally provides an Attitude and Heading Reference System (AHRS). The navigation filter provides estimates of position, velocity, attitude and gyro bias; whereas the AHRS provides attitude estimates only. The rotational state estimates are updated at 100 Hz and the translational states are updated at 20 Hz. Since the GPS receiver which is shipped with GX3 does not provide sufficient accuracy, a Novatel OEM4 FlexPak DGPS receiver (referred to as the OEM4) with 2 cm CEP position accuracy is used as an external input to the GX3. These position measurements are sent to the GX3 at 4 Hz. Although the GX3 is logically included as a part of the avionics, it is no longer mounted inside of the avionics box. As shown in Figure 2.1 the GX3 is mounted directly to the airframe. The reason for this change is to rigidly attach the inertial sensors to the airframe. Indeed, the avionics box is now coupled to the airframe using the elastic straps visible in Figure 2.1. This loose coupling prevents destructive resonance which results from the extended skids. Moving the GX3 out of the avionics box was done as a result of analyzing the experimental results presented in this thesis. Finally, the radio modem is a 2.4 GHz Microhard VIP2400 modem (referred to as the VIP) which provides RS-232 communication for the OEM4, and an Ethernet link for ground station communication. The ground station laptop is a Panasonic Toughbook CF-53 running the Linux operating system. The above avionics hardware is mounted in a custom made avionics box shown in Figure 2.2. Comparing the configurations in [35] and [39], the antenna for the OEM4 has been moved forward on the tail boom from its original location attached to the vertical stabilizer fin. This change was made after vibrations from the tail rotor caused the aluminum mounting bracket to break during flight.

The TS, which is shown in Figure 2.3, allows the pilot to select the source of the servo commands as either those from the radio control receiver (Pilot Manual Mode), or the ones generated by the embedded computer (Computer Control Mode). The TS is based on the Microbotics SSA20024 Servo Switch controller which sends commands to the servos and reads the pilot commands at 50 Hz. The TS also includes a custom designed PCB which provides connections to the receiver and the helicopter servos. Analog low pass filters have been integrated into the PCB in order

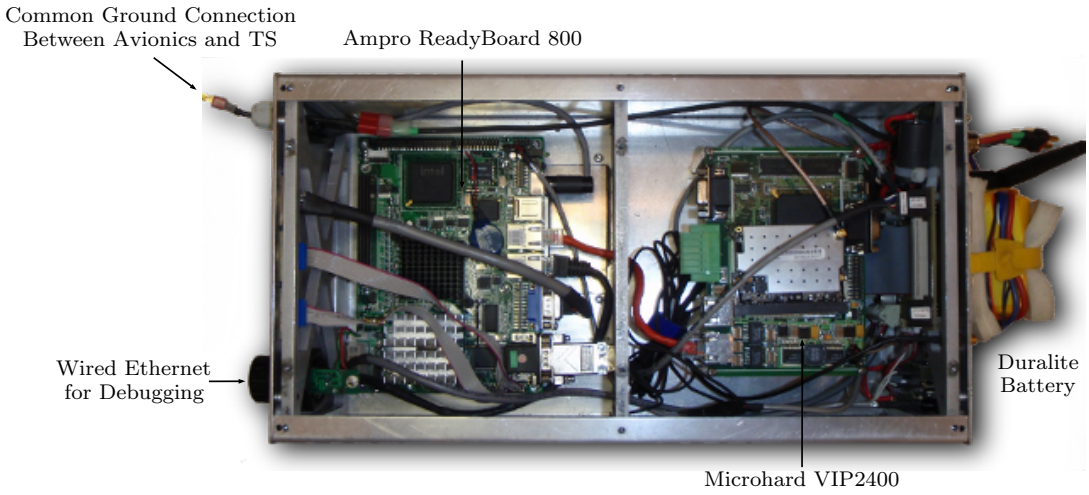


Figure 2.2: ANCL Helicopter avionics. Visible components include the Ampro Readyboard on the left and the Microhard VIP Modem on the right.

to eliminate the noise induced by the engine in the input and output channels of the TS. Additionally, a heavy-gauge ground cable was added to create a common ground bus between the TS and the avionics box. Without this ground connection, the servos were observed to move erratically. This connection is indicated in Figures 2.2 and 2.3. As shown in Figure 2.1 a Stator Gator speed sensor has been integrated into the system to measure the engine speed. This speed measurement is used by control algorithms to compute the main and tail rotor speeds which are related to the engine speed by a gear ratio. The Stator Gator connects to the TS using one of the servo connectors visible at the top of Figure 2.3. This connection is configured in the SSA20024 as an input.

2.2 Autopilot Software

As stated above, the purpose of the experimental platform and in particular the autopilot software is to facilitate control research for a helicopter UAV. Since this research requires frequent changes to the control algorithm as new designs are developed, a software framework is needed which allows controller implementation to be performed in an efficient manner. One possible approach would be to use Mathworks Real-Time Workshop (RTW) which generates C-code from Simulink diagrams and provides a familiar environment for developing control algorithms (see e.g., [42, 43]). In fact, a prior version of our platform used a RTW package [4], however we found it to be insufficiently reliable, and cumbersome for hardware integration and upgrades. Furthermore, Simulink diagrams are not well-suited for collaboration (either by internal team members or the open-source community). Therefore, in order to achieve

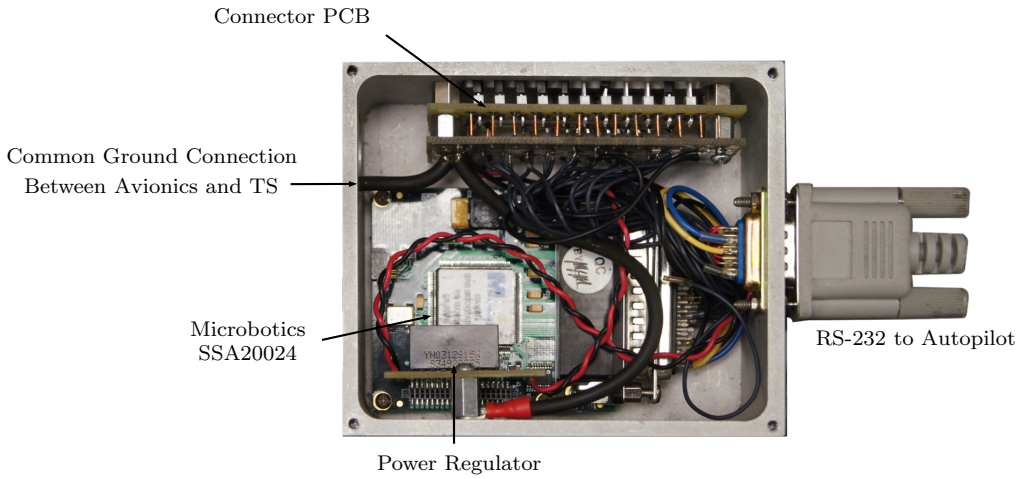


Figure 2.3: Takeover Switch (TS) packaged with power regulator and custom-built PCB for servo connectors.

maximal flexibility as well as reliability we have chosen to develop our own software framework.

2.2.1 Operating System and Development Environment

The main consideration for the choice of operating system is real-time performance. While the only way to guarantee real-time performance in software may be to program a microcontroller in assembler, this approach is very labour intensive and does not scale easily to large programs. Therefore, we selected a so-called real-time operating system which provides the convenience of task scheduling and a general layer of abstraction from the hardware, but also provides features which aid the developer in meeting timing deadlines not commonly available in desktop operating systems. In particular, we chose QNX 6.5.0 [44] since it is well documented, mature and already proven in the UAV field (see for instance [45]).

Another key benefit of using an operating system is access to a modern compiler suite. Since code footprint is not an issue for our hardware configuration, and the requisite compiler was provided with QNX we chose to write the autopilot in C++. Indeed, this language provides convenient high level features which promote code reuse and modularity but can also be used for low-level programming using C syntax and libraries. In addition, there are many mature libraries for performing common yet sophisticated tasks (e.g., thread handling and linear algebra). In our implementation we make extensive use of the Boost C++ Libraries, many of which are slated to be included in upcoming C++ standards [46].

In addition to language and library dependency considerations, a development environment was devised to facilitate team collaboration. This environment includes a revisioned source repository; online HTML documentation available on the local

network; and the QNX Momentics IDE which can be used to develop, compile, debug, and execute code. Indeed, an autopilot is not a trivial piece of software to design and develop, and substantial long-term benefit is envisioned by simplifying the task of contributing to the source. For example, a single centralized code repository, including revision history, allows current and future contributors to review what changes are being made to which source files and by whom. The Git source control management software [47] provides a solution for local development, and is also used to update the publicly available source. Documentation of the autopilot source is written using Doxygen markup [48]. This functionality ensures online documentation is always current and allows the documentation to be viewed without requiring a copy of the source.

2.2.2 Architecture

A governing principle of the software design is that it should be modular. This approach helps facilitate portability and simplifies major code changes (e.g., control algorithm changes or rewriting a hardware interface when a component of the experimental platform is changed). From a high level, the software employs standard object-oriented programming techniques insofar as the code is logically divided into classes. Although there are at present many classes implemented, there are only a few main ones: a main class for high level program management and control computation timing, a class to provide a public interface for the control implementation, and a class for each piece of external hardware which contains the code used to communicate with the device.

One method employed to provide modularity in the system is use of the Boost Signals2 library. This library defines a signal type which can be dynamically connected to one or more functions, and these functions are executed whenever the signal is emitted. An example of how signals are used to improve modularity, is to report status changes received from the GX3. Among other things, the GX3 reports several error conditions such as large covariances for its estimated quantities. Currently, these error conditions emit signals from the class handling communication with the unit. Since it was not possible to know which functions should be called in future controllers to handle estimate errors, the only alternative to using signals which does not require future modification of the GX3 class, would be to constantly poll the GX3 class. Polling for information which is unlikely to change is inefficient (estimate errors are expected to be infrequent), however their inclusion in the public interface means that these error conditions can be handled without requiring modification of the GX3 class.

Data logging is another important aspect of the autopilot. The logging is im-

plemented in a way which is flexible with respect to data type.[1] In addition, the logs are flushed to disk regularly to prevent memory from being entirely consumed, and to also prevent a software crash from causing a complete loss of data. The timing of these disk writes is chosen to minimize the amount of data in memory which would be lost in the event of a system failure, but also to ensure they are not interfering with overall system performance. Message logging is also available with varying levels of severity to provide insight into the sequence of events occurring within the autopilot. This logging facility has proven essential not only for identifying sources of failure, but also as a record of the experimental procedure during successful flights.

2.2.3 Controller Implementation

Since the autopilot is being developed to research advanced control designs, it must accommodate several different algorithms, possibly generating output from multiple controllers simultaneously for comparison by the operator. In order to standardize the implementation of these algorithms, all of the controller classes are derived from an abstract class named `ControllerInterface` which declares the main functions related to control computation. For instance, the control presented in Section 4.6, uses an inner-outer hierarchical approach. Therefore, two controllers have been implemented: an inner loop attitude controller and an outer loop position controller. Figure 2.4 shows the relationship between the control classes and the `ControllerInterface` abstract class.[2] The `reset` and `runnable` functions are used to reinitialize any controller states, and test whether a controller is capable of running (e.g., whether sufficient state measurements are available), respectively. The `operator()` function performs the control computation and is distinct from the `get_control_effort` function since it may contain an integration step (as is the case for PID control). Therefore, `operator()` should only be called once per iteration of the autopilot main loop whereas `get_control_effort` can be called many times (e.g., for logging and transmitting to the ground station in addition to updating the servo commands). This approach was chosen to reflect the design philosophy of implementing each function to perform a single, clearly defined task. The result is modular code which is easier to use and maintain.

The `Control` class is used to provide the rest of the autopilot with access to the controllers. It manages the various controllers by keeping track of a mode

[1] It is implemented using a template function which only assumes a container type with support for a Forward Iterator, and that the data in the container can be converted to `std::string` using `boost::lexical_cast`.

[2] The vector type used by `ControllerInterface` is in fact `boost::numeric::ublas::vector<double>` but is abbreviated to `vector` for presentational clarity.

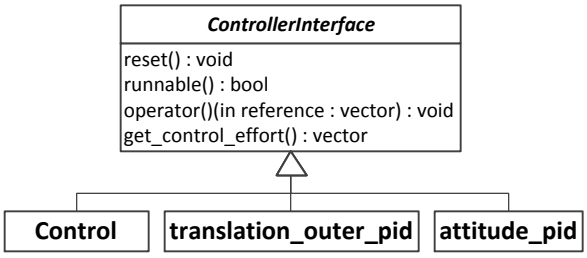


Figure 2.4: UML Inheritance Diagram showing the relationship between the conrtoller classes and the abstract ControllerInterface class.

that determines which controllers to call. In addition, this class performs ancillary functions such as maintaining an XML file to store configuration parameters and generating a list of parameters which can be exchanged with the ground station. Indeed, all configuration data in the autopilot is stored using the XML format. This format is used because it provides a flexible syntax for manual editing. In addition, many libraries for interfacing with XML files exist. One example is RapidXml [49] which is currently used in the autopilot.

2.2.4 Overview of Program Flow

An overview of the main program loop is shown in Figure 2.5. This loop is executed at 100 Hz and is contained in a class named `MainApp` which is spawned as a thread upon program initiation. The decisions in this loop are strictly high level in nature, and in particular relate to selecting the source of the servo commands and waiting for threads to exit cleanly upon program termination. This clean thread termination is performed by allowing threads to register themselves in a list. When the autopilot receives a kill command (from the operator) the `MainApp` class emits a terminate signal, the list of threads is then iterated and each element of the list is joined by the `MainApp` thread. This joining allows the registered threads to perform any final operations before ending. In practice, examples of thread cleanup include flushing remaining data logs and closing the log files, or sending on-board hardware commands to stop transmitting data and releasing serial ports.

The control computation is performed by calling the `Control` class. The decisions made by the `Control` class are shown in Figure 2.6. This operation can throw a `bad_control` exception which is handled by giving control of the servos directly to the pilot. Although this method of handling an error may not be successful in averting a disaster (e.g., the pilot may not be aware he has taken control), it should only be encountered upon a complete failure of all possible control modes. Indeed, it is preferable to use the pilot commands even without his knowledge rather than allowing an unhandled exception to cause the process to be terminated by the op-

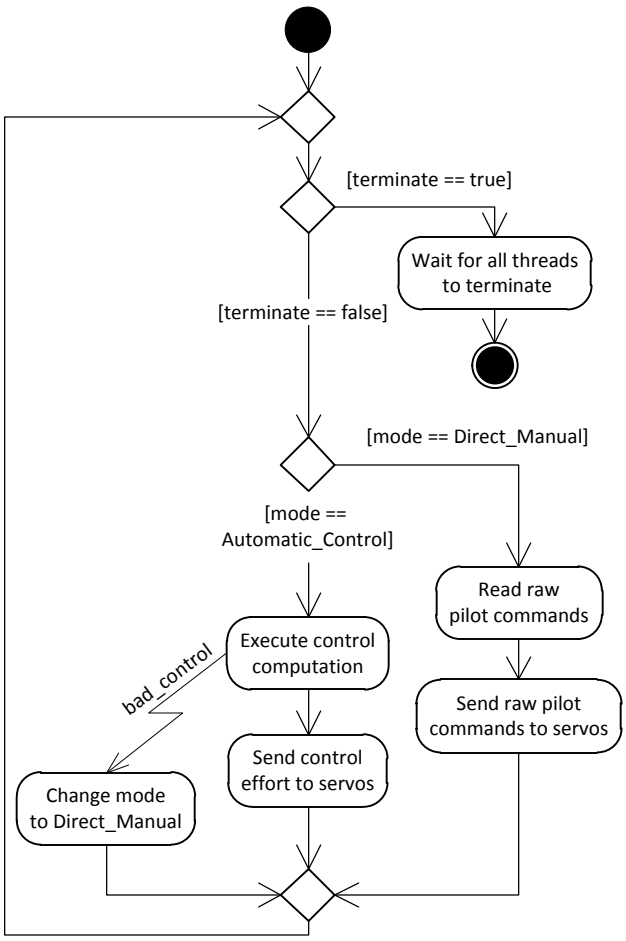


Figure 2.5: UML Activity Diagram showing how the servo command source is chosen.

erating system or continuing to attempt control computation when it is no longer possible (e.g., when state estimates are no longer reliable). This approach to error-handling is similar to [10, Sec. 4.2.4] where the proposed method is to command all servos to their trim value.

The details of how the control effort is computed based on the control mode are shown in Figure 2.6. Of particular interest is the built in error handling. An exception thrown by the position controller is handled by changing the mode to attitude stabilization only. The attitude control mode attempts to regulate the orientation which is tuned by the ground station operator, and should be set as closely as possible to the trim orientation. Thus, upon failure of the position controller the helicopter will attempt to stabilize to a hover. In practice the intention is that the pilot would observe an apparent failure in position control and regain manual control using the TS. However, this error handling allows the autopilot to give the pilot a chance to react by sending stabilizing commands to the servos.

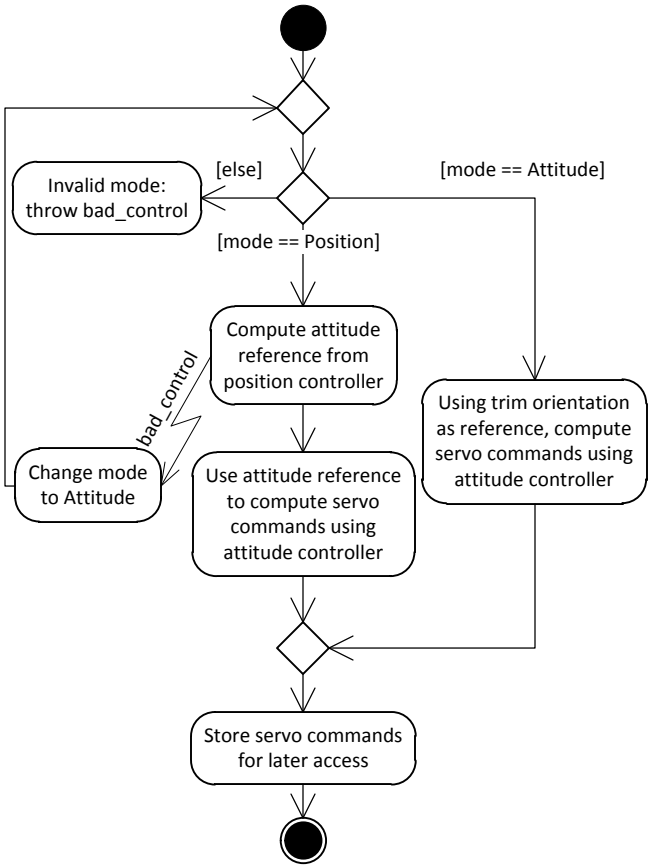


Figure 2.6: UML Activity Diagram showing an overview of the control computation.

2.2.5 Communication with On-board Hardware

The autopilot communicates with the GX3, OEM4, and TS (see Section 2.1) using RS-232. The majority of the communication with the on-board hardware involves the reception of data. Therefore in all cases a thread is spawned which reads data from the respective serial port. For the case of the OEM4, a command is only sent once when the program starts, to request that data transmission begins, and once when the program finishes to request that data transmission ceases. These commands are therefore simply implemented as part of the receiving thread. For the case of the TS, data is continually being sent as well as received. Therefore, a second thread used for sending data is executed in parallel with the receive thread. Sending commands to the GX3 is implemented slightly differently because we found that it was convenient to allow the ground station operator to have some direct control of the navigation filter. In particular, we implemented the ability for the operator to reset the navigation filter and initialize the attitude estimate based on the current AHRS measurement computed by the GX3. Since these user commands are asynchronous to the execution of the autopilot and could happen at any time (e.g., if the operator observes divergence of the state estimate he could request the pilot briefly land the helicopter while the filter is reinitialized), we designed the autopilot to spawn a thread for each command sent to the GX3. These threads do not contain a loop, instead they terminate after an acknowledgement is received from the GX3 indicating successful transmission of the command. Sequential writing to the serial port is ensured using a mutex.

Although the hardware communication essentially consists of reading and interpreting incoming serial data, along with the ground station communication it accounts for roughly 80% of the source code. This is the main reason that we chose C++ over a Simulink/RTW implementation. As is evidenced by the distribution between the code related to the control and the code related to hardware communication, in terms of implementation effort it is the software framework and not the control which requires the focus. Therefore, it is most efficient to use a development environment that allows direct access to the operating system.

2.2.6 Ground Station

Rather than write our own ground control software we chose to modify, and contribute to, the existing open source project QGroundControl (QGC) [50]. QGC natively supports an application layer message protocol called MAVLink [51], an implementation of which is distributed alongside QGC. It is outside the scope of this document to detail QGC, however we will describe the three main modifications we made which have particular relevance to our work. The operator screen used during flight tests is shown in Figure 2.8.

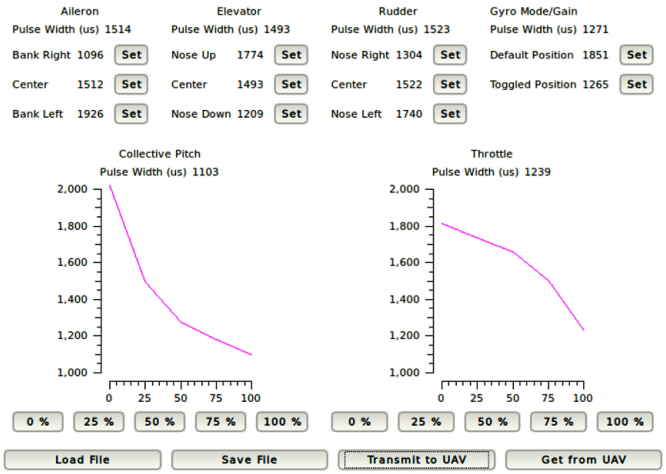


Figure 2.7: Window for reading pilot radio calibration parameters which are used to map normalized cyclic inputs to servo pulse commands.

The first addition we made to QGC is a window for reading the radio calibration parameters and is shown in Figure 2.7. These parameters define the transformation between the pilot stick position and the raw pulse width delivered to the servos. Due to environmental factors such as changes in temperature or wind conditions the pilot will often make adjustments to the radio calibration in the field. By using the radio calibration window, these changes can be read and updated in the autopilot without requiring recompilation of the source code, or manual configuration file editing. In practice we have found this feature invaluable during field experiments. As described in Chapter 3, correct measurements of servo travel and centers are critical to map normalized cyclic inputs to servo pulse widths. This window was integrated with the official QGC source.

The second addition we made to QGC was to add a dockable window which provides an overview of the state of the autopilot. This custom widget, shown on the right of Figure 2.8, is used to monitor and control aspects of our autopilot which are unique to our implementation. For example, this window allows the operator to view and change the servo command source, control mode, and attitude estimate source. The attitude estimate source refers to a selection between the AHRS and the navigation filter, both of which are provided by the GX3. This widget also allows the operator to view and change the location of the origin used to transform position measurements into the local navigation frame. Another function provided by this widget is to provide the interface mentioned in Section 2.2.5 which allows the operator to reset the navigation filter and initialize its attitude estimate from the AHRS. Additionally, the reference trajectory used for the control can be selected from a drop-down box. Finally, the operator can send a kill command to

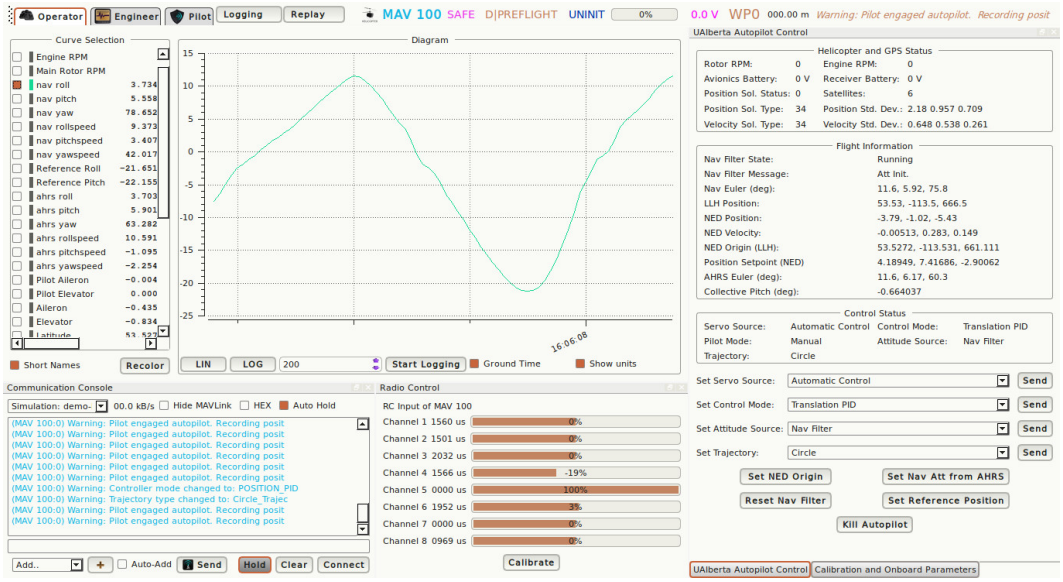


Figure 2.8: Screenshot of QGC. The UAlberta Autopilot Control widget on the right shows an overview of the autopilot's status. The Radio Control widget shows the output of the pilot's radio. The Curve Selection widget allows the operator to select which quantities are plotted in the Diagram widget. The current selection is the roll angle estimate provided by the navigation filter.

the autopilot, the result of which is described in Section 2.2.4.

The exchange of MAVLink messages between the autopilot and QGC is performed using UDP socket communication. This socket is accessed by a send and a receive thread which are implemented using the Boost Asio library. This link is fault tolerant in the sense that whenever possible, error conditions are detected and data retransmission is attempted. As much as is possible we make use of the existing MAVLink messages, however in order to fully support our extensions to QGC it was necessary to create custom messages. In QGC, the common message set is handled by the `UAS` class. The final modification we made to QGC was to specialize the `UAS` class in order to handle our custom messages.

2.3 Conclusions

The ANCL Helicopter experimental platform has been developed to facilitate applied control research. The hardware platform consists of a Bergen R/C Industrial Twin helicopter, an avionics box, navigation sensors and the Takeover Switch. Hardware problems have been solved by replacing and relocating components. Examples include the antenna for the OEM4 which was moved and the TS which replaced the previous Takeover Board [4]. The autopilot software was developed modularly to allow for changes to hardware components and control algorithms. In addition,

error-handling was built into the software in order to react to problems such as an unreliable state estimate. The QGC ground station software was extended to incorporate unique aspects of our autopilot and to record the calibration stored in the pilot's radio. This ground station development has proven invaluable in the field during flight testing.

Chapter 3

Helicopter UAV Modelling for Control Design

Our motivating interest in helicopter UAV research is to investigate nonlinear control methods. The first necessary step in this pursuit is to specify a model. Typically model specification is approached with the goal of maximizing accuracy [3, 7, 9]. However, the difficulty in establishing a model for the helicopter is to find expressions for the forces and torques. A helicopter is actuated by two rotors, referred to as the main and tail rotors. The aerodynamic modelling used to derive the form of this actuation typically results in complex expressions [1, 3, 5, 28] which are intractable for use in a control law. For example, the main rotor thrust typically depends on the flapping dynamics which describe the motion of the main rotor disk with respect to the airframe. However, the thrust is unlikely to be influenced in a significant way by the flapping angles or their derivatives and can reasonably be ignored [31]. Treating the actuator dynamics provides insight into the helicopter model, and improves the accuracy of simulators. However, in order to perform a control design a tractable model is required. Other dynamics also exist in the actuators due to effects such as motion of the servos and the flybar. In particular, the flybar dynamics have been a main focus of small helicopter models since they are unique to this variation of the aircraft. However, the action of the flybar is stabilizing which means its dynamics do not require any control action. Additionally, the flybar does not prevent aggressive manoeuvring as evidenced by acrobatic flight capabilities [2]. Furthermore, most new models of R/C helicopters do not have flybars. Given these considerations, our approach is to maintain a minimal model order and to focus on model aspects which a significant effect.

We begin our exposition of the model in Section 3.1 where we establish reference frames and notation, provide expressions for the orientation, and present the model used for control design. Section 3.2 presents the mappings used to relate the physical

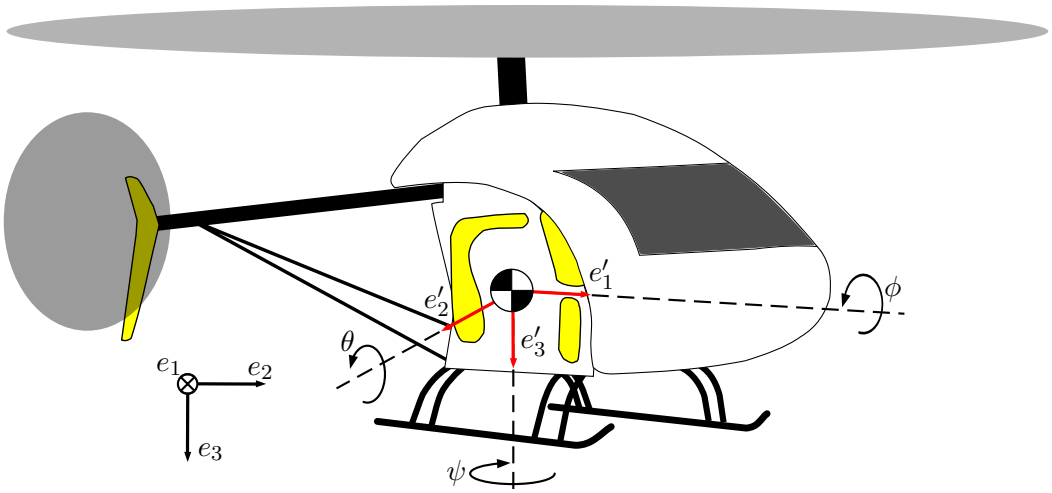


Figure 3.1: Diagram of the helicopter showing the coordinate systems used for navigation, the Euler angles, and the approximate location of the center of mass without a payload.

inputs to the inputs used for control design. This section includes original results in which experimental data is used to justify a simplification of aerodynamic modelling found in the literature. This simplified modelling provides mappings which are appropriate for implementation since they can be described analytically and have an explicit inverse.

3.1 Helicopter Dynamics

To describe the dynamics of the helicopter, we use a navigation coordinate frame $\mathcal{N}$ which is assumed inertial, has its origin fixed to the surface of the earth, and whose basis is the orthonormal set of vectors $\{e_1, e_2, e_3\}$ which are oriented north, east, and down respectively. Remark, that $\mathcal{N}$ is *assumed* inertial since this frame is fixed to the earth and therefore it is actually rotating slowly. The body-fixed frame $\mathcal{B}$ has its origin located at the helicopter's center of mass, and its basis $\{e_1', e_2', e_3'\}$ oriented forward, right and down respectively. These frames are shown in Figure 3.1.

The navigation and body-fixed frames are related by a rigid body transformation which includes a translation and a rotation. The rotation is of particular interest since it is used to transform the coordinate expressions of state vectors between reference frames, and it evolves on a non-Euclidean manifold which is discussed in the following section.

3.1.1 Representation of the Orientation

In order to present vectors precisely in the context of rotating reference frames, we must make explicit the distinction between what the vector represents physically,

and its coordinate representation with respect to a reference frame. Among many references on mechanics, see for example [52] where this concept is described rigorously. For example, the velocity of the helicopter, which for the purpose of this discussion we denote by $\vec{v}$, is an absolute physical quantity which exists independent of the notion of reference frames. Indeed, in the most general sense of a vector space a scalar product need not exist, and therefore a coordinate expression cannot be defined. Since this level of abstraction is not useful for our purpose we will use the usual definition of the scalar product in a Euclidean space to establish a coordinate representation of $\vec{v}$. Thus we have

$$\vec{v} = v_1 e_1 + v_2 e_2 + v_3 e_3,$$

where

$$v_i = \vec{v} \cdot e_i,\ i \in \{1, 2, 3\}.$$

Since in the above expression we have used the basis of $\mathcal{N}$ which is inertial, its basis vectors have constant directions. A similar expression using $\mathcal{B}$ is also possible, however the directions of the basis vectors depend on time. The position of the helicopter, which we define as the position of the origin of $\mathcal{B}$ with respect to the origin of $\mathcal{N}$, can be thought of similarly. This definition is independent of its coordinate expression in either frame, and it is an absolute physical quantity. Furthermore, its representation in the two frames is related by the rotation matrix $R : \mathcal{B} \to \mathcal{N}$ given by

$$p^n = R p^b \tag{3.1}$$

where p^n is the position vector expressed in $\mathcal{N}$, and p^b is its expression in $\mathcal{B}$. Thus, the orientation of the helicopter is described by the relative rotation from $\mathcal{B}$ to $\mathcal{N}$. This rotation belongs to the special orthogonal group $SO(3)$ whose elements are the real-valued 3×3 invertible matrices characterized by orthogonal rows and having a determinant of $+1$. Although $SO(3)$ is a three dimensional manifold, no three parameter coordinate chart of $SO(3)$ is globally defined [53]. The unit quaternions provide a nonsingular parameterization of $SO(3)$, however they form a double cover which leads to unwinding [54]. Indeed, many attitude representations exist in the literature [55]. Here, in addition to using R directly we will use the so-called ZYX Euler angle representation

$$R = R_z R_y R_x = \begin{pmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{pmatrix} \tag{3.2}$$

$$R = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\psi s\theta c\phi + s\psi s\phi \\ c\theta s\psi & s\psi s\theta s\phi + c\psi c\phi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{pmatrix} \tag{3.3}$$

where ϕ is roll, θ is pitch, ψ is yaw which we will consider to be synonymous with the heading, $s\phi = \sin(\phi)$, and $c\phi = \cos(\phi)$. The Euler angles are shown in Figure 3.1. For a detailed discussion of the Euler angles see e.g., [56]. We now consider the coordinate representation for the derivative of a vector in $\mathcal{B}$. Continuing from the example of the position vector defined above, we differentiate (3.1) to obtain

$$\dot{p}^n = \dot{R}p^b + R\dot{p}^b. \tag{3.4}$$

Thus, we need an expression for the derivative of R. Since this expression is standard in the literature (see e.g., [53] among many others) we will not derive it here. Indeed, the rotational kinematic equation is

$$\dot{R} = R\hat{\omega}^b$$

where ω^b is the (absolute) angular velocity of $\mathcal{B}$ represented in $\mathcal{B}$, and $\hat{\cdot}\colon \mathbb{R}^3 \to \mathfrak{so}(3)$ or $(\cdot)^\wedge\colon \mathbb{R}^3 \to \mathfrak{so}(3)$ is the isomorphism between vector spaces $\mathbb{R}^3$ and the 3×3 skew-symmetric matrices. We will denote its inverse by $(\cdot)^\vee\colon \mathfrak{so}(3) \to \mathbb{R}^3$. Thus, we expand (3.4) as

$$\dot{p}^n = R\hat{\omega}^b p^b + R\dot{p}^b$$

which when combined with the identities $RR^T = I$ and $\hat{\omega}^b p^b = \omega^b \times p^b$, as well as the fact that when represented in an inertial frame $\dot{p}^n = v^n$, we obtain

$$\dot{p}^b = -\omega^b \times p^b + v^b$$

where $v^b = R^T v^n$ is the coordinate representation of the velocity in $\mathcal{B}$. Although this kinematic expression was derived specifically for the position, the derivative of any vector in $\mathcal{B}$ is expressed similarly.

In the sequel, no distinction will be made between a physical vector and its coordinate representation, and the $\vec{\cdot}$ notation will not be used. A superscript ^{n} will indicate a vector expressed in $\mathcal{N}$, and a superscript ^{b} denotes a vector expressed in $\mathcal{B}$.

3.1.2 Model for Control Design

As discussed in the previous section, state vectors can be represented in either $\mathcal{N}$ or $\mathcal{B}$. However, a particular choice of coordinate frame is usually more convenient for a given application. The helicopter is modelled as a rigid body using either the

Euler-Lagrange equations of motion or the Newton-Euler formulation [13]. Since the former requires a parameterization of the configuration manifold, we choose the Newton-Euler expression of the dynamics [53]. When the states are expressed in $\mathcal{N}$ the dynamics are

$$\dot{p}^n = v^n \tag{3.5a}$$

$$m\dot{v}^n = mge_3 + Rf \tag{3.5b}$$

$$\dot{R} = \hat{\omega}^n R \tag{3.5c}$$

$$J^n\dot{\omega}^n = -\omega^n \times J^n\omega^n + R\tau \tag{3.5d}$$

where p^n is the position, v^n is the velocity, m is mass, g is acceleration due to gravity, f is applied force, ω^n is the angular velocity, and τ is the applied torque. The inertia matrix J^n is expressed relative to $\mathcal{N}$ and therefore it is time-varying. Indeed, it is related to the inertia matrix expressed in the body frame by the similarity transform $J^n = RJ^bR^T$. Since the actuators have fixed positions in $\mathcal{B}$, it is convenient to always express the applied forces and torques in the body-fixed frame. Thus, we must rotate these terms into the navigation frame in (3.5). Alternatively, when the states are expressed in the body-fixed frame the dynamics become

$$\dot{p}^b = -\omega^b \times p^b + v^b \tag{3.6a}$$

$$m\dot{v}^b = -\omega^b \times mv^b + f + mgR^Te_3 \tag{3.6b}$$

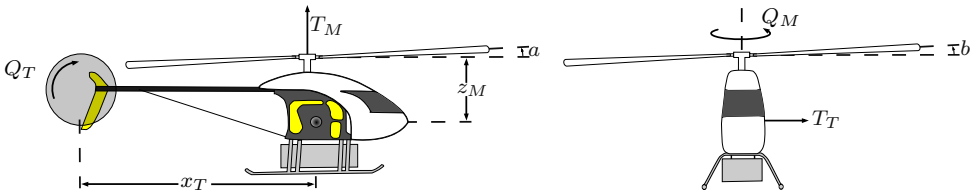
$$\dot{R} = R\hat{\omega}^b \tag{3.6c}$$

$$J^b\dot{\omega}^b = -\omega^b \times J^b\omega^b + \tau \tag{3.6d}$$

where p^b is the position, v^b is the velocity, ω^b is the body-fixed angular velocity, and J^b is the inertia matrix. In practice, a combination of the above is usually employed. Of note is the 'cross' term appearing in the rotational dynamics when the inertia is computed in $\mathcal{N}$ (3.5d). This term creates a nonlinearity in state similar to when the inertia and angular velocity are expressed in $\mathcal{B}$ (3.6d). Therefore, since both expressions for the rotational dynamics are nonlinear and J^n is time-varying, we will always consider the inertia expressed relative to $\mathcal{B}$, and denoted $J = J^b$. In addition, the angular velocity will be expressed in the body frame, and denoted $\omega = \omega^b$. However, we will express the translational states in whichever frame proves convenient. We will also make use of the Euler angle kinematic expression

$$\dot{\eta} = W(\eta)\omega \tag{3.7a}$$

$$W(\eta) = \begin{pmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{pmatrix} \tag{3.7b}$$



(a) Main rotor thrust, tail rotor countertorque, main and tail hub offsets, and longitudinal flapping angle.

(b) Lateral flapping angle, tail rotor thrust, and main rotor countertorque.

Figure 3.2: Front and side views of the helicopter showing the parameters relevant to actuation.

where $\eta = (\phi, \theta, \psi)^T$ and $t\phi = \tan(\phi)$.

A generally accepted model of the applied force and torque has been established in the literature (e.g., [11, 13]) and can be expressed as

$$f = \begin{pmatrix} -T_M a \\ T_M b - T_T \\ -T_M \end{pmatrix} \tag{3.8a}$$

$$\tau = l_T \times \begin{pmatrix} 0 \\ -T_T \\ 0 \end{pmatrix} + l_M \times \begin{pmatrix} -T_M a \\ T_M b \\ -T_M \end{pmatrix} + \begin{pmatrix} 0 \\ -Q_T \\ -Q_M \end{pmatrix} \tag{3.8b}$$

where T_M and T_T are the main and tail rotor thrusts respectively, a and b are the longitudinal and lateral main rotor flapping angles respectively, g is the acceleration due to gravity, l_M and l_T are the positions of the main and tail rotor hubs respectively expressed in $\mathcal{B}$, and Q_M and Q_T are the main and tail countertorques respectively. The countertorque is non-negative and oriented in the appropriate direction for the ANCL Helicopter. These parameters are shown in Figure 3.2 where z_M is the vertical offset of the main rotor hub, and x_T is the tail rotor hub offset in the e_1' direction. Note the appearance of a, b, and T_T in (3.8a). This coupling between the rotational and translational inputs is unique to helicopters among the class of vehicles with similar flight capabilities and is referred to as the Small Body Forces (SBF). From a mechanical perspective, the SBF are the result of actuating the rotational dynamics using linear forces acting on lever-arms. Some authors e.g., [11, 17] include additional terms in (3.8b) which account for restorative moments applied on the body by the rotor. These moments are due to the hingeless main rotor hub and are proportional to the flapping angles. In this work we have neglected these terms for simplicity similar to e.g., [13].

3.2 Physical Input Modelling and Identification

A straightforward control based on the modelling presented in Section 3.1 relies on our ability to use T_M, T_T, a, b as inputs and to know Q_M. In practice, these inputs cannot be actuated directly. Indeed, servo motors are used to control blade pitch and engine throttle which are the physical inputs. These servos are controlled by a PWM-like waveform wherein the orientation of the output shaft is proportional to the positive width of the pulse. However, unlike PWM, the period of the wave does not affect the encoded value. For example, a 1.5 ms pulse commands the same orientation in a 50 Hz or 70 Hz signal. We define δ_M and δ_T to be the pulse widths of the main and tail rotor collective pitch servos respectively. Furthermore, we define δ_r and δ_p to be the normalized cyclic inputs (i.e., swashplate tilt angles) which force the lateral and longitudinal flapping angles respectively. We now proceed by modelling the relationships between those used for control design T_M, T_T, a, b and the physical inputs δ_M, δ_T, δ_r, δ_p.

3.2.1 Mapping from Main Rotor Thrust to Collective Pitch Servo

The main rotor thrust is typically modelled using an expression derived from first principles. Examples of such derivations can be found in [1, 28]. However, without further simplification these models are structurally complex and possess a dependence on several physical quantities which is unlikely to be practically significant. This complexity renders the model intractable for control design. Therefore, we must simplify the model while retaining the principal structure and dependence. Here we will use the general model given in [5, p. 162] as the starting point for this simplification. This model is expressed as

$$T_M = \frac{\rho a_M c_M R_M N_b}{24}\Big\{\left(4(R_M\Omega_M)^2 + 6(u^2+v^2)\right)\Theta_M + 6R_M\Omega_M\left(w - v_i - va + ub\right) \\ -3R_M\left(u(p+\dot{b}) + v(q+\dot{a})\right)\Big\} \tag{3.9}$$

where Θ_M is the main rotor collective pitch; u, v, and w are the velocities in the e_1', e_2', and e_3' directions respectively; p, q, and r are the angular velocities in the e_1', e_2', and e_3' directions respectively; Ω_M is the main rotor speed; ρ is the air density; R_M is the radius of the main rotor disk; N_b is the number of blades; a_M is the main rotor lift curve slope; c_M is the main rotor blade chord length; we have neglected the rotor coning angle; and v_i is the induced velocity

$$v_i = \frac{T_M}{2\rho\pi R_M^2\sqrt{(V\cos\alpha)^2 + (V\sin\alpha + v_i)^2}} \tag{3.10}$$

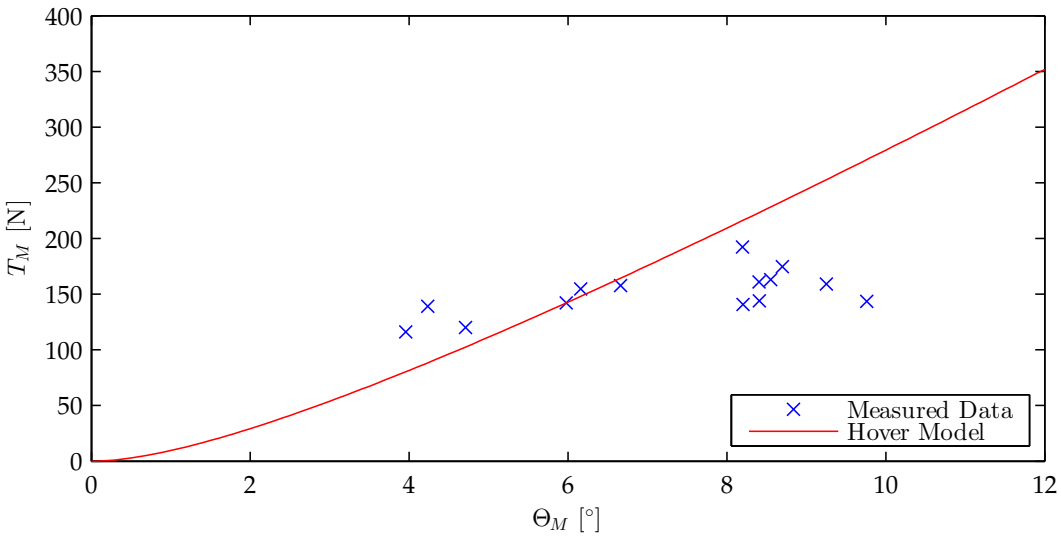


Figure 3.3: Results of the hover model validation experiment. The solid line shows the thrust predicted by (3.11) and the measured values are shown by the x's.

where V is the air flow speed and α is the angle of the airstream with respect to the rotor disk. Based on the mechanical configuration of the helicopter we expect the principal dependence of the main rotor thrust to be on collective pitch and rotor speed. For example, by ignoring all of the terms in (3.9) except the one containing Θ_M and Ω_M we obtain the thrust model used in [16]. The caveat to this approach is it ignores the effect of the induced velocity. A modification to this approach is derived in [5] which includes the effect of the induced velocity, but makes the assumption that the helicopter is in hover (all velocities are zero) to obtain a closed form expression for T_M. Indeed, when the general expression for the induced velocity is used an algebraic solution does not exist. A numerical solution can be used to overcome this obstacle [9]. However, we seek an analytical expression which allows for a rigorous statement of performance. In [35] we proposed a control design using the hover thrust model from [5]

$$T_M = C_M\Omega_M^2\Theta_M + \frac{D_M^2\Omega_M^2}{4\pi\rho R_M^2} - \frac{D_M\Omega_M}{\sqrt{2\pi\rho}R_M}\sqrt{C_M\Omega_M^2\Theta_M + \frac{D_M^2\Omega_M^2}{8\rho\pi R_M^2}} \tag{3.11a}$$

$$C_M = \frac{\rho a_M c_M R_M^3 N_b}{6} \tag{3.11b}$$

$$D_M = \frac{\rho a_M c_M R_M^2 N_b}{4}. \tag{3.11c}$$

In an attempt to validate this model we performed flight tests where the pilot would apply step inputs by hovering then quickly moving the stick up (or down) and holding the collective constant for as long as possible. Our expectation was that we would measure constant accelerations in response to these inputs. Figure 3.3

shows the results where the measured thrust is computed by differentiating the velocity. Except at hover (roughly 6° collective pitch), the model does not match the measured values. Indeed, the measured values contradict the assumption that the only important dependence is on collective pitch. Figure 3.4 shows the response when a step input is applied. We observe that the vertical velocity quickly saturates. Therefore, we propose the inclusion of vertical velocity dependence in our thrust model. To proceed we simplify (3.9) by assuming $u = v = p = q = 0$ and obtain

$$T_M = C_M \Omega_M^2 \Theta_M + D_M \Omega_M (w - v_i). \tag{3.12}$$

Then taking either of the special cases $\alpha = \frac{\pi}{2}$ and $V = -w$ for climb, or $\alpha = -\frac{\pi}{2}$ and $V = w$ for descent we obtain the same simplification of (3.10)

$$v_i^2 - w v_i - \frac{T_M}{2\rho\pi R_M^2} = 0$$

which can be solved to obtain

$$v_i = \frac{w}{2} \pm \sqrt{\frac{w^2}{4} + \frac{T_M}{2\rho\pi R_M^2}}. \tag{3.13}$$

Combining (3.13) with (3.12) we obtain the relationship between collective pitch and main rotor thrust

$$T_M = C_M \Omega_M^2 \Theta_M + \frac{D_M \Omega_M w}{2} \mp \frac{D_M \Omega_M}{\sqrt{2\rho\pi} R_M} \sqrt{T_M + \frac{\rho\pi w^2 R_M^2}{2}}. \tag{3.14}$$

We choose a solution by enforcing $\Theta_M = 0$ when $T_M = 0$ which leads to

$$\Theta_M = \frac{1}{C_M \Omega_M^2} \left(T_M - \frac{D_M \Omega_M w}{2} + \begin{cases} \frac{D_M \Omega_M}{\sqrt{2\rho\pi} R_M} \sqrt{T_M + \frac{\rho\pi w^2 R_M^2}{2}} & w \geq 0 \\ -\frac{D_M \Omega_M}{\sqrt{2\rho\pi} R_M} \sqrt{T_M + \frac{\rho\pi w^2 R_M^2}{2}} & w < 0 \end{cases} \right). \tag{3.15}$$

However, for our current purpose we desire the inverse relationship which gives main rotor thrust in terms of collective, thus we rearrange (3.14) as a quadratic and obtain

$$\begin{aligned} \sqrt{T_M + \frac{w^2 \rho\pi R_M^2}{2}} = & \mp \frac{D_M \Omega_M}{\sqrt{2\rho\pi} R_M} \\ & \pm \sqrt{C_M \Omega_M^2 \Theta_M + \frac{D_M^2 \Omega_M^2}{8\rho\pi R_M^2} + \frac{D_M \Omega_M w}{2} + \frac{\rho\pi w^2 R_M^2}{2}} \end{aligned} \tag{3.16}$$

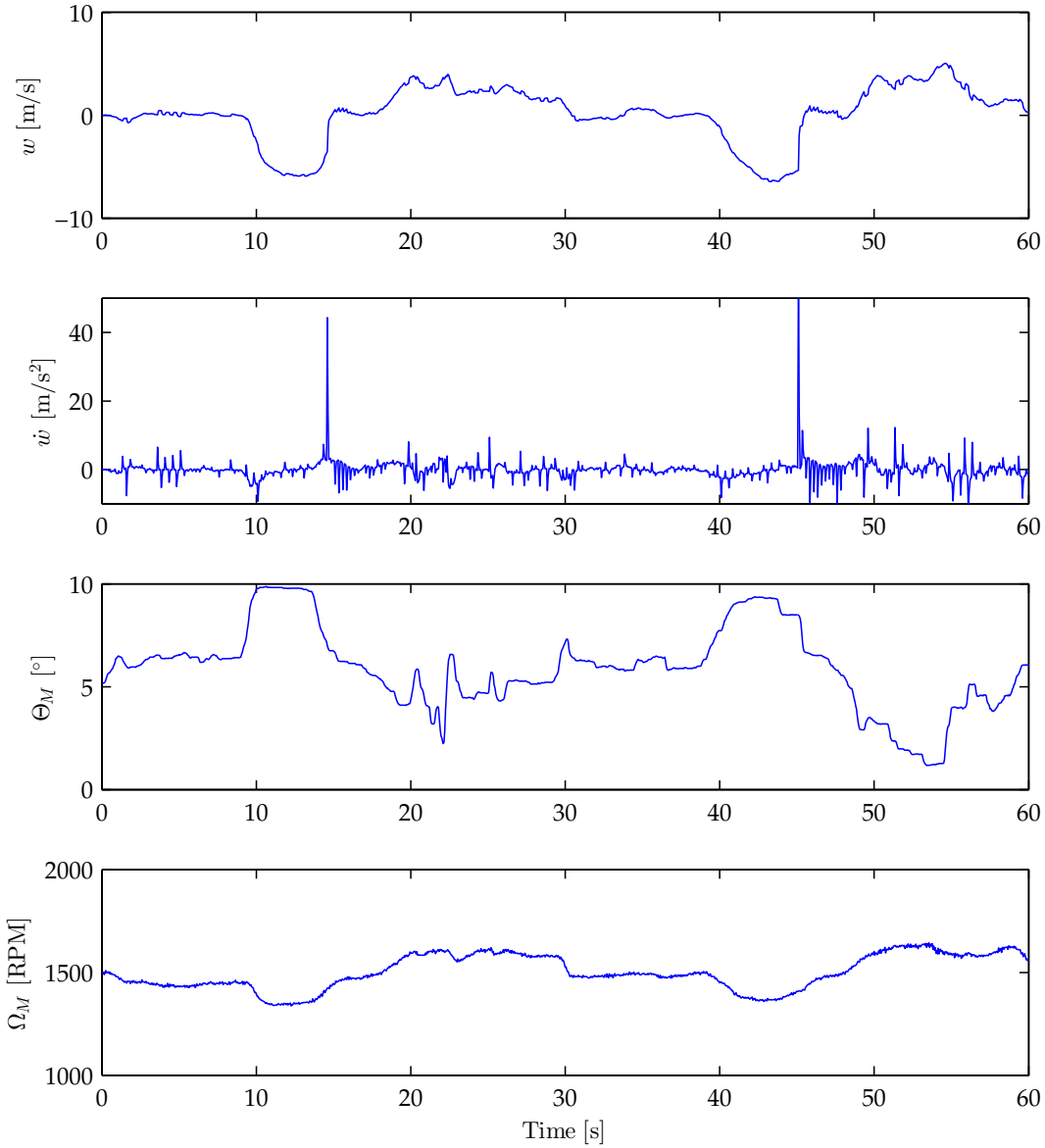


Figure 3.4: Example step response showing velocity saturation.

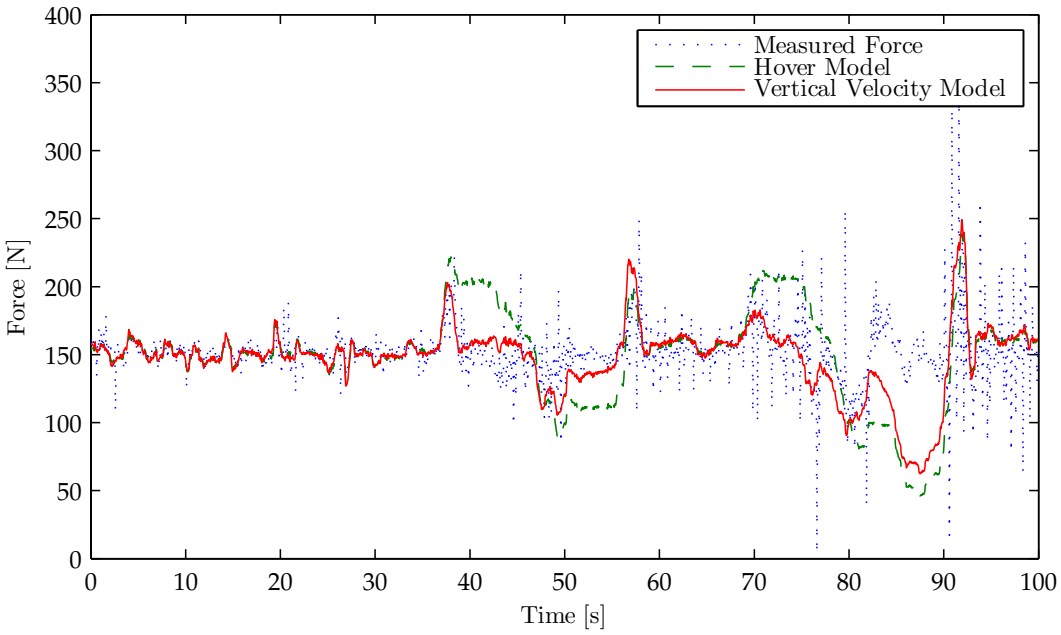
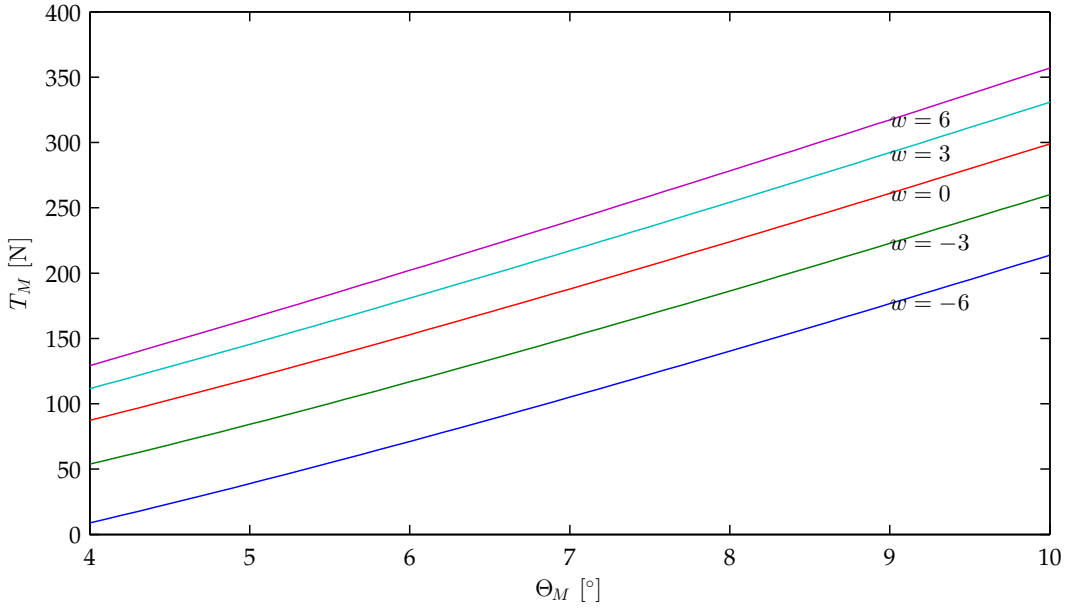


Figure 3.5: Comparison of thrust models (3.14) (solid line) and (3.11a) (dashed line), and the measured force (dotted line).

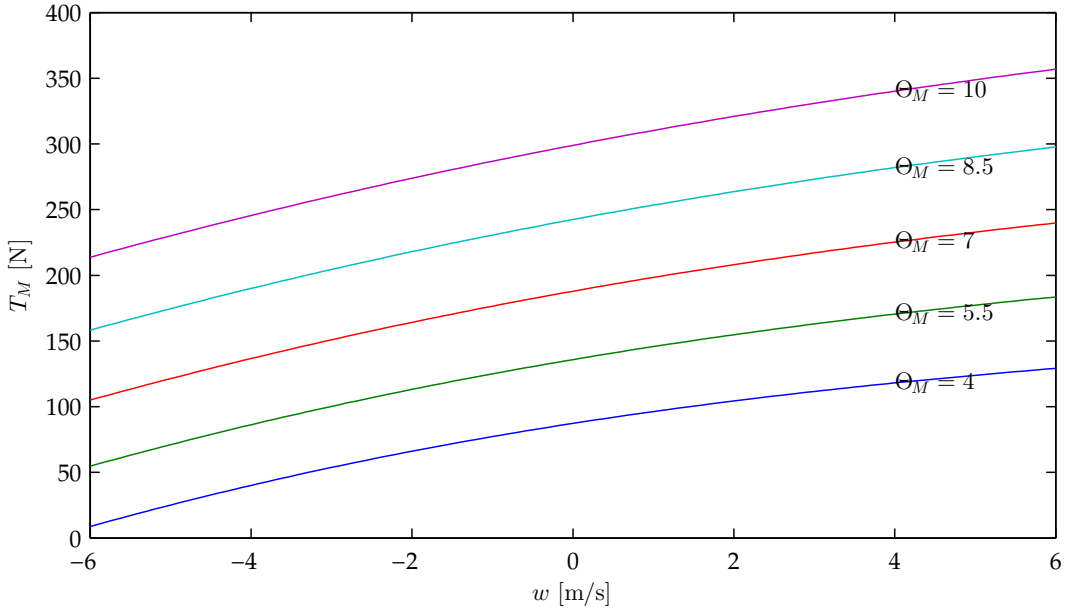
and thus

$$T_M = C_M \Omega_M^2 \Theta_M + \frac{D_M^2 \Omega_M^2}{4\rho\pi R_M^2} + \frac{D_M \Omega_M w}{2} - \frac{D_M \Omega_M}{\sqrt{2\rho\pi} R_M} \sqrt{C_M \Omega_M^2 \Theta_M + \frac{D_M^2 \Omega_M^2}{8\rho\pi R_M^2} + \frac{D_M \Omega_M w}{2} + \frac{\rho\pi w^2 R_M^2}{2}}. \tag{3.17}$$

Remark that when $w = 0$ we obtain (3.11a). In Figure 3.5 we see the difference between the above thrust models. Equation (3.14) captures the saturation behaviour observed at times 38 s and 48 s evidenced by a transient net acceleration, whereas (3.11a) predicts a steady-state net acceleration which is not supported by the results of the experiment. Between 80 s and 90 s there is an error in both models as compared to the measurements. This error is likely the result of a wind gust, which is equivalent to horizontal velocity. If the horizontal velocity causes additional thrust to be generated and the pilot reacts by decreasing collective to maintain altitude, it is expected that the models would predict a decreased thrust since they do not account for this effect. In order to help clarify the relationship of T_M to Θ_M and w, Figure 3.6 shows plots of (3.17) over practically relevant ranges of Θ_M and w. Remark that both families of curves are nearly linear. As mentioned in [7] the vertical velocity in the thrust model provides a stabilizing effect on the vertical velocity dynamics. Indeed, if we assume level, vertical flight we can express



(a)



(b)

Figure 3.6: Illustration of the dependence of main rotor thrust on collective and vertical speed.

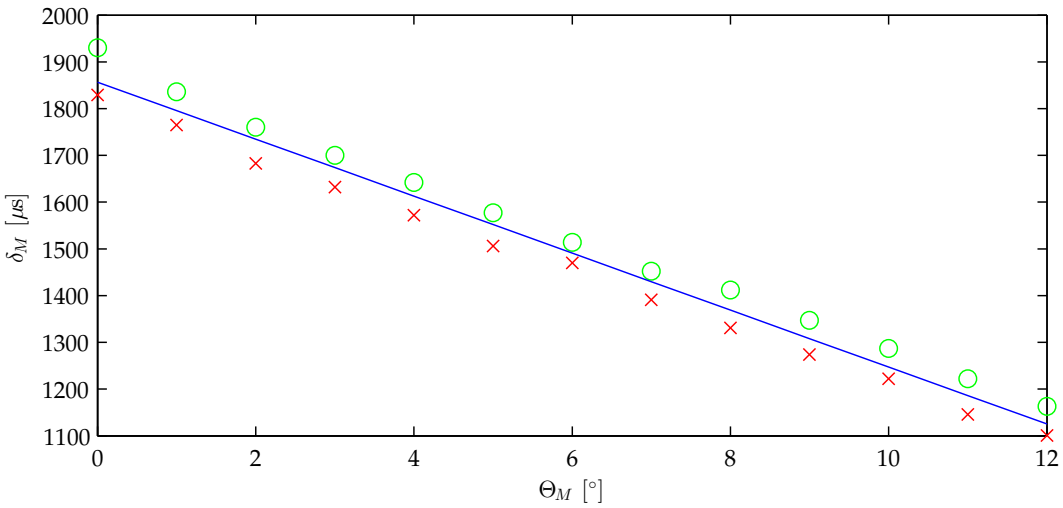


Figure 3.7: Identified mapping from collective pitch to servo pulse width. The circles show measurements made while the pitch was being decreased, while the x's show increasing pitch. The solid line shows the least squares fit.

these dynamics as

$$m\dot{w} = mg - T_M.$$

Since T_M is strictly increasing in w (as shown in Figure 3.6b), the w-dynamics will stay bounded due to this effect.

Finally, we must establish a mapping between the collective pitch and its corresponding servo pulse width since this is the actual physical input. This mapping must be identified for the particular mechanical configuration used, and the results for the ANCL Helicopter are shown in Figure 3.7. The servo was found to exhibit a hysteresis resulting in different measurements if the pitch was sequentially increased as compared to when it was decreased. However, this difference is not expected to be significant in practice and so we fit the line

$$\delta_M = \alpha_M \Theta_M + \beta_M \tag{3.18}$$

to the entire data set. The parameter values are given in Table A.1.

3.2.2 Mapping from Main Rotor Flapping Angles to Normalized Cyclic Inputs

In order to treat the cyclic inputs precisely, we must model the flapping angles as a dynamics system which is actuated cyclic inputs as well as a stabilizing effect from the flybar [1, 5, 28, 57]. However, both the main rotor flapping dynamics and the flybar dynamics are fast and stable. Since our goal is a simplified model which allows us to focus on the fundamental coupling present in the helicopter dynamics,

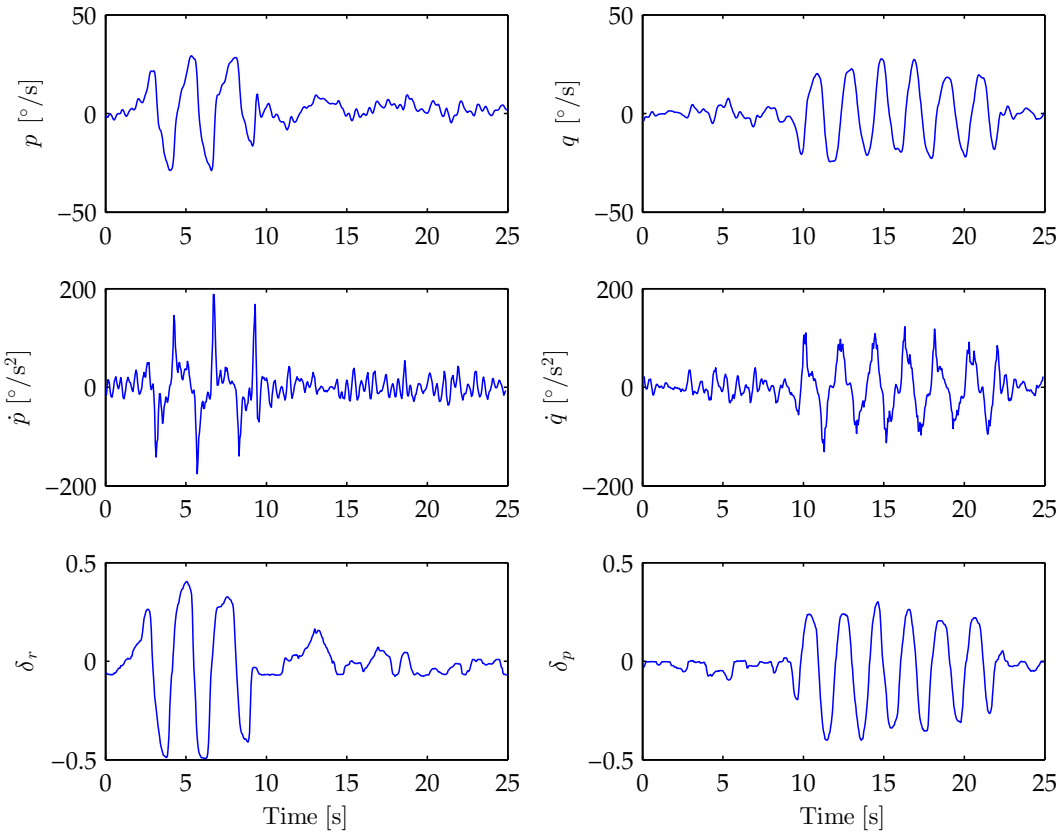


Figure 3.8: Cyclic input identification data. The pilot inputs are normalized and therefore dimensionless.

we choose to ignore these additional dynamics. We therefore take a steady-state approximation of the mapping between the flapping angles and the cyclic inputs by using flight data where the pilot forced the rotational dynamics using a sinusoidal input to the cyclics. The pilot inputs and the system response in the roll and pitch directions for this experiment are shown in Figure 3.8. We expand the rotational dynamics in the roll and pitch directions as

$$\begin{aligned} J_x\dot{p} &= qr(J_y - J_z) + z_M T_M b \\ J_y\dot{q} &= pr(J_z - J_x) + z_M T_M a \end{aligned}$$

and since $r = 0$ during this flight, we are able to isolate the flapping angles as

$$b = \frac{J_x\dot{p}}{z_M T_M} \tag{3.19a}$$

$$a = \frac{J_y\dot{q}}{z_M T_M} \tag{3.19b}$$

which results in the plots shown in Figure 3.9. We used a linear fit to approximate the mapping by the identified functions

$$a = k_p \delta_p \tag{3.20a}$$

$$b = k_r \delta_r \tag{3.20b}$$

where the values of the gains are given in Table A.1. We remark that the gains differ by an order of magnitude. If we did not identify this relationship control would still be possible since these values would be lumped into the controller gains. However, by isolating this effect from the controller gains it allows for a more consistent tuning between channels.

3.2.3 Tail Rotor Thrust and Main Rotor Countertorque

We treat the tail rotor thrust modelling similarly to the main rotor thrust. Once again we begin by using the hover model from [5], however this model neglects negative collective pitch. Whereas for non-inverted flight considering negative main rotor collective pitch is not practically relevant, for the tail rotor it must be included in the modelling. We include the negative collective pitch by defining T_T piecewise as

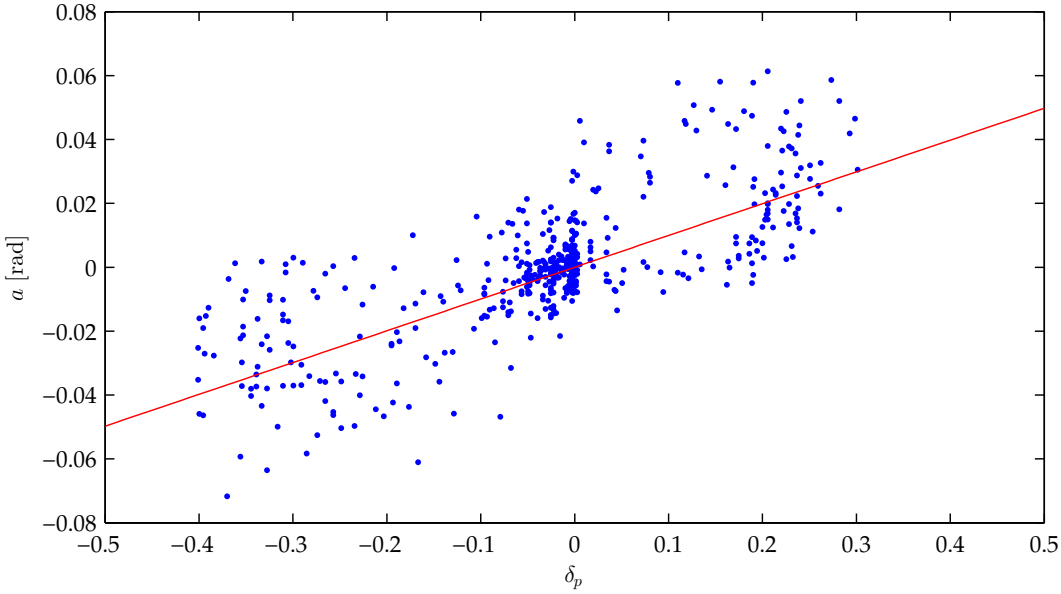
$$T_T = \left(C_T \Omega_T^2 |\Theta_T| + \frac{D_T^2 \Omega_T^2}{4\pi\rho R_T^2} - \frac{D_T \Omega_T}{\sqrt{2\pi\rho} R_T} \sqrt{C_T \Omega_T^2 |\Theta_T| + \frac{D_T^2 \Omega_T^2}{8\rho\pi R_T^2}} \right) \cdot \begin{cases} 1 & \Theta_T \geq 0 \\ -1 & \Theta_T < 0 \end{cases} \tag{3.21a}$$

$$C_T = \frac{\rho a_T c_T R_T^3 N_b}{6} \tag{3.21b}$$

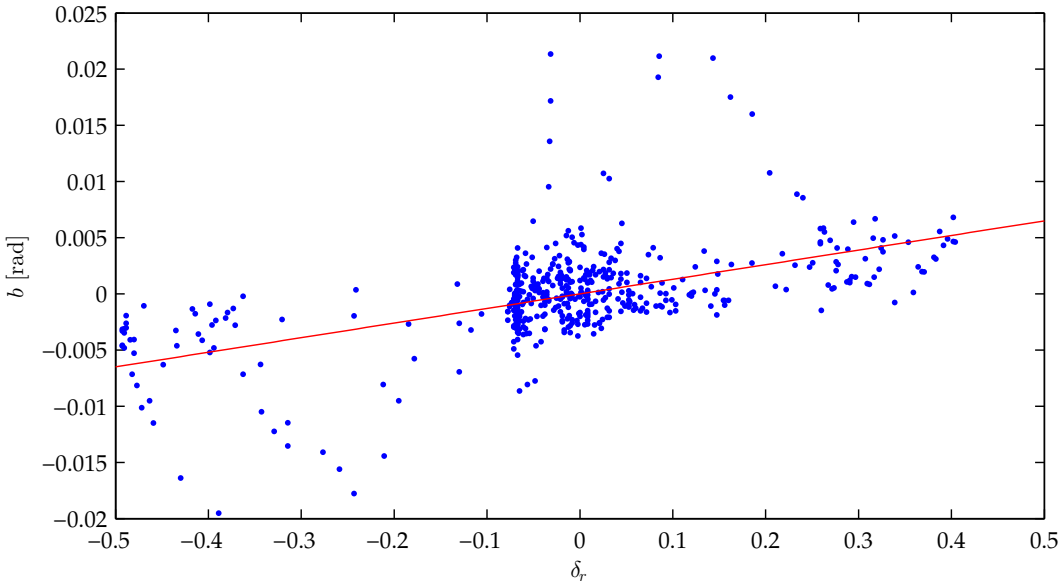
$$D_T = \frac{\rho a_T c_T R_T^2 N_b}{4} \tag{3.21c}$$

where Ω_T is the tail rotor speed, Θ_T is the tail rotor collective pitch, a_T is the tail blade lift curve slope, c_T is the tail blade chord length, and R_T is the radius of the tail rotor disk. This function is shown over a practical range of values in Figure 3.10. Due to the presence of the main rotor countertorque in the yaw dynamics, measuring the angular acceleration about the e_3' axis does not give a direct relationship to the tail rotor thrust. Indeed, in the absence of rolling and pitching motion the dynamics are

$$J_z \dot{r} = x_T T_T - Q_M. \tag{3.22}$$



(a) Identification data for longitudinal flapping angle. The solid line shows $a = k_p \delta_p$.



(b) Identification data for lateral flapping angle. The solid line shows $b = k_r \delta_r$.

Figure 3.9: Identification results for cyclic input to flapping angle mapping.

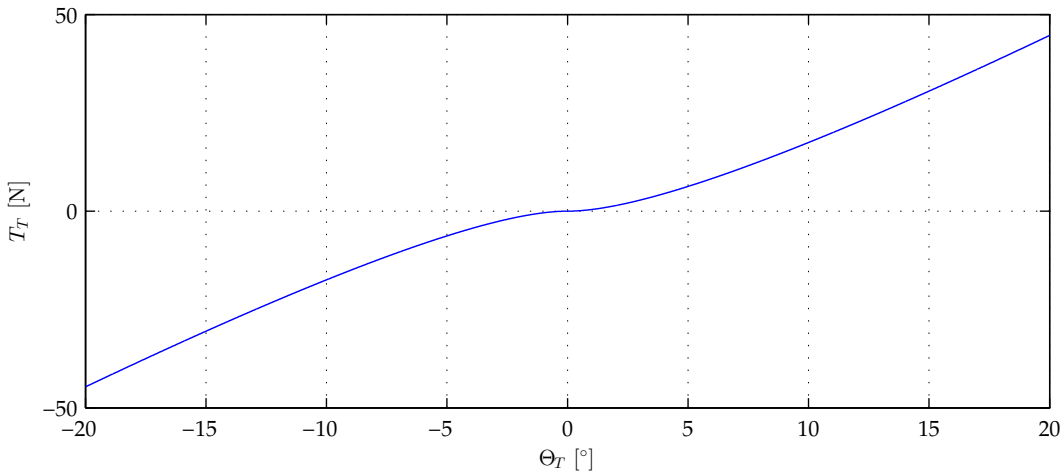


Figure 3.10: Illustration of the dependence of T_T on Θ_T. The curve is very close to linear away from the origin, however as opposed to the main rotor the tail collective is operated at or close to zero under normal flight conditions.

Thus we must consider the tail rotor thrust and the main rotor countertorque models simultaneously. We use the countertorque model given in [5, Eq. 5.40]

$$Q_M = C_M^Q \frac{(T_M)}{\Omega_M}^{(3/2)} + D_M^Q \Omega_M^2 \tag{3.23a}$$

$$C_M^Q = \frac{1}{\sqrt{2\rho\pi} R_M} \tag{3.23b}$$

$$D_M^Q = \frac{\rho c_M R_M^4 C_D N_b}{8} \tag{3.23c}$$

where C_D is the drag coefficient. When Ω_M is assumed constant (3.23) is the same model as the one used in [11].

As is widely discussed in the literature, it is common practice for a pilot to control the heading of a helicopter by providing a velocity reference to a gyro which controls the collective pitch of the tail rotor. Some authors (e.g., [1, 7, 10]) choose to leave the gyro in the loop during automatic control and therefore must identify the dynamics of the gyro and invert them in order to achieve tail rotor collective control. The standard argument for leaving the gyro in the loop is to allow the human pilot to take control in an emergency. Indeed, it is necessary for the gyro to be in the loop for the pilot to safely fly the helicopter. However, we have chosen a configuration such that the output of the gyro passes through the takeover switch (see Chapter 2) allowing our autopilot direct control of the tail collective while retaining the ability for the pilot to regain control. Using this configuration, we only require a mapping from Θ_T to δ_T (which is analogous to the main rotor)

$$\delta_T = \alpha_T \Theta_T + \beta_T \tag{3.24}$$

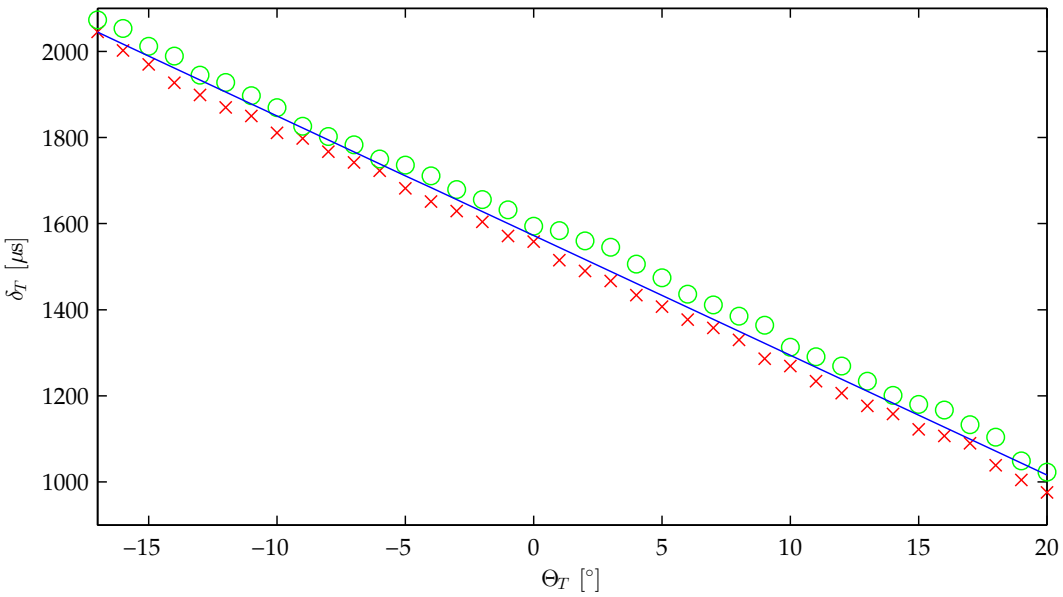


Figure 3.11: Identified mapping between tail rotor collective pitch and servo pulse width. The circles show measurements made while the pitch was being decreased, while the xs show increasing pitch. The solid line shows the least squares fit.

which is found by fitting data measured directly from the ANCL Helicopter and shown in Figure 3.11. The parameter values are given in Table A.1.

The thrust model (3.21) and countertorque model (3.23) are plotted using flight data where the pilot hovered the helicopter at a constant altitude and rapidly changed the heading using a large amplitude motion. The intention of the experiment was to leave Q_M constant while varying T_T. In order to calibrate Q_M for the period of constant heading, C_D was decreased from 0.016 as given in [5] to 0.005. The results are shown in Figure 3.12 where the measured torque is computed by differentiating the heading. Observe the discrepancy between the measured torque and the torque due to T_T at approximately time $t = 55$. This abrupt increase in T_T is due to a corresponding increase in Θ_T. However, we do not measure any resulting motion which indicates this thrust is due to an external disturbance such as a wind gust being rejected by the gyro. To further validate our model we use a vertical climb flight which provides a variation in Q_M (without removing the gyro it is impossible to perform and experiment where T_T remains constant). As shown in Figure 3.13, the results using (3.23) provided a poor prediction of the countertorque during the climb. As we did for T_M we return to the general model of Q_M in [5] and preserve the vertical translational velocity terms while removing the effects of the lateral and angular velocities, as well as the flapping and coning angles. The

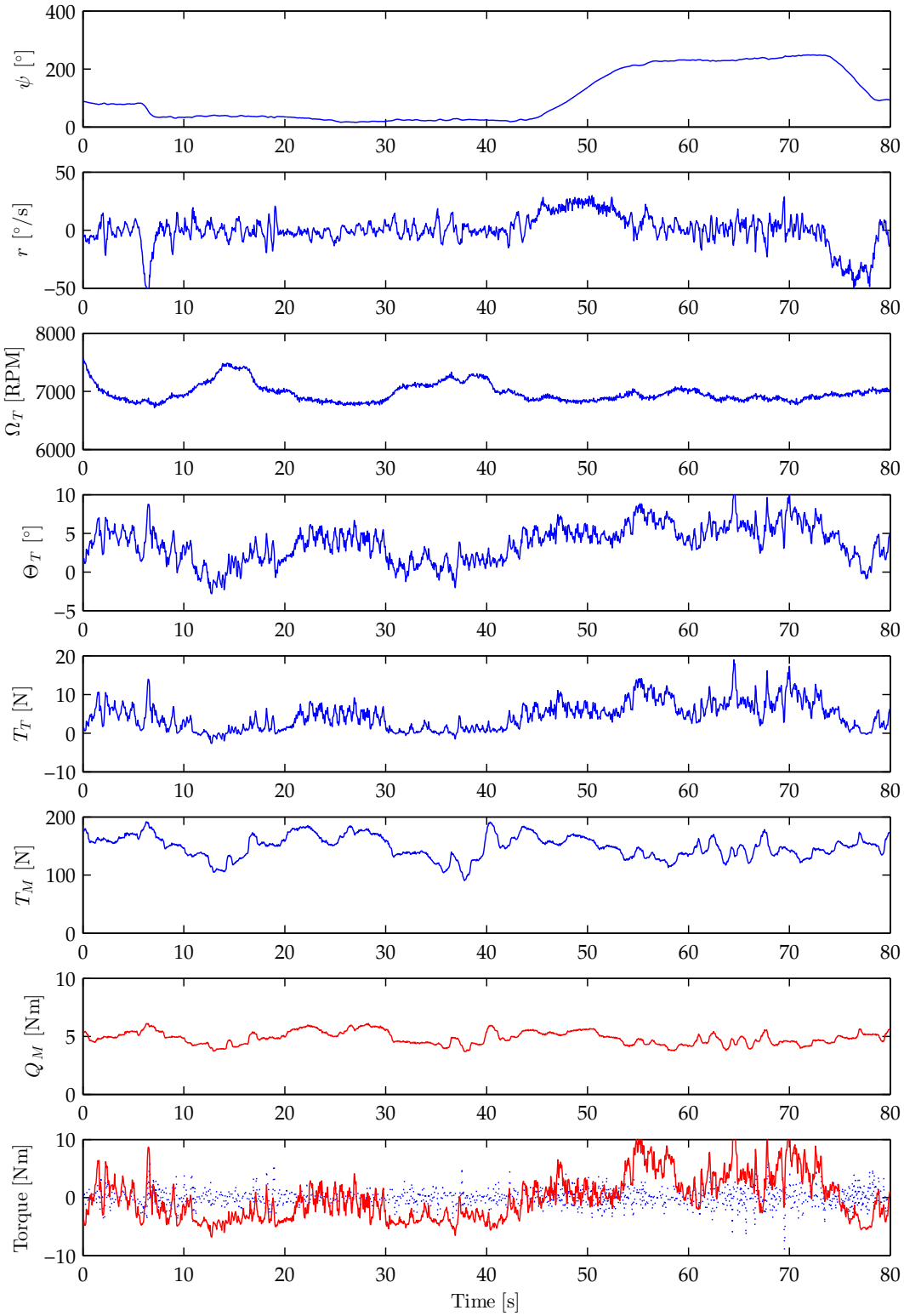


Figure 3.12: Tail thrust model validation data showing a hover flight with fast large amplitude yaw motion. In the bottom plot the dotted line shows the measured net torque while the solid shows the difference between the predicted torque due to the tail rotor and the main rotor countertorque.

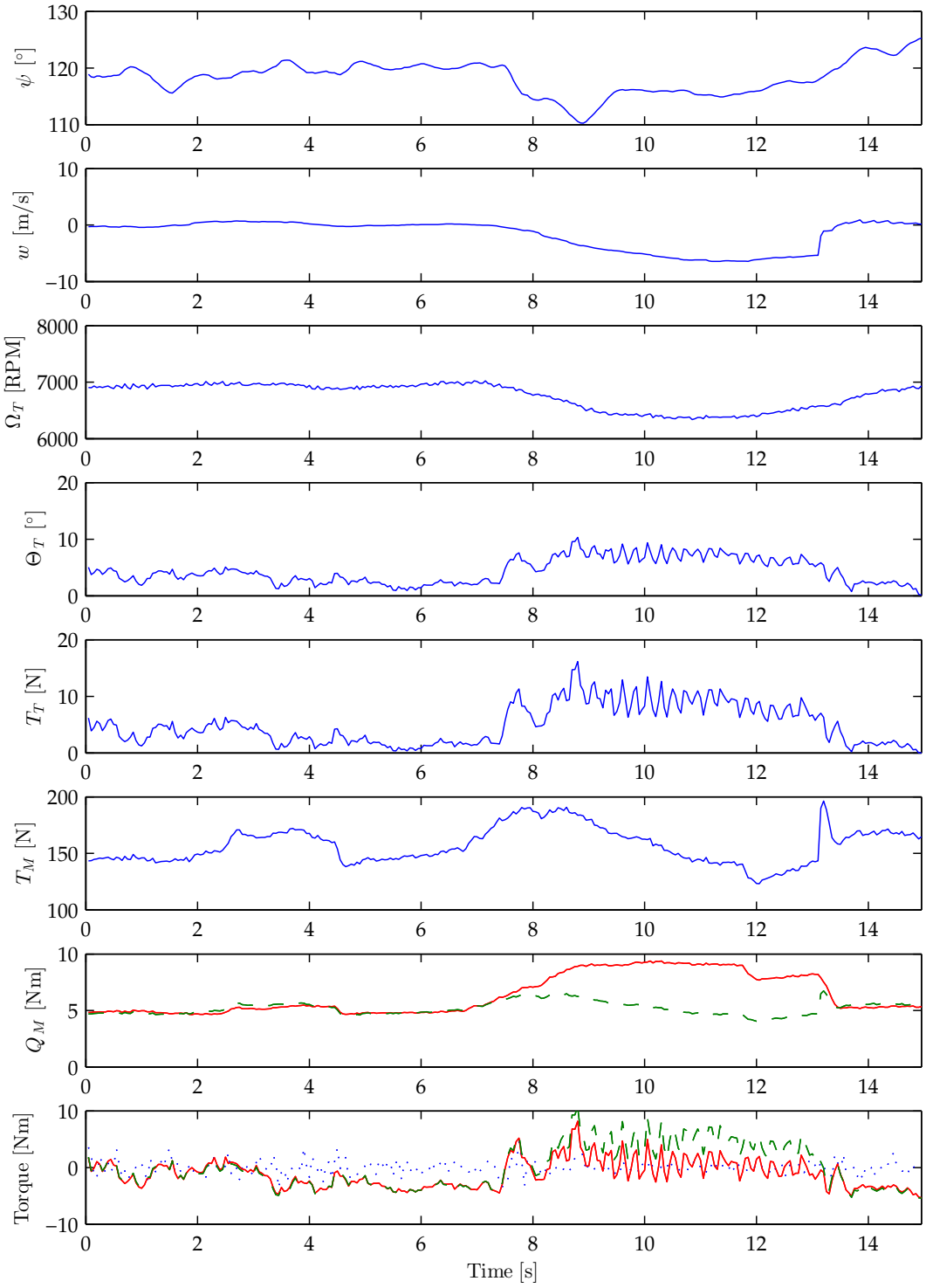


Figure 3.13: Tail thrust and main rotor countertorque model validation data showing large variation in Q_M during fast vertical climb flight. In the bottom two plots the solid line uses the countertorque model with velocity dependence and the dashed line uses the hover model. In the bottom plot the dotted line shows the measured net torque.

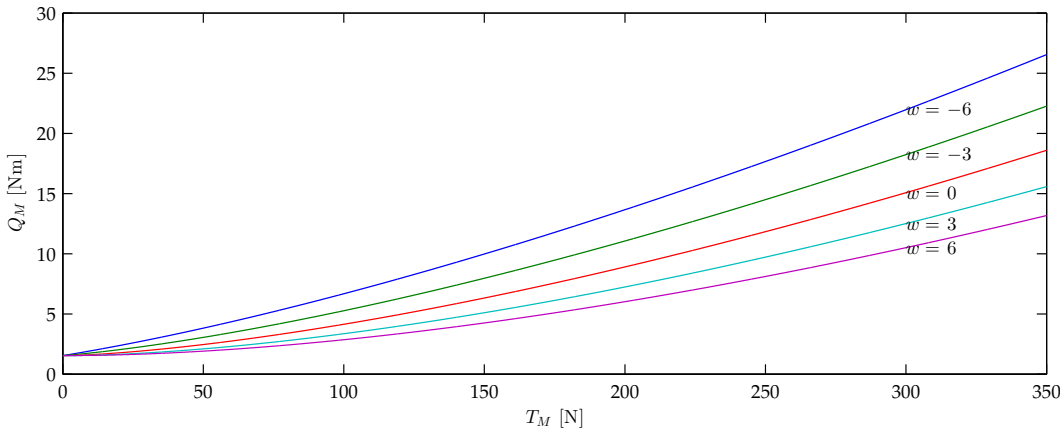


Figure 3.14: Illustration showing how Q_M depends on T_M and w.

resulting expression is

$$Q_M = -\frac{\rho a_M c_M R_M^2 N_b (w - v_i)^2}{4} - \frac{\rho a_M c_M R_M^3 N_b \Omega_M (w - v_i)}{6}\Theta_M + \frac{\rho c_M R_M^4 C_D N_b \Omega_M^2}{8}$$

which when combined with (3.12) and (3.13) gives

$$Q_M = \left(-\frac{w}{2} + \sqrt{\frac{w^2}{4} + \frac{T_M}{2\rho\pi R_M^2}} \right) \frac{T_M}{\Omega_M} + D_M^Q \Omega_M^2. \tag{3.25}$$

This function is plotted as a family of curves over practical ranges of T_M and w in Figure 3.14. Figure 3.13 shows that this model provides a much better fit for the climb data while reducing to (3.23) when $w = 0$ (as is the case for the flight shown in Figure 3.12).

3.3 Conclusions

The rigid body dynamics can be treated with states represented in either the body-fixed or navigation frames. For the purpose of control, considering the translational states in either of these frames can be useful for a given application. However, the rotational states do not have a similar benefit and are therefore only considered in $\mathcal{B}$. When aerodynamic effects are included, the complexity of the expressions of the helicopter dynamics renders them intractable for control design. In order to design controls it is helpful to define non-physical inputs which can then be related to the physical inputs to apply the control. We define the control inputs T_M, T_T, a, b and consider Q_M to be known. These inputs can then be mapped to the physical inputs δ_M, δ_T, δ_r, δ_p using simple relationships we have proposed which can be easily implemented in practice.

Chapter 4

Model-Based PID Control

Before proceeding to the nonlinear control development in Chapters 5 and 6, it is desirable to derive a simple linear control to be used for platform validation. The helicopter dynamics in (3.6) or (3.8) are naturally divided into two subsystems in cascade [16, 27, 58]: an inner-loop system which tracks an orientation reference, and an outer-loop system which tracks a position reference. This hierarchical approach transforms the underactuated open-loop dynamics into two fully actuated subsystems by assuming the outer-loop has direct control of the orientation. An overview of the complete closed-loop system is shown in Figure 4.1. The control objective is to track a position reference $p_d^n : \mathbb{R} \to \mathbb{R}^3$ and a heading reference $\psi_d : \mathbb{R} \to \mathbb{R}$. Both reference trajectories are of class C^4. The inner-outer approach is used throughout this thesis and provides the practical benefit of allowing controls to be tested modularly. For instance, the PID control derived in this chapter was the first design to be tested and tuned in flight. The modularity of the control allowed us to use the working PID inner loop with the outer-loop control from Section 5.2.2 resulting in a more incremental approach to experimental testing.

The organization of this chapter is as follows. Section 4.1 presents the system modelling and explains the assumptions we use to perform the design. The inner-

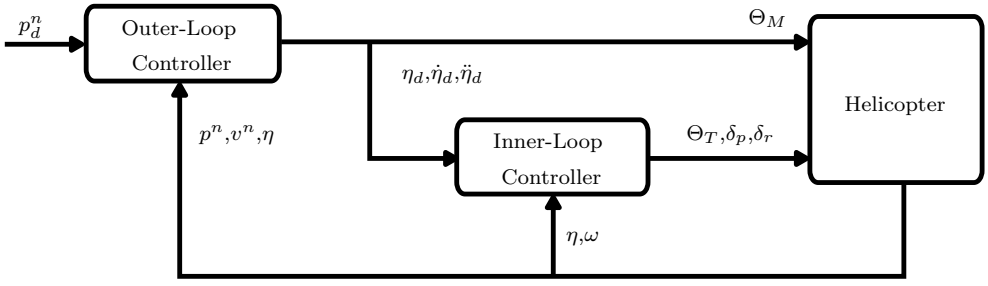


Figure 4.1: Block diagram of the closed-loop dynamics.

loop rotational control is designed in Section 4.2 where, as discussed in Chapter 3, we will only consider the rotational states in the body-fixed frame. We will present the design of the outer-loop control in Sections 4.3 and 4.4 using both the body-fixed and navigation frame representations of the translational states respectively. Simulation results are presented in Section 4.5 which investigate the effect of the velocity dependence in the main rotor thrust and countertorque models. Modifications made to the control design for implementation on the ANCL Helicopter are described in Section 4.6 and experimental results for hover as well as line and circle trajectory tracking flights are given.

4.1 Modelling

As mentioned above, we will perform the translational control design twice by considering the position and velocity in both $\mathcal{N}$ and $\mathcal{B}$. Since the actuation of the translational dynamics appears differently in these coordinate representations, we will not present the model generally.

The objective of the control is to track a position and heading reference. Although the control is computed using two coordinate frames, the reference position p_d^n is always defined in $\mathcal{N}$. We also define a heading reference ψ_d to be tracked by the rotational control. To perform the design, we will use the inputs T_M, T_T, a, b. However, the feedback computed using these inputs will be transformed to the physical inputs Θ_M, Θ_T, δ_r, δ_p using the modelling in Chapter 3. In addition, we make the following assumptions: the SBF and Q_T are negligible, and $l_M \approx (0,\, 0,\, -z_M)^T$ and $l_T \approx (-x_T,\, 0,\, 0)^T$ where z_M and x_T are the distances between the center of mass and the main and tail rotors in the e_3' and e_1' directions respectively.

4.2 Inner-Loop Control

The inner-loop rotational controller will track reference orientation $\eta_d = (\phi_d,\, \theta_d,\, \psi_d)^T$ which consists of the heading reference given in the problem statement and the orientation reference provided by the outer-loop control. The rotational dynamics are

$$\begin{aligned} \dot{\eta} &= W(\eta)\omega \\ J\dot{\omega} &= -\omega \times J\omega + \begin{pmatrix} z_M T_M b \\ z_M T_M a \\ x_T T_T - Q_M \end{pmatrix} \end{aligned}$$

where η is the Euler angle parametrization of R (as in (3.3)); $W(\eta)$ is the transformation from the body-fixed angular velocity to the Euler rates; T_T, a, b are the inputs; and Q_M and T_M are considered to be known time-varying parameters. We

define the input transformation

$$u^r = J^{-1}(z_M T_M b,\, z_M T_M a,\, x_T T_T - Q_M)^T \tag{4.1}$$

and take a linear approximation about the equilibrium $\omega = 0$, $\phi = \theta = 0$, which results in the simplified second order dynamics

$$\begin{aligned} \dot{\eta} &= \omega \\ \dot{\omega} &= u^r. \end{aligned}$$

Introducing the error coordinates $\tilde{\eta} = \eta - \eta_d$ and an integrator state, we obtain the error dynamics

$$\dot{\xi}^r = \tilde{\eta} \tag{4.2a}$$

$$\dot{\tilde{\eta}} = \tilde{\omega} \tag{4.2b}$$

$$\dot{\tilde{\omega}} = u^r - \ddot{\eta}_d. \tag{4.2c}$$

The dynamics (4.2) are stabilized by

$$u^r = \ddot{\eta}_d - K_d^r \dot{\tilde{\eta}} - K_p^r \tilde{\eta} - K_i^r \xi^r \tag{4.3}$$

where we have chosen the gain matrices to be diagonal. This choice is made since there is no intuitive benefit to coupling the feedback. From (4.3), we remark that in addition to η_d we require the outer-loop control to provide the first two derivatives of the orientation reference in order to track a time-varying trajectory.

4.3 Outer-Loop Control in the Navigation Frame

When the translational states are expressed in the navigation frame and the orientation is assumed to be the reference R_d, the open-loop dynamics are

$$\begin{aligned} \dot{p}^n &= v^n \\ m\dot{v}^n &= R_d \begin{pmatrix} 0 \\ 0 \\ -T_M \end{pmatrix} + mge_3 \end{aligned}$$

where the inputs are ϕ_d, θ_d, and T_M. Since the value of ψ_d is not constrained by the open-loop equilibrium set of the complete dynamics (i.e., there is no expectation for ψ to be small), we wish to approximate the functional dependence on the inputs without making an assumption regarding the heading. Indeed, while practical trajectories require small variations in ϕ_d and θ_d, ψ_d can take any value. We thus

define the input force vector

$$\nu = R_{yd} R_{xd} f$$

where we have used the fact that the orientation can be decomposed into three rotations (3.2). Expanding this product we have

$$\nu = -T_M \begin{pmatrix} \cos\phi_d \sin\theta_d \\ -\sin\phi_d \\ \cos\theta_d \cos\phi_d \end{pmatrix} + mge_3 \tag{4.4}$$

which is approximated by

$$\nu \approx \begin{pmatrix} -mg\theta_d \\ mg\phi_d \\ -T_M + mg \end{pmatrix}. \tag{4.5}$$

Finally, we define the input transformation

$$u^{tn} = R_{zd}\nu, \tag{4.6}$$

introduce the error states $\tilde{p}^n = p^n - p_d^n$ and $\tilde{v}^n = v^n - \dot{p}_d^n$, and include an integrator state in the control which results in the triple integrator error dynamics

$$\dot{\xi}^{tn} = \tilde{p}^n \tag{4.7a}$$

$$\dot{\tilde{p}}^n = \tilde{v}^n \tag{4.7b}$$

$$m\dot{\tilde{v}}^n = u^{tn} - m\ddot{p}_d^n. \tag{4.7c}$$

We stabilize these dynamics using the feedback

$$u^{tn} = m\ddot{p}_d^n - K_d^t \tilde{v}^n - K_p^t \tilde{p}^n - K_i^t \xi^{tn} \tag{4.8}$$

where the gain matrices K_j^t, $j \in \{d, p, i\}$ are symmetric positive definite. Remark that if we make the stronger (practical) assumption that the gain matrices are diagonal we obtain three decoupled SISO third order systems which are known to be stable for (scalar) $k_d k_p > k_i$. However, for the sake of generality we will show stability for the former case. Consider the possible Lyapunov Function Candidate (LFC)

$$V = \frac{1}{2} \begin{pmatrix} \tilde{v}^n \\ \tilde{p}^n \\ \xi^{tn} \end{pmatrix}^T \begin{pmatrix} mI & \alpha mI & 0 \\ \alpha mI & K_p^t + \alpha K_d^t & K_i^t \\ 0 & K_i^t & \alpha K_i^t \end{pmatrix} \begin{pmatrix} \tilde{v}^n \\ \tilde{p}^n \\ \xi^{tn} \end{pmatrix} \tag{4.9}$$

which is based on [59]. In order to establish V is an LFC we must show that it is positive definite. To proceed we make use of the following lemmas whose proofs can be found in the respective reference.

Lemma 4.1 ([60])**.** *Given a real symmetric matrix which is partitioned as*

$$\begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$$

where A and C are square. This matrix is positive definite if and only if A is positive definite and $C > B^T A^{-1} B$.

Lemma 4.2 ([59])**.** *If N, P, and Q are positive definite symmetric matrices and $\alpha > 0$ then*

$$N - P - \frac{1}{\alpha} Q > 0$$

implies

(a) $N - P > 0$

(b) $\alpha Q - Q(N - P)^{-1} Q > 0$.

Following [59] if we take $A = mI$, $B = \begin{pmatrix} \alpha mI & 0 \end{pmatrix}$, and $C = \begin{pmatrix} K_p^t + \alpha K_d^t & K_i^t \\ K_i^t & \alpha K_i^t \end{pmatrix}$ then since mI is trivially positive definite we only need

$$C - B^T A^{-1} B = \begin{pmatrix} K_p^t + \alpha K_d^t - \alpha^2 mI & K_i^t \\ K_i^t & \alpha K_i^t \end{pmatrix}$$

to be positive definite. Invoking Lemma 4.1 a second time with $A = K_p^t + \alpha K_d^t - \alpha^2 mI$, $B = K_i^t$, and $C = \alpha K_i^t$ we are left with the requirements

$$K_p^t + \alpha K_d^t - \alpha^2 mI > 0 \tag{4.10}$$

$$\alpha K_i^t - K_i^t (K_p^t + \alpha K_d^t - \alpha^2 mI)^{-1} K_i^t > 0. \tag{4.11}$$

Conditions which ensure (4.11) holds are found using Lemma 4.2 with $N = K_p^t + \alpha K_t^d$, $P = \alpha^2 mI$, and $Q = K_i^t$ which results in the condition

$$K_p^t + \alpha K_d^t - \alpha^2 mI - \frac{1}{\alpha} K_i^t > 0.$$

Therefore, if we choose

$$\alpha K_d^t - \alpha^2 mI > 0 \tag{4.12}$$

then (4.10) is satisfied, and if in addition we choose

$$K_p^t - \frac{1}{\alpha} K_i^t > 0 \tag{4.13}$$

we also satisfy (4.11). Since we have now established the LFC in (4.9) is positive definite, it remains to compute the derivative of V along the closed loop trajectories

of (4.7)

$$\begin{aligned}\dot{V} &= (\tilde{v}^n)^T(-K_d^t\tilde{v}^n - K_p^t\tilde{p}^n - K_i^t\xi^{tn}) + \alpha(\tilde{p}^n)^T(-K_d^t\tilde{v}^n - K_p^t\tilde{p}^n - K_i^t\xi^{tn}) \\ &\quad + \alpha m(\tilde{v}^n)^T\tilde{v}^n + (\tilde{p}^n)^T(K_p^t + \alpha K_d^t)\tilde{v}^n + (\tilde{p}^n)^T K_i^t\tilde{p}^n + (\xi^{tn})^T K_i^t\tilde{v}^n \\ &\quad + \alpha(\xi^{tn})^T K_i^t\tilde{p}^n \\ &= -(\tilde{v}^n)^T(K_d^t - \alpha mI)\tilde{v}^n - (\tilde{p}^n)^T(\alpha K_p^t - K_i^t)\tilde{p}^n.\end{aligned}$$

Thus, we see $\dot{V} < 0$ as long as (4.12) and (4.13) are satisfied. In order to make the control law explicit we will solve for the inputs. From (4.5), (4.6), and (4.8) we find

$$\begin{pmatrix}\phi_d \\ \theta_d \\ T_M\end{pmatrix} = \begin{pmatrix}0 & \frac{1}{mg} & 0 \\ -\frac{1}{mg} & 0 & 0 \\ 0 & 0 & -1\end{pmatrix} R_{zd}^T\left(m\ddot{p}_d^n - K_d^t\tilde{v}^n - K_p^t\tilde{p}^n - K_i^t\xi^{tn}\right) + mge_3. \quad (4.14)$$

We remark that due to the appearance of the desired heading the expressions for ϕ_d and θ_d it is necessary for ψ_d to be class C^4.

4.4 Outer-Loop Control in the Body-Fixed Frame

The control design in the body-fixed frame will follow a similar approach to the design in the navigation frame presented in Section 4.3. Beginning with the translational dynamics presented in (3.6) and using the assumptions stated in Section 4.1, we have

$$\begin{aligned}\dot{p}^b &= -\omega \times p^b + v^b \\ m\dot{v}^b &= -\omega \times mv^b + \begin{pmatrix}0 \\ 0 \\ -T_M\end{pmatrix} + mgR_d^T e_3\end{aligned}$$

where the last term is the orientation of gravity in the body-fixed frame and can be expressed using the Euler angles as

$$mgR_d^T e_3 = mg\begin{pmatrix}-\sin\theta_d \\ \cos\theta_d\sin\phi_d \\ \cos\theta_d\cos\phi_d\end{pmatrix} \approx mg\begin{pmatrix}-\theta_d \\ \phi_d \\ 1\end{pmatrix}.$$

We make the input transformation

$$u^{tb} = \begin{pmatrix}0 \\ 0 \\ -T_M\end{pmatrix} + mg\begin{pmatrix}-\theta_d \\ \phi_d \\ 1\end{pmatrix}, \quad (4.15)$$

which is the same as (4.4). Next, we define the error coordinates $\tilde{p}^b = p^b - p_d^b$ and $\tilde{v}^b = v^b - v_d^b$ where $p_d^b = R^T p_d^n$, $v_d^b = R^T \dot{p}_d^n$ and $\ddot{p}_d^b = R^T \ddot{p}_d^n$, and introduce an integrator state to obtain

$$\dot{\xi}^{tb} = -\omega \times \xi^{tb} + \tilde{p}^b \tag{4.16a}$$

$$\dot{\tilde{p}}^b = -\omega \times \tilde{p}^b + \tilde{v}^b \tag{4.16b}$$

$$m\dot{\tilde{v}}^b = -\omega \times m\tilde{v}^b + u^{tb} - m\ddot{p}_d^b. \tag{4.16c}$$

We choose the control law

$$u^{tb} = m\ddot{p}_d^b - K_d^t \tilde{v}^b - K_p^t \tilde{p}^b - K_i^t \xi^{tb}, \tag{4.17}$$

and the LFC

$$V = \frac{1}{2} \begin{pmatrix} \tilde{v}^b \\ \tilde{p}^b \\ \xi^{tb} \end{pmatrix}^T \begin{pmatrix} mI & \alpha mI & 0 \\ \alpha mI & R^T(K_p^t + \alpha K_d^t)R & R^T K_i^t R \\ 0 & R^T K_i^t R & \alpha R^T K_i^t R \end{pmatrix} \begin{pmatrix} \tilde{v}^b \\ \tilde{p}^b \\ \xi^{tb} \end{pmatrix}.$$

By employing the fact that for $x \in \{\mathbb{R}^3 \colon x \neq 0\}$, $x^T M x = 0$ if M is skew symmetric, and the vector identity $a \cdot (b \times c) = c \cdot (a \times b)$ for $a, b, c \in \mathbb{R}^3$, $\dot{V}$ can be found to be

$$\dot{V} = -(\tilde{v}^b)^T (K_d^t - \alpha mI)\tilde{v}^b - (\tilde{p}^b)^T R^T (\alpha K_p^t - K_i^t) R \tilde{p}^b.$$

Therefore, $V > 0$ and $\dot{V} < 0$ if (4.12) and (4.13) are satisfied. We remark that the $\omega\times$ terms in (4.16) do not affect stability. This result is due to the fact that these terms are passive [61]. We complete the body-fixed frame design by solving for the inputs using (4.15)

$$\begin{pmatrix} \phi_d \\ \theta_d \\ T_M \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{mg} & 0 \\ -\frac{1}{mg} & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} (m\ddot{p}_d^b - K_d^t \tilde{v}^b - K_p^t \tilde{p}^n - K_i^t \xi^{tb}) + mge_3. \tag{4.18}$$

As a comparison between the designs given in the two coordinate frames, we can approximate the orientation by $R \approx R_{zd}$ and relate the gain matrices by $\tilde{K}_j^t = R^T K_j^t R$, $j \in \{d, p, i\}$ then (4.17) becomes $u_t^b = R_{zd}^T(\ddot{p}_d^n - \tilde{K}_d^t \tilde{v}^n - \tilde{K}_d^t \tilde{p}^n - \tilde{K}_i^t \xi^{tn})$ and the inputs are

$$\begin{pmatrix} \phi_d \\ \theta_d \\ T_M \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{mg} & 0 \\ -\frac{1}{mg} & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} R_{zd}^T (m\ddot{p}_d^n - \tilde{K}_d^t \tilde{v}^n - \tilde{K}_p^t \tilde{p}^n - \tilde{K}_i^t \xi^{tn}) + mge_3$$

which is the same as (4.14). Therefore, the translational control laws designed in $\mathcal{N}$

and $\mathcal{B}$ are locally equivalent. We note that the desired heading is used as a result of the inner-outer approximation which assumes the orientation is tracking its reference perfectly.

4.5 Simulation Results

The system response using the control law given above is simulated to ensure validity of the design. For clarity, we present a summary of the control law which is used and how the physical inputs are calculated. Since the control laws (4.14) and (4.18) are approximately equivalent, we only need to perform simulations using one of them to verify the design. Furthermore, from a practical perspective they require the same amount of code to implement. The principal difference between the controls is the coordinate system in which the gains are chosen. Since we will use diagonal matrices in our implementation, each non-zero entry of the gain matrix represents a scaling along a particular coordinate axis. However, it is more intuitive to choose these gains along the body-fixed axes. Therefore we use (4.18) to define the outer-loop control

$$\begin{pmatrix} \phi_d \\ \theta_d \\ T_M \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{mg} & 0 \\ -\frac{1}{mg} & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} (m\ddot{p}_d^b - K_d^t\tilde{v}^b - K_p^t\tilde{p}^b - K_i^t\xi^t) + \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} \tag{4.19}$$

where the subscript $_d$ denotes reference quantities for the rotational control, $\xi^t = \xi^{tb}$, and we will also use the notation $u^t = u^{tb}$. In order to fully define the reference orientation we must compute two derivatives of ϕ_d and θ_d. These derivatives are easily computed using the first two coordinates of (4.15). Thus, we must compute two derivatives of u^t. The first derivative is

$$\dot{u}^t = m(p_d^b)^{(3)} + ((K_d^t)^2 - K_p^t)\tilde{v}^b + (K_d^tK_p^t - K_i^t)\tilde{p}^b + K_d^tK_i^t\xi^t, \tag{4.20}$$

and the second derivative is

$$\begin{aligned} \ddot{u}^t = {} & m(p_d^b)^{(4)} + (-(K_d^t)^3 + 2K_d^tK_p^t - K_i^t)\tilde{v}^b + (-(K_d^t)^2K_p^t + K_d^tK_i^t + (K_p^t)^2)\tilde{p} \\ & + (-(K_d^t)^2K_i^t + K_p^tK_i^t)\xi^t. \end{aligned} \tag{4.21}$$

Then given η_d and its derivatives we can evaluate the inner loop control using (4.1) and (4.3) to obtain

$$\begin{pmatrix} a \\ b \\ T_M \end{pmatrix} = \begin{pmatrix} 0 & \frac{J_y}{z_M T_M} & 0 \\ \frac{J_x}{z_M T_M} & 0 & 0 \\ 0 & 0 & \frac{J_z}{x_T} \end{pmatrix} (\ddot{\eta}_d - K_d^r\dot{\tilde{\eta}} - K_p^r\tilde{\eta} - K_i^r\xi^r) + \begin{pmatrix} 0 \\ 0 \\ \frac{Q_M}{x_T} \end{pmatrix}.$$

r_0	7 m
T_0	60 s
α_0	-0.1 m/s^2

Table 4.1: Reference trajectory parameters used for simulation.

We can now compute the physical inputs. From (3.20a), the cyclics are simply

$$\delta_p = \frac{a}{k_p}$$
$$\delta_r = \frac{b}{k_r}.$$

The main rotor collective pitch is given in (3.15) as

$$\Theta_M = \frac{1}{C_M \Omega_M^2}\left(T_M - \frac{D_M \Omega_M w}{2} + \begin{cases} \frac{D_M \Omega_M}{\sqrt{2\rho\pi} R_M}\sqrt{T_M + \frac{\rho\pi w^2 R_M^2}{2}} & w \geq 0 \\ -\frac{D_M \Omega_M}{\sqrt{2\rho\pi} R_M}\sqrt{T_M + \frac{\rho\pi w^2 R_M^2}{2}} & w < 0 \end{cases}\right),$$

and the tail rotor collective pitch is from (3.21) to be

$$\Theta_T = \frac{1}{C_T \Omega_T^2}\left(|T_T| + \frac{D_T \Omega_T}{\sqrt{2\pi\rho} R_T}\sqrt{|T_T|}\right) \cdot \begin{cases} 1 & T_T \geq 0 \\ -1 & T_T < 0 \end{cases}. \tag{4.22}$$

For implementation on the helicopter, the main and tail rotor collective pitch is then converted to a servo pulse width using (3.18) and (3.24) respectively. However, for the purpose of the simulation we use collective pitch angle as the input to the plant since it carries intuitive meaning. In addition, we assume a constant main rotor speed $\Omega_M = 1500$ RPM which implies a tail rotor speed $\Omega_T = 7000$ RPM. These rotor speeds correspond to the ANCL helicopter as in Figures 3.4 and 3.12. The reference trajectory is helical with a constant upwards acceleration

$$p_d = R^T \begin{pmatrix} r_0 \cos\left(\frac{2\pi}{T_0} t\right) \\ r_0 \sin\left(\frac{2\pi}{T_0} t\right) \\ \frac{\alpha_0 t^2}{2} \end{pmatrix} \tag{4.23}$$

where r_0 is the radius, T_0 is the period, and α_0 is the vertical acceleration. The values of the parameters are given in Table 4.1. The heading reference was the constant $\psi_d = 0$ which requires time-varying rotational inputs in steady-state as opposed to the case when the heading is always tangent to the circle. The controller gains used for the simulation are given in Table 4.2. The results of the simulation are shown in Figure 4.2. These results show the system response when the controller uses the exact main rotor thrust and countertorque models (including vertical velocity

K_i^r	$0.1I$ s^{-3}
K_p^r	diag(10, 10, 7) s^{-2}
K_d^r	$5I$ s^{-1}
K_i^t	$0.1I$ kg/s^3
K_p^t	$2I$ kg/s^2
K_d^t	$3I$ kg/s

Table 4.2: Controller gains used for simulation.

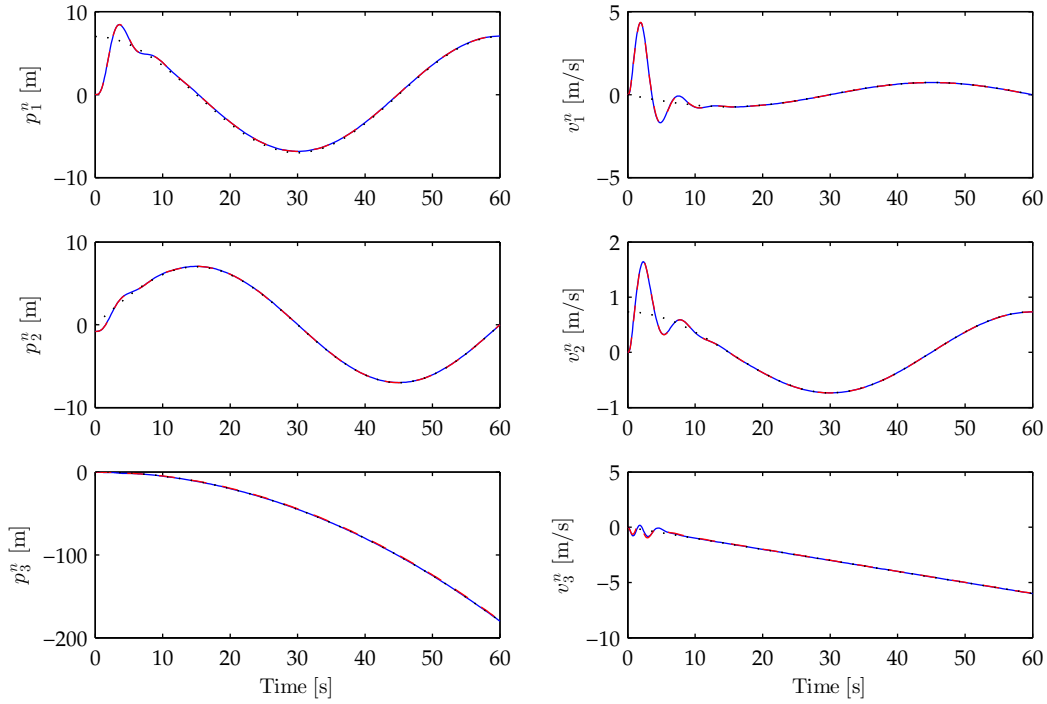
dependence) and when the controller uses the hover approximation. When the hover approximation is used, the countertorque disturbs the yaw dynamics with a time-varying magnitude which prevents the integrator states in the controller from removing the effect. The errors in the vertical position and velocity are shown in Figure 4.2d. In Figure 4.2e the same errors are shown except for simulations where the integrator states have been removed from the control. We observe that without the help of the integrators the position error is unbounded when the hover model is used. However, as expected the control using the exact model is able to track the reference.

4.6 Experimental Results

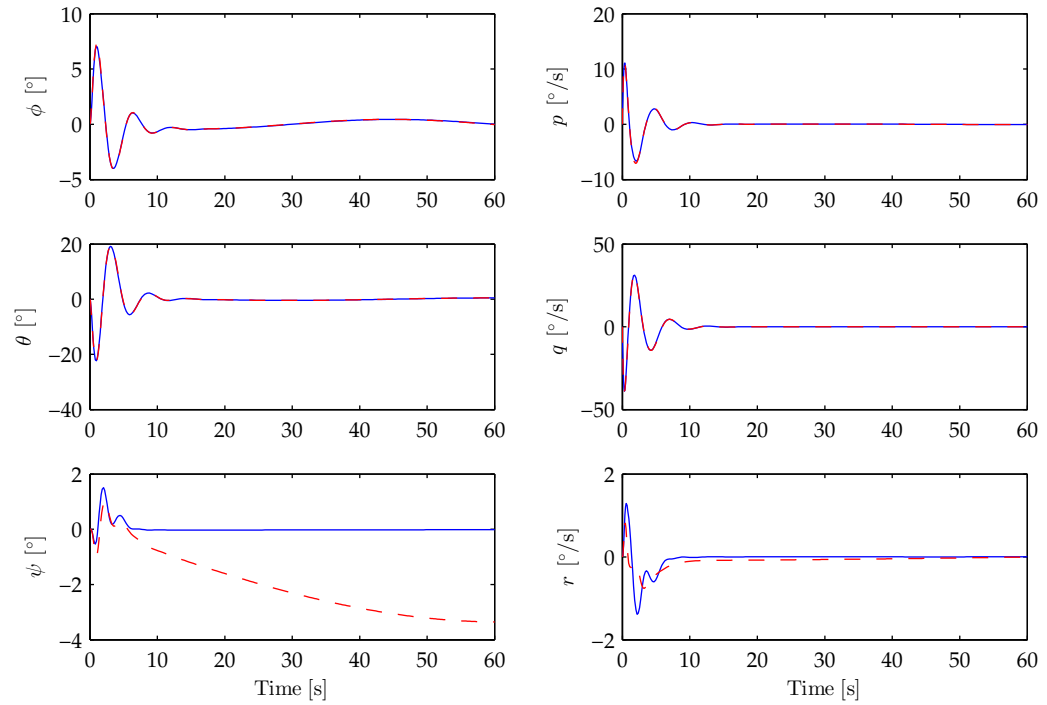
While simulation is a useful first step in testing a design, it is necessary to conduct flight experiments to fully validate the autopilot. For safety, the autopilot is programmed to control the cyclic inputs while allowing the pilot to control the main and tail collective pitch as well as the engine throttle. Therefore, we only consider the first two components of the position, velocity, orientation, and angular velocity. Consequently, the dimension of the gain matrices and inputs used in the control laws is reduced. Let $\underline{\eta} = (\phi, \theta)^T$ be the reduced dimension orientation, $\underline{\eta}_d$ and $\tilde{\underline{\eta}}$ be the corresponding references and errors respectively, and $\underline{\xi}$ be the reduced dimension integrator state. For implementational simplicity we assume $\dot{\underline{\eta}}_d = \ddot{\underline{\eta}}_d = 0$. Indeed, this assumption is reasonable for slow time-varying trajectories and reduces the potential for software bugs by simplifying the control law. We then rewrite the control law for the rotational subsystem (4.3) as

$$\underline{u}^r = -K_d^r \dot{\underline{\eta}} - K_p^r \tilde{\underline{\eta}} - K_i^r \underline{\xi}^r \tag{4.24}$$

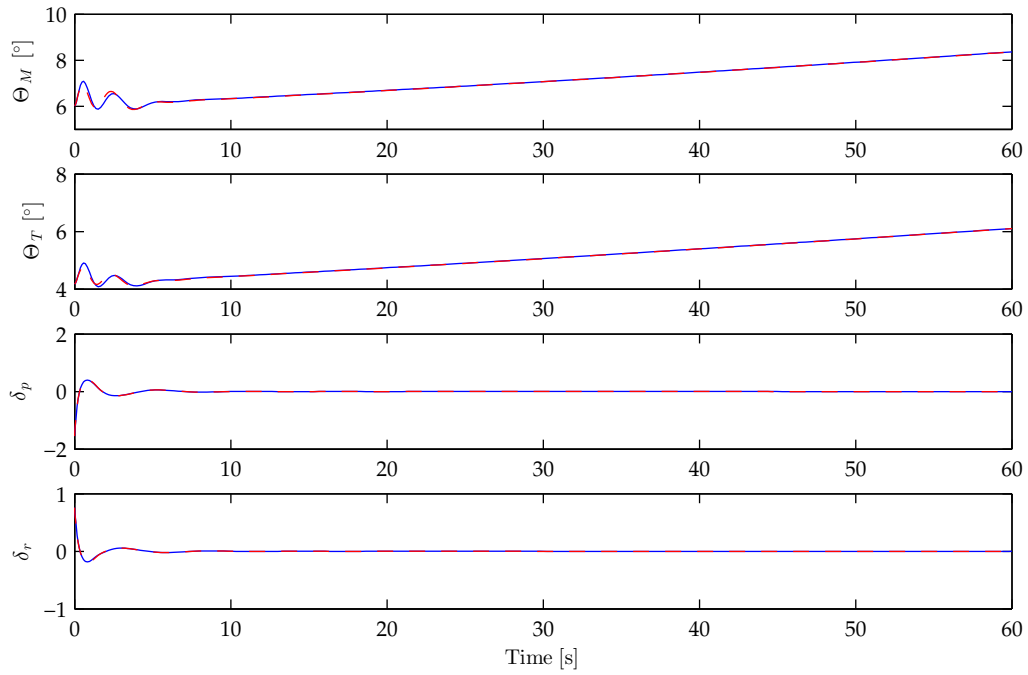
where $\underline{u}^r$ is the reduced dimension control input and we have slightly abusively used the previous notation for the reduced dimension gain matrices. To complete the control implementation we must use the physical inputs. By assuming the pilot will



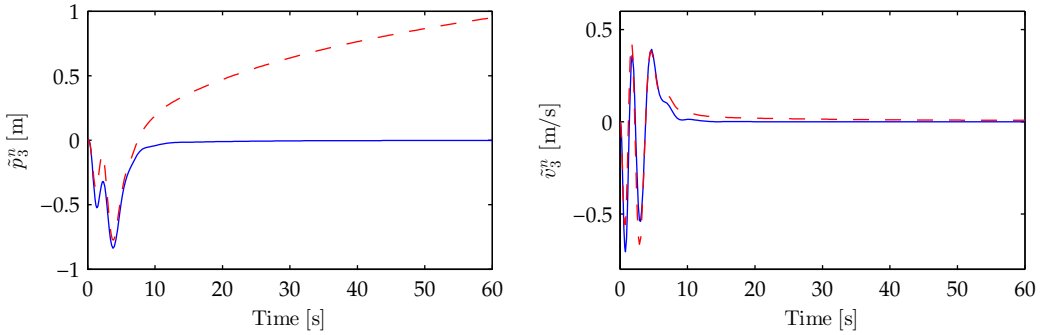
(a) Translational states. The solid line shows the response to the control using the exact models, the dashed line shows the hover models, and the dotted line shows the reference.



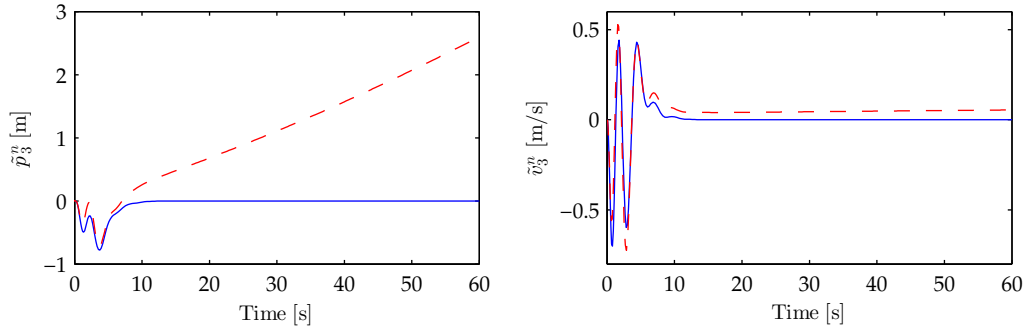
(b) Rotational states. The solid line shows the control using the exact models, and the dashed line shows the response using the hover models.



(c) Control effort. The solid line shows the exact model, and the dashed line shows the hover model.



(d) Vertical position and velocity errors. The solid line shows the exact model, and the dashed line shows the hover model.



(e) Vertical position and velocity errors when the integrator states have been removed from the control. The solid line shows the exact model, and the dashed line shows the hover model.

Figure 4.2: Simulated system response for the control including the physical input mappings. Two simulations were run: one using the exact thrust and countertorque models in the control and the other using the hover thrust and countertorque models in the control. In both cases the plant was simulated using the exact models.

$\bar{K}_d^r$	$0.3I$ s/rad
$\bar{K}_p^r$	$0.8I$ rad^{-1}
$\bar{K}_i^r$	$0.1I$ rad^{-1}s^{-1}
$\bar{K}_d^t$	$0.2I$ rad s/m
$\bar{K}_p^t$	$0.05I$ rad/m
$\bar{K}_i^t$	$0.005I$ rad/m s

Table 4.3: Controller gains used for experimental flight testing.

maintain a close to constant altitude we use $T_M = mg$ and obtain

$$\begin{pmatrix} \delta_r \\ \delta_p \end{pmatrix} = -\bar{K}_d^r \dot{\underline{\eta}} - \bar{K}_p^r \tilde{\underline{\eta}} - \bar{K}_i^r \underline{\xi}^r \tag{4.25}$$

where

$$\bar{K}_j^r = \frac{1}{z_M mg} \begin{pmatrix} \frac{J_x}{k_r} & 0 \\ 0 & \frac{J_y}{k_p} \end{pmatrix} K_j^r, \qquad j \in \{d, p, i\}.$$

Similarly, let $\underline{p}^b$, $\underline{v}^b$ be the first two components of p^b, v^b, and $\underline{p}_d$ and $\tilde{\underline{p}}^b$ be its corresponding references and errors respectively. Then the translational control law (4.19) becomes

$$\begin{pmatrix} \phi_d \\ \theta_d \end{pmatrix} = -\bar{K}_d^t \underline{v}^b - \bar{K}_p^t \tilde{\underline{p}}^b - \bar{K}_i^t \underline{\xi}^t \tag{4.26}$$

in terms of the physical inputs where

$$\bar{K}_j^t = \frac{1}{g} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} K_j^t, \qquad j \in \{d, p, i\}.$$

In order to prevent the controller's integrator states from diverging, they were programmed to reset when an upper bound was reached. The bounds were $|\xi_j^r| \leq 10$ rad s and $|\xi_j^t| \leq 10$ m s where $j \in \{1, 2\}$ denotes the component.

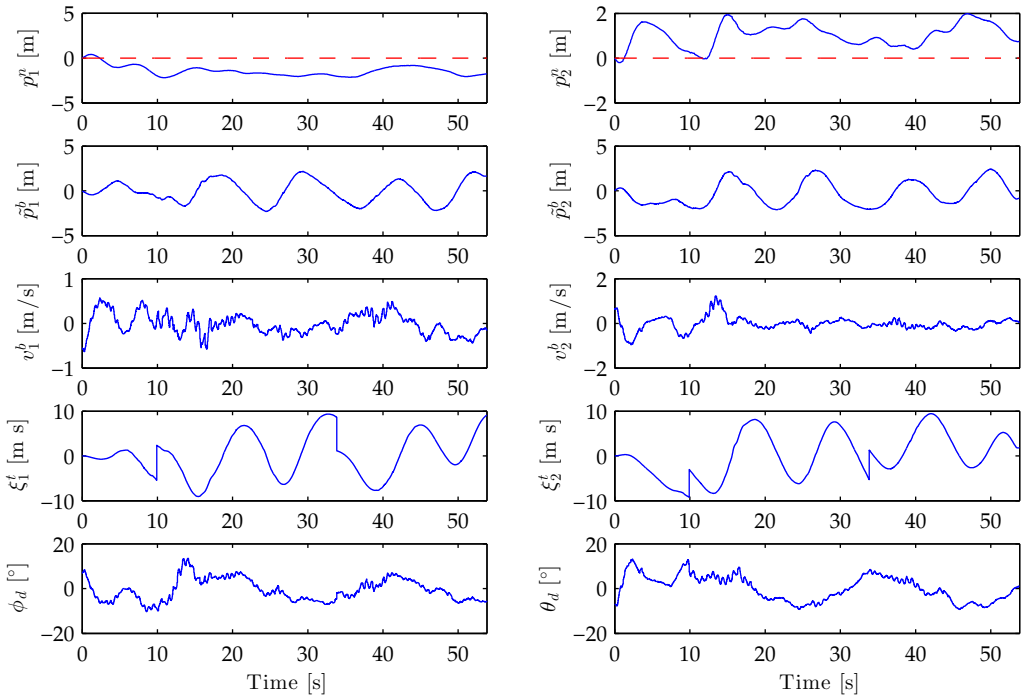
The testing was performed over several flight sessions wherein the authority of the autopilot was mixed in slowly while the controller gains were tuned. The final gains used for the validation flights are given in Table 4.3. During the flights described in the following, there was a 25 km/h wind coming from the north-west with strong gusts.

The first reference we used to validate the control was a constant setpoint (i.e., hover). This reference was implemented to record the position when the pilot engaged the autopilot, thus minimizing initial condition error. Figure 4.3a shows the translational states and inputs, Figure 4.3b shows the rotational states and inputs, and Figure 4.3c shows the heading and height measurements. We observe that the control behaves as expected with translation tracking error of a few meters (ap-

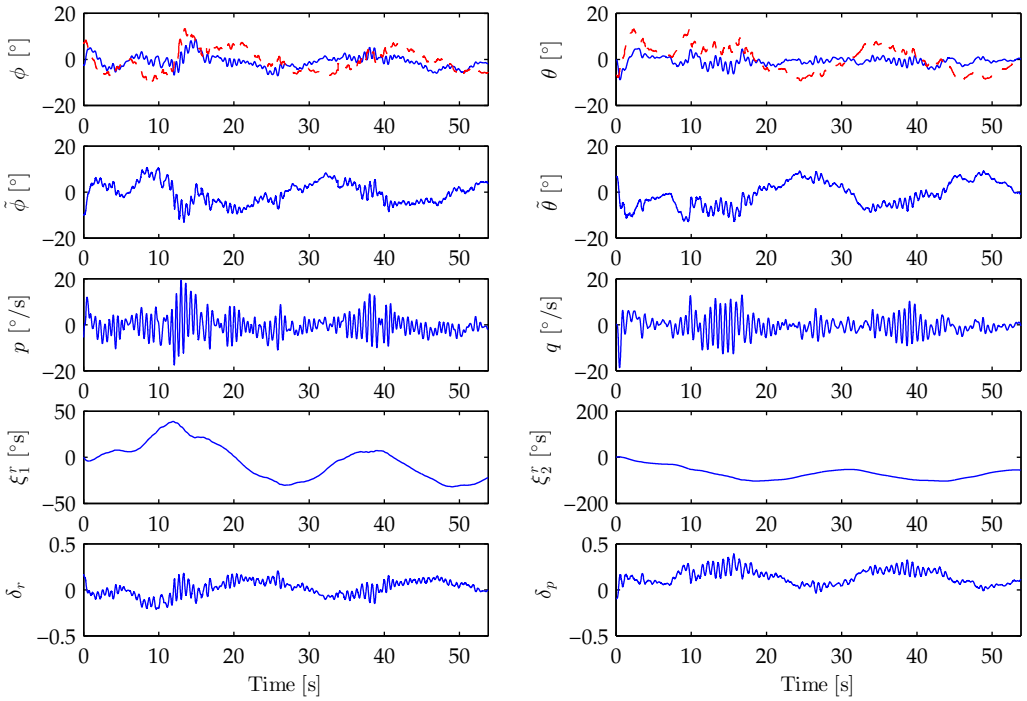
proximately one rotor diameter). This performance is achieved despite a constant change of heading as the pilot performs two rotations. This rotation emphasizes the dependence of ϕ_d and θ_d on the heading measurement, but it also causes the direction of the wind expressed in $\mathcal{B}$ to change, which forces the integrator states to oscillate as they attempt to compensate for the disturbance. This inability for the integrator states to converge is responsible for the steady state error evident in Figure 4.3d. The general drift to the south-east is likely due to the effect of the wind disturbance.

The second reference is a line trajectory which is implemented as follows. The operator specifies an initial hover duration, speed, and displacements in the e'_1 and e'_2 directions. When the autopilot is engaged, the initial position is recorded and the reference remains constant for the hover duration. The displacements set by the operator in $\mathcal{B}$ is transformed into $\mathcal{N}$ to obtain a final position. A linear trajectory is then computed with the speed configured by the operator and passed as a reference to the autopilot. When the reference reaches the final position it remains constant for all time. This procedure gives a natural coordinate frame for the operator to specify the trajectory. However, once the trajectory begins it is not affected by changes in heading. If the trajectory is generated in $\mathcal{B}$, changes in heading would warp the resulting trajectory which would not give the expected result from the point of view of the operator. The results are shown in Figure 4.4, where Figure 4.4a shows the response of the translational subsystem, Figure 4.4b shows the response of the rotational subsystem, Figure 4.4c shows the heading and height measurements, and Figure 4.4d shows the position in $\mathcal{N}$ plotted parametrically. These results once again show tracking error within a few meters. At 22 s the pilot yawed the aircraft by more than 100°. This significant change of heading caused the error evident near the end of the trajectory in Figure 4.4d. The probable cause of the error is the integrator states requiring time to converge to the new wind direction, similar to the effect observed during hover stabilization.

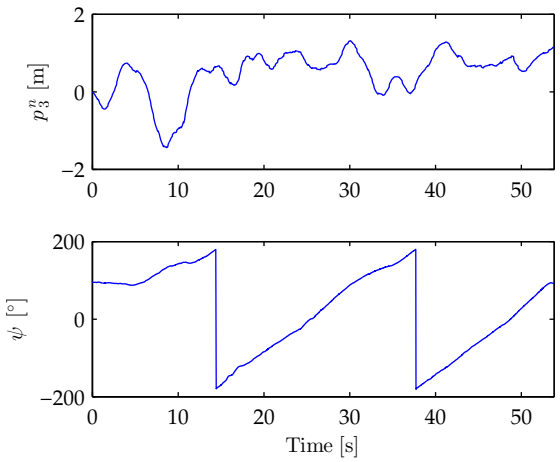
The third and final reference trajectory used for validation is a circle which is implemented as follows. Similar to the line, the operator specifies an initial hover duration, radius, and tangential speed. When the autopilot is engaged the initial position is recorded. The location of the center of the circle is computed by adding a vector to the initial position whose length was the radius of the circle and has the same direction as the heading of the helicopter (i.e., the helicopter is initially always facing the center of the circle). Once again, since the reference is envisioned with respect to the orientation of the helicopter, the reference is transformed into $\mathcal{N}$ to prevent warping if the heading is changed during the flight. Therefore, the initial position must be converted into an initial angle in the e_1-e_2 plane. After the initial hover, this angle is varied to provide the desired tangential reference speed. The



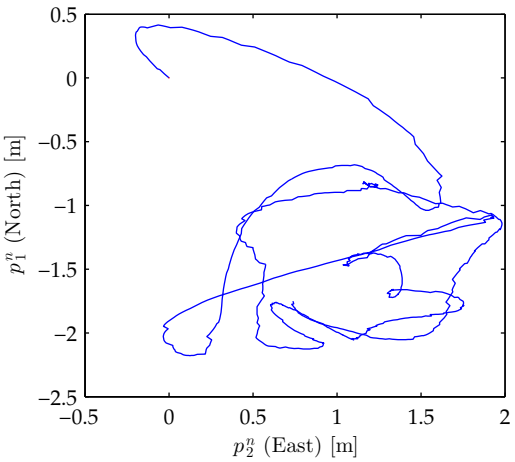
(a) Translational states, errors and control effort. The reference position is shown by the dashed line.



(b) Rotational states, errors and control effort. The reference is shown by the dashed line.



(c) Heading and height measurements for setpoint regulation with full yaw rotation.



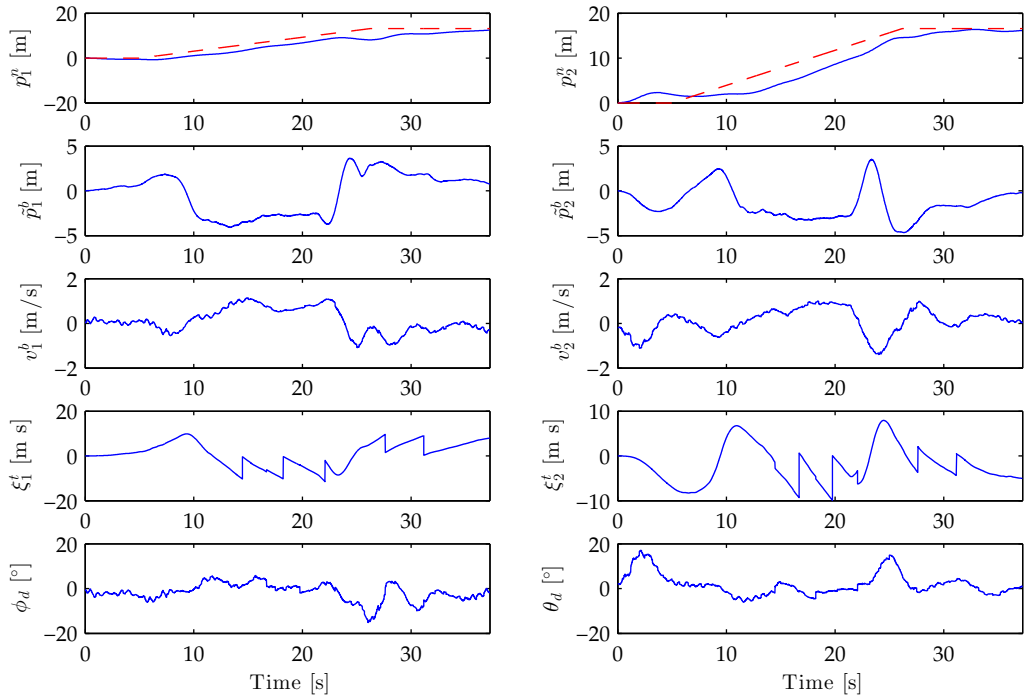
(d) Parametric plot of the horizontal position in $\mathcal{N}$.

Figure 4.3: System response for setpoint regulation with a full yaw rotation. The reference location is the origin of the navigation frame.

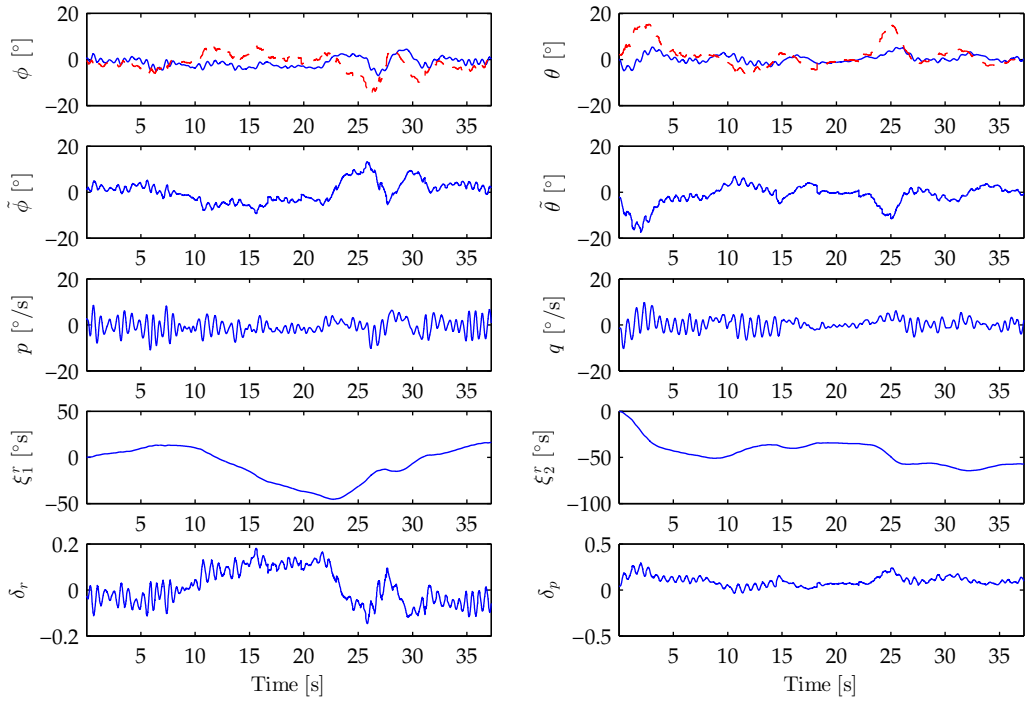
results are shown in Figure 4.5. Figures 4.5a and 4.5b show the translational and rotational controller responses respectively, Figure 4.5c shows the heading and height measurements, and Figure 4.5d shows a parametric plot of the horizontal position. As in the case of the line tracking, the integrator reset is reached more frequently than in the case of hover regulation. A likely explanation for this difference is due to the time-varying reference. Since the control assumes a constant reference, as the reference velocity increases some amount of error is expected. However, despite any additional disturbance, we remark that the performance is consistent between the two complete periods of the circular trajectory and the error remains within approximately 2-3 m.

4.7 Conclusions

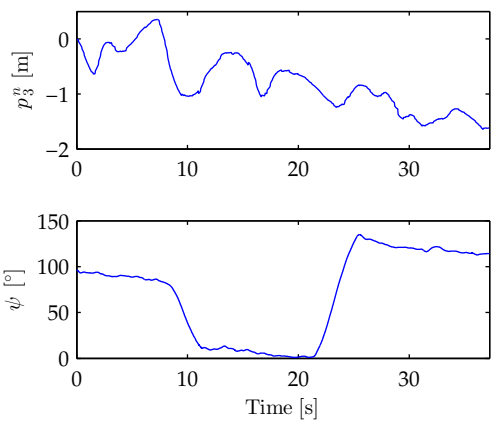
We have derived a simple model-based linear control law and successfully used it in experimental flight tests. The simulation results presented in this chapter emphasize the importance of including an accurate thrust model in the control. Overall, the experimental results showed good tracking performance despite a strong wind disturbance. The flight tests also show correct hardware and software functionality. Furthermore, this control is not only useful for platform validation but can also serve as a basis of comparison for other control methods.



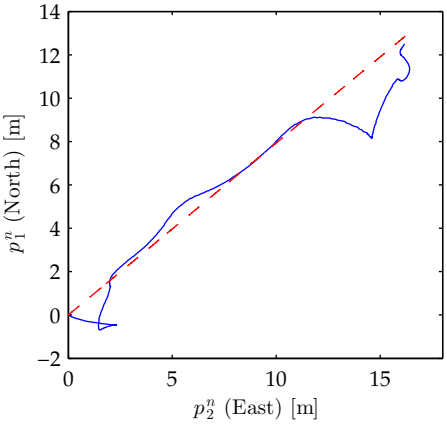
(a) Translation states, errors and control effort. The reference is shown by the dashed line.



(b) Rotational states, errors and control effort for line tracking. The reference is shown by the dashed line.

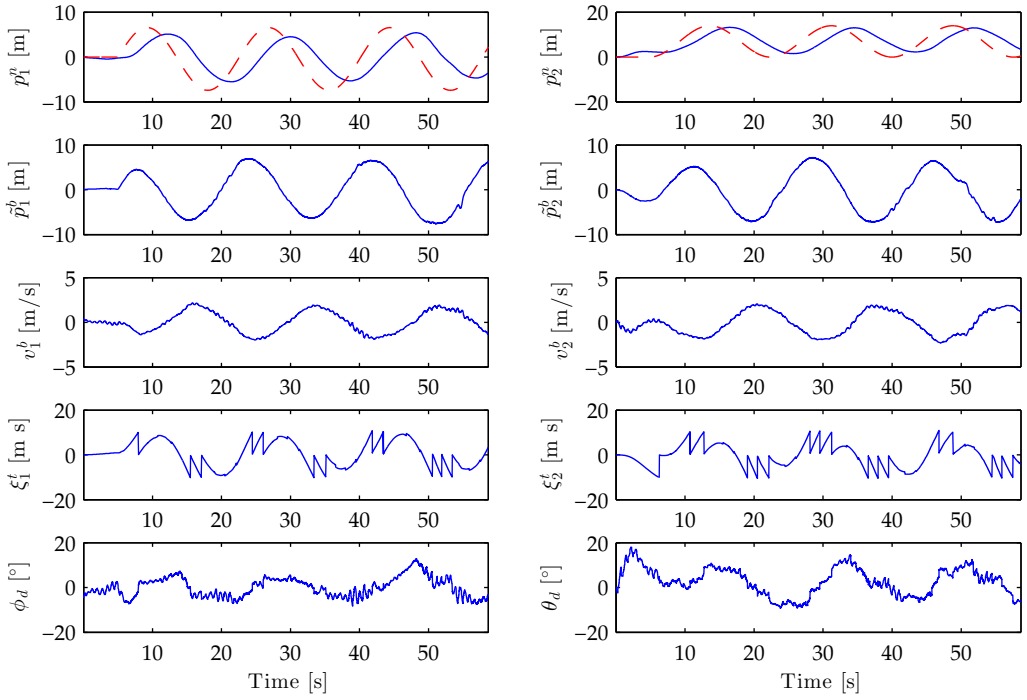


(c) Heading and height measurements.

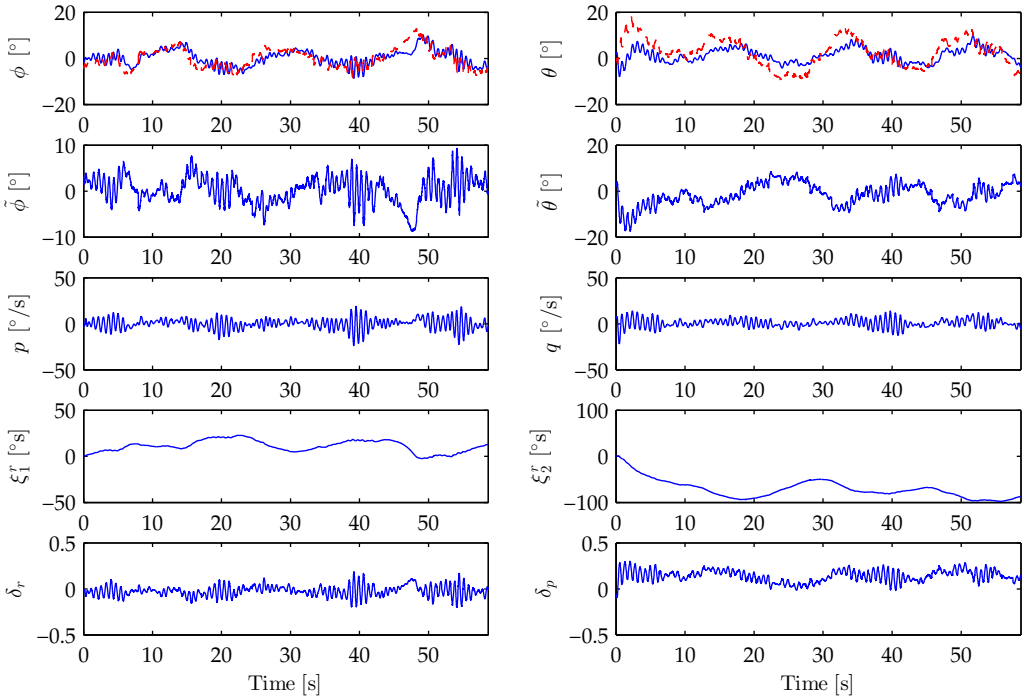


(d) Parametric plot of the horizontal position in $\mathcal{N}$. The reference is shown by the dashed line.

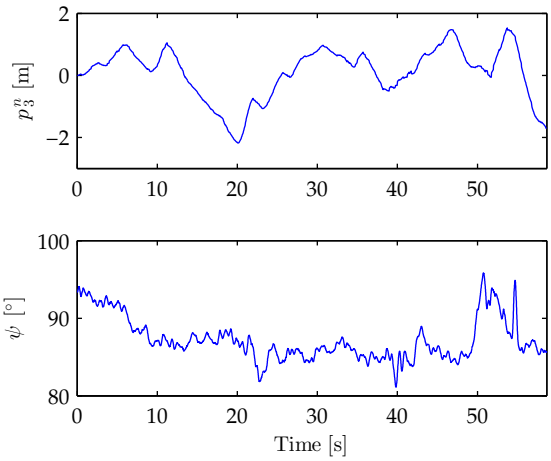
Figure 4.4: System response for line tracking experiment with a 15 m travel in the e_1' direction and a -15 m travel in the e_2' direction at a speed of 1 m/s.



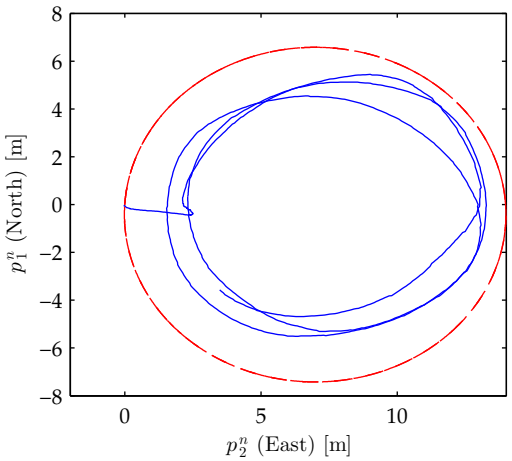
(a) Translation states, errors, and control effort. The reference is shown by the dashed line.



(b) Rotational states, errors and control effort. The reference is shown by the dashed line.



(c) Heading and height measurements.



(d) Parametric plot of the horizontal position in $\mathcal{N}$. The reference is shown by the dashed line.

Figure 4.5: System response for circle tracking experiment with a 7 m radius a tangential speed of 2.5 m/s.

Chapter 5

Nonlinear Control with Small Body Force Compensation

The generally accepted dynamic model of a helicopter with actuation (3.8) includes a coupling between the rotational inputs a, b, T_T and the translational dynamics. These coupling forces are the so-called Small Body Forces (SBF). We remark that the SBF are particular to the traditional helicopter and do not appear in other vehicles such as multirotors [13]. In fact, the reason SBF arise is due to the rotational inputs being linear forces acting on a lever-arm about the center of mass. The flapping angles parameterize the tilting of the main rotor disk and are shown in Figure 3.2. This relative orientation between the main rotor thrust and the vertical body-fixed axis creates a moment about the vehicle's center of mass. However, this lever-arm also creates a horizontal force and hence translational motion. A similar effect occurs due to the tail rotor thrust. The SBF coupling makes the control problem challenging. For example, in [11] SBF result in unstable tracking dynamics making input-output linearization not possible. It is therefore common practice to neglect SBF in the control, e.g., [13, 16, 17]. In [12] the SBF are neglected to derive a control, however, their effect is analyzed as a disturbance. Ultimate boundedness of the trajectory tracking error is concluded. A similar analysis and conclusion is obtained in [11].

The contribution of this work is to derive a control law which *compensates* the SBF associated with the tail rotor, and to provide an analysis of the stability in the presence of the remaining SBF. The robustness analysis leads to specific statement about the performance tradeoffs of the proposed design. Section 5.1 presents the model of the helicopter, and establishes the relevant notation. In Section 5.2 the proposed control is designed by treating the translational and rotational dynamics as two subsystems in cascade, and deriving a stabilizing feedback using an inner-outer control scheme. As described in [58] the design principle is to treat the rotational

subsystem as an actuator for the translational dynamics. Therefore, we do not consider the state of the rotational dynamics in the translational control design. For an example of a quadrotor control which uses an inner-outer approach see [27]. We note that the principal difference between the quadrotor and helicopter models is the absence of the SBF appearing in the quadrotor model (see [13] for a description of both models). Since our approach does not compensate the SBF associated with the main rotor flapping angles, we analyze their effect on the closed loop stability in Section 5.3 and show that the system remains uniformly ultimately bounded. Simulation results are presented in Section 5.4, and in Section 5.5 experimental results are given.

5.1 Modelling

In this chapter we express the translational dynamics in $\mathcal{N}$. From Section 3.1 we have

$$\dot{p}^n = v^n \tag{5.1a}$$

$$m\dot{v}^n = mge_3 + f \tag{5.1b}$$

$$\dot{R} = R\hat{\omega} \tag{5.1c}$$

$$J\dot{\omega} = -\omega \times J\omega + \tau \tag{5.1d}$$

where f and τ are given by (3.8).

Next, we make practically reasonable model assumptions which simplify the derivation of a control which compensates the SBF. In particular, we consider the tail rotor countertorque to be negligible $Q_T = 0$, the inertia matrix is diagonal $J = \text{diag}(J_x, J_y, J_z)$, and the rotor hub positions as $l_M = (0\ 0\ -z_M)^T$ and $l_T = (-x_T\ 0\ 0)^T$. Therefore (3.8b) simplifies to

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} = \begin{pmatrix} z_M T_M b \\ z_M T_M a \\ x_T T_T - Q_M \end{pmatrix} \tag{5.2a}$$

which is a bijective mapping with inverse

$$\begin{pmatrix} a \\ b \\ T_T \end{pmatrix} = \begin{pmatrix} \frac{\tau_2}{z_M T_M} \\ \frac{\tau_1}{z_M T_M} \\ \frac{\tau_3 + Q_M}{x_T} \end{pmatrix}. \tag{5.2b}$$

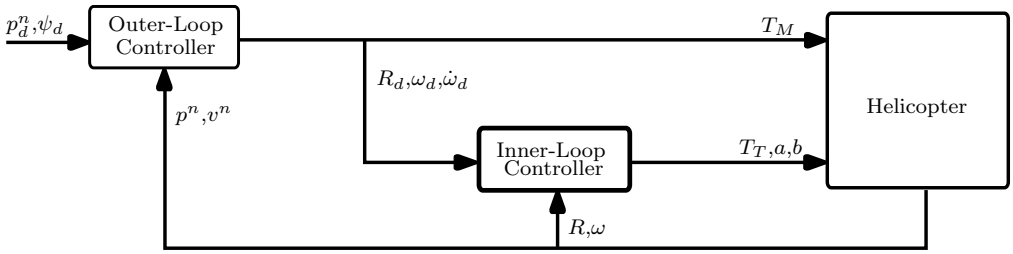


Figure 5.1: Block diagram showing the closed loop system.

Hence, we can use (5.2b) to obtain

$$f = R \begin{pmatrix} -\frac{\tau_2}{z_M} \\ \frac{\tau_1}{z_M} - \frac{\tau_3}{x_T} - \frac{Q_M}{x_T} \\ -T_M \end{pmatrix}. \tag{5.3}$$

From (5.3) we see the SBF is in part due to the applied torque and in part due to the countertorque. Finally, we assume the main rotor countertorque is linear in the main rotor thrust $Q_M = \sigma_M T_M$ for some $\sigma_M > 0$.

The assumptions made in this section are practically reasonable and consistent with current designs in the literature. The tail countertorque is not considered in [17, 62]; and in [13] similar assumptions are made regarding the rotor hub offsets. Furthermore, based on the geometry of the helicopter it is reasonable to expect that the body-fixed frame $\mathcal{B}$ is aligned with the principal axes, which justifies the diagonal inertia matrix.

5.2 Control Design

The goal of the controller is to track class C^4 reference trajectories denoted $p_d^n : \mathbb{R} \to \mathbb{R}^3$ for position, and $\psi_d : \mathbb{R} \to \mathbb{R}$ for heading. We perform the control design using an inner-outer hierarchical approach. As shown in Figure 5.1, the inner-loop controls the orientation, while the outer-loop controls the position. In order to perform the outer-loop design we make the assumption that the orientation reference generated by the outer-loop is tracked perfectly by the inner-loop.

5.2.1 Inner-Loop Control

Control of the rotational dynamics can be performed by applying an existing design from the literature. Our design is based on [63] which is further refined in [64]. This design is chosen since it does not suffer from singularities or other topological problems such as unwinding [15], and it provides almost global stability. To begin, we define error coordinates $\tilde{R} = R_d^T R$ where R_d is the desired orientation. The

desired angular velocity ω_d is defined to be compatible with the attitude kinematics $\dot{R}_d = R_d\hat{\omega}_d$. Therefore, the error kinematics are found to be

$$\begin{aligned}\dot{\tilde{R}} &= \dot{R}_d^T R + R_d^T \dot{R} \\ &= R_d^T R\hat{\omega} - \hat{\omega}_d R_d^T R \\ &= \tilde{R}(\omega - \tilde{R}^T\omega_d)^\wedge\end{aligned}$$

which implies $\tilde{\omega} = \omega - \tilde{R}^T\omega_d$. We remark that $\tilde{\omega}$ is a vector expressed in $\mathcal{B}$ while ω_d is expressed in the desired frame (whose orientation is R_d). Thus we have $\dot{\tilde{\omega}} = \dot{\omega} - \tilde{R}\dot{\omega}_d + \hat{\omega}\tilde{R}^T\omega_d$ and therefore the open-loop error dynamics are

$$\dot{\tilde{R}} = \tilde{R}\hat{\tilde{\omega}} \tag{5.4a}$$

$$J\dot{\tilde{\omega}} = -\omega \times J\tilde{\omega} + \tau - J\tilde{R}^T\dot{\omega}_d + J\hat{\omega}\tilde{R}^T\omega_d. \tag{5.4b}$$

We will show that these dynamics are stabilized by using torque feedback

$$\tau = -\operatorname{skew}(K_p^r\tilde{R})^\vee - K_d^r\tilde{\omega} - J\hat{\omega}\tilde{R}^T\omega_d + J\tilde{R}^T\dot{\omega}_d + (\tilde{R}^T\omega_d)^\wedge J\tilde{\omega} \tag{5.5}$$

where K_p^r and K_d^r are diagonal, positive definite gain matrices; and $\operatorname{skew}(A) = \frac{1}{2}(A - A^T)$ projects a matrix onto the skew-symmetric subspace. In order to show stability we require a potential function for the attitude error which we define in the following proposition.

Proposition 5.1 ([64])**.** *The function*

$$V_1 = \frac{1}{2}\operatorname{tr}(K_p^r(I - R)) \tag{5.6}$$

is an LFC since $V_1(R) = 0$ when $R = I$ and $V_1(R) > 0$ otherwise.

Proof. The property $V_1(R) = 0$ for $R = I$ is obvious. When $R \neq I$ all of the diagonal elements of $I - R$ are nonnegative and at least one is positive. □

Next we derive the relevant properties of V_1.

Proposition 5.2 ([64])**.** *The derivative of V_1 given by* (5.6) *is*

$$\dot{V}_1 = \operatorname{skew}(K_p^r R)^\vee \cdot \omega.$$

Proof. From (5.6) we compute

$$\begin{aligned}\dot{V}_1 &= -\frac{1}{2}\operatorname{tr}(K_p^r\dot{R}) \\ &= -\frac{1}{2}\operatorname{tr}\left(\left(\operatorname{skew}(K_p^r R) + \operatorname{sym}(K_p^r R)\right) R^T\dot{R}\right)\end{aligned}$$

where $\text{sym}(A) = \frac{1}{2}(A + A^T)$ and we have used the fact that any matrix can be decomposed as $A = \text{skew}(A) + \text{sym}(A)$ [65]. Since symmetric and skew-symmetric matrices are orthogonal under the trace norm, and $\hat{\omega} = R^T\dot{R}$ we have

$$\dot{V}_1 = -\frac{1}{2}\,\text{tr}\left(\text{skew}(K_p^r R)\hat{\omega}\right).$$

Finally, since for $a, b \in \mathbb{R}^3$, $a \cdot b = -\frac{1}{2}\,\text{tr}(\hat{a}\hat{b})$ we have the result. □

Note that $\cdot$ in Proposition 5.2 denotes the usual inner product on $\mathbb{R}^3$. We are now ready to present our Lyapunov stability proof.

Proposition 5.3. *The equilibrium* $(\tilde{R}, \tilde{\omega}) = (I, 0)$ *of the error dynamics* (5.4) *with the torque feedback* (5.5) *is asymptotically stable.*

Proof. To begin we compute the closed-loop dynamics to be

$$\begin{aligned}\dot{\tilde{R}} &= \tilde{R}\hat{\tilde{\omega}}\\ J\dot{\tilde{\omega}} &= -\tilde{\omega} \times J\tilde{\omega} - \text{skew}(K_p^r\tilde{R})^\vee - K_d^r\tilde{\omega}.\end{aligned}$$

We choose the LFC

$$V = \frac{1}{2}\,\text{tr}\left(K_p^r(I - \tilde{R})\right) + \frac{1}{2}\tilde{\omega} \cdot J\tilde{\omega}$$

whose derivative is

$$\dot{V} = \text{skew}(K_p^r\tilde{R})^\vee \cdot \tilde{\omega} + \tilde{\omega} \cdot (-\tilde{\omega} \times J\tilde{\omega} - \text{skew}(K_p^r\tilde{R})^\vee - K_d^r\tilde{\omega})$$

thus

$$\dot{V} = -\tilde{\omega} \cdot K_d^r\tilde{\omega}.$$

□

Following [66] we note that any smooth function on $SO(3)$ has at least 4 critical points. This result can be obtained by Lusternik-Schnirelmann Category theory, which states that any smooth function on a manifold M with $\text{cat}(M) = n$ will have at least $n + 1$ critical points [67]. Furthermore, from [68, Table 1] where we see that $\text{cat}(SO(3)) = 3$. Critical points are where the gradient field is zero and therefore a feedback using the gradient of V_1 will not provide global stability. In fact, no continuous feedback can provide global stability on $SO(3)$ [15]. In order to compute the critical points of (5.6) we use the following lemma. This lemma makes use of the quaternion formulation of R. For details on this parametrization see for example, [53, 69].

Lemma 5.4 (Critical points of V_1 [64])**.** *Let* V_1 *be given by* (5.6) *where* K_p^r *is positive definite and* $K_p^r = \text{diag}(k_{p_1}, k_{p_2}, k_{p_3})$, $k_{p_i} \neq k_{p_j}$ *for each* $i \neq j$. *Then, the critical*

points of V_1 are the identity and the rotations by 180° about the coordinate axes.

Proof. Let $q = (q_0, q_v)$ be the quaternion representation of R where $q_0 = \cos(\theta/2)$, $q_v = \sin(\theta/2)n$, and R is a rotation by θ about the fixed axis n. First, we will show

$$V_1(R) = q_v^T \bar{K} q_v \tag{5.7}$$

where $\bar{K} = \operatorname{diag}(k_{p_2} + k_{p_3},\, k_{p_1} + k_{p_3},\, k_{p_1} + k_{p_2})$. Using Rodrigues' formula, $R = I + 2q_0\hat{q}_v + 2\hat{q}_v^2$ gives

$$\begin{aligned} V_1(R) &= -\operatorname{tr}(K_p^r q_0 \hat{q}_v) - \operatorname{tr}(K_p^r \hat{q}_v^2) \\ &= -tr(K_p^r \hat{q}_v^2) \\ &= -tr\left(\left(\operatorname{skew}(K_p^r \hat{q}_v) + \operatorname{sym}(K_p^r \hat{q}_v)\right)\hat{q}_v\right) \\ &= -tr\left(\operatorname{skew}(K_p^r \hat{q}_v)\hat{q}_v\right) \\ &= 2\operatorname{skew}(K_p^r \hat{q}_v)^\vee \cdot q_v \end{aligned}$$

and by direct computation we have (5.7). Then, recalling the kinematic expression

$$\dot{q}_v = \frac{1}{2}(q_0\omega + q_v \times \omega)$$

we have

$$\begin{aligned} \dot{V}_1 &= \omega^T(q_0\bar{K}q_v - \hat{q}_v\bar{K}q_v) \\ &= (q_0\bar{K}q_v - \hat{q}_v\bar{K}q_v)^\wedge \cdot \hat{\omega}. \end{aligned}$$

The gradient of V_1 is therefore

$$dV_1 = (q_0\bar{K}q_v - \hat{q}_v\bar{K}q_v)^\wedge.$$

Clearly the identity $q = (\pm 1, 0, 0, 0)$ is a critical point. The other critical points are more interesting. Since the eigenvalues of K are distinct, the critical points are where $q_0 = 0$ *and* $q_v \times \bar{K}q_v = 0$. To see where this latter condition holds we use $q_v = (q_1,\, q_2,\, q_3)^T$ and expand the cross product

$$q_v \times \bar{K}q_v = \begin{pmatrix} -q_3(k_{p_1} + k_{p_3})q_2 + q_2(k_{p_1} + k_{p_2})q_3 \\ q_3(k_{p_2} + k_{p_3})q_1 - q_1(k_{p_1} + k_{p_2})q_3 \\ -q_2(k_{p_2} + k_{p_3})q_1 + q_1(k_{p_1} + k_{p_3})q_2 \end{pmatrix}$$

which is only zero when q_v is one of the basis vectors (note q_v is of unit length when $q_0 = 0$). □

Remark 5.5. In the context of a Riemannian manifold, the gradient of V_1 is not the

covector dV_1 [70]. A more precise statement is to say the gradient of V_1 is the vector field

$$\begin{aligned}\operatorname{grad} V_1 &= dV_1^\sharp \\ &= M^{-1} dV_1\end{aligned}$$

where M is the metric. However, this distinction does not affect the result since the critical points remain the same.

Remark 5.6. The requirement that the eigenvalues of K_p^r be distinct prevents the critical points of V_1 from being a connected submanifold of $SO(3)$. For example, if we assume $K_p^r = k_p I$ then

$$dV_1 = q_0 \bar{K} q_v$$

which is zero whenever $q_0 = 0$. This condition characterizes the rotations by 180° about any axis.

5.2.2 Outer-Loop Control

To design the outer-loop control we will treat the translational dynamics as an independent subsystem forced by the desired roll ϕ_d, pitch θ_d and main rotor thrust T_M. To begin we consider the dynamics (5.1a)-(5.1b) where f is taken as the input. These dynamics are essentially three double integrators, and therefore stabilization is trivial. However, to apply such a control requires its expression in terms ϕ_d, θ_d, T_M. Writing this expression leads to unstable dynamics for ϕ_d, θ_d. Hence, we must make a model approximation in order to define a control which does not depend on dynamics.

The first step in the design is to define the error states $\tilde{p}^n = p^n - p_d^n$ and $\tilde{v}^n = v^n - \dot{p}_d^n$ which results in the error dynamics

$$\dot{\tilde{p}}^n = \tilde{v}^n \tag{5.8a}$$

$$m\dot{\tilde{v}}^n = mge_3 - m\ddot{p}_d^n + f. \tag{5.8b}$$

We therefore choose the feedback

$$f = m\ddot{p}_d^n - mge_3 - K_d^t \tilde{v}^n - K_p^t \tilde{p}^n \tag{5.9}$$

where K_d^t and K_p^t are symmetric positive definite gain matrices. If we consider the LFC

$$V = \frac{m}{2}(\tilde{v}^n)^T \tilde{v}^n + \frac{1}{2}(\tilde{p}^n)^T K_p^t \tilde{p}^n$$

then the derivative of V along the trajectories of (5.8) is

$$\dot{V} = -(\tilde{v}^n)^T K_d^t \tilde{v}^n$$

and thus without further qualification the closed loop is globally asymptotically stable. In Section 5.2.2 we attempt to apply the feedback (5.9) where f is given as in (5.3). We will then approximate f in Section 5.2.2 to obtain a control which compensates the SBF due to the tail rotor while treating the SBF due to the flapping angles as a disturbance.

Unstable Reference Dynamics

Next, we derive the differential equations for ϕ_d and θ_d which must be satisfied to apply (5.9). We deduce the instability of these differential equations from their linearization. In order to isolate ϕ_d and θ_d we define a transformed input

$$\nu = R_{zd}^T f \tag{5.10}$$

where the subscript $_d$ denotes a desired quantity, and f is given in (3.8a). We must now relate the stabilizing force given by (5.9) to the outer-loop inputs: ϕ_d, θ_d, and T_M

$$\nu_1 = s\phi_d s\theta_d \left(\frac{\tau_1}{z_M} - \frac{\tau_3}{x_T} - \frac{Q_M}{x_T}\right) - c\theta_d \frac{\tau_2}{z_M} - T_M c\phi_d s\theta_d \tag{5.11a}$$

$$\nu_2 = c\phi_d \left(\frac{\tau_1}{z_M} - \frac{\tau_3}{x_T} - \frac{Q_M}{x_T}\right) + s\phi_d T_M \tag{5.11b}$$

$$\nu_3 = s\theta_d \frac{\tau_2}{z_M} + c\theta_d s\phi_d \left(\frac{\tau_1}{z_M} - \frac{\tau_3}{x_T} - \frac{Q_M}{x_T}\right) - c\phi_d s\phi_d T_M. \tag{5.11c}$$

Thus, in order to evaluate (5.11) we require an expression for the applied torque. This expression is found by evaluating (5.5) when the rotational dynamics are tracking the reference

$$\tau = J\dot{\omega}_d. \tag{5.12}$$

To express τ in terms of the reference roll and pitch, we begin with the rotational kinematics from (3.7)

$$\omega_d = \begin{pmatrix} 1 & 0 & -s\theta_d \\ 0 & c\phi_d & s\phi_d c\theta_d \\ 0 & -s\phi_d & c\phi_d c\theta_d \end{pmatrix} \begin{pmatrix} \dot{\phi}_d \\ \dot{\theta}_d \\ \dot{\psi}_d \end{pmatrix} \tag{5.13}$$

then we differentiate (5.13) to obtain

$$\dot{\omega}_d = \begin{pmatrix} 1 & 0 & -s\theta_d \\ 0 & c\phi_d & s\phi_d c\theta_d \\ 0 & -s\phi_d & c\phi_d c\theta_d \end{pmatrix} \begin{pmatrix} \ddot{\phi}_d \\ \ddot{\theta}_d \\ \ddot{\psi}_d \end{pmatrix} + \begin{pmatrix} 0 & 0 & -c\theta_d \\ -s\phi_d & c\phi_d c\theta_d & -s\phi_d s\theta_d \\ -c\phi_d & -s\phi_d c\theta_d & -c\phi_d s\theta_d \end{pmatrix} \begin{pmatrix} \dot{\phi}_d\dot{\theta}_d \\ \dot{\phi}_d\dot{\psi}_d \\ \dot{\theta}_d\dot{\psi}_d \end{pmatrix}. \tag{5.14}$$

Furthermore, we assume an approximation of $Q_M|_{w=0}$ which is linear in T_M and passes through the origin. Figure 3.14 shows that such an approximation is qualitatively reasonable in the practically relevant region of $T_M \approx mg$. Thus, Q_M is denoted

$$Q_M = \sigma_M T_M. \tag{5.15}$$

Using the above, the expressions in (5.11) can be solved to obtain

$$T_M = c\phi_d\nu_2 - s\theta_d c\phi_d\nu_1 - c\phi_d s\theta_d\nu_3 \tag{5.16a}$$

$$\ddot{\theta}_d = -t\phi_d c\theta_d\ddot{\psi}_d + t\phi_d\dot{\phi}_d\dot{\theta}_d - c\theta_d\dot{\phi}_d\dot{\psi}_d + t\phi_d s\theta_d\dot{\theta}_d\dot{\psi}_d - \frac{c\theta_d z_M\nu_1}{c\phi_d J_y} + \frac{s\phi_d z_M\nu_3}{c\phi_d J_y} \tag{5.16b}$$

$$\begin{aligned} \ddot{\phi}_d = &- \frac{z_M J_z t\phi_d s\phi_d\dot{\phi}_d\dot{\theta}_d}{J_x x_T} - \frac{c\phi_d z_M J_z\dot{\phi}_d\dot{\theta}_d}{J_x x_T} \\ &+ \left(\frac{z_M^2 J_z t\phi_d c\theta_d}{J_x x_T J_y} - \frac{z_M\sigma_M c\phi_d s\theta_d}{J_x x_T} + \frac{z_M s\phi_d s\theta_d}{J_x}\right)\nu_1 \\ &+ \left(\frac{z_M\sigma_M s\phi_d}{J_x x_T} - \frac{z_M s\phi_d t\phi_d}{J_x} + \frac{z_M}{J_x c\phi_d}\right)\nu_2 \\ &+ \left(\frac{z_M s\phi_d c\theta_d}{J_x} - \frac{c\phi_d c\theta_d z_M\sigma_M}{J_x x_T} - \frac{z_M^2 J_z t\phi_d s\theta_d}{J_x x_T J_y}\right)\nu_3 \\ &+ \left(c\theta_d\dot{\theta}_d - \frac{t\phi_d s\phi_d s\theta_d\dot{\theta}_d z_M J_z}{J_x x_T} - \frac{c\phi_d s\theta_d\dot{\theta}_d z_M J_z}{J_x x_T}\right)\dot{\psi}_d \\ &+ \left(\frac{z_M J_z t\phi_d s\phi_d c\theta_d}{J_x x_T} + \frac{s\theta_d + c\phi_d z_M c\theta_d J_z}{J_x x_T}\right)\ddot{\psi}_d \end{aligned} \tag{5.16c}$$

where $t\phi = \tan(\phi)$. The right hand side of these dynamics is completely defined, therefore, they could be integrated in a controller to define the reference for the rotational dynamics. However, we will show that the dynamics (5.16) are unsuitable for a control law since they have a linearization which is not asymptotically stable for at least a practically relevant special case.

Proposition 5.7. *Consider a constant position and heading reference. Suppose the dynamics* (5.8) *are initialized on the reference. In other words* $\dot{\psi}_d = \ddot{\psi}_d = \nu_1 = \nu_2 = 0$ *and* ν_3 *is constant. Let* $\zeta_1 = \phi_d$, $\zeta_2 = \theta_d$, $\zeta_3 = \dot{\phi}_d$, *and* $\zeta_4 = \dot{\theta}_d$. *Then, the reference dynamics* (5.16) *have a linearization which is not asymptotically stable.*

Proof. Under the assumptions of the proposition, the reference dynamics (5.16)

become

$$\dot{\zeta}_1 = \zeta_3 \tag{5.17a}$$

$$\dot{\zeta}_2 = \zeta_4 \tag{5.17b}$$

$$\dot{\zeta}_3 = z_M \nu_3 \left(\frac{s\zeta_1 c\zeta_2}{J_x} - \frac{c\zeta_1 c\zeta_2 \sigma_M}{J_x x_T} - \frac{z_M J_z t\zeta_1 s\zeta_2}{J_x J_y x_T} \right) - \frac{z_M J_z \zeta_3 \zeta_4}{J_x x_T} \left(t\zeta_1 s\zeta_1 + c\zeta_1 \right) \tag{5.17c}$$

$$\dot{\zeta}_4 = t\zeta_1 \zeta_3 \zeta_4 + \frac{s\zeta_2 z_M \nu_3}{c\zeta_1 J_y}. \tag{5.17d}$$

The equilibrium of (5.17) is $\zeta_1 = \arctan\left(\frac{\sigma_M}{x_T}\right)$, $\zeta_2 = \zeta_3 = \zeta_4 = 0$. The Jacobian of (5.17) evaluated at the equilibrium is

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{z_M \nu_3 \sqrt{\sigma_M^2 + x_T^2}}{J_x x_T} & -\frac{z_M^2 J_z \sigma_M \nu_3}{J_x x_T^2 J_y} & 0 & 0 \\ 0 & \frac{z_M \nu_3 \sqrt{\sigma_M^2 + x_T^2}}{J_y x_T} & 0 & 0 \end{pmatrix}$$

The eigenvalues of A are two repeated pairs

$$\lambda(A) = \pm \sqrt{\frac{z_M \sqrt{\sigma_M^2 + x_T^2}}{2 x_T J_x J_y} \left(\nu_3 (J_x + J_y) + \sqrt{\nu_3^2 (J_x - J_y)^2} \right)}$$

where z_M, x_T, J_x, J_y are positive by definition. □

Main Result: Tail SBF Compensation

Since we cannot apply a control law which is based on the critically stable dynamics (5.17), we partition (5.3) in order to derive an orientation reference. In particular, we treat any terms in f with dependence on derivatives of ϕ_d or θ_d as a disturbance in the design. Effectively, we break f into two terms $f = R_{zd}\nu + R_d \bar{\tau}$ where we have redefined ν and defined a disturbance $\bar{\tau}$. That is

$$\nu = R_{yd} R_{xd} \begin{pmatrix} 0 \\ -\frac{c\phi_d c\theta_d J_z \ddot{\psi}_d}{x_T} - \frac{\sigma_M T_M}{x_T} \\ -T_M \end{pmatrix}, \tag{5.18}$$

and

$$\bar{\tau} = \begin{pmatrix} -\frac{\tau_2}{z_M} \\ \frac{\tau_1}{z_M} + \frac{J_z}{x_T} \left(s\phi_d \ddot{\theta}_d + c\phi_d \dot{\phi}_d \dot{\theta}_d + s\phi_d c\theta_d \dot{\phi}_d \dot{\psi}_d + c\phi_d s\theta_d \dot{\theta}_d \dot{\psi}_d \right) \\ 0 \end{pmatrix}, \tag{5.19}$$

where we have used (5.12) to express τ as a function of ϕ_d, θ_d, ψ_d, and their derivatives. In other words, we have neglected the effect of the SBF due to the flapping angles as well as the coupling of the roll and pitch into the expression for the tail rotor thrust in the SBF. However, we have retained the principal effect of the SBF due to the tail rotor. Indeed, for many practical trajectories only constant steady-state roll-pitch speeds are required (e.g., hover, constant velocity forward flight, and circles where the heading is tangent to the position trajectory), however, the main rotor countertorque is always present, and $\ddot{\psi}_d$ is used to change the heading at low velocity and hover. To complete the control design, we expand (5.18)

$$\begin{aligned}
\nu_1 &= s\phi_d s\theta_d \left(-\frac{c\phi_d c\theta_d J_z \ddot{\psi}_d}{x_T} - \frac{\sigma_M T_M}{x_T} \right) - c\phi_d s\theta_d T_M \\
\nu_2 &= c\phi_d \left(-\frac{c\phi_d c\theta_d J_z \ddot{\psi}_d}{x_T} - \frac{\sigma_M T_M}{x_T} \right) + s\phi_d T_M \\
\nu_3 &= c\theta_d s\phi_d \left(-\frac{c\phi_d c\theta_d J_z \ddot{\psi}_d}{x_T} - \frac{\sigma_M T_M}{x_T} \right) - c\phi_d c\theta_d T_M
\end{aligned}$$

and solve to obtain

$$T_M = -\frac{(c\theta_d)^2 s\phi_d c\phi_d J_z \ddot{\psi}_d + \nu_3 x_T}{c\theta_d (s\phi_d \sigma_M + x_T c\phi_d)} \tag{5.20a}$$

$$\theta_d = \arctan\left(\frac{\nu_1}{\nu_3}\right) \tag{5.20b}$$

$$\phi_d = \arctan\left(\frac{J_z \ddot{\psi}_d \nu_3 - \sqrt{\nu_1^2 + \nu_3^2} \left(\sigma_M \sqrt{\nu_1^2 + \nu_3^2} + x_T \nu_2 \right)}{\sqrt{\nu_1^2 + \nu_3^2} \left(\sigma_M \nu_2 - x_T \sqrt{\nu_1^2 + \nu_3^2} \right)} \right) \tag{5.20c}$$

where we have assumed $\nu_3 < 0$. We note that for practically relevant trajectories $|\nu_1| \ll |\nu_3|$ and $|\nu_2| \ll |\nu_3|$ since the force of gravity will be applied mainly in the e_3' direction. We formally summarize the control design in the following proposition.

Proposition 5.8. *Let p_d^n and ψ_d be the position and heading references respectively, and the outer-loop subsystem be*

$$\dot{p}^n = v^n \tag{5.21a}$$

$$m\dot{v}^n = mge_3 + R_{zd}\nu \tag{5.21b}$$

where the inputs are (ϕ_d, θ_d, T_M) as given by (5.18)*. Then the control*

$$\nu = R_{zd}^T \left(m\ddot{p}_d^n - mge_3 - K_p^t \tilde{p}^n - K_d^t \tilde{v}^n \right), \tag{5.22}$$

which is applied using (5.20)*, will cause* (5.21) *to track the reference with globally*

asymptotically stable error dynamics on the domain where $\nu_3 < 0$.

Proof. Following the above derivation, we use the Lyapunov function

$$V = \frac{1}{2}(\tilde{p}^n)^T K_p^t \tilde{p}^n + \frac{m}{2}\|\tilde{v}^n\|^2.$$

The derivative of V along the trajectories of the closed loop system (5.21) is given by

$$\dot{V} = -(\tilde{v}^n)^T K_d^t \tilde{v}^n. \tag{5.23}$$

□

Remark 5.9. The restriction of the state space to the domain where $\nu_3 < 0$ is necessary in order to prevent T_M from crossing zero. If $T_M = 0$ the helicopter becomes uncontrollable which is a situation we do not consider in this work. In practice, for non-inverted flight this case is not relevant. Indeed, experimental data we have collected during an investigation of main rotor thrust modelling where the pilot attempted to maximize downward acceleration while retaining control showed $T_M > \frac{mg}{2}$ [36].

5.3 Robustness Analysis

As part of the control design we neglected some of the terms in the SBF. The purpose of this section is to determine if conditions exist where stability can be guaranteed in the presence of the disturbance $\bar{\tau}$. We will show that for a general reference the dynamics are uniformly ultimately bounded, and for a constant-velocity, constant-heading reference the result specializes to asymptotic stability.

The undisturbed closed loop error dynamics are a stable linear system $\dot{\xi} = A\xi$ where

$$A = \begin{pmatrix} 0 & I \\ -K_p^t & -K_d^t \end{pmatrix},$$

and

$$\xi = \begin{pmatrix} \tilde{p}^n \\ \tilde{v}^n \end{pmatrix}.$$

Therefore we know for any symmetric positive definite Q there exists a symmetric positive definite P such that

$$V = \xi^T P \xi \tag{5.24}$$

is a Lyapunov function with derivative $\dot{V} = -\xi^T Q \xi$. When the effect of $\bar{\tau}$ is included, we have

$$\dot{V} = -\xi^T Q \xi + 2\xi^T P \begin{pmatrix} 0 \\ R_d \bar{\tau} \end{pmatrix}. \tag{5.25}$$

In (5.25) we see that the first term is stabilizing while the second is indefinite. Therefore, we seek a comparison of the magnitude of these terms. Let $\lambda_{\min}(M)$ ($\lambda_{\max}(M)$), $M \in \mathbb{R}^{n\times n}$ be the minimum (maximum) eigenvalue of a matrix M, then

$$\dot{V} \leq -\lambda_{\min}(Q)\|\xi\|^2 + 2\lambda_{\max}(P)\|\xi\|\|\bar{\tau}\|. \tag{5.26}$$

In order to determine whether this inequality can guarantee stability we must compute an upper bound on $\bar{\tau}$ in terms of the magnitude of the error states.

The relationship between $\bar{\tau}$ and the error state ξ is defined by the composition

$$\bar{\tau}(\xi) = \bar{\tau} \circ \dot{\omega}_d \circ \nu \circ f(\xi)$$

where $\bar{\tau}(\dot{\omega}_d)$ is found using $\tau = J\dot{\omega}_d$, $\dot{\omega}_d(\nu)$ is a differential operator from (5.20b)-(5.20c), $\nu(f) = R_{zd}^T f$, and $f(\xi)$ is given by (5.9). Therefore, to derive a growth bound on $\bar{\tau}$ in terms of $\|\xi\|$ we proceed by deriving a bound on each of these functions. Since two derivatives of the control are necessary for $\dot{\omega}_d$, beginning with f we compute

$$\begin{aligned}
\dot{f} &= m(p_d^n)^{(3)} + \left(\frac{1}{m}\left(K_d^t\right)^2 - K_p^t\right)\tilde{v} \\
\ddot{f} &= m(p_d^n)^{(4)} + \frac{1}{m}\left(2K_d^tK_p^t - \frac{1}{m}(K_d^t)^3\right)\tilde{v} + \frac{1}{m}\left(\left(K_p^t\right)^2 - \frac{1}{m}\left(K_d^t\right)^2 K_p^t\right)\tilde{p}
\end{aligned}$$

where we have assumed K_p^t and K_d^t are diagonal. Then, we have the bounds

$$\|f\| \leq m\|\ddot{p}_d^n\| + mg + (\bar{k}_d + \bar{k}_p)\|\xi\| \tag{5.27a}$$
$$\|\dot{f}\| \leq m\|(p_d^n)^{(3)}\| + (\bar{k}_d^2 + \bar{k}_d\bar{k}_p + \bar{k}_p)\|\xi\| \tag{5.27b}$$
$$\|\ddot{f}\| \leq m\|(p_d^n)^{(4)}\| + (\bar{k}_d^3 + 2\bar{k}_d\bar{k}_p + \bar{k}_d^2\bar{k}_p + \bar{k}_p^2)\|\xi\| \tag{5.27c}$$

where $\bar{k}_d = \lambda_{\max}(K_d^t)$ and $\bar{k}_p = \lambda_{\max}(K_p^t)$. Next, we compute two derivatives of $\nu(f)$

$$\begin{aligned}
\nu &= R_{dz}^T f \\
\dot{\nu} &= R_{zd}^T\dot{f} - \hat{\omega}_{zd}R_{zd}^T f \\
\ddot{\nu} &= R_{zd}^T\ddot{f} - 2\hat{\omega}_{zd}R_{zd}^T\dot{f} + (\hat{\omega}_{zd}^2 - \dot{\hat{\omega}}_{zd})R_{zd}^T f
\end{aligned}$$

and thus

$$\|\nu\| = \|f\| \tag{5.28a}$$
$$\|\dot{\nu}\| \leq \|\dot{f}\| + |\dot{\psi}_d|\|f\| \tag{5.28b}$$
$$\|\ddot{\nu}\| \leq \|\ddot{f}\| + 2|\dot{\psi}_d|\|\dot{f}\| + (|\dot{\psi}_d|^2 + |\ddot{\psi}_d|)\|f\|. \tag{5.28c}$$

A bound on $\dot{\omega}_d(\nu)$ is found using (5.20b) and (5.20c). However, these bounds are complex to compute so we leave them until last. It easy to see $\|\bar{\tau}\| \leq \|\tau\|$, and from (5.12) we know $\|\tau\| \leq \|J\|\|\dot{\omega}_d\|$. From (5.14) we have

$$\|\dot{\omega}_d\| \leq \sqrt{3}\left(\|\ddot{\eta}_d\| + \|\dot{\eta}_d\|^2\right) \tag{5.29}$$

where $\eta_d = (\phi_d,\, \theta_d,\, \psi_d)^T$. Thus

$$\|\bar{\tau}\| \leq \sqrt{3}\|J\|\left(|\ddot{\phi}_d| + |\ddot{\theta}_d| + |\ddot{\psi}_d| + |\dot{\phi}_d|^2 + |\dot{\theta}_d|^2 + |\dot{\psi}_d|^2\right). \tag{5.30}$$

Since ψ_d is defined as part of the problem statement (see Section 5.2) we need to only compute expressions for the derivatives of ϕ_d and θ_d. The derivative of the reference roll is found from the expression given in (5.20c) and is

$$\dot{\phi}_d = \frac{c_1 + c_2 + c_3}{d} \tag{5.31a}$$

$$c_1 = -\left(\sigma_M^2 + x_T^2\right)\left(\nu_1^2 + \nu_3^2\right)\left(-\dot{\nu}_2\nu_3^2 + \nu_2\dot{\nu}_3\nu_3 + \nu_2\nu_1\dot{\nu}_1 - \nu_1^2\dot{\nu}_2\right) \tag{5.31b}$$

$$\begin{aligned} c_2 = \ddot{\psi}_d J_z\Bigg(& x_T\sqrt{\nu_1^2 + \nu_3^2}\left(\dot{\nu}_3\nu_3^2 - \nu_1^2\dot{\nu}_3 + 2\nu_3\nu_1\dot{\nu}_1\right) \\ & - \sigma_M\left(\nu_3\dot{\nu}_2\nu_1^2 - \dot{\nu}_3\nu_2\nu_1^2 + \nu_3\nu_2\nu_1\dot{\nu}_1 + \nu_3^3\dot{\nu}_2\right)\Bigg) \end{aligned} \tag{5.31c}$$

$$c_3 = -\left(\nu_1^2 + \nu_3^2\right)J_z\nu_3\psi_d^{(3)}\left(x_T\sqrt{\nu_1^2 + \nu_3^2} - \sigma_M\nu_2\right) \tag{5.31d}$$

$$\begin{aligned} d = \sqrt{\nu_1^2 + \nu_3^2}\Bigg(& \sqrt{\nu_1^2 + \nu_3^2}\left(\|u\|^2(\sigma_M^2 + x_T^2) - 2\nu_3 J_z\ddot{\psi}_d\left(x_T\nu_2 + \sigma_M\sqrt{\nu_1^2 + \nu_3^2}\right)\right) \\ & + \nu_3^2 J_z^2\ddot{\psi}_d^2\Bigg) \end{aligned} \tag{5.31e}$$

where we have split up the terms in order to improve the presentation of the second derivative. The second derivative $\ddot{\phi}_d$ can be computed from the above expressions. However, due to complexity we will not present the full expression, instead we will use the derivatives of (5.31b)-(5.31e) directly to compute a bound on $\ddot{\phi}_d$. The derivatives of c_1 and c_2 are

$$\begin{aligned} \dot{c}_1 = & \left(\sigma_M^2 + x_T^2\right)\Big(\ddot{\nu}_2\nu_3^4 + \left(3\dot{\nu}_2\dot{\nu}_3 - \ddot{\nu}_3\nu_2\right)\nu_3^3 + \big(3\nu_1\dot{\nu}_2\dot{\nu}_1 - 3\dot{\nu}_3^2\nu_2 + 2\nu_1^2\ddot{\nu}_2 - \dot{\nu}_1^2\ddot{\nu}_2 \\ & - \nu_1\ddot{\nu}_1\nu_2\big)\nu_3^2 + \left(-\nu_1^2\ddot{\nu}_3\nu_2 - 4\nu_1\dot{\nu}_1\dot{\nu}_3\nu_2 + 3\nu_1^2\dot{\nu}_2\dot{\nu}_3\nu_2 + 3\nu_1^2\dot{\nu}_2\dot{\nu}_3\right)\nu_3 - \nu_1^3\ddot{\nu}_1\nu_2 \\ & + 3\nu_1^3\dot{\nu}_2\dot{\nu}_1 - 3\nu_1^2\dot{\nu}_1^2\nu_2 + \ddot{\nu}_2\nu_1^4 - \nu_1^2\dot{\nu}_3^2\nu_2\Big) \end{aligned}$$

$$\begin{aligned} \dot{c}_2 = \frac{J_z x_T}{\sqrt{\nu_1^2 + \nu_3^2}}\Bigg(& \Big(\ddot{\nu}_3\nu_3^4 + \left(2\nu_1\ddot{\nu}_1 + 2\dot{\nu}_1^2 + 3\dot{\nu}_3^2\right)\nu_3^3 + 3\nu_1\dot{\nu}_1\dot{\nu}_3\nu_3^2 + \left(\dot{\nu}_3^2\nu_1^2 + 2\nu_1^3\ddot{\nu}_1\right. \\ & \left. + 4\nu_1^2\dot{\nu}_1^2\right)\nu_3 - \ddot{\nu}_3\nu_1^4 - \dot{\nu}_3\nu_1^3\dot{\nu}_1\Big)\ddot{\psi}_d + \left(\nu_3^4\dot{\nu}_3 + 2\nu_3^3\nu_1\dot{\nu}_1 + 2\nu_3\nu_1^3\dot{\nu}_1 - \dot{\nu}_3\nu_1^4\right)\psi_d^{(3)}\Bigg) \end{aligned}$$

$$+ J_z\sigma_M\Bigg(\Big(- \nu_3^3\ddot{\nu}_2 - 3\nu_3^2\dot{\nu}_2\dot{\nu}_3 - \big(\nu_1^2\ddot{\nu}_2 + 3\nu_1\dot{\nu}_2\dot{\nu}_1 + \dot{\nu}_1^2\nu_2 + \nu_1\ddot{\nu}_1\nu_2\big)\nu_3$$
$$+ \nu_1^2\ddot{\nu}_3\nu_2 + \nu_1\dot{\nu}_1\dot{\nu}_3\nu_2\Big)\ddot{\psi}_d + \Big(- \nu_3^3\dot{\nu}_2 - \big(\nu_1^2\dot{\nu}_2 + \nu_1\dot{\nu}_1\nu_2\big)\,\nu_3 + \dot{\nu}_3\nu_2\nu_1^2\Big)\psi_d^{(3)}\Bigg)$$

and the derivatives of c_3 and d are

$$\dot{c}_3 = J_z\Bigg(\Bigg(- \frac{\nu_3^4 x_T\dot{\nu}_3}{\sqrt{\nu_1^2+\nu_3^2}} + \left(\sigma_M\dot{\nu}_2 - \frac{x_T\nu_1\dot{\nu}_1}{\sqrt{\nu_1^2+\nu_3^2}}\right)\nu_3^3 + \left(3\sigma_M\nu_2\dot{\nu}_3 - \frac{x_T\nu_1^2\dot{\nu}_3}{\sqrt{\nu_1^2+\nu_3^2}}\right.$$
$$\left. - 3x_T\dot{\nu}_3\sqrt{\nu_1^2+\nu_3^2}\right)\nu_3^2 + \left(\sigma_M\dot{\nu}_2\nu_1^2 - \frac{x_T\nu_1^3\dot{\nu}_1}{\sqrt{\nu_1^2+\nu_3^2}} - 2\nu_1\dot{\nu}_1 x_T\sqrt{\nu_1^2+\nu_3^2}\right.$$
$$\left. + 2\sigma_M\nu_1\dot{\nu}_1\nu_2\right)\nu_3 - x_T\dot{\nu}_3\nu_1^2\sqrt{\nu_1^2+\nu_3^2} + \sigma_M\dot{\nu}_3\nu_2\nu_1^2\Bigg)\psi_3^{(3)} + \Bigg(\Big(\sigma_M\nu_2$$
$$- x_T\sqrt{\nu_1^2+\nu_3^2}\Big)\nu_3^3 + \Big(\sigma_M\nu_2\nu_1^2 - x_T\nu_1^2\sqrt{\nu_1^2+\nu_3^2}\Big)\nu_3\Bigg)\psi_d^{(4)}\Bigg)$$

$$\dot{d} = J_z^2\left(\left(\frac{\dot{\nu}_3\nu_3^3}{\sqrt{\nu_1^2+\nu_3^2}} + \frac{\nu_1\dot{\nu}_1\nu_3^2}{\sqrt{\nu_1^2+\nu_3^2}} + 2\nu_3\dot{\nu}_3\sqrt{\nu_1^2+\nu_3^2}\right)\ddot{\psi}_d^2 + 2\nu_3^2\psi_d^{(3)}\ddot{\psi}_d\sqrt{\nu_1^2+\nu_3^2}\right)$$
$$+ J_z\Bigg(\Bigg(\Bigg(- \frac{2\dot{\nu}_3\nu_3^4}{\sqrt{\nu_1^2+\nu_3^2}} - \frac{2\nu_1\dot{\nu}_1\nu_3^3}{\sqrt{\nu_1^2+\nu_3^2}} - \left(6\dot{\nu}_3\sqrt{\nu_1^2+\nu_3^2} + \frac{2\nu_1^2\dot{\nu}_3}{\sqrt{\nu_1^2+\nu_3^2}}\right)\nu_3^2$$
$$- \left(4\nu_1\dot{\nu}_1\sqrt{\nu_1^2+\nu_3^2} + \frac{2\nu_1^3\dot{\nu}_1}{\sqrt{\nu_1^2+\nu_3^2}}\right)\nu_3 - 2\dot{\nu}_3\nu_1^2\sqrt{\nu_1^2+\nu_3^2}\Bigg)\ddot{\psi}_d$$
$$- 2\nu_3\big(\nu_3^2+\nu_1^2\big)^{\frac{3}{2}}\psi_d^{(3)}\Bigg)\sigma_M - 2\Big(\Big(2\nu_2\dot{\nu}_3\nu_3^2 + \big(\nu_1\dot{\nu}_1\nu_2 + \nu_2\nu_1\dot{\nu}_1$$
$$+ \dot{\nu}_2\left(\nu_1^2+\nu_3^2\right)\big)\nu_3 + \nu_2\dot{\nu}_3\left(\nu_1^2+\nu_3^2\right)\Big)\ddot{\psi}_d + \left(\nu_2\nu_3^3 + \nu_1^2\nu_2\nu_3\right)\psi_d^{(3)}\Big)x_T\Bigg)$$
$$+ \sqrt{\nu_1^2+\nu_3^2}\Big(5\dot{\nu}_3\nu_3^3 + (5\nu_1\dot{\nu}_1 + 2\nu_2\dot{\nu}_2)\nu_3^2 + \Big(\big(\nu_1^2+\nu_2^2\big)\dot{\nu}_3 + 2\big(\left(\nu_1^2+\nu_2^2\right)\dot{\nu}_3$$
$$+ \dot{\nu}_3\nu_1^2\big)\Big)\nu_3 + 3\left(\nu_1^2+\nu_2^2\right)\nu_1\dot{\nu}_1 + 2\nu_1^2(\nu_1\dot{\nu}_1 + \nu_2\dot{\nu}_1)\Big)\left(\sigma_M^2 + x_T^2\right)$$

Bounds on $|\dot{\phi}_d|$ and $|\ddot{\phi}_d|$ can be readily computed using these expressions. However, due to their complexity, explicit expressions will not be presented here since they provide little insight. What is important to note is the independence of $|\dot{\phi}_d|$ and $|\ddot{\phi}_d|$ on the upper bound of $|\nu_3|$. Indeed, this component of the control has a constant offset due to the force of gravity. If this constant were included in the bound it would preclude asymptotic stability regardless of any conditions on the reference. We clarify this idea using a special case for which we will also present a complete analysis. Consider a constant-heading and constant-velocity reference. Since this type of reference results in a significant simplification of (5.20c) and its derivatives,

we compute the bounds for this case as well. When $\ddot{\psi}_d = 0$ (5.20c) becomes

$$\phi_d = -\arctan\left(\frac{\sigma_M\sqrt{\nu_1^2+\nu_3^2}+x_T\nu_2}{\sigma_M\nu_2 - x_T\sqrt{\nu_1^2+\nu_3^2}}\right).$$

The first derivative is

$$\dot{\phi}_d = \frac{\dot{\nu}_2\nu_1^2 + \dot{\nu}_2\nu_3^2 - \nu_1\dot{\nu}_1\nu_2 - \nu_3\dot{\nu}_3\nu_2}{\|u\|^2\sqrt{\nu_1^2+\nu_3^2}}$$

and

$$|\dot{\phi}_d| \leq \frac{3\|\dot{\nu}\|}{|\nu_3|}. \tag{5.32a}$$

The second derivative is

$$\begin{aligned}\ddot{\phi}_d = \frac{1}{\|u\|^2\sqrt{\nu_1^2+\nu_3^2}}\Bigg(&- \dot{\nu}_1^2\nu_2 - \nu_1\ddot{\nu}_1\nu_2 + \nu_1\dot{\nu}_1\dot{\nu}_2 - \dot{\nu}_3^2\nu_2 - \nu_3\ddot{\nu}_3\nu_2 + \nu_3\dot{\nu}_3\dot{\nu}_2 + \ddot{\nu}_2\nu_1^2 \\ &+ \ddot{\nu}_2\nu_3^2 - \frac{2\left(\nu_1\dot{\nu}_1\nu_2 + \nu_3\dot{\nu}_3\nu_2 - \dot{\nu}_2\nu_1^2 - \dot{\nu}_2\nu_3^2\right)\left(\nu_2\dot{\nu}_2 + \nu_1\dot{\nu}_1 + \nu_3\dot{\nu}_3\right)}{\|u\|^2} \\ &- \frac{\left(\nu_1\dot{\nu}_1\nu_2 + \nu_3\dot{\nu}_3\nu_2 - \dot{\nu}_2\nu_1^2 - \dot{\nu}_2\nu_3^2\right)\left(\nu_1\dot{\nu}_1 + \nu_3\dot{\nu}_3\right)}{\nu_1^2+\nu_3^2}\Bigg)\end{aligned}$$

and

$$|\ddot{\phi}_d| \leq \frac{6\|\dot{\nu}\|^2}{|\nu_3|^3} + \frac{\|\ddot{\nu}\|}{|\nu_3|} + \frac{18\|\dot{\nu}\|}{|\nu_3|^3} + \frac{3\|\dot{\nu}\||\nu_1|}{|\nu_3|^3} \tag{5.32b}$$

Remark that (5.32) is independent on the upper bound of $|\nu_3|$. Instead, we are only concerned with a lower bound on $|\nu_3|$. Therefore, there is no constant term due to the force of gravity. The derivatives of θ_d are found by differentiating (5.20b) to obtain

$$\begin{aligned}\dot{\theta}_d &= \frac{\dot{\nu}_1\nu_3 - \dot{\nu}_3\nu_1}{\nu_3^2+\nu_1^2} \\ \ddot{\theta}_d &= \frac{(\ddot{\nu}_1\nu_3 - \ddot{\nu}_3\nu_1)(\nu_3^2+\nu_1^2) - 2(\nu_3\dot{\nu}_3 + \dot{\nu}_1\nu_1)(\dot{\nu}_1\nu_3 - \dot{\nu}_3\nu_1)}{(\nu_3^2+\nu_1^2)^2}\end{aligned}$$

which results in the bounds

$$|\dot{\theta}_d| \leq \frac{\|\dot{\nu}\|}{|\nu_3|} + \frac{\|\dot{\nu}\||\nu_1|}{\nu_3^2} \tag{5.33a}$$

$$|\ddot{\theta}_d| \leq \frac{\|\ddot{\nu}\|}{|\nu_3|} + \frac{\|\ddot{\nu}\||\nu_1|}{\nu_3^2} + \frac{2\|\dot{\nu}\|^2}{\nu_3^2} + \frac{4\|\dot{\nu}\|^2|\nu_1|}{|\nu_3|^3} + \frac{2\|\dot{\nu}\|^2|\nu_1|^2}{\nu_3^4} \tag{5.33b}$$

The main result of our analysis is to prove uniform ultimate boundedness of the solutions to the error dynamics in the presence of $\bar{\tau}$. The definition of uniform ultimate boundedness is given in Definition 5.10. We further show that this re-

sult specializes to asymptotic stability when the reference is constant-velocity and constant-heading (e.g., hover).

Definition 5.10 ([71])**.** The dynamics

$$\dot{x} = f(t, x)$$

where $f\colon [0,\infty) \times D \to \mathbb{R}^n$ is continuous in t and differentiable in x on D, and $D \subset \mathbb{R}^n$ contains the origin, are *uniformly ultimately bounded* with ultimate bound b if there exists $b, c > 0$, which are independent of $t_0 \geq 0$, such that for every $a \in (0, c)$, there exists $T \geq 0$ where

$$\|x(t_0)\| \leq a \implies \|x(t)\| \leq b, \quad \forall\, t \geq t_0 + T.$$

Theorem 5.11. *Let ψ_d be a heading reference and p_d^n be a position reference. If $|\psi_d^{(i)}| \leq l_1$ for $i \geq 1$, $\|(p_d^n)^{(i)}\| \leq l_2$ for $i \geq 2$, and $l = \max\{l_1, l_2\}$, then the dynamics (5.8) with $f = R_{zd}\nu$ and the inputs ϕ_d, θ_d and T_M given by (5.20), are uniformly ultimately bounded.*

Proof. The bound on $\bar{\tau}$ is found to be of the form

$$\|\bar{\tau}\| \leq \sum_{n=0}^{10} c_n \|\xi\|^n.$$

Following the method of proof found in [71, Thm. 4.7] we observe

$$\lim_{\|\xi\| \to 0} \frac{\sum_{n=2}^{10} c_n \|\xi\|^n}{\|\xi\|} = 0$$

which implies that for any $\rho > 0$ there exists $r > 0$ such that

$$\sum_{n=2}^{10} c_n \|\xi\|^n \leq \rho \|\xi\|.$$

Using the Lyapunov Function (5.24) whose derivative is bounded as in (5.26) it follows that

$$\dot{V} \leq -(1 + 2\lambda_{\max}(P)(c_1 + \rho))\|\xi\|^2 + 2\lambda_{\max}(P) c_0 \|\xi\|, \quad \text{when } \|\xi\| < r$$

where we have chosen $Q = I$ in order to maximize the ratio $\frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)}$ [72]. Thus $\dot{V} < 0$ for

$$\rho < \frac{1}{2\lambda_{\max}(P)} - c_1$$

and we can conclude uniform ultimate boundedness as long as

$$\frac{2\lambda_{\max}(P)c_0}{1-2\lambda_{\max}(P)(c_1+\rho)} \leq r$$

which can always be satisfied by choosing l sufficiently small since $c_0 = \sum_{n=1}^{12} c_n' l^n$. □

Next, we present the case of a constant-heading, constant-velocity reference trajectory which is practically relevant.

Corollary 5.12 (Specialization to constant velocity, constant heading)**.** *Let p_d^n be a reference trajectory with constant velocity, and ψ_d be a constant heading reference. Then the dynamics* (5.8) *with $f = R_{zd}\nu$ and the inputs ϕ_d, θ_d and T_M given by* (5.20) *are asymptotically stable.*

Proof. Beginning with (5.30), we derive a growth bound on $\|\bar{\tau}\|$ in terms of $\|\xi\|$ by simplifying (5.28) and (5.27) using the given assumptions regarding the reference trajectory. Indeed, we obtain the bounds

$$\begin{aligned}
|\nu_1| &\leq (\bar{k}_d + \bar{k}_p)\|\xi\| \\
\|\dot{\nu}\| &\leq (\bar{k}_d^2 + \bar{k}_d\bar{k}_p + \bar{k}_p)\|\xi\| \\
\|\ddot{\nu}\| &\leq (\bar{k}_d^3 + 2\bar{k}_d\bar{k}_p + \bar{k}_d^2\bar{k}_p + \bar{k}_p^2)\|\xi\|
\end{aligned}$$

which we combine with (5.32) and (5.33) to obtain

$$\|\bar{\tau}\| \leq \sum_{n=1}^{4} c_n \|\xi\|^n \tag{5.34}$$

where

$$\begin{aligned}
c_1 &= 2\sqrt{3}\|J\|\left(\frac{\bar{k}_d^3 + 2\bar{k}_d\bar{k}_p + \bar{k}_d^2\bar{k}_p + \bar{k}_p^2}{|\nu_3|} + \frac{9\bar{k}_p + 9\bar{k}_d^2 + 9\bar{k}_d\bar{k}_p}{|\nu_3^3|}\right) \\
c_2 &= \sqrt{3}\|J\|\Bigg(\frac{19\bar{k}_d^2\bar{k}_p^2 + 22\bar{k}_d^2\bar{k}_p + 2\bar{k}_d^4\bar{k}_p + 11\bar{k}_d^4 + 8\bar{k}_d\bar{k}_p^3 + 22\bar{k}_d^3\bar{k}_p + 23\bar{k}_d\bar{k}_p^2 + 10k_p^2}{\nu_3^2} \\
&\quad + \frac{2k_d^6 + 10\bar{k}_p^2 + 4\bar{k}_d^2\bar{k}_p^3 + 4\bar{k}_d^5\bar{k}_p + 12\bar{k}_d^3\bar{k}_p^2 + 2\bar{k}_p^4 + 2\bar{k}_d^4\bar{k}_p^2 + \bar{k}_p^3}{\nu_3^2} + \frac{6\bar{k}_d^2\bar{k}_p^2 + 6\bar{k}_d^4}{|\nu_3^3|} \\
&\quad + \frac{15\bar{k}_d\bar{k}_p^2 + 3\bar{k}_d^3 + 9\bar{k}_p^2 + 12\bar{k}_d^3\bar{k}_p + 3\bar{k}_d\bar{k}_p + 18\bar{k}_d^2\bar{k}_p}{|\nu_3^3|}\Bigg) \\
c_3 &= \frac{6\sqrt{3}\|J\|\left(\bar{k}_p + \bar{k}_d\right)\left(\bar{k}_d^2 + \bar{k}_d\bar{k}_p + \bar{k}_p\right)^2}{|\nu_3^3|} \\
c_4 &= \frac{3\sqrt{3}\|J\|\left(\bar{k}_p + \bar{k}_d\right)^2\left(\bar{k}_d^2 + \bar{k}_d\bar{k}_p + \bar{k}_p\right)^2}{\nu_3^4}.
\end{aligned}$$

In this case we have

$$\lim_{\|\xi\|\to 0} \frac{\sum_{n=2}^{4} c_n \|\xi\|^n}{\|\xi\|} = 0.$$

Thus

$$\dot{V} < -\left(1 - 2\lambda_{\max}(P)(c_1 + \rho)\right) \|\xi\|^2, \quad \forall \|\xi\| < r$$

where we have again used $Q = I$. Finally, we can conclude asymptotic stability by choosing

$$\rho < \frac{1}{2\lambda_{\max}(P)} - c_1. \tag{5.35}$$

□

Remark 5.13. As an alternative to the latter part of the proof, if we restrict our interest to $\|\xi\| < 1$ we can express the growth bound (5.34) as

$$\|\bar{\tau}\| \leq \bar{c}\|\xi\| \sum_{n=1}^{3} \|\xi\|^n$$

where $\bar{c} = \max\{c_n\}$. Thus,

$$\dot{V} \leq -\left(1 - \frac{2\lambda_{\max}(P)\bar{c}}{1-r}\right) \|\xi\|^2 - \frac{2\lambda_{\max}(P)\bar{c}\|\xi\|^5}{1-r}, \quad \text{when } \|\xi\| < r$$

which implies asymptotic stability for

$$\frac{1}{2\lambda_{\max}(P)} \geq \frac{\bar{c}}{1-r}. \tag{5.36}$$

Remark 5.14. For a particular choice of controller gains, the bounds (5.35) and (5.36) provide a design trade-off between r and a lower bound on ν_3. To clarify the relationship consider a particular numerical example. If we choose $K_d^t = \bar{k}_d I$ and $K_p^t = \bar{k}_p I$ where $\bar{k}_d = 0.3$ and $\bar{k}_p = 0.2$ then using the value of $\|J\|_2 = 1.48$, $\bar{c} = c_1$, and

$$c_1 \approx \frac{1.05}{|\nu_3|} + \frac{16.2}{|\nu_3|^3}.$$

In addition, when $Q = I$ then $\frac{1}{2\lambda_{\max}(P)} \approx 0.0464$. Thus (5.36) becomes

$$r < 1 - \frac{22.6}{|\nu_3|} - \frac{350}{|\nu_3|^3}.$$

Therefore, by increasing the lower bound on $|\nu_3|$ we can increase the region for which stability is ensured. For example, we can achieve stability for $r = \frac{1}{4}$ if we assume $|\nu_3| > 30 \approx \frac{mg}{5}$. From Figure 3.5 we see $\frac{mg}{2}$ is approximately the smallest measured force during a test where the pilot was attempting to let the helicopter descend as quickly as he could while maintaining control. Figure 5.2 shows this relationship

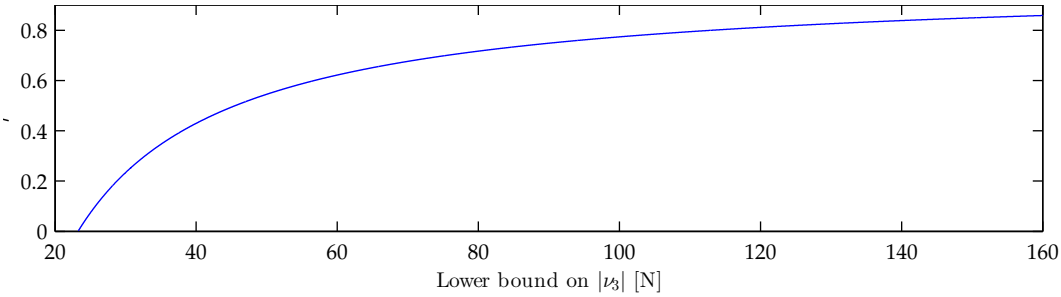


Figure 5.2: Relationship between the region of attraction r and the lower bound on ν_3 given by (5.36) for $\bar{k}_d = 0.3$ and $\bar{k}_p = 0.2$.

K_p^r	diag(20, 16, 18) s^{-2}
K_d^r	diag(7, 7, 6) s^{-1}
K_p^t	$3I$ kg/s
K_d^t	$5.5I$ kg/s^2

Table 5.1: SBF controller gains used for simulation.

over a range of lower bounds on $|\nu_3|$.

5.4 Simulation Results

The system is simulated using the dynamics (5.1) including the SBF (3.8a), where Q_M is given by (5.15) in the torque expression (3.8b). The outer-loop control is (5.20) where ν is from (5.10) and f is given by (5.9). The inner-loop control is computed using (5.5) which is related to the inputs using (5.2b). In the following, we refer to this control as the SBF control. For comparison, the response to the PID controller described in Section 4.5 is also simulated, except the inputs are T_M, T_T, a, and b. The parameter values are given in Table A.1. Simulations were run for setpoint, helix, and figure-8 trajectories. The SBF controller gains are given in Table 5.1, and the PID controller gains are shown in Table 5.2.

Figure 5.3 shows the system response for the hover simulation. The initial conditions were $p_0 = (0.2,\ -0.3,\ -0.1)^T$ m, $v_0 = 0$ m/s, $\eta_0 = (10°,\ 5°,\ 30°)^T$, and

K_i^r	$0.5I$ s^{-3}
K_p^r	diag(10, 10, 7) s^{-2}
K_d^r	$5I$ s^{-1}
K_i^t	$0.2I$ kg/s^3
K_p^t	$2I$ kg/s^2
K_d^t	$3I$ kg/s

Table 5.2: PID controller gains used for simulation.

r_0	10 m
T_0	12 s
w_0	1 m/s

Table 5.3: Helix and Figure-8 reference trajectory parameters.

$\omega_0 = 0$. The state response in Figures 5.3a and 5.3b shows both controls have similar performance. This similarity is due to the SBF converging to constant values in the steady state. We observe from Figure 5.3c that the integrator states in the PID control converge to constants which allows the PID control to compensate the effect of the SBF.

Figure 5.4 shows the response to the helical trajectory. This position trajectory is circular with a constant vertical velocity

$$p_d^n = \begin{pmatrix} r_0 \cos\left(\frac{2\pi}{T_0} t\right) \\ r_0 \sin\left(\frac{2\pi}{T_0} t\right) \\ w_0 t \end{pmatrix}$$

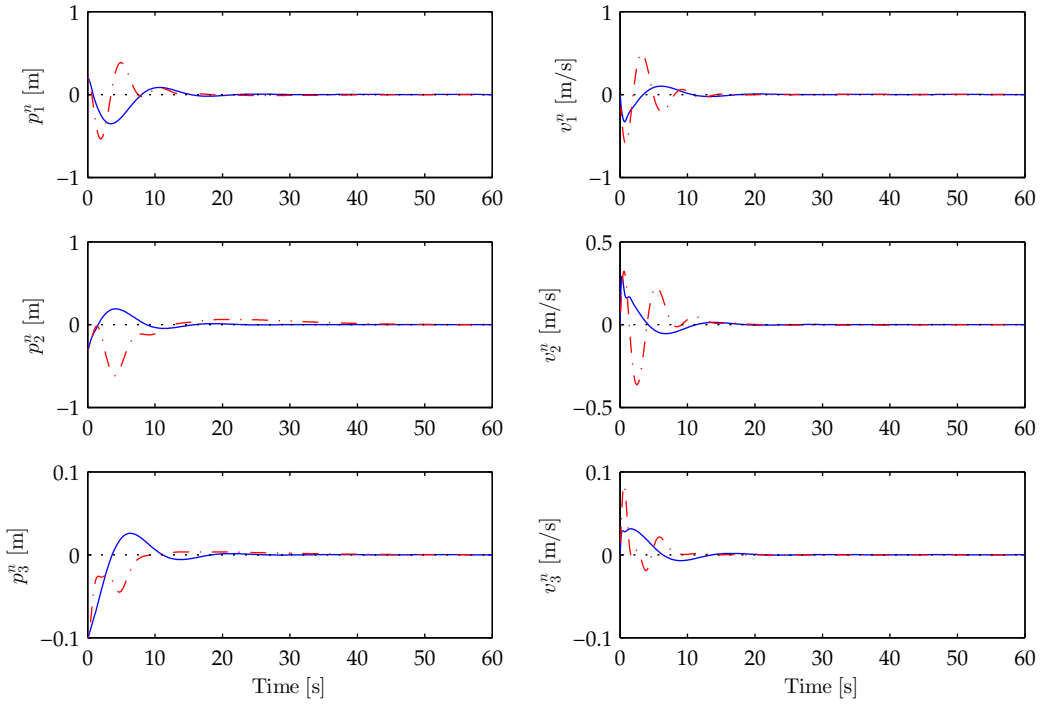
and the heading is always tangent to the circle

$$\psi_d = \frac{2\pi}{T_0} t + \frac{\pi}{2}$$

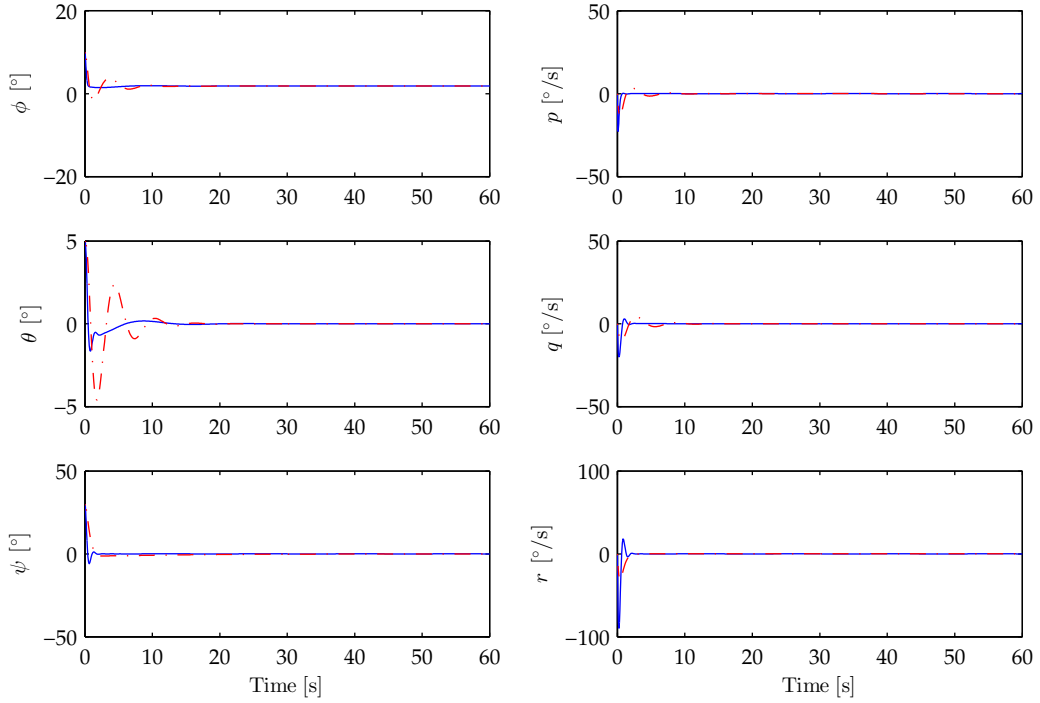
where the parameter values are given in Table 5.3. The initial conditions were $p_0 = (15,\ 0,\ -0.3)^T$ m, $v_0 = (0,\ 5.23,\ 0)^T$ m/s, $\eta_0 = (10^\circ,\ 5^\circ,\ 30^\circ)^T$, and $\omega_0 = 0$. Figures 5.4a and 5.4b show the state response. These figures show both controllers exhibit similar performance. However, once again the SBF converge to constants. Indeed, recalling (5.3) we see that time-varying terms do not appear in the SBF because only constant rotational velocities are necessary for this trajectory. In particular, comparing Figures 5.3b and 5.4b we see that the helical trajectory requires non-zero pitching and yawing speeds in the steady-state, but in both Figures 5.3d and 5.4d the inputs become constant which results in constant SBF. We remark that if a different heading trajectory was used, such as a constant heading, the flapping angle inputs would be time-varying which would disturb the translational dynamics through the SBF.

The final trajectory which was simulated was a figure-8 given by

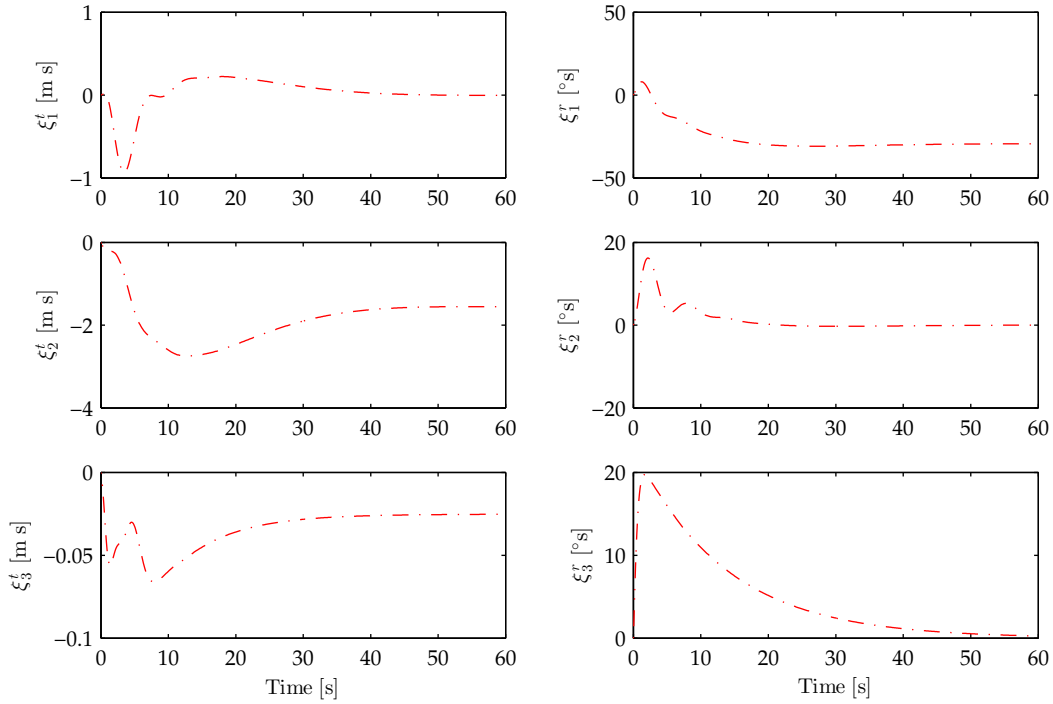
$$p_d^n = \begin{pmatrix} r_0 \sin\left(\frac{2\pi}{T_0} t\right) \\ \frac{r_0}{4} \sin\left(\frac{4\pi}{T_0} t\right) \\ 0 \end{pmatrix} \tag{5.37}$$



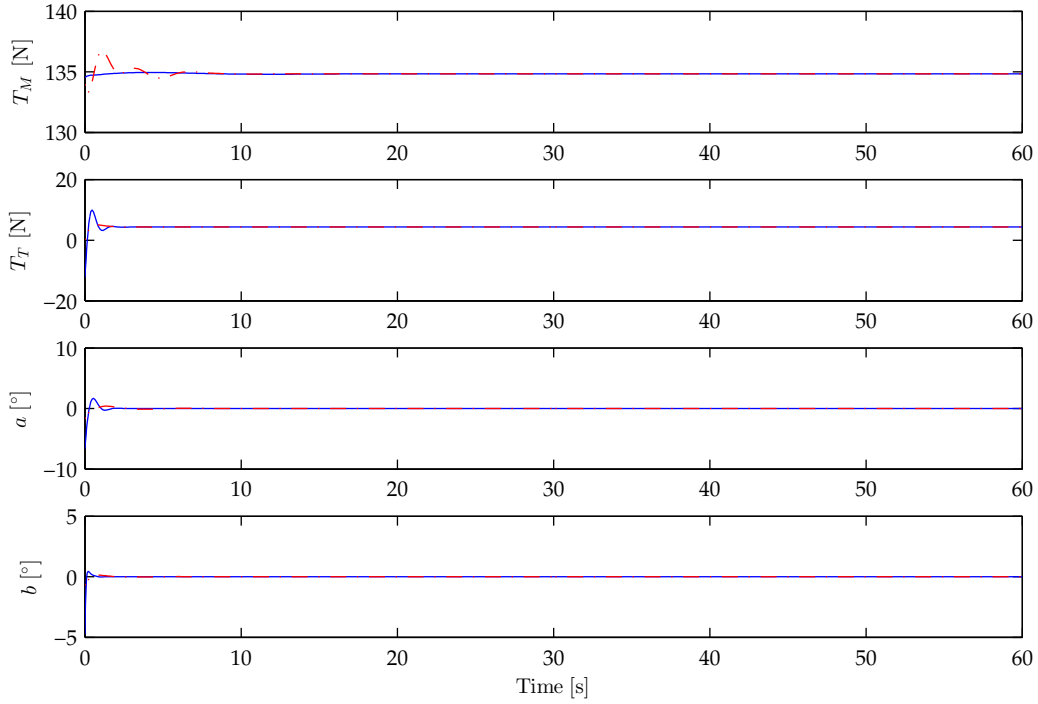
(a) Translational states.



(b) Rotational states.

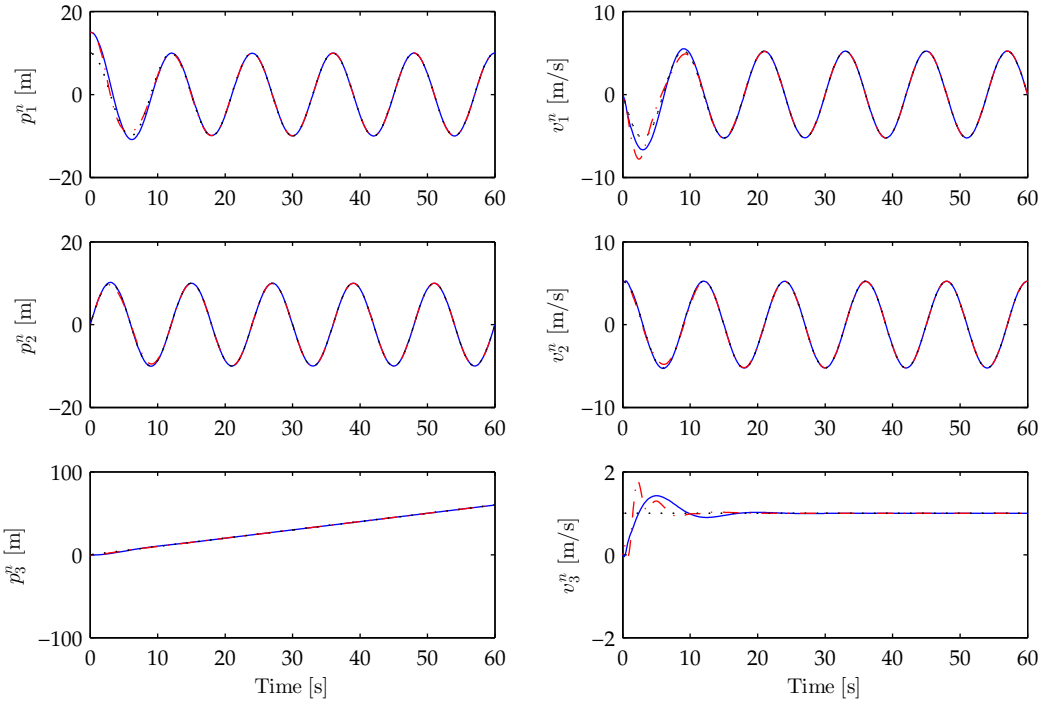


(c) PID controller states.

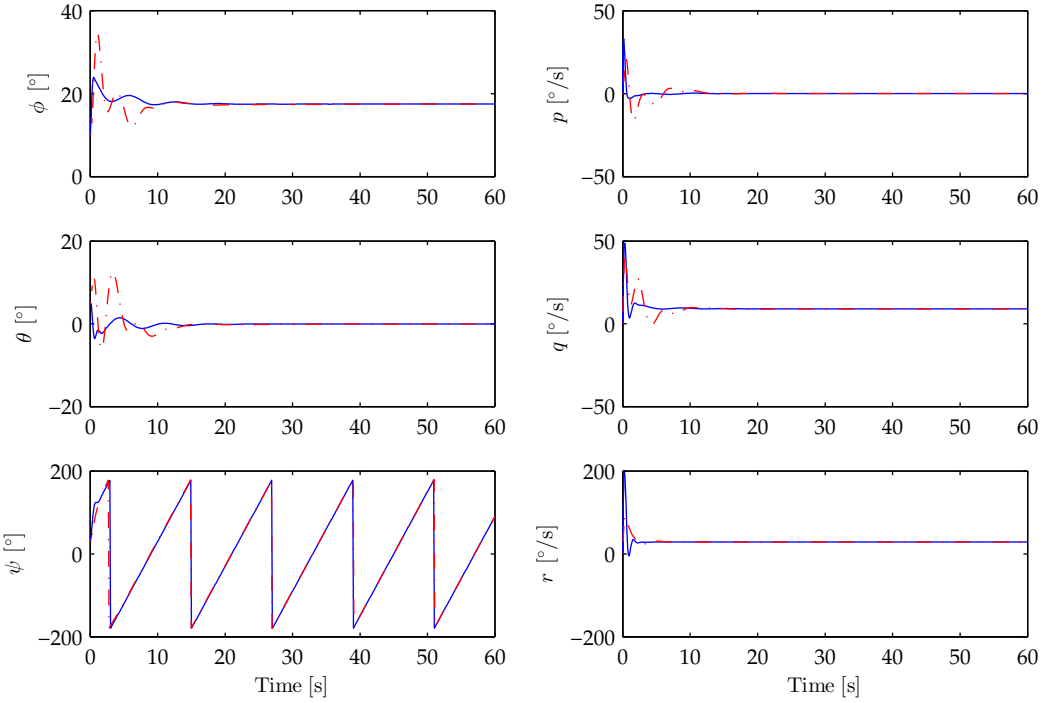


(d) Control effort.

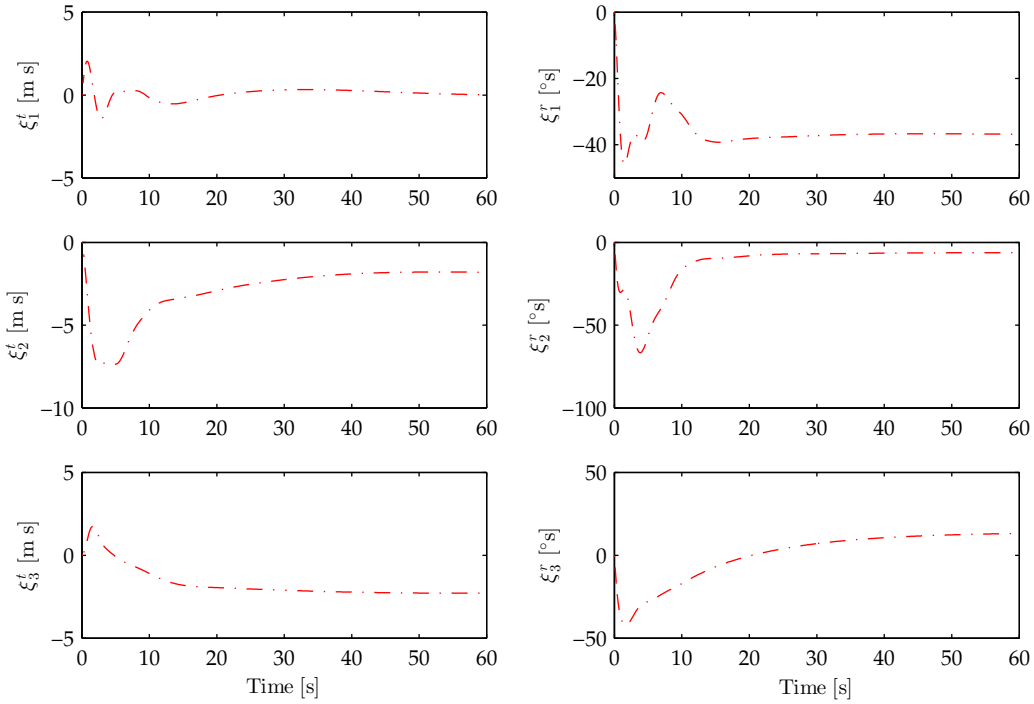
Figure 5.3: Simulated system response to regulating the origin. The solid lines show the SBF control, and the dashed-dotted lines show the PID control. The dotted lines show the reference.



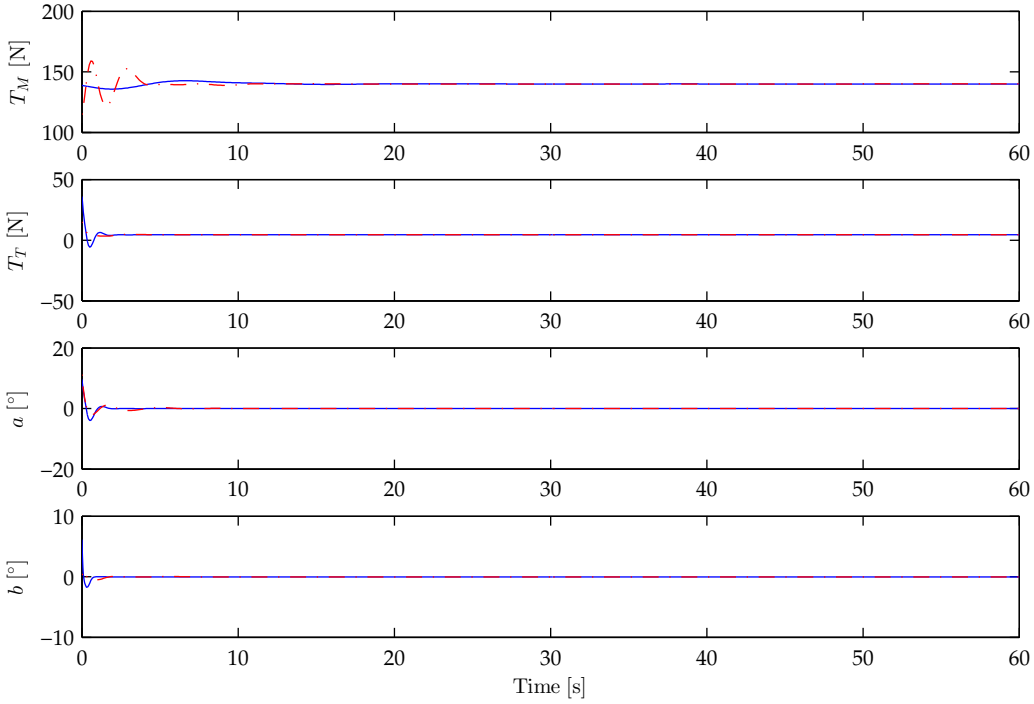
(a) Translational states.



(b) Rotational states.



(c) PID controller states.



(d) Control effort.

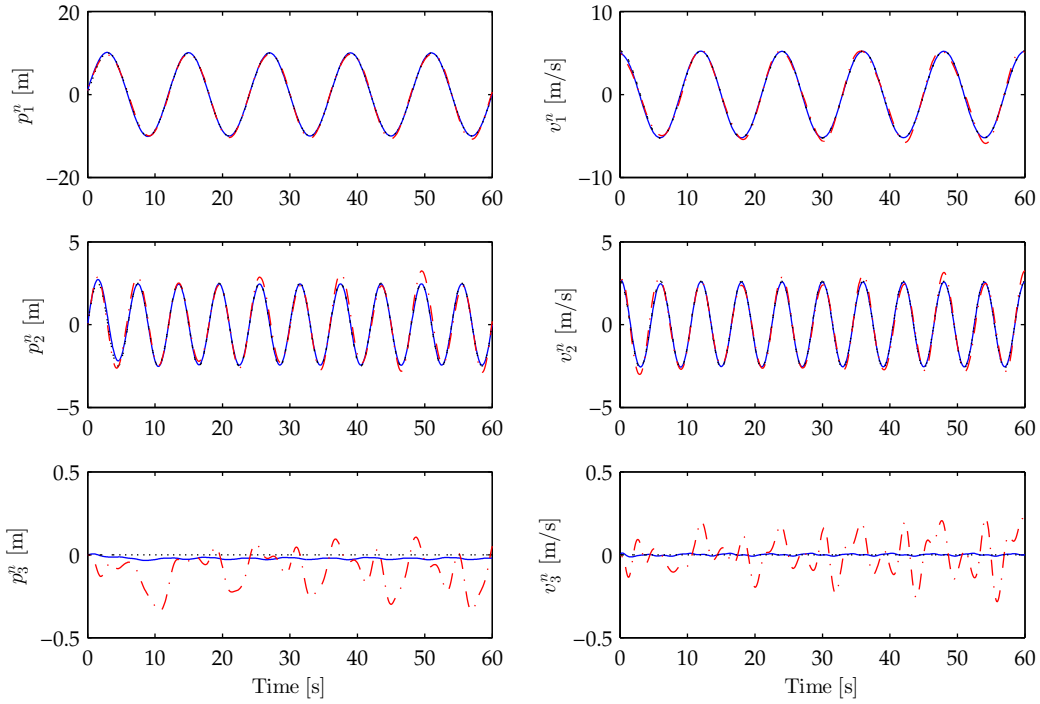
Figure 5.4: Simulated system response to tracking a helical trajectory. The solid lines show the SBF control, and the dashed-dotted lines show the PID control. The dotted lines show the reference.

with the heading always tangent

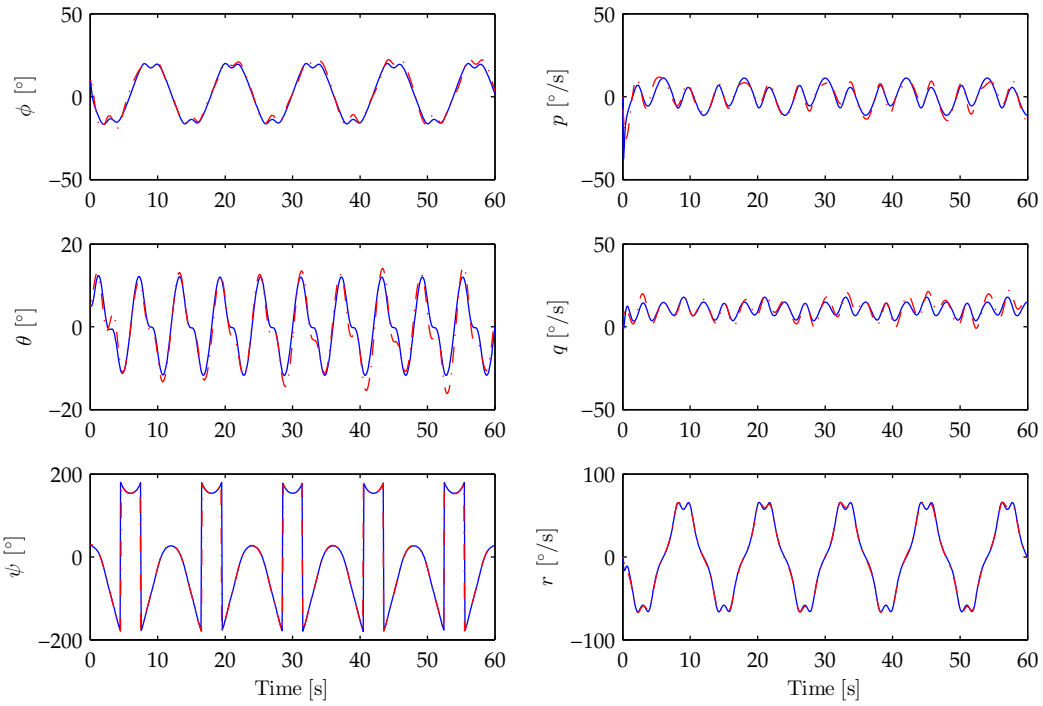
$$\psi_d = \arctan\left(\frac{\cos\left(\frac{4\pi}{T_0}t\right)}{2\cos\left(\frac{2\pi}{T_0}t\right)}\right) \tag{5.38}$$

where the parameters are the same as those given in Table 5.3. The initial conditions were $p_0 = (1,\ 0,\ 0)^T$ m, $v_0 = (5.23,\ 2.61,\ 0)^T$ m/s, $\eta_0 = (10^\circ,\ 5^\circ,\ 30^\circ)^T$, $\omega_0 = 0$. Figure 5.5 shows the system response. This trajectory requires steady-state time-varying inputs as seen in Figure 5.5d, and therefore the SBF do not converge. As a result, the PID integrator states shown in Figure 5.5c do not converge which causes the poor performance of the PID control evident in Figures 5.5a and 5.5b. The performance of the SBF controller is superior to the PID controller, however it only provides boundedness for this trajectory, as expected from Theorem 5.11.

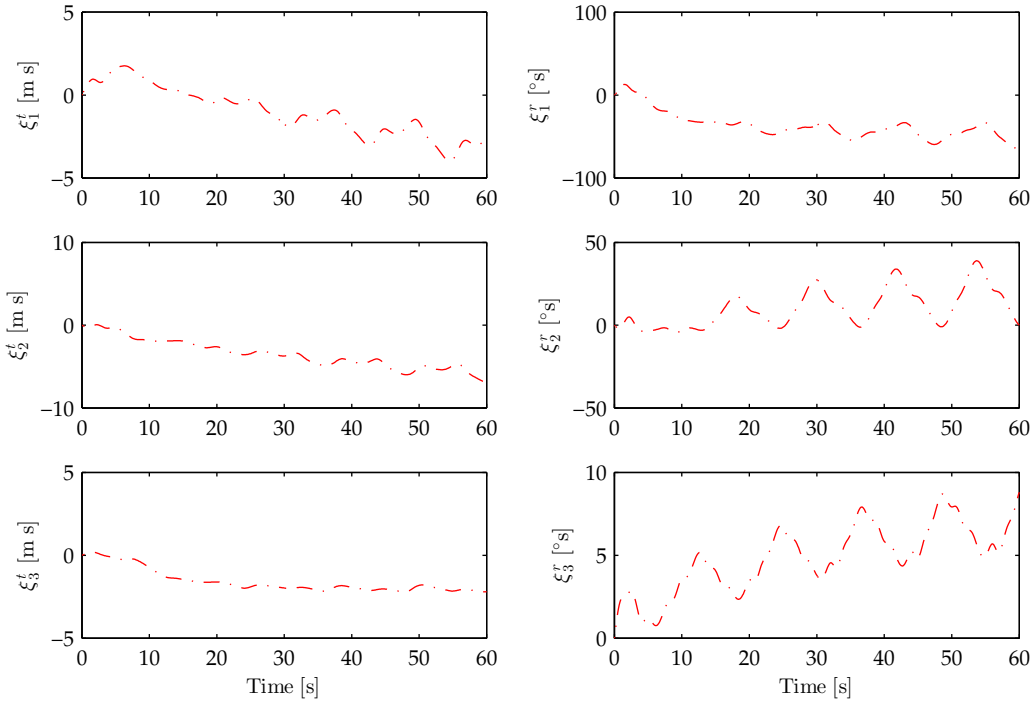
In addition to simulating the closed-loop response, we investigate the results from Section 5.3 using a numerical approach. Our interest arises since the bounds used for the analysis are likely to be conservative and the dynamics may possess a region of convergence larger than what is guaranteed. Using (5.25) we evaluate $\dot{V}$ over a range of errors in each degree of freedom independently. In other words, we compute $\dot{V}$ in the phase planes of the error coordinates along the e_1, e_2, and e_3 directions using different controller gains. Figure 5.6 shows the contours where $\dot{V} = 0$ in the phase plane of the e_1 direction when the reference has a constant velocity and heading, and the errors in the other directions are zero. These contours show that choosing gains less than one cause the region of convergence to be large but not global for this choice of Lyapunov function. While simulating the closed-loop we found that choosing gains larger than one improved the response (see Table 5.1). Therefore, we computed $\dot{V}$ using three multiples of the gains used for Figure 5.6. Figure 5.7 shows the contours where $\dot{V} = 0$ for a constant-velocity, constant-heading reference where the gains are $K_d^t = 0.1\gamma I$, and $K_p^t = 0.9\gamma I$, $\gamma \in \{4,\ 10,\ 14\}$. For the plots of the e_1 and e_2 directions, the errors in the other directions are zero. For the plot in the e_3 direction the errors are $\tilde{p}_1^n = 5$ m, $\tilde{p}_2^n = 10$ m, $\tilde{v}_1^n = -3$ m/s, and $\tilde{v}_2^n = 7$ m/s. This difference is due to the fact that the effect of the disturbance (5.19) does not enter in the e_3 direction when the errors in the other directions are zero. These plots show that as the gains are increased the region of convergence decreases. Furthermore, they suggest that the region of convergence could be made arbitrarily large by decreasing the gains which is the necessary condition for semiglobal stability. Finally, we consider the time-varying reference given by (5.37) and (5.38). In particular, we choose the values of the reference derivatives when $t = 8$ s. The results are shown in Figure 5.8 where for each plot the errors in the other directions are zero. As shown in Figure 5.8, this reference creates a region where $\dot{V} > 0$ whose boundary includes



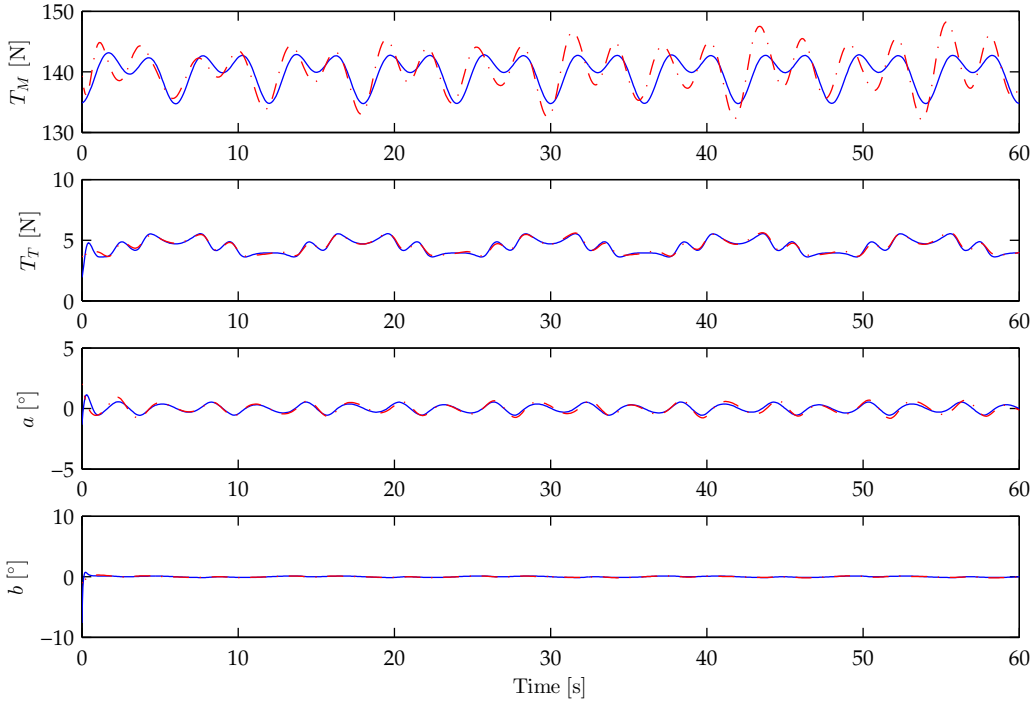
(a) Translational states.



(b) Rotational states.



(c) PID controller states.



(d) Control effort.

Figure 5.5: Simulated system response to tracking a figure 8 trajectory. The solid lines show the SBF control, and the dashed-dotted lines show the PID control. The dotted lines show the reference.

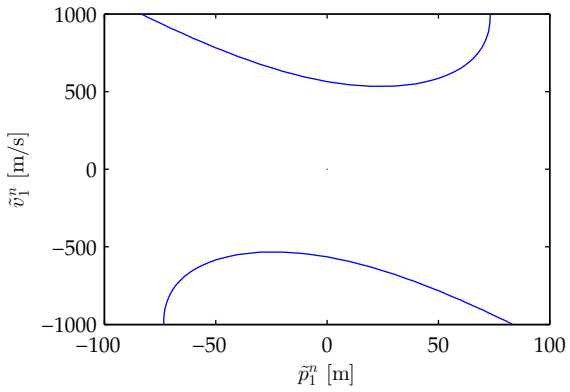


Figure 5.6: Contours where $\dot{V} = 0$ in the e_1 direction for a constant-velocity, constant-heading reference where $K_d^t = 0.1I$, and $K_p^t = 0.9I$.

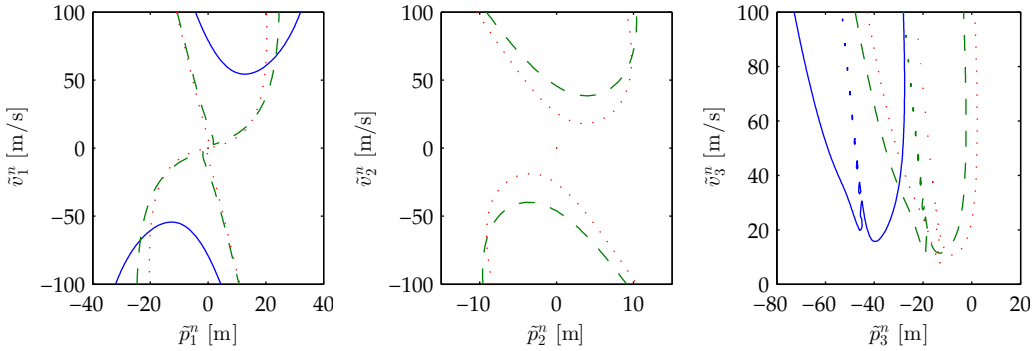


Figure 5.7: Contours where $\dot{V} = 0$ for a constant-heading, constant-velocity reference. The solid line shows contours for $K_d^t = 0.4I$, and $K_p^t = 3.6I$; the dashed line shows contours for $K_d^t = 1I$, and $K_p^t = 9I$; and the dotted line shows contours for $K_d^t = 1.4I$, and $K_p^t = 12.6I$.

the origin. These regions imply that ultimate boundedness is the strongest stability condition which can be guaranteed for a general time-varying reference.

5.5 Experimental Results

Similar to the PID control in Chapter 4, the control derived in the present chapter is tested in experiment using the ANCL Helicopter described in Chapter 2. For safety, the autopilot is programmed to control the cyclic inputs while the pilot controls the main and tail rotor thrusts. Since we are mainly interested in the outer-loop control, we use the inner-loop PID control (4.25) for the attitude. For the outer-loop control we compute ϕ_d and θ_d as in (5.20c) and (5.20b). However, we assume a constant heading which simplifies (5.20c). Instead of (5.22) we compute ν using

$$\nu = R_z^T \left(-mge_3 - K_d^t v^n - K_p^t \tilde{p}^n - K_i^t \xi^t \right)$$

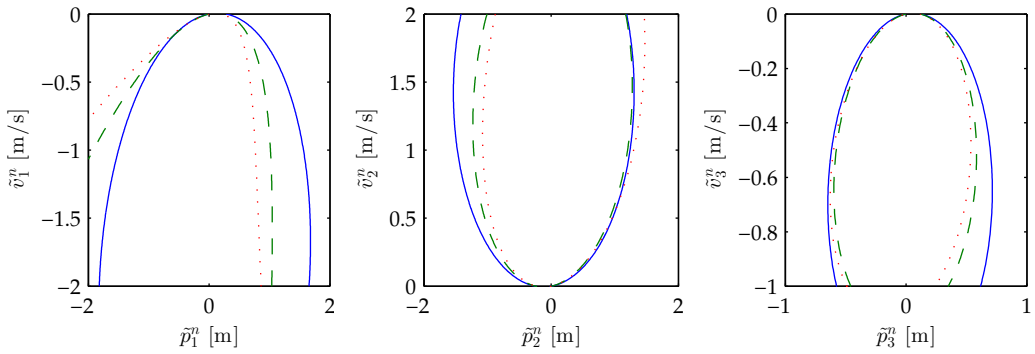
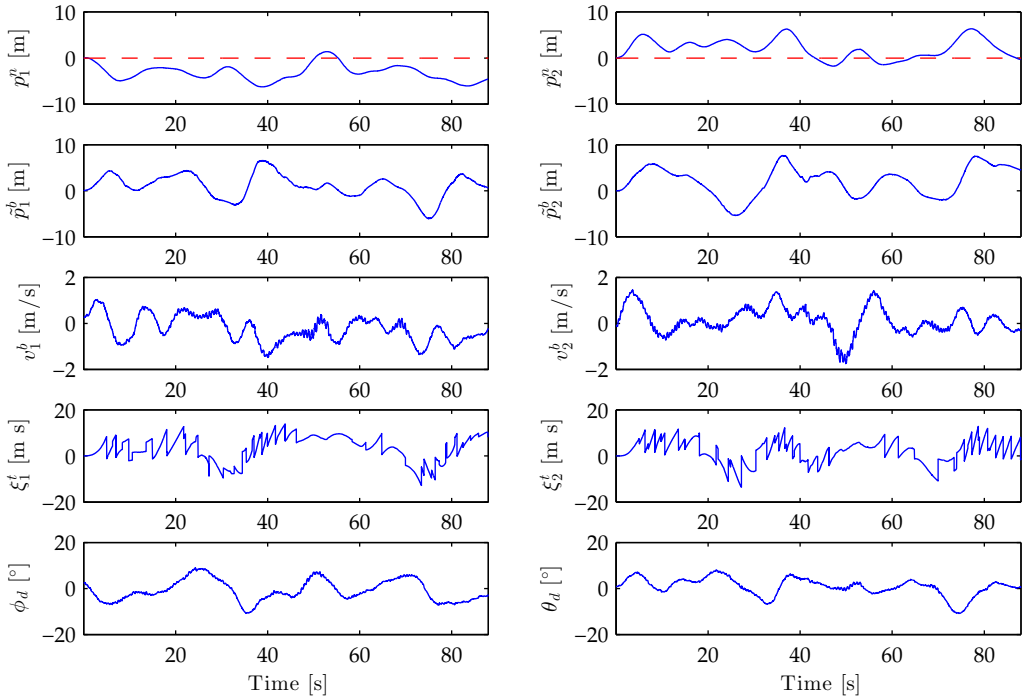


Figure 5.8: Contours where $\dot{V} = 0$ near the origin when the reference is given by (5.37) and (5.38) at $t = 8$ s. The solid line shows contours for $K_d^t = 0.4I$, and $K_p^t = 3.6I$; the dashed line shows contours for $K_d^t = 1I$, and $K_p^t = 9I$; and the dotted line shows contours for $K_d^t = 1.4I$, and $K_p^t = 12.6I$.

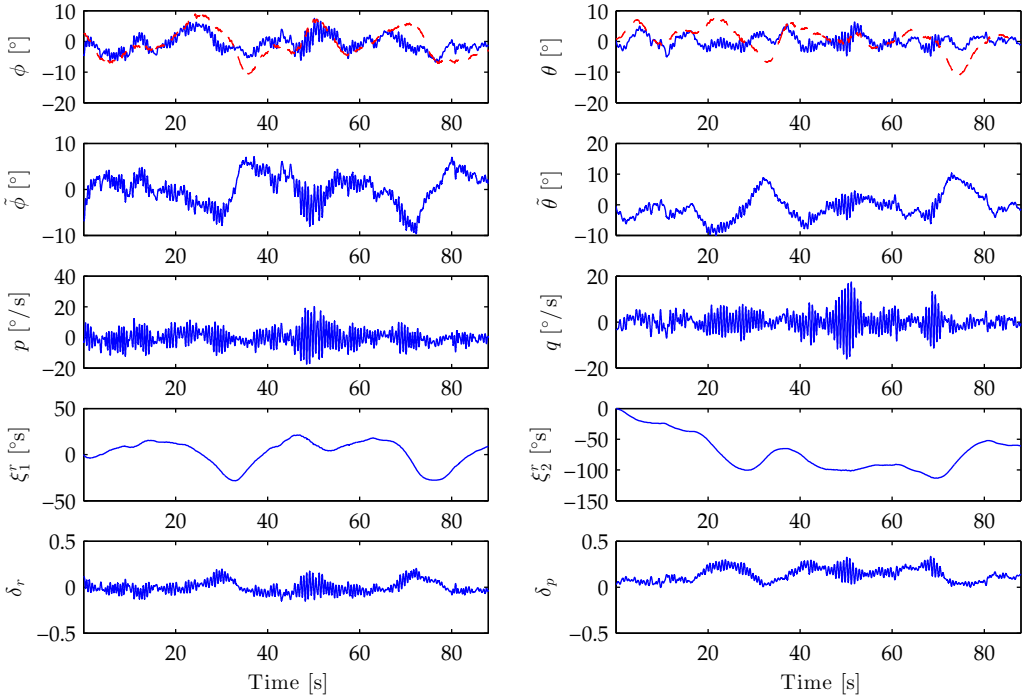
K_d^t	$0.22I$ kg/s
K_p^t	$0.5I$ kg/s^2
K_i^t	0.009 kg/s^3

Table 5.4: Controller gains used for experimental testing.

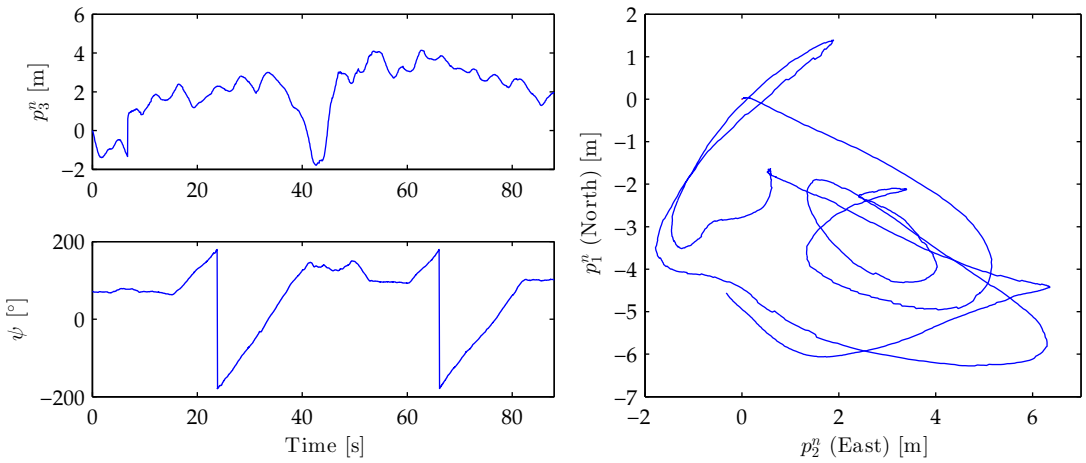
where we have introduced the integrator state ξ^t for robustness. In addition, we assume the pilot maintains the reference altitude and therefore use $\nu_3 = -mg$. Furthermore, the dimension of the gain matrices is reduced to 2×2. The values used for the control gains are given in Table 5.4. Figure 5.9 shows the response to regulating the origin while the pilot conducts two complete yaw rotations. These experiments were conducted concurrently with those in Section 4.6 and were therefore subject to same strong wind conditions which explains why the helicopter is pushed to the south-east. Despite the disturbance it is clear from the results that the control is working correctly. The integrator states shown in Figure 5.9a are computed in the navigation frame and do not oscillate as the helicopter rotates, as opposed to those in Figure 4.3a which computed in the body frame. The second reference was a line in the horizontal plane. The parameters for the trajectory were 15 m travel in the e_1' direction and a -15 m travel in the e_2' direction at a speed of 1 m/s. The results for this trajectory are shown in Figure 5.10. Once again, it is clear that the control is performing correctly. The final trajectory was circular with a 7 m radius and a tangential speed of 2.5 m/s. The results of this experiment are shown in Figure 5.11. It is possible that the translational tracking error resulted from the ultimate boundedness performance limitation of the control method. Overall, although the control works, it is difficult to compare it with the PID control in Chapter 4. Experimental helicopter flight is subject to many disturbances and unmodelled effects. Therefore, as opposed to simulation, it is difficult to attribute the cause of performance



(a) Translational states, errors and control effort. The reference position is shown by the dashed line.



(b) Rotational states, errors and control effort. The reference is shown by the dashed line.



(c) Heading and height measurements for setpoint regulation with full yaw rotation.

(d) Parametric plot of the horizontal position in $\mathcal{N}$.

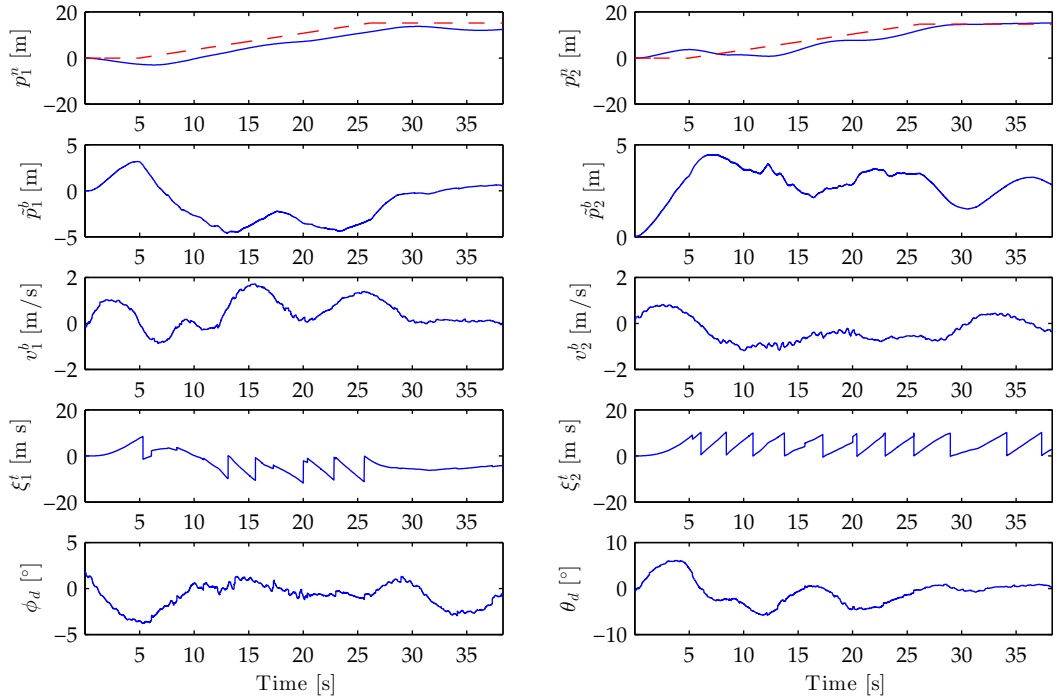
Figure 5.9: System response for setpoint regulation with a full yaw rotation. The reference location is the origin of the navigation frame.

differences.

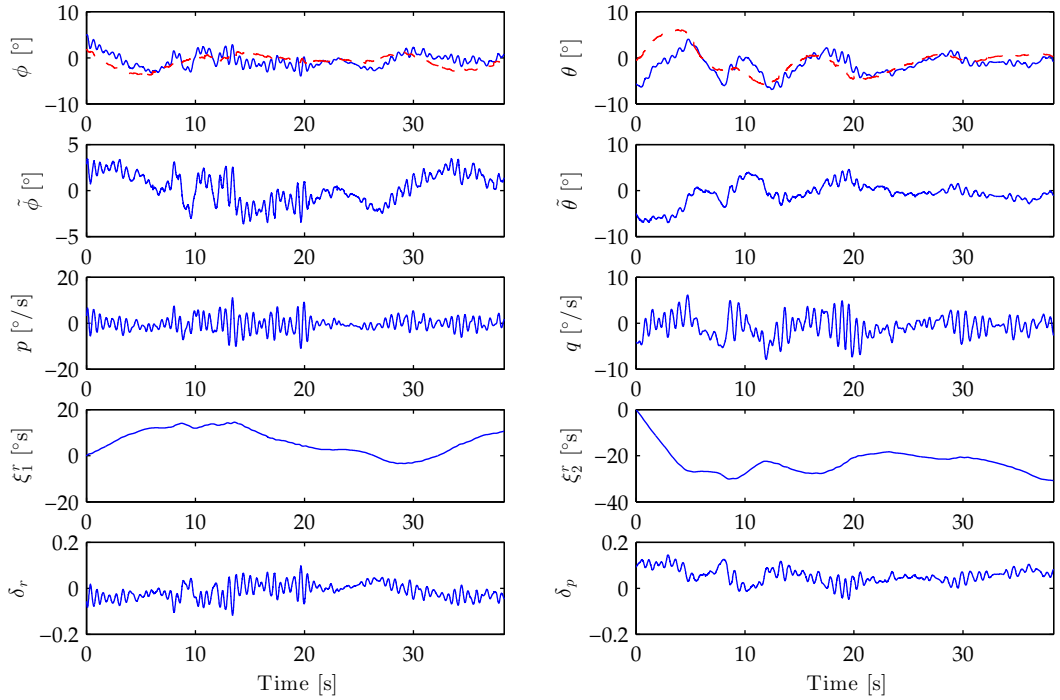
5.6 Conclusions

The generally accepted model for small helicopters includes a disturbance to the translational dynamics by the inputs to the rotational dynamics called the SBF. Despite its acceptance, existing work neglects this effect for control design. We provide a design which compensates the contribution of the tail rotor to the SBF using an inner-outer hierarchical approach. We further show that the closed-loop system will remain uniformly ultimately bounded in the presence of the SBF due to the main rotor flapping angles for a general reference. The specific statement of the disturbance bounds given in the robustness analysis provides insight into performance tradeoffs of the design. When the reference is constant heading, constant velocity (e.g., hover) we show asymptotic stability.

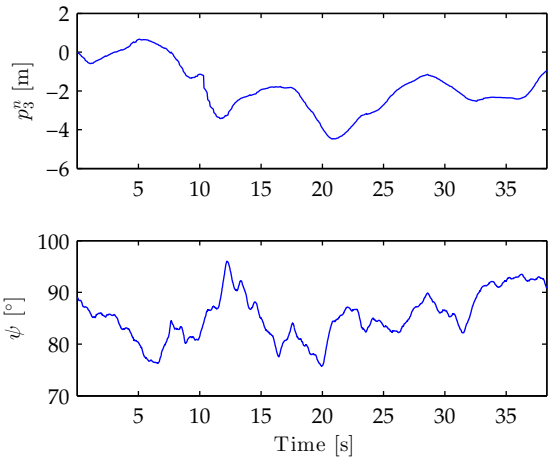
The design presented in this chapter was validated by simulation and in flight tests. The simulation results show our design has performance benefits over a classical PID solution for a figure-8 trajectory. In this case, the SBF control tracking error remains bounded as predicted by the analysis. Although both controls had bounded tracking errors, the SBF achieved a smaller bound.



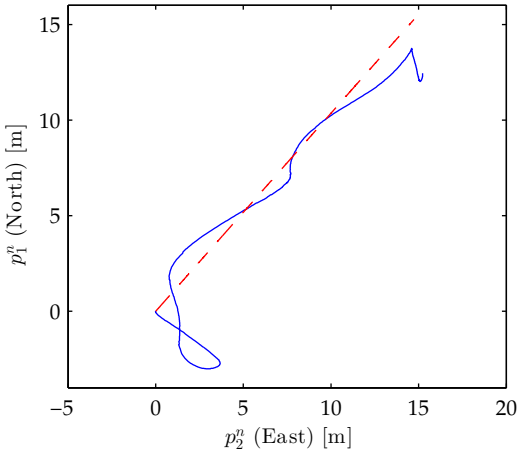
(a) Translation states, errors and control effort. The reference is shown by the dashed line.



(b) Rotational states, errors and control effort for line tracking. The reference is shown by the dashed line.

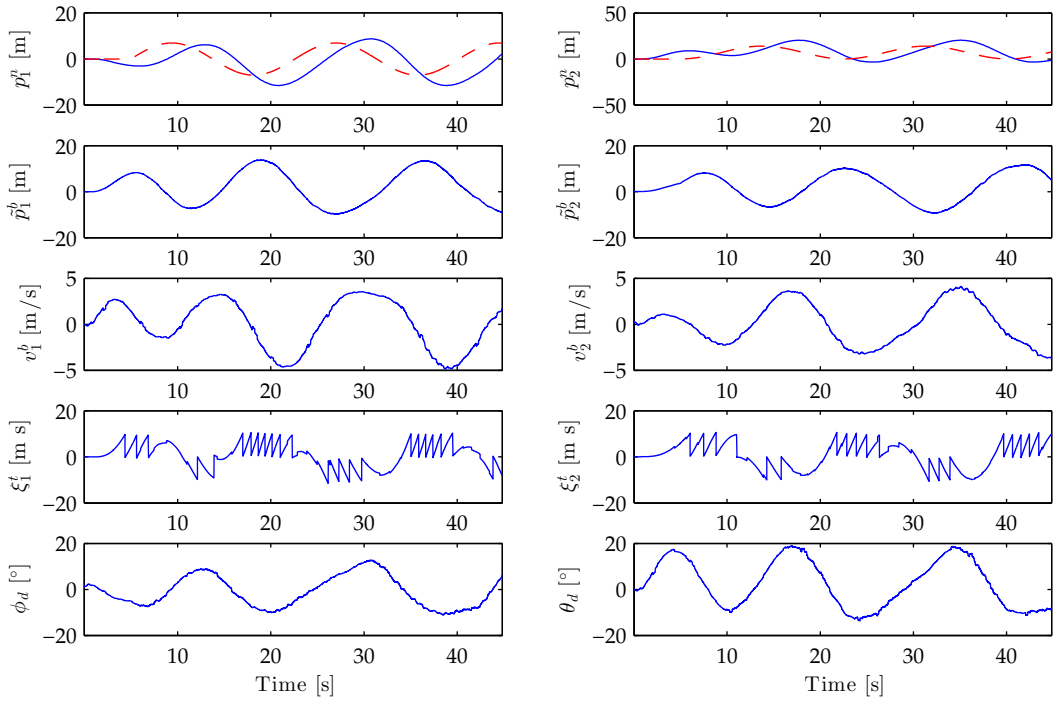


(c) Heading and height measurements.

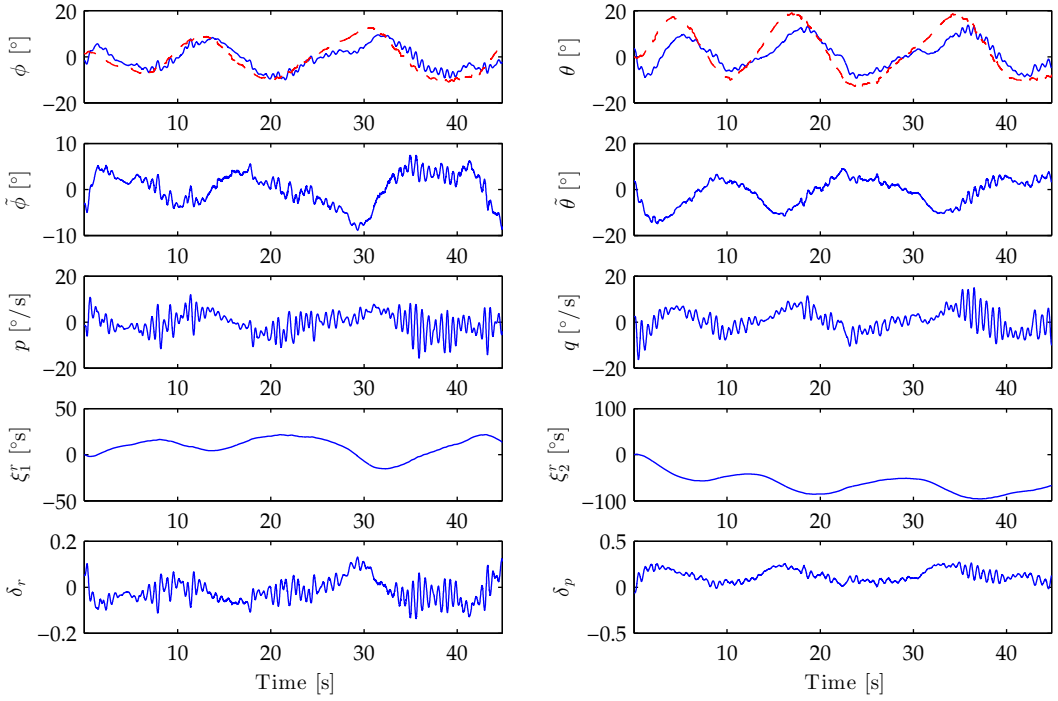


(d) Parametric plot of the horizontal position in $\mathcal{N}$. The reference is shown by the dashed line.

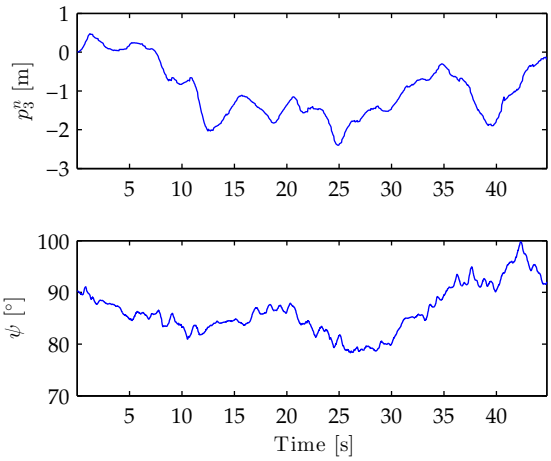
Figure 5.10: System response for line tracking experiment with a 15 m travel in the e_1' direction and a -15 m travel in the e_2' direction at a speed of 1 m/s.



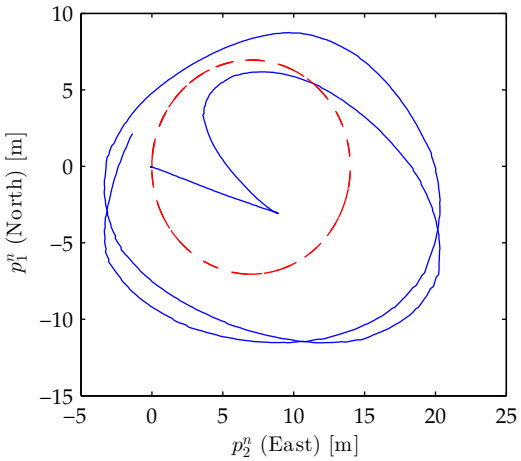
(a) Translation states, errors, and control effort. The reference is shown by the dashed line.



(b) Rotational states, errors and control effort. The reference is shown by the dashed line.



(c) Heading and height measurements.



(d) Parametric plot of the horizontal position in $\mathcal{N}$. The reference is shown by the dashed line.

Figure 5.11: System response for circle tracking experiment with a 7 m radius a tangential speed of 2.5 m/s.

Chapter 6

Practically-Motivated Nonlinear Control with State Constraints

Real-world flight is subject to constraints. Input constraints could be considered since factors such as servo travel limit control authority. Examples of designs which consider this effect are [16, 17]. Considering Figures 3.8, 4.5, and 5.11 where aggressive manouevering is performed with relatively little control effort, we conclude input saturation is not practically relevant for non-inverted flight. This conclusion is also supported by manual flight wherein performance does not seem limited by the controls. In this chapter we consider the problem of stabilizing the helicopter subject to state constraints. The motivation for the design is to allow an operator to use setpoints to guide the helicopter, while also being able to configure upper bounds on velocity and roll-pitch. For instance, if the helicopter were carrying a payload used for inspection or surveying (e.g., camera or Lidar), large translational velocity may cause insufficient data coverage. In addition, if the sensor is fixed to the airframe or mounted with a limited pan-tilt travel it would be desirable to be able to prevent the roll and pitch of the aircraft from exceeding bounds. We apply a recently published method found in [32] and originally applied to helicopter control in [33]. Their approach to helicopter control is based on the design presented in [11] which uses a dynamic extension to decouple the inputs and an input-output linearization. However, as discussed in [73] control algorithms which use dynamic extension are not desirable in practice. Therefore, we exploit our hierarchical design used throughout this thesis to remove the need for dynamic extension and construct a control which is appropriate for practical implementation. The background for the analysis is presented in Section 6.1. In Section 6.2 the model is given in order make explicit the notation and parametrization used in this design. The outer-loop control is developed in Section 6.3, and the inner-loop control is given in Section 6.4. Finally, in Section 6.5 we present the simulation results and implementation details.

6.1 General Approach

We base our method of satisfying state constraints on recent work [32] which was expanded to the multivariable case in [33] and applied to the input-output linearization of the helicopter dynamics given in [11]. The method can be roughly described as follows. The first step is to define state constraints as fictitious outputs. The dynamics can then be transformed to the tracking form found in [74]. Using these coordinates, a backstepping approach is applied to derive conditions on the input which will satisfy the original constraints.

We begin by defining some basic notation. Consider the system

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)u_i \tag{6.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ is the input, and f and g_i are smooth for $1 \leq i \leq m$ on some domain of interest. We model the constraints as outputs

$$y_j = h_j(x), \qquad 1 \leq j \leq l \tag{6.2}$$

where $y \in \mathbb{R}^l$. We assume that each output has a well-defined relative degree ρ_j

$$\begin{aligned} L_{g_i} L_f^k h_j(x) &= 0, \qquad 1 \leq i \leq m,\ 0 \leq k \leq \rho_j - 2 \\ L_{g_i} L_f^{\rho_j - 1} h_j(x) &\neq 0, \qquad \text{for some } i \in \{1, \ldots, m\} \end{aligned}$$

where $L_f^i h$ is the ith Lie derivative of h along the vector field f (see e.g., [75] for a general reference, among many others), and a transformation exists such that

$$\zeta_j = (h_j, \ldots, L_f^{\rho_j - 1} h_j) \tag{6.3}$$

results in the dynamics

$$\begin{aligned} \dot{\zeta}_{j,i} &= \zeta_{j,i+1} \qquad 1 \leq i \leq \rho_j - 1 \\ \dot{\zeta}_{j,\rho_j} &= L_f^{\rho_j} h_j + \sum_{i=1}^{m} u_i L_{g_i} L_f^{\rho_j - 1} h_j. \end{aligned}$$

We are now ready to establish some useful results.

Lemma 6.1 (Comparison Lemma [71])**.** *Given the scalar dynamics*

$$\dot{v}(t) \leq f(t, v(t)), \qquad v_0 = v(0)$$

where $v\colon [0,\infty) \to \mathbb{R}$ *and* $f\colon [0,\infty)\times\mathbb{R} \to \mathbb{R}$ *is smooth, and the comparison dynamics*

$$\dot{w}(t) = f(t, w(t)), \qquad w_0 = w(0)$$

where $w\colon [0,\infty) \to \mathbb{R}$*, if* $v_0 \leq w_0$ *then* $v(t) \leq w(t)$ *for all* $t \in [0,\infty)$*.*

Proof. See [71, Lem. 3.4]. □

Lemma 6.2 ([32])**.** *Given the first order dynamics*

$$\dot{v}(t) = -\alpha(v(t)) + \beta(t) \tag{6.4}$$

where $v\colon [0,\infty) \to \mathbb{R}$*,* α *is class* $\mathcal{K}$*, and* $\beta(t) \leq 0$ *for all* $t \in [0,\infty)$*; if* $v(0) \leq 0$ *then* $v(t) \leq 0$ *for all* $t \in [0,\infty)$*.*

Proof. The proof is to show that the solutions of (6.4) are always less than the solutions of a globally asymptotically stable first order system. Consider the comparison dynamics

$$\dot{w}(t) = -\alpha(w(t)) \tag{6.5}$$

where $w\colon [0,\infty) \to \mathbb{R}$, and the Lyapunov function $V = \frac{1}{2}w^2$. The derivatives of V along the trajectories of (6.5) are

$$\dot{V} = -\alpha(w)w$$

which implies global asymptotic stability. Since the dynamics are first order, there will not be overshoot so if $w(0) \leq 0$ then $w(t) \leq 0$ for all $t \in [0,\infty)$. Therefore, by Lemma 6.1 if $v(0) \leq w(0)$ then $v(t) \leq 0$ for all $t \in [0,\infty)$. □

The following theorem allows us to convert a state constraint of relative degree ρ_j into $\rho_j - 1$ conditions on the initial state and a new state constraint with relative degree one. A control law can then be derived based on these new conditions which satisfies the original state constraint.

Theorem 6.3 ([33])**.** *Consider the dynamics* (6.1)*. For each constraint* $h_j(x) \leq 0$ *which we treat as an output with a well-defined relative degree* ρ_j *we can derive* $\rho_j - 1$ *additional inequality conditions on the initial state* $x(0)$ *and a state constraint with relative degree one. Satisfaction of the derived conditions implies the satisfaction of the original constraint.*

Proof. The proof follows a backstepping approach to finding the constraints. The first condition on $x(0)$ is $\zeta_{j,1}(x(0)) = h_j(x(0)) \leq 0$. We know

$$\dot{\zeta}_{j,1}(x) = \zeta_{j,2}(x)$$

so we can add and subtract a term $\alpha_{j,0}\zeta_{j,1}$ to obtain

$$\dot{\zeta}_{j,1}(x) = -\alpha_{j,0}\zeta_{j,1}(x) + v_{j,1}(x)$$

where $\alpha_{j,0} > 0$ and $v_{j,1}(x) = \zeta_{j,2}(x) + \alpha_{j,0}\zeta_{j,1}(x)$. If we impose the constraint $v_{j,1}(x) \leq 0$, from Lemma 6.2 we have that for any initial condition $x(0)$ which satisfies $\zeta_{j,1}(x(0)) \leq 0$ then $\zeta_{j,1}(x) \leq 0$ for all $t \in [0, \infty)$. We proceed in a similar manner. The derivative of $v_{j,1}$ is

$$\begin{aligned}\dot{v}_{j,1}(x) &= \zeta_{j,3}(x) + \alpha_{j,0}\zeta_{j,2}(x) \\ &= -\alpha_{j,1}v_{j,1}(x) + v_{j,2}(x)\end{aligned}$$

where $v_{j,2} = \zeta_{j,3}(x) + \alpha_{j,0}\zeta_{j,2}(x) + \alpha_{j,1}v_{j,1}(x)$. Thus, as long as $v_{j,2}(x)$ is kept non-positive, an initial condition satisfying $v_{j,1}(x(0)) \leq 0$ and $\zeta_{j,1}(x(0))$ will guarantee $v_{j,1}(x) \leq 0$ for all $t \in [0, \infty)$. The process continues until we reach $v_{j,\rho_j-1}(x)$. This function will depend on ζ_{j,ρ_j} and therefore its derivative will contain the input. We write this expression as

$$\dot{v}_{j,\rho_j-1} = L_f^{\rho_j} h_j + \sum_{i=1}^{m} u_i L_{g_i} L_f^{\rho_j - 1} h_j + v_j(x)$$

where $v_j(x)$ can be computed recursively. Thus we have $\rho_j - 1$ conditions on the initial state $\zeta_{j,1}(x(0)), \ldots, v_{j,\rho_j-2}(x(0)) \leq 0$ and a state constraint with relative degree one $v_{j,\rho_j-1}(x) \leq 0$. □

In order to make use of Theorem 6.3 we must now construct an input which will guarantee $v_{j,\rho_j-1}(x) \leq 0$. First, we define an active constraint to be one which satisfies

$$-\varepsilon \leq v_{j,\rho_j-1}(x) \leq 0$$

for some $\varepsilon > 0$ which provides appropriate resolution. Then we define the number of active constraints $c \leq m$ and require there to be at least as many inputs as active constraints. Let

$$C = \begin{pmatrix} v_{1,\rho_1-1} \\ \vdots \\ v_{c,\rho_c-1} \end{pmatrix}$$

be a vector of the active constraints, and define

$$\dot{C} \triangleq B(x) + A(x)u$$

where $B\colon \mathbb{R}^n \to \mathbb{R}^c$ and $A\colon \mathbb{R}^n \to \mathbb{R}^{m \times c}$. Let u_{nom} be the control which stabilizes (6.1) inside the constraint admissible set. We will now define the control which

renders the admissible set invariant. Following [33, 76] we choose the invariance control to minimize the difference from the nominal control by moving the state along the boundary

$$\min\|W^{-1}(u_{inv} - u_{nom})\|, \qquad \text{such that } A(x)u_{inv} + B(x) = 0$$

where W is a chosen weighting matrix. This minimization has an explicit solution defined using the generalized inverse [60]

$$u_{inv} = (I - W(AW)^{\dagger}A)u_{nom} - W(AW)^{\dagger}B. \tag{6.6}$$

We complete the definition of the invariant control by adding the case where the nominal control already satisfies the constraints

$$u_{inv} = \begin{cases} u_{nom}, & A(x)u_{nom} + B(x) \leq 0 \\ (I - W(AW)^{\dagger}A)u_{nom} - W(AW)^{\dagger}B, & A(x)u_{nom} + B(x) \succ 0 \end{cases}$$

where $\leq$ indicates all elements of the vector are less than zero and $\succ$ means at least one element is greater than zero. Finally, we introduce a blending function to create a smooth transition between controls when the system approaches a boundary. When two constraints are active the control becomes

$$\begin{aligned} u = {} & (1 - \sigma_1(x))(1 - \sigma_2(x))u_{nom} + \sigma_1(x)\sigma_2(x)u_{inv} + (1 - \sigma_2(x))\sigma_1(x)u_{1,inv} \\ & + (1 - \sigma_1(x))\sigma_2(x)u_{2,inv} \end{aligned}$$

where u_{inv} is the invariance control for all active constraints, $u_{i,inv}$ is the invariance control when only the ith constraint is active, and the σ terms are sigmoid functions which provide a smooth transition between controls. These transition functions have the property that they are zero for some ε distance from the boundary and one on the boundary. When three constraints are active the control becomes

$$\begin{aligned} u = {} & (1 - \sigma_1(x))(1 - \sigma_2(x))(1 - \sigma_3(x))u_{nom} + \sigma_1(x)\sigma_2(x)\sigma_3(x)u_{inv} + \\ & + \sigma_1(x)(1 - \sigma_2(x))(1 - \sigma_3(x))u_{1,inv} + (1 - \sigma_1(x))\sigma_2(x)(1 - \sigma_3(x))u_{2,inv} \\ & + (1 - \sigma_1(x))(1 - \sigma_2(x))\sigma_3(x)u_{3,inv} + \sigma_1(x)\sigma_2(x)(1 - \sigma_3(x))u_{12,inv} \\ & + \sigma_1(x)(1 - \sigma_2(x))\sigma_3(x)u_{13,inv} + (1 - \sigma_1(x))\sigma_2(x)\sigma_3(x)u_{23,inv} \end{aligned}$$

where $u_{ij,inv}$ is the invariance control when the ith and jth constraints are active.

6.2 Modelling

The model we consider here treats the translational dynamics in $\mathcal{N}$ and the orientation parametrized by the Euler angles as in (3.7). This model is

$$\dot{p}^n = v^n \tag{6.7a}$$

$$m\dot{v}^n = mge_3 - R_z \begin{pmatrix} s\theta c\phi \\ -s\phi \\ c\theta c\phi \end{pmatrix} T_M \tag{6.7b}$$

$$\dot{\eta} = W(\eta)\omega \tag{6.7c}$$

$$J\dot{\omega} = -\omega \times J\omega + \begin{pmatrix} z_M T_M b \\ z_M T_M a \\ x_T T_T - Q_M \end{pmatrix} \tag{6.7d}$$

where we have neglected the SBF (see Chapter 5) and made the usual assumptions on the lever arms and tail countertorque: $l_M \approx (0,\, 0,\, -z_M)^T$, $l_T \approx (-x_T,\, 0,\, 0)^T$, $Q_T = 0$ (see Chapter 4). The thrusts T_M and T_T are mapped to Θ_M and Θ_T using (3.15) and (3.21). The flapping angle inputs a and b are mapped to normalized cyclic δ_p and δ_r inputs using (3.20).

6.3 Outer-Loop Control

The objective of the outer-loop control is to track the position reference p_d^n subject to constant velocity bounds. In order to perform the outer-loop control design we will make use of the linearizing input transformation

$$\nu^t = -R_{zd} \begin{pmatrix} s\theta_d c\phi_d \\ -s\phi_d \\ c\theta_d c\phi_d \end{pmatrix} T_M \tag{6.8}$$

where the subscript $_d$ denotes a reference quantity. We remark that a similar mapping to (6.8) appears in [27]. Using this transformation, we are left with

$$\begin{aligned} \dot{\tilde{p}}^n &= \tilde{v}^n \\ m\dot{\tilde{v}}^n &= mge_3 - m\ddot{p}_d^n + \nu^t \end{aligned}$$

where $\tilde{p}^n = p^n - p_d^n$ and $\tilde{v}^n = v^n - v_d^n$. We must now express the velocity bounds as fictitious outputs. We define the bounds $v^{b*} = (u^*,\, v^*,\, w^*)^T > 0$. Since these bounds are defined in $\mathcal{B}$, when they are transformed to the navigation frame, they

are time-varying. The corresponding outputs are

$$h_{\bar{u}} = v_1^n - v_1^{n*} \qquad h_{\bar{v}} = v_2^n - v_2^{n*} \qquad h_{\bar{w}} = v_3^n - v_3^{n*}$$
$$h_{\underline{u}} = -v_1^n - v_1^{n*} \qquad h_{\underline{v}} = -v_2^n - v_2^{n*} \qquad h_{\underline{w}} = -v_3^n - v_3^{n*}$$

where $v^{n*} = Rv^{b*}$. We remark that all of these outputs have relative degree one. Since the translational dynamics have been linearized, the nominal control is

$$\nu_{nom}^t = m\ddot{p}_d^n - mge_3 - K_d^t\tilde{v}^n - K_p^t\tilde{p}^n$$

where K_d^t and K_p^t are gain matrices. In order to illustrate the implementation of the invariance control we will consider the case where v_2^n approaches v_2^{n*} and v_3^n approaches $-v_3^{n*}$. In this case

$$C = \begin{pmatrix} v_2^n - v_2^{n*} \\ -v_3 - v_3^{n*} \end{pmatrix},$$

and

$$\dot{C} = \frac{1}{m}\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}\nu^t - \begin{pmatrix} \dot{v}_2^{n*} \\ g + \dot{v}_3^{n*} \end{pmatrix}.$$

If we weight the inputs equally using $W = I$ then we can compute the invariance controls

$$\nu_{inv}^t = \begin{pmatrix} \nu_{nom,1}^t \\ m\dot{v}_2^{n*} \\ -mg - m\dot{v}_3^{n*} \end{pmatrix} \qquad \nu_{1,inv}^t = \begin{pmatrix} \nu_{nom,1}^t \\ m\dot{v}_2^{n*} \\ \nu_{nom,3}^t \end{pmatrix} \qquad \nu_{2,inv}^t = \begin{pmatrix} \nu_{nom,1}^t \\ \nu_{nom,2}^t \\ -mg - m\dot{v}_3^{n*} \end{pmatrix}$$

where $\nu_{nom,i}^t$ denotes the ith component of the nominal control. The final control is then

$$\nu^t = (1-\sigma_1)(1-\sigma_2)\nu_{nom}^t + \sigma_1(1-\sigma_2)\nu_{1,inv}^t + (1-\sigma_1)\sigma_2\nu_{2,inv}^t + \sigma_1\sigma_2\nu_{inv}^t.$$

6.4 Inner-Loop Control

The inner-loop control is designed to track an orientation reference $\eta_d = (\phi_d,\, \theta_d,\, \psi_d)^T$ subject to bounds on roll and pitch. The error dynamics are

$$\ddot{\tilde{\eta}} = \dot{W}(\eta)\omega + W(\eta)J^{-1}\left(-\omega \times J\omega + \begin{pmatrix} z_M T_M b \\ z_M T_M a \\ x_T T_T - Q_M \end{pmatrix}\right) \tag{6.9}$$

where T_M and Q_M can be treated as known functions of time. Since the dynamics (6.9) have been written as a double integrator we can perform a simple feedback linearization

$$\nu^r = \dot{W}(\eta)\omega + W(\eta)J^{-1}\left(-\omega \times J\omega + \begin{pmatrix} z_M T_M b \\ z_M T_M a \\ x_T T_T - Q_M \end{pmatrix}\right). \tag{6.10}$$

The nominal stabilizing control is

$$\nu^r_{nom} = \ddot{\eta}_d - K^r_d\dot{\tilde{\eta}} - K^r_p\tilde{\eta}$$

where K^r_d and K^r_p are controller gains. The fictitious outputs for the constant bounds $\phi^*, \theta^* > 0$ are defined as

$$h_{\bar{\phi}} = \phi - \phi^* \qquad h_{\bar{\theta}} = \theta - \theta^* \tag{6.11a}$$

$$h_{\underline{\phi}} = -\phi - \phi^* \qquad h_{\underline{\theta}} = -\theta - \theta^* \tag{6.11b}$$

Since these outputs are relative degree two, we must apply Theorem 6.3. Considering $h_{\bar{\phi}}$ we can define the linearized coordinates $\zeta_{\bar{\phi},1} = \phi - \phi^*$ and $\zeta_{\bar{\phi},2} = \dot{\phi}$. We then have

$$\begin{aligned}\dot{\zeta}_{\bar{\phi},1} &= \zeta_{\bar{\phi},2} \\ &= -\alpha_{\bar{\phi},0}\zeta_1 + v_{\bar{\phi},1}\end{aligned}$$

where $v_{\bar{\phi},1} = \zeta_{\bar{\phi},2} + \alpha_{\bar{\phi},0}\zeta_{\bar{\phi},0}$. Thus for $\zeta_{\bar{\phi},1}(0) = \phi(0) - \phi^* \leq 0$ and $v_{\bar{\phi},1} \leq 0$ we are guaranteed to satisfy $h_{\bar{\phi}}$. Using the original coordinates and rearranging the inequality, this condition is

$$\dot{\phi} \leq -\alpha_{\bar{\phi},0}(\phi - \phi^*). \tag{6.12}$$

Equation (6.12) provides intuition regarding the effect of the α terms. In particular, a large value of $\alpha_{\bar{\phi},0}$ will cause the control to avoid the bound more aggressively. The complete set of relative degree one constraints are

$$v_{\bar{\phi},1} = \dot{\phi} + \alpha_{\bar{\phi},0}(\phi - \phi^*) \qquad v_{\bar{\theta},1} = \dot{\theta} + \alpha_{\bar{\theta},0}(\theta - \theta^*)$$

$$v_{\underline{\phi},1} = -\dot{\phi} - \alpha_{\underline{\phi},0}(\phi + \phi^*) \qquad v_{\underline{\theta},1} = -\dot{\theta} - \alpha_{\underline{\theta},0}(\theta + \theta^*)$$

where each expression must remain non-positive and the initial conditions must be compatible with (6.11). For an illustration of the control, suppose ϕ approaches ϕ^*

and θ approaches $-\theta^*$. Then we have

$$C = \begin{pmatrix} \dot{\phi} + \alpha_{\bar{\phi},0}(\phi - \phi^*) \\ -\dot{\theta} - \alpha_{\underline{\theta},0}(\theta + \theta^*) \end{pmatrix}$$

and

$$\dot{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \nu^r + \begin{pmatrix} \alpha_{\bar{\phi},0}\dot{\phi} \\ -\alpha_{\underline{\theta},0}\dot{\theta} \end{pmatrix}.$$

Thus from (6.6) we have

$$\nu^r_{inv} = \begin{pmatrix} -\alpha_{\bar{\phi},0}\dot{\phi} \\ -\alpha_{\underline{\theta},0}\dot{\theta} \\ \nu^r_{nom,3} \end{pmatrix} \qquad \nu^r_{1,inv} = \begin{pmatrix} -\alpha_{\bar{\phi},0}\dot{\phi} \\ \nu^r_{num,2} \\ \nu^r_{nom,3} \end{pmatrix} \qquad \nu^r_{2,inv} = \begin{pmatrix} \nu^r_{num,1} \\ -\alpha_{\underline{\theta},0}\dot{\theta} \\ \nu^r_{nom,3} \end{pmatrix}$$

and finally

$$\nu^r = (1-\sigma_1)(1-\sigma_2)\nu^r_{nom} + \sigma_1(1-\sigma_2)\nu^r_{1,inv} + (1-\sigma_1)\sigma_2\nu^r_{2,inv} + \sigma_1\sigma_2\nu^r_{inv}.$$

6.5 Simulation Results

The above design is verified in simulation by regulating the origin for the case of bounded translational velocity, roll, and pitch. In particular, we are interested in the behaviour when a reference approaches a state boundary. We choose the transition function

$$\sigma(\xi) = \frac{1}{1 + e^{-\frac{5}{\varepsilon}(\xi + \frac{\varepsilon}{2})}}$$

where ε is the distance from the boundary inside which the constraint becomes active. We can compute the inputs to the translational system by inverting (6.8) assuming $T_M > 0$ to obtain

$$\begin{pmatrix} \phi_d \\ \theta_d \\ T_M \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{c\psi_d\nu^t_2 - s\psi_d\nu^t_1}{\sqrt{(c\psi_d\nu^t_1 + s\psi_d\nu^t_2)^2 + (\nu^t_3)^2}}\right) \\ \arctan\left(\frac{c\psi_d\nu^t_1 + s\psi_d\nu^t_2}{\nu^t_3}\right) \\ \|\nu^t\| \end{pmatrix} \tag{6.13}$$

and the rotational inputs by inverting (6.10) which gives

$$\begin{pmatrix} a \\ b \\ T_T \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{z_M T_M} & 0 \\ \frac{1}{z_M T_M} & 0 & 0 \\ 0 & 0 & \frac{1}{x_T} \end{pmatrix} \left(\omega \times J\tilde{\omega} + Q_M e_3 + JW^{-1}(\eta)(\nu^r - \dot{W}(\eta)\omega)\right)$$

where $\dot{W}(\eta)$ is computed to be

$$\dot{W}(\eta) = \begin{pmatrix} 0 & c\phi t\theta\dot{\phi} + \frac{s\phi}{c^2\theta}\dot{\theta} & \frac{c\phi}{c^2\theta}\dot{\theta} - s\phi t\theta\dot{\phi} \\ 0 & -s\phi\dot{\phi} & -c\phi\dot{\phi} \\ 0 & \frac{c\phi\dot{\phi}+s\phi t\theta\dot{\theta}}{c\theta} & \frac{c\phi t\theta\dot{\theta}-s\phi\dot{\phi}}{c\theta} \end{pmatrix}.$$

In order to fully specify the reference for the rotational control we require two derivatives of η_d. Since we have defined ϕ_d and θ_d using a composition, we begin by computing the derivative of the first two components of (6.13). These derivatives will depend on $\dot{\nu}^t$ and $\ddot{\nu}^t$. The expressions for the derivatives of the nominal control are similar to (4.20) and (4.21). We must also include derivatives of the invariance control. The complete set of outputs for the translational control can be written

$$H^t = \begin{pmatrix} v^n - v^{n*} \\ -v^n - v^{n*} \end{pmatrix}$$

where we have used H rather than C since we are including more constraints than can be active simultaneously. We define

$$\dot{H}^t = \frac{1}{m}\begin{pmatrix} I \\ -I \end{pmatrix}\nu^t - \begin{pmatrix} \dot{v}^{n*} + ge_3 \\ \dot{v}^{n*} - ge_3 \end{pmatrix} \triangleq A_H^t\nu^t + B_H^t.$$

If we let A^t (B^t) be the rows of A_H^t (B_H^t) corresponding to the active constraints, then when the invariance controller is applied we can compute its derivatives to be

$$\dot{\nu}_{inv}^t = (I - (A^t)^\dagger A^t)\dot{\nu}^t - (A^t)^\dagger \dot{B}^t$$
$$\ddot{\nu}_{inv}^t = (I - (A^t)^\dagger A^t)\ddot{\nu}^t - (A^t)^\dagger \ddot{B}^t$$

where we have used the fact that in this case A^t is constant. Finally, we treat the derivative of σ_i as negligible in order to prevent the expressions from becoming intractable.

The simulation results are shown in Figure 6.1. The controller gains are given in Table 6.1 and the physical parameters are given in Table A.1. The initial conditions are $p_0 = (-40, 40, 10)^T$, $v_0 = \eta_0 = \omega_0 = 0$, and the gains used for computing the rotational boundaries were $\alpha_{i,0} = 3$ where $i \in \{\bar{\phi}, \underline{\phi}, \bar{\theta}, \underline{\theta}\}$. The simulation time was chosen to focus on the system behaviour while constraints were active. Therefore, it does not show complete convergence. However, once the system is operating within all boundaries the control becomes standard and is known to work. Figure 6.1a shows the position trajectory. This trajectory possesses a linear region due to the saturation of the velocity. In Figure 6.1b we remark that the velocity bounds have been violated. This overshoot is due to the inner-outer approximation.

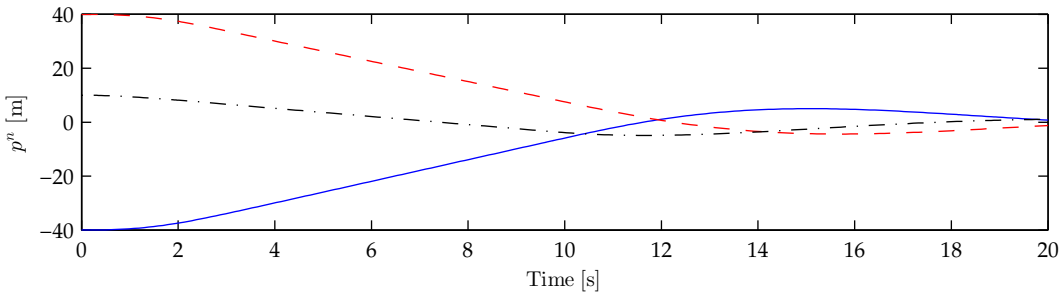
K_p^r	diag(20, 20, 14) s^{-2}
K_d^r	$10I$ s^{-1}
K_p^t	$2I$ $\mathrm{kg/s}^2$
K_d^t	$4.5I$ kg/s

Table 6.1: Controller gains used for simulation.

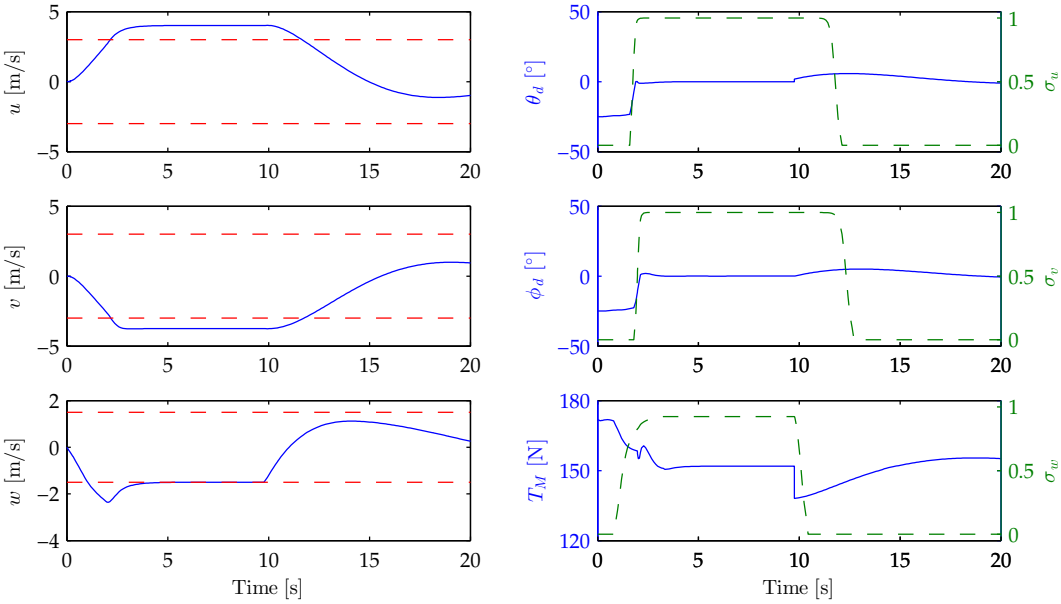
The control satisfies the velocity constraints by reducing the acceleration to zero as the velocity approaches the boundary. In the plots to the right, the roll and pitch references go to zero but the tracking error in the rotational dynamics requires time to converge. This effect can be reduced by increasing the rotational controller gains. However, in practice the overshoot is minimal and satisfies the design objective of preventing the velocity from increasing proportionally to the initial position error. Most importantly, the effect of the inner-outer approximation error is transient and not destabilizing. Figure 6.1c shows the roll and pitch trajectories as well as the flapping angle inputs. Since the inner-loop controller has direct control of the control inputs, the constraints can be satisfied absolutely.

6.6 Conclusions

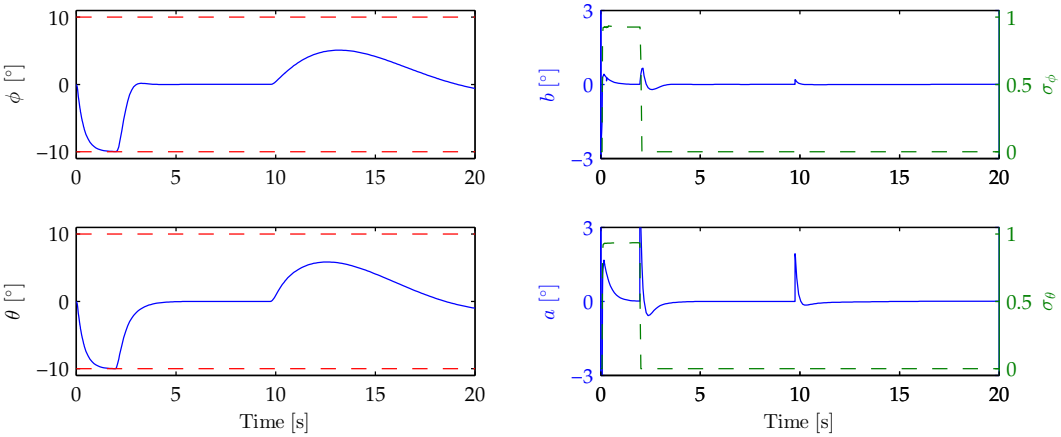
The design proposed in this chapter is motivated by practical experience gained during autopilot testing. It provides the relative implementational simplicity of setpoint regulation (as opposed to trajectory generation) and allows the operator to configure velocity and orientation bounds. These bounds prevent the respective states from becoming proportionally large due to initial condition error. For testing purposes bounds could be used to prevent overturn or otherwise unsafe roll-pitch as well as high velocity which may impede the pilot's ability to recover control of the aircraft in an emergency. From an application point of view, especially for any type of survey or inspection task, limiting translational velocity and roll-pitch would be very useful in practice. The simulation results show that the design functions correctly and is appropriate for practical implementation.



(a) Position trajectory showing linear region where the velocity is saturated. The solid line is in the e_1 direction, the dashed line in the e_2 direction, and the dash-dot line is in the e_3 direction.



(b) Velocity trajectory and respective control inputs. In the left plots the dashed line shows the bounds and on the right plot the dashed line shows the transition functions.



(c) Roll and pitch trajectories and respective inputs. The dashed lines in the left plots show the state constraints and the dashed lines in the right plots show the transition functions.

Figure 6.1: State response for simulated setpoint regulation subject to state constraints in translational velocity, roll, and pitch.

Chapter 7

Conclusions and Future Work

Experimental helicopter UAV control research presents a challenging control problem due to nonlinear, underactuated dynamics and platform development. Chapter 2 presents our hardware configuration and gives details regarding decisions made to address problems which arose during field tests. Additionally, details are provided regarding our software framework which is developed with an emphasis on modularity and error handling. This design facilitates control research by allowing parts of the code to be easily replaced. Another benefit of the modularity was realized when hardware changes became necessary. For instance, both the servo switch and IMU were replaced. The only code modifications necessary to interface with the new hardware were inside the relevant classes. Since this new code provided the same logical functionality, compatibility with the rest of the system was ensured. Correct functionality of the platform is demonstrated by extensive flight testing. The autopilot software is available to the community for download. Customizing QGroundControl was another aspect of the development which required a significant time investment. This effort also provided clear benefit during experimental testing. Outdoor helicopter field work is subject to constraints such as weather and daylight as well as pilot fatigue and vehicle maintenance. Therefore, once the system is operating correctly and there is an opportunity to perform experiments, it is critical to work efficiently. We found that the ability to monitor the autopilot status and to adjust controller gains without landing the helicopter enabled us to achieve the successful tests.

The helicopter dynamics is presented in Chapter 3. The actuation of a small helicopter is well established in the literature. However, this modelling usually assumes direct control of the main and tail rotor thrusts and the main rotor flapping angles. Therefore, it is necessary to establish a relationship between these inputs and the physically-controllable inputs. In particular, the thrust and countertorque models available in the literature suffer from intractability due to complexity. We derive a simplification of these general models based on experimental data. We

remark that it is not necessary to use a model based on physics in order to apply control laws in practice. Many of the system parameters will multiply the controller gains which means they do not require identification. In addition, the helicopter will often operate in a region where the model is approximately linear allowing PID to effectively stabilize the system. However, when significant terms in the model can be included in the control it will allow better performance since these effects can be compensated directly in feedforward rather than creating a disturbance to be countered by feedback. For example, the velocity terms in the thrust model are practically significant and if they can be accounted for in the control they will allow for better performance, especially for tracking aggressive trajectories.

In Chapter 4 we derive a PID control law which is justified by the model and intended for platform validation. This control is applied using errors computed in the body-fixed frame. This choice of frame is made to allow the operator to choose gains intuitively. However, for disturbances such as wind which have a constant direction in the navigation frame it is better to integrate the position error in the navigation frame as is done in Chapter 5. The tradeoff occurs between compensating effects such as the SBF which have a constant direction in $\mathcal{B}$ and effects such as wind. In Chapter 5 we derive a control that compensates the coupling which occurs when the translational dynamics are forced by the rotational inputs. In particular, this control accounts for the SBF due to the tail rotor which is usually present due to the main rotor countertorque. Both designs are validated in experiment for hover as well as time-varying trajectory flights. The experimental results presented in Chapters 4 and 5 show good tracking performance. However, comparing these results is misleading since the intention of the experiment was to demonstrate correct operation of the control. This intention is in contrast to designing an experiment for comparing the SBF design to PID. Probable reasons for differences in performance between the controls include tuning variations, and rapidly changing wind conditions during experimental testing. Indeed, many obstacles to acquiring experimental results exist such as crashes and adverse environmental conditions endemic to northern Canada. Ultimately, the helicopter and ground station were transported to Bergen R/C Helicopters in Cassopolis, MI in the fall of 2012 for repairs and testing. Although these tests were successful, there was no further opportunity to extend the results once the data had been analyzed. Although these controls were designed with the derivatives of the reference incorporated into the feedback, the experiments were performed assuming a constant reference. This simplification was made for simplicity and it is expected that including the derivatives will improve performance, particularly for curved trajectories. Additionally, it was assumed for the design that the reference trajectories were of class C^4. However, in practice it is expected that this differentiability condition need only hold piecewise. Indeed, based on the experimental

results were reasonable tracking was observed without including derivatives of the reference anywhere, good performance can be expected by eliminating the reference derivative terms only at isolated points.

Experience gained during experimental work provided the motivation for the design in Chapter 6 which satisfies bounds on translational velocity as well as roll and pitch. This control is expected to be useful for experimental testing. For instance, an inner-loop control which satisfies roll and pitch bounds can guarantee the helicopter will not rotate beyond a recoverable limit during tests of an outer-loop control. This example also provides justification for the hierarchical approach used for control design as well as the modularity of the autopilot software. The simulation results of this control show that the inner-outer approximation causes the translational velocity bounds to be violated. However, this error can be decreased by tuning the rotational controller. In addition, it does not prevent the translational control from achieving its objective. In practice, this error is expected to be insignificant and still guarantees the velocity does not grow proportionally to the initial condition error.

Future work includes testing the control using the thrust inputs as well as the cyclic. To enhance the safety of future tests the constrained control should be included, particularly for the inner-loop which will help prevent a poorly tuned outer-loop from overturning the helicopter. Additionally, experimental work is envisioned which uses trajectories designed to excite the SBF. For example, an S shaped curve provides a trajectory where the velocity continuously changes in direction and magnitude. Whereas, a circle requires a constant change in direction only. This trajectory would be essentially half of a figure-8. While a figure-8 provides a good demonstration of the SBF for simulation, in practice it requires the pilot to be able to recover the aircraft while it is facing nose-in. This orientation is challenging for pilots during manual flights. Therefore, the flight tests should be designed with the helicopter facing away from the pilot in order to provide a configuration which facilitates recovery in an emergency.

Bibliography

[1] B. Mettler, *Identification Modeling and Characteristics of Miniature Rotorcraft*. Norwell, MA: Kluwer Academic Publishers, 2003.

[2] V. Gavrilets, "Autonomous Aerobatic Maneuvering of Miniature Helicopters," Ph.D. dissertation, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2003.

[3] I. A. Raptis and K. P. Valavanis, *Linear and Nonlinear Control of Small-Scale Unmanned Helicopters*, ser. Intelligent Systems, Control, and Automation: Science and Engineering. London, UK: Springer Science+Business Media Inc., 2011.

[4] D. R. Kastelan, "Design and Implementation of a GPS-aided Inertial Navigation System for a Helicopter UAV," Master's thesis, Dept. of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, 2009.

[5] M. Barczyk, "Nonlinear State Estimation and Modeling of a Helicopter UAV," Ph.D. dissertation, Dept. of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, 2012.

[6] F. Kendoul, "Survey of Advances in Guidance, Navigation, and Control of Unmanned Rotorcraft Systems," *Journal of Field Robotics*, vol. 29, pp. 315–378, March/April 2012.

[7] H. Shim, "Hierarchical Flight Control System Synthesis for Rotorcraft-based Unmanned Aerial Vehicles," Ph.D. dissertation, Dept. of Mechanical Engineering, University Of California, Berkeley, CA, 2000.

[8] P. Abbeel, "Apprenticeship Learning and Reinforcement Learning with Application to Robotic Control," Ph.D. dissertation, Stanford University, Stanford, CA, 2008.

[9] B. Ahmed, "Autonomous landing of lightweight helicopters on moving platforms such as ships," Ph.D. dissertation, The University of New South Wales at The Australian Defence Force Academy, 2009.

[10] G. Cai, B. M. Chen, and T. H. Lee, *Unmanned Rotorcraft Systems*, ser. Advances in Industrial Control. London, UK: Springer-Verlag, 2011.

[11] T. J. Koo and S. Sastry, "Output Tracking Control Design of a Helicopter Model Based on Approximate Linearization," in *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, FL, 1998, pp. 3635–3640.

[12] R. Mahony and T. Hamel, "Robust trajectory tracking for a scale model autonomous helicopter," *International Journal of Robust and Nonlinear Control*, vol. 14, pp. 1035–1059, August 2004.

[13] P. Castillo, R. Lozano, and A. E. Dzul, *Modelling and Control of Mini-Flying Machines*. London, UK: Springer-Verlag, 2005.

[14] E. Frazzoli, M. A. Dahleh, and E. Feron, "Trajectory tracking control design for autonomous helicopters using a backstepping algorithm," in *Proceedings of the American Control Conference, 2000*, Chicago, IL, 2000, pp. 4102–4107.

[15] S. P. Bhat and D. S. Bernstein, "A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon," *Systems & Control Letters*, vol. 39, pp. 63–70, January 2000.

[16] L. Marconi and R. Naldi, "Robust Full Degree-of-Freedom Tracking Control of a Helicopter," *Automatica*, vol. 43, pp. 1909–1920, November 2007.

[17] I. A. Raptis, K. P. Valavanis, and W. A. Moreno, "A Novel Nonlinear Backstepping Controller Design for Helicopters Using the Rotation Matrix," *IEEE Transactions on Control Systems Technology*, vol. 19, pp. 465–473, March 2011.

[18] H. J. Kim, D. H. Shim, and S. Sastry, "Nonlinear Model Predictive Tracking Control for Rotorcraft-Based Unmanned Aerial Vehicles," in *Proceedings of the American Control Conference, 2002*, Anchorage, AK, 2002, pp. 3576–3581.

[19] A. Isidori, L. Marconi, and A. Serrani, "Robust Nonlinear Motion Control of a Helicopter," *IEEE Transactions on Automatic Control*, vol. 48, pp. 413–426, March 2003.

[20] F. Mazenc, R. E. Mahony, and R. Lozano, "Forwarding control of a scale model autonomous helicopter: A Lyapunov control design," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, HI, 2003, pp. 3960–3965.

[21] R. W. Prouty, *Helicopter Performance, Stability, and Control*. Malabar, FL: Krieger Publishing Company, 1986.

[22] K. Kondak, C. Deeg, G. Hommel, M. Musial, and V. RemuB, "Mechanical Model and Control of an Autonomous Small Size Helicopter with a Stiff Main Rotor," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 2469–2474.

[23] B. Ahmed and F. Kendoul, "Flight control of a small helicopter in unknown wind conditions," in *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, GA, December 2010, pp. 3536–3541.

[24] K. Danapalasingam, J. Leth, A. la Cour-Harbo, and M. Bisgaard, "Robust Helicopter Stabilization in the Face of Wind Disturbance," in *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, GA, December 2010, pp. 3832–3837.

[25] T. Cheviron, F. Plestan, and A. Chriette, "A robust guidance and control scheme of an autonomous scale helicopter in presence of wind gusts," *International Journal of Control*, vol. 82, pp. 2206–2220, December 2009.

[26] S. K. Kannan, "Adaptive Control of Systems in Cascade with Saturation," Ph.D. dissertation, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 2005.

[27] F. Kendoul, I. Fantoni, and R. Lozano, "Asymptotic Stability of Hierarchical Inner-Outer Loop-Based Flight Controllers," in *17th IFAC World Congress*, Seoul, KR, 2008, pp. 1741–1746.

[28] M. Bisgaard, "Modeling, Estimation, and Control of Helicopter Slung Load System," Ph.D. dissertation, Dept. of Electronic Systems Section for Automation and Control, Aalborg University, Aalborg, DK, 2007.

[29] R. D. Garcia, "Designing an Autonomous Helicopter Testbed: From Conception Through Implementation," Ph.D. dissertation, Dept. of Computer Science and Engineering, University of South Florida, Tampa, FL, 2008.

[30] ANCL/autopilot - Github. Accessed 5 June 2013. [Online]. Available: http://github.com/ANCL/Autopilot

[31] V. Gavrilets, E. Frazzoli, B. Mettler, M. Piedmonte, and E. Feron, "Aggressive Maneuvering of Small Autonomous Helicopters: A Human-Centered Approach," *The International Journal of Robotics Research*, vol. 20, pp. 795–807, October 2001.

[32] M. Bürger and M. Guay, "Robust Constraint Satisfaction for Continuous-Time Nonlinear Systems in Strict Feedback Form," *IEEE Transactions on Automatic Control*, vol. 55, pp. 2597–2601, November 2010.

[33] ——, "A Backstepping Approach to Multivariable Robust Constraint Satisfaction With Application to a VTOL Helicopter," in *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, 2009, pp. 5239–5244.

[34] B. Godbolt, N. Vitzilaios, and A. F. Lynch, "Design and Validation of a Helicopter UAV Autopilot," in *Proceedings of the 2012 International Conference on Unmanned Aircraft Systems*, Philadelphia, PA, 2012.

[35] B. Godbolt, N. I. Vitzilaios, and A. F. Lynch, "Experimental Validation of a Helicopter Autopilot Design using Model-Based PID Control," *Journal of Intelligent & Robotic Systems*, vol. 70, pp. 385–399, 2013.

[36] B. Godbolt and A. F. Lynch, "Physical Input Modelling and Identification for a Helicopter UAV," in *Proceedings of the 2013 International Conference on Unmanned Aircraft Systems*, Atlanta, GA, 2013, pp. 879–885.

[37] ——, "Control-Oriented Physical Input Mapping for a Helicopter UAV," *Journal of Intelligent & Robotic Systems*, 2014, to appear.

[38] ——, "Model-Based Helicopter UAV Control: Experimental Results," *Journal of Intelligent & Robotic Systems*, 2014, to appear.

[39] B. Godbolt, N. Vitzilaios, C. Bergen, and A. F. Lynch, "Experimental Validation of a Helicopter Autopilot: Time-Varying Trajectory Tracking," in *Proceedings of the 2013 International Conference on Unmanned Aircraft Systems*, Atlanta, GA, 2013, pp. 391–396.

[40] B. Godbolt and A. F. Lynch, "A Novel Cascade Controller for a Helicopter UAV with Small Body Force Compensation," in *Proceedings of the American Control Conference, 2013*, Washington, DC, 2013, pp. 802–807.

[41] ——, "An Unmanned Aerial Vehicle (UAV) Helicopter Control with Small Body Force Compensation: Experimental Results," *Automatica*, 2013, submitted.

[42] M. Bernard, K. Kondak, and G. Hommel, "Framework for Development and Test of Embedded Flight Control Software for Autonomous Small Size Helicopters," in *Embedded Systems - Modeling, Technology, and Applications*. Springer-Verlag, 2006, pp. 159–168.

[43] J. del Cerro, A. Barrientos, J. Artieda, E. Lillo, P. Gutierez, and R. S. Martin, "Embedded Control System Architecture applied to an Unmanned Aerial Vehicle," in *IEEE International Conference on Mechatronics*, Budapest, July 2006, pp. 254–259.

[44] QNX Software Systems Online Infocenter. Accessed 5 June 2013. [Online]. Available: http://www.qnx.com/developers/docs/6.5.0/index.jsp

[45] D. H. Shim, H. J. Kim, and S. Sastry, "Hierarchical Control System Synthesis for Rotorcraft-Based Unmanned Aerial Vehicles," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, 2000.

[46] Boost 1.46.1 Library Documentation. Accessed 5 June 2013. [Online]. Available: http://www.boost.org/doc/libs/1_46_1/

[47] Git - Fast Version Control System. Accessed 5 June 2013. [Online]. Available: http://git-scm.com/

[48] Doxygen. Accessed 5 June 2013. [Online]. Available: http://doxygen.org

[49] RAPIDXML Manual. Accessed 5 June 2013. [Online]. Available: http://rapidxml.sourceforge.net/manual.html

[50] QGroundControl. Accessed 5 June 2013. [Online]. Available: http://qgroundcontrol.org/

[51] Micro Air Vehicle Communication Protocol. Accessed 5 June 2013. [Online]. Available: http://qgroundcontrol.org/mavlink/start

[52] J. C. van der Ha and M. D. Shuster, "A Tutorial on Vectors and Attitude [Focus on Education]," *IEEE Control Systems Magazine*, vol. 29, no. 2, pp. 94–107, April 2009.

[53] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.

[54] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch, "Rigid-Body Attitude Control: Using Rotation Matrices for Continuous, Singularity-Free Control Laws," *IEEE Control Systems Magazine*, vol. 31, no. 3, pp. 30–51, June 2011.

[55] M. D. Shuster, "A Survey of Attitude Representations," *The Journal of the Astronautical Sciences*, vol. 41, pp. 439–517, October 1993.

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

[57] S. K. Kim, "Modeling, identification, and trajectory planning for a model-scale helicopter," Ph.D. dissertation, University of Michigan, Ann Arbour, MI, 2001.

[58] M. J. van Nieuwstadt and R. M. Murray, "Outer Flatness: Trajectory Generation for a Model Helicopter," in *Proceedings of the European Control Conference 1997 (CD-ROM)*, Brussels, BE, 1997.

[59] R. Kelly, "A tuning procedure for stable PID control of robot manipulators," *Robotica*, vol. 13, pp. 141–148, March 1995.

[60] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, NY: Cambridge University Press, 1985.

[61] P. Tsiotras, "Further Passivity Results for the Attitude Control Problem," *IEEE Transactions on Automatic Control*, vol. 43, pp. 1597–1600, November 1998.

[62] B. Ahmed, H. R. Pota, and M. Garratt, "Flight control of a rotary wing UAV using backstepping," *International Journal of Robust and Nonlinear Control*, vol. 20, pp. 639–658, April 2010.

[63] D. E. Koditschek, "Application of a New Lyapunov Function to Global Adaptive Attitude Tracking," in *Proceedings of the 27th IEEE Conference on Decision and Control*, Austin, TX, 1988, pp. 63–68.

[64] F. Bullo and R. M. Murray, "Tracking for fully actuated mechanical systems: a geometric framework," *Automatica*, vol. 35, pp. 17–35, 1999.

[65] K. Hoffman and R. Kunze, *Linear Algebra*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1971.

[66] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, "Quaternion-Based Hybrid Control for Robust Global Attitude Tracking," *IEEE Transactions on Automatic Control*, vol. 56, pp. 2555–2566, November 2011.

[67] O. Cornea, G. Lupton, J. Oprea, and D. Tanré, *Lusternik-Schnirelmann Category*, ser. Mathematical Surveys and Monographs. American Mathematical Society, 2003, vol. 103.

[68] N. Iwase, M. Mimura, and T. Nishimoto, "Lusternik-Schnirelmann category of non-simply connected compact simple Lie groups," *Topology and its Applications*, vol. 150, pp. 111–123, May 2005.

[69] J. B. Kuipers, *Quaternions and Rotation Sequences*. Princeton, NJ: Princeton University Press, 1999.

[70] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature*, ser. Graduate Texts in Mathematics. New York, NY: Springer-Verlag, 1997, vol. 176.

[71] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[72] R. V. Patel and M. Toda, "Quantitative Measures of Robustness for Multivariable Systems," in *Proceedings of the Joint Automatic Control Conference, 1980*. American Automatic Control Council, 1980, pp. TP8–A.

[73] M.-D. Hua, T. Hamel, P. Morin, and C. Samson, "Introduction to Feedback Control of Underactuated VTOL Vehicles," *IEEE Control Systems Magazine*, vol. 33, pp. 61–75, February 2013.

[74] R. Marino and P. Tomei, *Nonlinear Control Design: Geometric, Adaptive and Robust*. Hertfordshire, UK: Prentice Hall, 1995.

[75] J. M. Lee, *Introduction to Smooth Manifolds*, ser. Graduate Texts in Mathematics. Springer Science+Business Media Inc., 2006, vol. 218.

[76] M. Sobotka, J. Wolff, and M. Buss, "Invariance controlled balance of legged robots," in *Proceedings of the European Control Conference 2007*, 2007.

Appendix A

Model Parameters

Symbol	Value	Units	Description
			ANCL Helicopter and Environmental Parameters
J_x	0.36	kg m^2	Moment of inertia about e_1' axis
J_y	1.48	kg m^2	Moment of inertia about e_2' axis
J_z	1.21	kg m^2	Moment of inertia about e_3' axis
m	15.5	kg	Mass
ρ	1.2	kg/m^3	Air density
z_M	0.32	m	Main rotor hub vertical offset from center of mass
x_T	1.06	m	Tail rotor hub horizontal offset from center of mass
g	9.81	m/s^2	Acceleration due to gravity
σ_M	0.04	m	Linearized countertorque model coefficient
			Main Rotor Parameters
R_M	0.89	m	Rotor disk radius
N_b	2		Number of blades
a_M	6.6		Lift curve slope
c_M	0.066	m	Chord length
C_D	0.005		Coefficient of drag
α_M	-3490	μs/rad	Slope of $\Theta_M \mapsto \delta_M$ map
β_M	1860	μs	Intercept of $\Theta_M \mapsto \delta_M$ map
k_p	0.10	rad	$\delta_p \mapsto a$ gain
k_r	0.013	rad	$\delta_r \mapsto b$ gain
			Tail Rotor Parameters
R_T	0.175	m	Rotor disk radius
a_T	6.4		Lift curve slope
c_T	0.0325	m	Chord length

Symbol	Value	Units	Description
α_T	-1590	μs/rad	Slope of $\Theta_T \mapsto \delta_T$ map
β_T	1570	μs	Intercept of $\Theta_T \mapsto \delta_T$ map

Table A.1: Model parameters for the ANCL Helicopter.