Developing a Framework for Control of Agile Aircraft Platforms in Autonomous Hover

Kyle J. Krogh

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

Master of Science in Aeronautics & Astronautics

University of Washington

2009

Program Authorized to Offer Degree: Aeronautics & Astronautics

University of Washington
Graduate School

This is to certify that I have examined this copy of a master's thesis by

Kyle J. Krogh

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____
Kristi A. Morgansen

_____
Juris Vagners

Date: _____

University of Washington

Abstract

Developing a Framework for Control of Agile Aircraft Platforms in Autonomous Hover

Kyle J. Krogh

Chair of the Supervisory Committee:
Professor Kristi A. Morgansen
Aeronautics & Astronautics

This thesis extends previous work on autonomous agile aircraft to develop a framework for control in both linear and non-linear flight profiles. Two dynamical systems are considered, a scale YAK-54 aerobatic remote control aircraft and the Flexrotor concept developed by Aerovel. Both models are capable of hover and level flight, requiring complex control transitions between the flight regimes. Linear controllers are developed for both models in a hover configuration. Open-loop control sequences for the transitions from hover to level flight and back are discussed. All controllers are optimized, and robustness considered; additionally, a procedure for providing near-optimal non-linear control sequences is proposed. Finally, a general framework for generating control laws of agile aircraft systems is put forth. Simulation results are shown to support the theoretical analysis.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

The author wishes to express sincere appreciation his parents, Sue Combs and Charlie Krogh, for their words of encouragement, and to Kristi Morgansen and Juris Vagners for advising throughout the process. Additonally, the author wishes to thank Tad McGeer and Kris Gauksheim of Aerovel for their assistance and generous support.

DEDICATION

To all those who made this thesis possible.

DISCLAIMER

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or U.S. Government.

Chapter 1
INTRODUCTION

Control theory is the study of dynamical systems and generating inputs. Types of dynamical systems include thermal, electrical, and mechanical systems. Mechanical systems are of specific concern, as unmanned aerial vehicles (UAVs) have become increasingly prevalent and important in the twenty-first century. With the growth of UAVs, new flight regimes are possible that require the application of linear and nonlinear control theory for automatic input generation. The topic of this thesis is the design of controllers for agile aircraft systems and the creation of a broad framework applicable to this class of problems.

1.1 Motivation

The rapid development of unmanned aerial vehicles has been made possible, in part, by recent advances in communication, computation, and sensing. The demand for unmanned systems has encouraged growth as well, and UAVs have numerous military and civil applications. Military applications include reconnaissance, surveillance, intelligence, communications, and tactical strike of ground targets, while civil applications include research, weather forecasting, border patrol, and others.

Unmanned aircraft also offer a unique set of advantages when compared to piloted aircraft. Without a pilot, the aircraft can be made smaller and lighter weight by eliminating life support systems. Unmanned systems may also be designed around specific missions and maneuvers that are incompatible with a human pilot on board the aircraft, such as extended hover in fixed-wing aircraft. By eliminating the pilot, systems can be made smaller, allowing them to operate in tight spaces, to be man

portable, or to enjoy unique configurations.

Recent interest has been focused on UAVs capable of sustained hover, allowing a unique set of capabilities. Sustained hovering flight in a vertical orientation is termed thrust-borne flight, as the propulsion system counteracts gravity to keep the aircraft aloft. Traditional flight with lift produced by the wings is termed wing-borne flight. Systems capable of both thrust-borne flight and wing-borne flight strike a unique balance between the endurance of fixed-wing aircraft and the ability to operate in confined spaces of rotary-wing aircraft. Consider the task of observation and recon-

Figure 1.1: A ScanEagle UAV and the SkyHook Retrieval System [28].

naissance in an urban environment. An agile UAV could employ a belly camera able to track a target in wing-borne and thrust-borne flight, and in an urban environment could track while in thrust-borne flight. Alternatively, a UAV could employ thrust-borne flight as an object avoidance maneuver while operating in an urban environment, or any other similarly constrained environment. Lastly, a thrust-borne capability allows for vertical take-off and landing, further extending the operating envelope of agile UAVs. The Insitu/Boeing ScanEagle demonstrates the potential for

the concept of non-traditional launch and recovery methods via its SkyHook Retrieval System, as shown in Figure 1.1.

1.2 Problem Statement

The goal of this thesis is to address the control of two agile autonomous systems, a scale YAK-54 (see Figure 1.2) and the Aerovel Flexrotor (see Figure 1.3). Both systems are capable of wing-borne and thrust-borne flight, though they differ significantly in design. The application of previous work, combined with the analysis of both systems, provides the basis for a framework describing control methods of agile autonomous systems.

Figure 1.2: A YAK-54 in hover in the Real Flight RC Simulator [24].

Existing theory applies to various iterations of fixed-wing aircraft capable of autonomous hover, such as the YAK-54, but the design of the Flexrotor presents a new approach to agile UAVs and the controls required. Recent work has been conducted into the various aspects of thrust-borne flight and transition to wing-borne flight, and this thesis attempts to build upon those efforts.

The main challenge in these problems is controlling aircraft attitude and position

while in thrust-borne flight. Initially, controllers are designed with the assumption of full-state, non-noisy feedback, but the final control laws for each system should be robust against system and measurement noise, and environmental uncertainty, such as wind effects.

The proposed framework for agile UAVs should be applicable to a wide range of agile aircraft. Additionally, control methods for both linear and non-linear flight regimes should be identified. Linear controls should be optimal under the guidelines of LQ theory, while non-linear flight regimes should be near-optimal for the desired maneuver. Various machine-learning schemes can be used to ensure near-optimality. All proposed control methods should be robust to noise and disturbances as well.

1.3 Previous Work

The general class of problems considered here combines elements of linear control and non-linear control. Specific examples of both control elements have been previously studied. Existing work on both linear control and non-linear control for various agile autonomous systems will be summarized in this section.

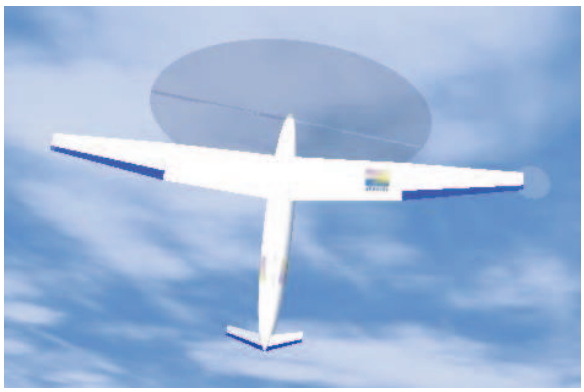

Figure 1.3: The Aerovel Flexrotor concept.

1.3.1 Linear Control Methods for Hover

The flight regimes that can be reasonably approximated as linear in behavior comprise two different categories, thrust-borne and wing-borne flight. Aircraft behavior while in wing-borne flight has been extensively studied and is cataloged by Stevens [25], McCormick [16], Yechout [30], and others. The case of thrust-borne flight represents the general class of problems addressed here and is motivated by various attempts and approaches to controlling an agile aircraft in a hover, or thrust-borne, configuration.

In the work by Green [9], a micro-air-vehicle (MAV) is considered, and simplified dynamics are proposed to model a MAV in transition from level flight to hover. While the MAV is in hovering flight, the thrust must balance the aircraft weight and drag forces, which are expected to be small. The dynamics fail to consider the issue of controlling the aircraft while in hover, or how to account for rotational effects due to propeller inertia.

A novel tail-sitter UAV design proposed by Stone [27] used multi-disciplinary optimization schemes. Subsequent work [26, 6] developed a hover control model using Linear Quadratic Regulator (LQR) techniques to stabilize the vehicle in the presence of prescribed wind gusts. The four low-level controllers that make up the vertical flight controller are a velocity elevator controller, a velocity rudder controller, a vertical roll rate controller, and a vertical velocity throttle controller. The controller is then incorporated directly into the optimization problem.

In a related work [8], a small foam agile autonomous aircraft is considered that relies on motion capture cameras for state sensing and computes all controls off-board. The hover controller uses full state feedback from the sensing system, and controller gains are optimized using linear quadratic control techniques. Large penalties are assigned to pitch, roll, and their respective derivatives to ensure that the vehicle maintains thrust-borne flight. Small penalties on the position deviations prevent the vehicle from making sudden movements should the position change. A large penalty

is used on altitude deviation, ensuring that the airplane can accurately maintain a prescribed altitude. As foam airplanes are low in structural rigidity, airframe twist is a concern and is mitigated by focusing on minimizing rapid changes in control surface position.

The same experimental setup is used in [21] to investigate a controls framework based on logic to operate in the various flight regimes of an agile aircraft. Using the motion capture system, local state-feedback stabilizers are proposed for set-point hovering flight and for set-point hovering flight at angle θ. The stabilizers are designed following optimal LQR techniques.

A different approach was explored by Johnson [10], in which the use of dynamic inversion with neural network adaptation was considered. The adaptive controller capable of transitioning to and from hovering flight in a nearly stationary position was applied to the GTEdge UAV. The controller was validated with eight successful transitions from forward horizontal flight to hover and back in varying atmospheric conditions, though lateral and rotational drifts occurred while in hover.

These results have demonstrated a wide approach to controlling an agile aircraft in hover configuration, though linear quadratic based approaches are favored. Less focus has been paid to controlling the translation or rotation of an aircraft in hover, or to the use of reduced order controllers.

1.3.2 Non-Linear Control Methods

Transferring an agile aircraft from one flight regime to another and back poses significant difficulties, as the aircraft transits highly non-linear ranges of its dynamics. In [8], transitions were performed in essentially an open-loop manner. Transitioning to level flight was performed by an instant movement of the reference way point to a distant location combined with a 5% reduction in throttle. The transition to hover was performed in a prescribed pitch-up maneuver, with the ailerons controlled by a feedback loop to keep the aircraft on the predefined trajectory. The transitions incor-

porated into the hybrid control scheme developed by Naldi [21] are again open loop, but are based upon commanded human pilot inputs from a large sequence of flight tests.

Taking a different approach, the transition controllers implemented by Johnson [10] involved the use of a trajectory generator. Two different transitions were considered for each transition, one slower and one faster. The slower transition involved a gradual and linear ramping of the commanded pitch angle to 90 degrees combined with a decrease in commanded velocity over a ten second period. The faster scheme involved a step change in pitch angle to 120 degrees and then back to 90 degrees while the commanded velocity decreased to zero in only five seconds. The transitions back to level flight were essentially inverses of these two schemes.

While open loop controllers are popular for their ease of use and simplicity, optimizing the performance of such controllers is much more challenging than optimizing linear feedback controllers. In the attempt to provide measures of optimality, machine learning approaches have been developed, investigated, and tested. Kearns and Singh [11] presented a reinforcement learning (RL) algorithm to find near-optimal returns in Markov decision processes (MDP). They felt that previous work lacked provable bounds on the resources required to learn an optimal (or near-optimal) policy. The algorithm they developed, known as E^3 for Explicit Explore or Exploit, provided guarantees on the resources and time necessary to find a near-optimal policy through a mix of balanced wandering and exploitation of areas within the state space.

While E^3 was a significant step forward, Abbeel and Ng [4] felt that the bias on exploration made E^3 incompatible with a real system. To focus more on exploitation, they developed a three step algorithm referred to as apprenticeship learning. First, a teacher demonstrates a policy, and all state-action trajectories are recorded. Second, using any RL algorithm and all state-action trajectories in the training set, a near-optimal policy is found for the system. Finally, the near-optimal policy is compared to the teacher's policy. If the performance meets or exceeds the teacher's, the algorithm

is finished; otherwise, the trajectories are added to the state-action trajectories from the unsuccesful test to the training step and the algorithm loops back to the second step.

This approach to reinforcement learning was subsequently applied to an autonomous aerobatic helicopter [2, 3], which represented the first successful autonomous completion of a forward flip, sideways roll, tail-in funnel and nose-in funnel for a helicopter. They first used a pilot to fly the helicopter to generate a dynamics model and cost function, then applied the apprenticeship learning algorithm to develop a near-optimal policy to perform the difficult non-linear aerobatic maneuvers. This state-of-the-art work in autonomous helicopter flight presents an intriguing approach to the non-linear transitions required to navigate between thrust-borne and wing-borne flight for an agile autonomous aircraft and is a step above and beyond simple open-loop control sequences.

1.4 Contribution

The main contributions of this work are as follows. Thrust-borne controllers are developed for the YAK-54 and Aerovel Flexrotor aircraft via linearization and decoupling of the resulting linear systems. Linear quadratic control techniques are used, with the addition of integral terms in certain cases. Control methods for autonomous transitions between wing-borne and thrust-borne flight regimes are also discussed. The problem of reduced order control is contemplated for the YAK-54 with recommendations for a sensing/control suite to be used for hardware implementation and flight testing.

Additionally, a general framework for control of agile aircraft capable of sustained autonomous hover is proposed based on the developed controllers. Simulation results indicate the satisfactory performance of the LQ control laws for thrust-borne flight and validate the proposed framework.

Throughout this work, noise-free sensing and full state feedback are assumed,

unless otherwise noted. Also assumed is a disturbance free environment (no wind in simulations).

1.5 Thesis Organization

This thesis is organized as follows. Mathematical preliminaries necessary for the understanding of theoretical contributions of this thesis are detailed in Chapter 2. Next, Chapter 3 contains the theoretical contributions of this work in detail. Simulation results supporting the theory are presented in Chapter 4. Finally, Chapter 5 contains concluding remarks.

Chapter 2
MATHEMATICAL PRELIMINARIES

The work here is an instance of established theory in vehicle control techniques. Mathematical details of previous work and mathematical techniques in use throughout this thesis are the topic of this chapter.

2.1 Linear Systems Theory

The motion and control of agile aircraft is dictated by the aircraft's nonlinear dynamics, described by equations of the form

$$\dot{x}_i = f_i(x_1, \ldots, x_n, u_1, \ldots, u_m),\ i = 1, \ldots, n, \tag{2.1}$$

$$y_i = g_i(x_1, \ldots, x_n, u_1, \ldots, u_m),\ i = 1, \ldots, p, \tag{2.2}$$

where u_i, $i = 1, \ldots, m$, denotes the system inputs; y_i, $i = 1, \ldots, p$, denote the system outputs; x_i, $i = 1, \ldots, n$, denote the system states; $\dot{x}_i$ denotes the time derivative of the states, x_i; and f_i and g_i are real-valued functions [5]. Time is not explicitly present in equations (2.1) and (2.2) as the aircraft dynamics are assumed be time-invariant. The system dynamics can be represented in vector form as

$$\dot{\vec{x}} = f(\vec{x}, \vec{u}), \tag{2.3}$$

$$\vec{y} = g(\vec{x}, \vec{u}), \tag{2.4}$$

where $\vec{x} \in \mathbb{R}^n$ is the state vector, $\vec{u} \in \mathbb{R}^m$ is the input vector, and $\vec{y} \in \mathbb{R}^p$ is the output vector [5].

Nonlinear system dynamics can then be approximated as linear around equilibrium or trim solutions, such as wings-level flight or hover for an agile aircraft. The

approximations are linear as the state derivatives change according to a linear function of the states and controls. Whether the linear dynamics are found via numerical linearization or through a Taylor Series approximation, the linear dynamics have the form

$$\dot{\vec{x}} = A\vec{x} + B\vec{u}, \tag{2.5}$$

$$\vec{y} = C\vec{x} + D\vec{u}, \tag{2.6}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, and $D \in \mathbb{R}^{p\times m}$.

Approximating nonlinear systems as linear allows the use of linear control techniques, such as optimal feedback control. Additionally, the simplicity provided by linearization combined with agile aircraft likely to spend the majority of missions in linear flight regimes makes it a powerful and useful tool.

2.2 Optimal Feedback Control

In order to satisfy the mission requirements or goals for any agile aircraft system, some form of control signals are required. When the aircraft is operating in a flight regime in which a linear approximation of the dynamics is valid, optimal feedback control signals can be generated according to the theory of the Linear-Quadratic-Regulator (LQR) problem or the Linear-Quadratic-Gaussian (LQG) problem. Linear quadratic approaches are so called since the system of dynamics of concern are linear, while the performance measure involves the square of a signal, hence quadratic [7].

For a deterministic linear system of the form (2.5)-(2.6), a control law of the form $u = -Kx$ that minimizes the performance measure

$$V = \int_0^T (x'Qx + u'Ru)dt + x'(T)Mx(T) \tag{2.7}$$

is desired. In this formulation of the LQR problem, Q and M are typically positive-semidefinite matrices, and R is a positive-definite matrix [7]. The matrix Riccati

equation[1] and final boundary value for the symmetric matrix $P(t)$

$$\begin{aligned} -\dot{P} &= A'P + PA + Q - PBR^{-1}B'P; \\ P(T) &= M \end{aligned} \tag{2.8}$$

lead to the optimal state-feedback control given by

$$\begin{aligned} u &= -K(t)x; \\ K(t) &= R^{-1}B'P(t). \end{aligned} \tag{2.9}$$

The LQG problem is an extension of the linear quadratic approach to linear stochastic systems of the form

$$\dot{x} = Ax + Bu + \xi \tag{2.10}$$

$$y = Cx + \theta \tag{2.11}$$

where $\xi(t)$ and $\theta(t)$ are uncorrelated zero-mean, Gaussian, white-noise random vectors with correlation matrices

$$E\{\xi(t)\xi'(t)\} = \Xi\delta(t-\tau);\ E\{\theta(t)\theta'(t)\} = \Theta\delta(t-\tau). \tag{2.12}$$

The random vector $\xi(t)$ is referred to as the process noise, and $\theta(t)$ is referred to as the measurement noise. Finding a control u that minimizes the cost function

$$V = \lim_{t\to\infty} E\{x'Qx + u'Ru\} \tag{2.13}$$

represents the LQG problem.

---

[1]The formulation of the Riccati equation and a discussion of the Hamilton-Jacobi optimization equation has been omitted. The information given here is sufficient for understanding of the remainder of this thesis, but interested readers may wish to refer to Dorato [7] for more detailed information.

2.3 Open Loop Control

While LQ techniques can be used for feedback control of flight regimes approximated as linear, the transitions between linear flight regimes require a different approach. The simplest form of open loop control is a predetermined signal designed by the system operator or creator. In order to improve on the basic open loop control sequences, machine learning schemes can be used.

2.3.1 Markov Decision Processes

A Markov decision process (or MDP) is the specification of a sequential decision problem for a fully observable environment with a Markovian transition model and additive rewards [23]. An MDP is defined by three components: an initial state S_0, a transition model $T(s, a, s')$, and a reward function $R(s)$. The transition model denotes the probability of reaching the state s' if action a is executed in state s. The assumption of a Markovian transition model implies that the probability of reaching s' from s depends only on s and not on any earlier states visited by the system.

A solution to an MDP is termed a policy π, which specifies what actions to take for any state that the system may reach. The action specified by a policy π for a system at state s is depicted as $\pi(s)$. When executing a policy, the sum of the rewards for all the visited states is the utility of that policy. Given the stochastic nature of a system and its environment, repeatedly executing a policy from an initial condition leads to different trajectories and different utilities. The expected utility of a policy can then be thought of as a measure of the quality of a policy [23]. The optimal policy, denoted π^*, is the policy that yields the highest expected utility for an MDP.

2.3.2 Reinforcement/Apprenticeship Learning

Reinforcement learning is a general category of machine learning in which feedback in the form of rewards or reinforcement is used to improve the performance of a

system. Reinforcement learning problems also include the subproblem of learning how the environment works. Numerous algorithms are available to find optimal policies for solving MDPs that include value iteration, policy iteration, adaptive dynamic programming, temporal difference schemes, Q-learning agent, policy search, and the E^3 algorithm.

An alternative approach to the reinforcement learning problem developed in [1] is that of apprenticeship learning. Whereas reinforcement learning algorithms focus on a combination of exploitation or exploration of the state space, apprenticeship learning focuses on exploiting regions of the state space near a policy provided by a teacher. The algorithm works to find a near-optimal policy based on the teachers that avoids dangerous parts of the state space. The apprenticeship learning algorithm is:

1. Run N trials under the teacher's policy π_T of system M. Save the state-action trajectories of these trials and compute $\hat{U}_M(\pi_T)$, an estimate of the utility of the teacher's policy, by averaging the sum of rewards in each trial.

2. Using all state-action trajectories collected, estimate the system dynamics T using regularized linear regression for the linearly parameterized dynamics case. Call the estimated dynamics $\hat{T}^{(i)}$.

3. Find an $\alpha/8$ (where $\alpha > 0$) optimal policy for the MDP $\hat{M}^{(i)} = (S, A, \hat{T}^{(i)}, H, D, R)$, and call this policy $\pi^{(i)}$.

4. Evaluate the utility of policy $\pi^{(i)}$ on real system M for k trials. Save all state-action trajectories, and let $\hat{U}_M(\pi^{(i)})$ be the average sum of rewards accumulated.

5. If $\hat{U}_M(\pi^{(i)}) \geq \hat{U}_M(\pi_T) - \alpha/2$, return $\pi^{(i)}$, otherwise set $i = i + 1$, and go back to step 2 [1].

2.4 Aircraft Orientation

In modeling the six degrees of freedom of an aircraft in flight, the aircraft orientation and position must be oriented to some fixed frame, traditionally the vehicle-carried Earth axis system. The common approach to depicting an aircraft's orientation is to use Euler angles, an ordered set of rotations; however, Euler angles present a few problems if the system is capable of sustained vertical flight. As a result, alternate methods of depicting the aircraft orientation must be considered and used to replace or supplement the Euler angle depictions of attitude.

2.4.1 Euler Angles

The order dependent Euler angle transformation starts from the standard North-East-Down (NED) axes and translates the aircraft's orientation to the body axes. First, a right-handed yaw rotation (ψ) is performed about the z-axis. Second, a right-handed pitch rotation (θ) is performed about the new position of the y-axis. Finally, a right-handed roll rotation (ϕ) is performed about the new position of the x-axis [25].

The plane rotation matrices can be written according to the prescribed rules of the rotations as

$$\vec{u}^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{u}^r \tag{2.14}$$

where $\vec{u}^b$ is a vector expressed in the body frame, and $\vec{u}^r$ is the vector expressed in the reference frame. Note also that an abbreviating convention is used where cosine and sine are shortened to c and s, respectively. The three rotation matrices can by multiplied together to form an invertible transformation matrix from the reference

system to the body system as

$$C_{b/r} = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ -c\theta s\psi + s\phi s\theta c\psi & c\phi c\psi + s\phi s\theta s\psi & s\phi c\theta \\ s\phi s\psi + c\phi s\theta c\psi & -s\phi c\psi + c\phi s\theta s\psi & c\phi c\theta \end{bmatrix}. \tag{2.15}$$

Problems arise with Euler angle determination of aircraft orientation when the the pitch angle (θ) is 90^o, as the roll and heading angles are not uniquely determined. Additionally, wrap-around difficulties present themselves when the yaw and roll angles exceed their bounds of $\pm 180^o$ [6].

The kinematic relationships for rotation[2] can be found by letting the Euler angles have time derivatives $(\dot{\psi}, \dot{\theta}, \dot{\phi})$, and using the components of the aircraft's angular-velocity vector,

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & t\theta s\phi & t\theta c\phi \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{2.16}$$

where P, Q, R are the standard symbols for roll, pitch, and yaw rate of an aircraft [25].

2.4.2 Vertical Euler Angles

One approach to avoiding the deficiencies of the Euler angles is to develop a second set of rotations, termed the vertical Euler angles. This second set of angles can be used to provide unambiguous attitude depiction while aircraft are operating in a vertical flight regime. As with the Euler angles, the vertical Euler angles are an ordered set of right-handed rotations. Starting with the aircraft in a vertical attitude with the body x-axis pointing opposite the NED z-axis and the body z-axis pointing North, aligned with the NED x-axis, perform a roll rotation (ϕ_v) about the body x-axis.

[2] A complete derivation of this kinematic relationship is omitted for succinctness. Please see [25] for more information.

Next, perform a pitch rotation (θ_v) about the new y-axis. Finally, perform a yaw rotation (ψ_v) about the twice moved z-axis, as seen in Figure (2.1).

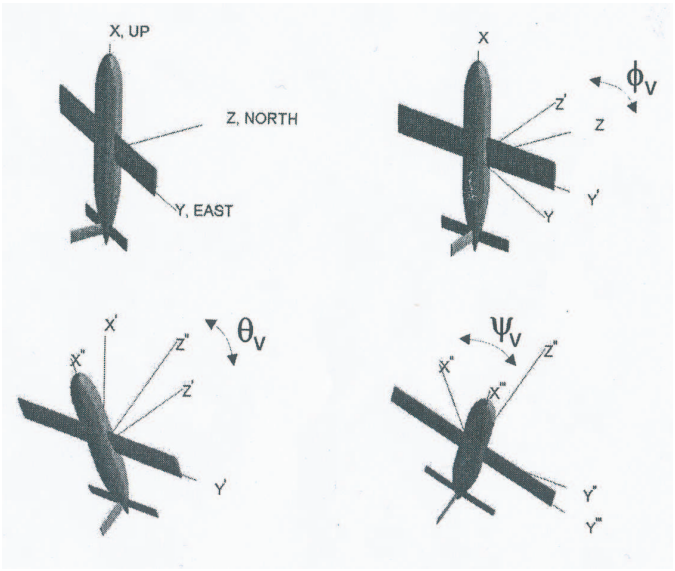

Figure 2.1: Depiction of the rotations of the vertical Euler method of tracking orientation [6].

In the vertical Euler angle system, the order of rotations is roll, pitch, yaw; the opposite of the yaw, pitch, roll rotations of the traditional Euler angle rotations. Additionally, since the same idea is applied to rotate the axes in both sets of rotations, singularities are present in the vertical Euler angle depiction as well. Here, the vertical roll and yaw angles are not uniquely determined when the vertical pitch angle is 90^o, when the aircraft is in level horizontal flight.

The vertical Euler angles allow intuitive angular representations of an aircraft's orientation in the vertical regimes with flight, while traditional Euler angles provide the same intuitive representation in horizontal flight regimes. By using both in conjunction with one another, attitude representations can be made throughout the whole spectrum of flight; however, this approach complicates calculation of the time

derivatives of the Euler angles.

2.4.3 Quaternions

In order to avoid the singularity and computational issues of the Euler angle approach, quaternions can be used. Though much less intuitive, the quaternion approach is more robust, and still allows the use of Euler angles for visualization of the aircraft's orientation. Quaternions, of the form $x_0+x_1i+x_2j+x_3k$ allow an alternate expression of the kinematical equations:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -P & -Q & -R \\ P & 0 & R & -Q \\ Q & -R & 0 & P \\ R & Q & -P & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}. \tag{2.17}$$

For compound rotations involving a combination of pitch, roll, and yaw, the quaternion rotation axis is not evident [25], so quaternions are often initialized from Euler angles. Further, while the orientation is tracked via the quaternion, it is possible to obtain the Euler angles (or vertical Euler angles) as needed or desired. The relationship between the quaternions and Euler angles is given by

$$q_0 = \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \tag{2.18}$$

$$q_1 = \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \tag{2.19}$$

$$q_2 = \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \tag{2.20}$$

$$q_3 = \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2), \tag{2.21}$$

and the relationship between the quaternions and vertical Euler angles is given by

$$q_0 = \sqrt{\frac{1-\sin(\theta_v)}{2}}\cos\left(\frac{\psi_v-\phi_v}{2}\right) \tag{2.22}$$

$$q_1 = \sqrt{\frac{1+\sin(\theta_v)}{2}}\sin\left(\frac{\psi_v+\phi_v}{2}\right) \tag{2.23}$$

$$q_2 = \sqrt{\frac{1+\sin(\theta_v)}{2}}\cos\left(\frac{\psi_v+\phi_v}{2}\right) \tag{2.24}$$

$$q_3 = \sqrt{\frac{1-\sin(\theta_v)}{2}}\sin\left(\frac{\psi_v-\phi_v}{2}\right). \tag{2.25}$$

To obtain the Euler or vertical Euler angles from quaternions, first the transformation matrix is formed,

$$\mathbf{B} = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}, \tag{2.26}$$

the elements b_{ij} of which are used to find the Euler angles:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \tan^{-1}(\frac{b_{23}}{b_{33}}) \\ -\sin^{-1}(b_{13}) \\ \tan^{-1}(\frac{b_{12}}{b_{11}}) \end{bmatrix}, \tag{2.27}$$

or the vertical Euler angles:

$$\begin{bmatrix} \phi_v \\ \theta_v \\ \psi_v \end{bmatrix} = \begin{bmatrix} -\tan^{-1}(\frac{b_{23}}{b_{13}}) \\ \sin^{-1}(b_{33}) \\ tan^{-1}(\frac{b_{12}}{b_{31}}) \end{bmatrix}. \tag{2.28}$$

This approach of internally calculating the quaternions in the aircraft dynamics, but using a combination of the normal Euler angles and the vertical Euler angles to present vehicle orientation is used for the modeling of the Tandem-Wing Tail-Sitter UAV [6], in which Euler angles are used for pitch angles less than 45^o and the vertical Euler angles are used for pitch angles greater than 45^o.

2.4.4 Incremental Rotations

A final method of depicting the orientation of the aircraft is to use incremental rotations. In the case of the Aerovel Flexrotor, ϕ_x, ϕ_y, and ϕ_z are incremental rotations

about the respective body axes. In other words, they are values resulting from the integration of the respective rotation rates of the Flexrotor dynamics [18].

This simpler method of tracking and presenting orientation can be derived from equation (2.16) through the small angle approximation. If ϕ, θ, and ψ are all close to zero, then the cosines are approximately 1 and the sines are approximately the angular values, or zero, so equation (2.16) becomes

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix}. \tag{2.29}$$

The task of tracking and presenting aircraft orientation is complicated by studying an agile-aircraft capable of sustained vertical flight; however, by combining the various approaches here, full and accurate attitude information is available.

2.5 Wing-Borne Flight

Aircraft dynamics are prescribed by the general six degree of freedom formulation,

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} + g \begin{bmatrix} -S\theta \\ S\phi C\theta \\ C\phi C\theta \end{bmatrix} + \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} \tag{2.30}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix}^{-1} \begin{bmatrix} L + (I_{yy} - I_{zz})qr + I_{xz}pq \\ M + (I_{zz} - I_{xx})pr + I_{xz}(r^2 - p^2) \\ N + (I_{xx} - I_{yy})pq - I_{xz}qr \end{bmatrix} \tag{2.31}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & T\theta S\phi & T\theta C\phi \\ 0 & C\phi & -S\phi \\ 0 & S\phi/C\theta & C\phi/C\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.32}$$

$$\begin{bmatrix} \dot{P}_N \\ \dot{P}_E \\ -\dot{h} \end{bmatrix} = \mathbf{B}^{-1} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{2.33}$$

where the forces, $F_x,\ F_y,\ F_z$, moments, $L,\ M,\ N$, and the moments and products of inertia $I_{xx},\ I_{yy},\ I_{xz},\ I_{zz}$ are specific to each aircraft [25], and the transformation matrix $\mathbf{B}$ is given by Equation 2.26. In this formulation (Equation 2.32), the Euler kinematic equation is used, though the quaternion implementation could be used in its place.

In wing-borne flight, the wings provide lift to balance the force of gravity, while the motor provides a propulsive force. In a trimmed, wings-level unaccelerated flight condition, the four forces of flight balance out so that $L = W$ and $T = D$.

2.6 Thrust-Borne Flight

Aircraft capable of sustained vertical flight, or thrust-borne flight, present a unique set of challenges when compared to traditional aircraft. Consider an agile platform in a trimmed hover configuration, so that thrust equals weight ($T = W$), and lift and drag forces are reduced to magnitudes near zero and can be ignored. While the 6-DOF equations (2.30, 2.31, 2.32, and 2.33) still apply, special attention needs to be made to the modeling of the aircraft. In a hover configuration, the throttle is used to control altitude, ailerons control the vertical roll rate, the rudder controls the velocity in the body-axis y direction, and the elevator controls the velocity in the body-axis z direction.

Typically, to simplify modeling, control surface contributions to aircraft dynamics are a function of the free stream velocity, but in hover, the free stream velocity goes to zero. To accurately model the control surfaces, the prop wash must be modeled, as it provides necessary airflow over the ailerons, rudder, and elevator to allow control while in hover. Some aircraft are further restricted in their controllability if the ailerons are out-board of the prop wash. The rotational effects of the prop wash and engine must also be taken into account, and balanced by the ailerons (or some other means) while in a stabilized hover.

Chapter 3
SYSTEM MODELING AND CONTROLLER DESIGN

The development of various controllers for the YAK-54 and the Aerovel Flexrotor is the focus of this chapter. Initially, each system is described, including the derivation of system dynamics and linearization. Based on the linearization of each system, linear controllers will be considered for thrust-borne and wing-borne flight regimes. Finally, open loop controllers will be proposed.

3.1 System Description

The two aircraft under consideration here are the YAK-54 and the Aerovel Flexrotor. While both have thrust to weight ratios in excess of one, they vary significantly in terms of design and performance. The YAK-54 is a highly agile aircraft capable of complex aerobatic maneuvers. The Flexrotor is designed for long endurance flight and presents a different approach to thrust-borne flight through the use of tip thrusters for roll control.

3.1.1 YAK-54

The Yakovlev YAK-54 is a Russian built aerobatic aircraft that first flew on the 23rd of December, 1993 [29]. Scale models to the YAK-54 have enjoyed prominence in the remote control aircraft world for their exceptional maneuverability and ability to perform complex 3D acrobatics.

The YAK-54 studied here is a 69 inch almost-ready-to-fly (ARF) remote control aircraft manufactured by NitroModels (see Figure 3.1). While this size of RC aircraft typically employs small two stroke gas engines in the 1.0 to 2.0 cubic centimeter size,

an electric motor was selected. The Turnigy 63-64-B motor provides a high level of reliability and is easier to install and operate. The motor, combined with a 20 inch diameter prop with 10 inches of pitch, provides the YAK-54 with a thrust to weight ratio greater than one. The significant limitation of the electric motor is a short endurance - roughly five to ten minutes depending on throttle settings throughout the flight.

3.1.2 Aerovel Flexrotor

The Aerovel Flexrotor was designed with long range, long endurance, and a vertical take off and landing (VTOL) capability. Similar in design and concept to tail-sitter aircraft developed and tested during the 1950s, such as the Convair XFY-1 Pogo, Lockheed XFV-1 Salmon, and the Ryan X-13A Vertijet [22], the Flexrotor concept has these distinctive features and design goals:

- a large hybrid propeller/rotor with low disc loading,
- a variable-pitch rotor hub,
- the capability to perform stationary launch in a vertically oriented position,
- a high speed cruise in a typical aircraft configuration,
- standard control surfaces in wing-borne flight,
- a unique combination of a rotor cyclic and wing-tip thrusters for control in thrust-borne flight, and
- the capability to recover autonomously from thrust-borne flight[19].

The ability to launch and recover from a vertical orientation is significantly different from that of the YAK-54, which can only takeoff and land in a conventional manner. Figure 1.3 shows the general layout of the Flexrotor, including the tractor propeller and the wing-tip thrusters, along with a small V-tail for rudder and elevator control.

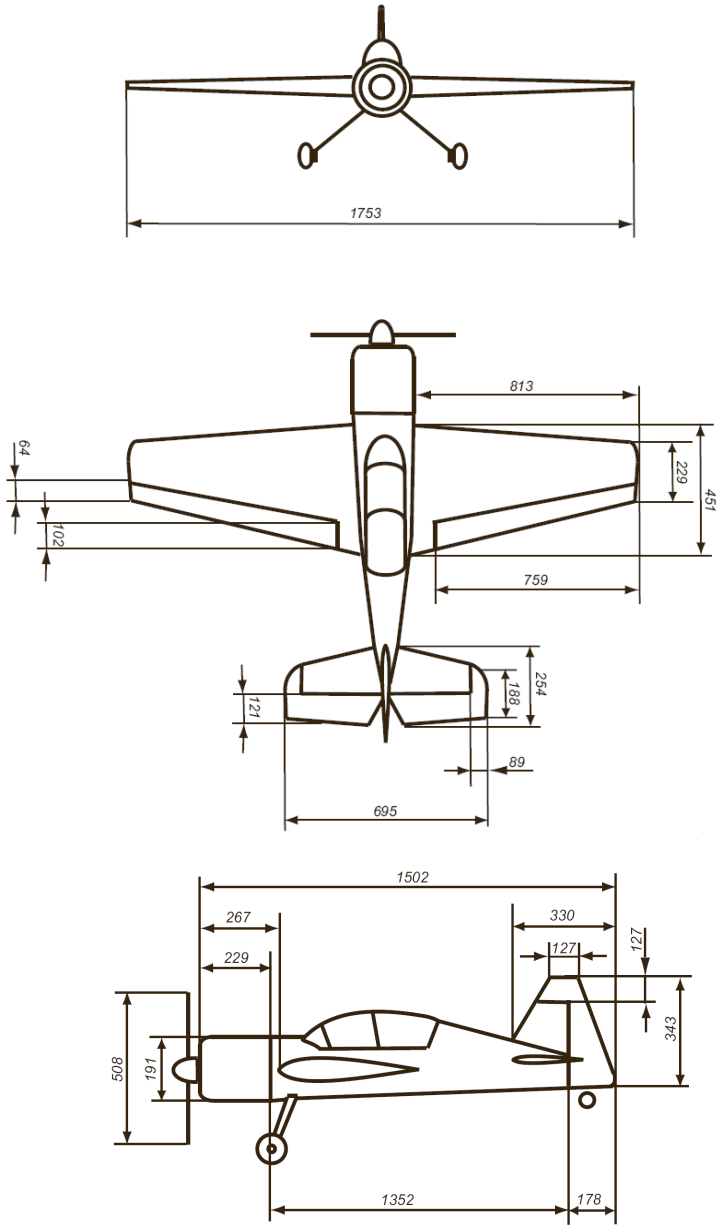

Figure 3.1: A three-view of the YAK-54 showing all geometry and dimensions (in mm) [15].

3.2 System Dynamics and Modeling

Just as both aircraft are different in terms of design, both are modeled in different manners. Though the basic dynamics of any aircraft are quite similar, the modeling and specifics of the dynamics vary significantly. The YAK-54 is a conventional aircraft in design, and can be easily modeled for wing-borne flight; however, flight near vertical or in unusual angles of attack presents some difficulties. The dynamics of the YAK-54[1] are modeled in a Matlab/Simulink environment that makes use of the Aerosim Blockset. Other modeling challenges include the lack of airspeed over the wings while in thrust-borne flight. As the only airflow is provided by the propeller, prop wash and its effects must be accounted for. The Flexrotor dynamics[2] are complicated by the helicopter style prop and the use of tip thrusters for roll control. The dynamics of the Flexrotor are modeled in a proprietary software package developed by Aerovel.

3.2.1 YAK-54

The state vector for the Yak-54 is

$$x = [\ u \quad v \quad w \quad p \quad q \quad r \quad P_N \quad P_E \quad P_D \quad \phi \quad \theta \quad \psi\ ]^T \tag{3.1}$$

or

$$x = [\ u \quad v \quad w \quad p \quad q \quad r \quad P_N \quad P_E \quad P_D \quad \phi_v \quad \theta_v \quad \psi_v\ ]^T \tag{3.2}$$

depending on the current flight regime, where $u,\ v,\ w$ represent the aircraft velocity, $p,\ q,\ r$ represent the aircraft roll, pitch, and yaw rates, and $P_N,\ P_E,\ P_D$ represent the aircraft's position in the NED frame. In wing-borne flight, Euler angles are used, whereas in thrust-borne flight the vertical Euler angles are used for position tracking.

---

[1]For a complete dialogue on the derivation of the YAK-54's modeling and dynamics, please see [15].

[2]For more information regarding the dynamics, please see [19], [20], and [17] and for more information regarding the proprietary nature of the modeling and dynamics of the Aerovel Flexrotor contact Dr. Tad McGeer of Aerovel.

The input vector of controls is

$$u = [\ \delta_T \quad \delta_e \quad \delta_r \quad \delta_a\ ]^T, \tag{3.3}$$

where δ_T is the throttle input and $\delta_e,\ \delta_r,\ \delta_a$ are the elevator, rudder, and aileron inputs, respectively.

The aileron range has limits of ±35 degrees, while the elevator and rudder both have ranges of ±45 degrees. The throttle fractional varies between 0 and 1, with 0 corresponding to a fully closed throttle, and 1 corresponding to maximum throttle.

3.2.2 Aerovel Flexrotor

The state vector for the Flexrotor is

$$x = [\ u \quad v \quad w \quad p \quad q \quad r \quad \phi_x \quad \phi_y \quad \phi_z \quad P_N \quad P_E \quad P_D\ ]^T \tag{3.4}$$

where $u,\ v,\ w,\ p,\ q,\ r,\ P_N,\ P_E$, and P_D represent the same variables as in the YAK-54 dynamics, and orientation is tracked through the incremental rotations $\phi_x,\ \phi_y$, and ϕ_z. The Flexrotor's input vector is

$$u = [\ P_{0\ thr\ frac} \quad R_{0\ hub\ beta} \quad dp'c \quad dq'c \quad dr'c\ ]^T \tag{3.5}$$

where $P_{0\ thr\ frac}$ is the throttle fractional, $R_{0\ hub\ beta}$ is the collective pitch angle for the main rotor, and $dp'c,\ dq'c$, and $dr'c$ are diagonalized control inputs. A mixing matrix then specifies the map from the diagonalized inputs to the physical controls [18]. The remaining physical controls are the aileron, elevator, and rudder inputs: $\delta_a,\ \delta_e,\ \delta_r$, the cyclic settings of the main rotor: $R_{0\ cyc\ y},\ R_{0\ cyc\ z}$, and the power settings for the tip thrusters: $P_{1\ power},\ P_{2\ power}$.

3.3 Linearization

Both the YAK-54 and the Flexrotor make use of numerical linearization techniques to find trim conditions and the matrices necessary to define an LTI system.

3.3.1 YAK-54

For thrust-borne flight using the vertical Euler angles and neglecting the navigation equations, the YAK-54 dynamics are given by [15]:

$$\dot{u} = rv - qw - g\cos(\psi_v)\cos(\theta_v) + \frac{F_x}{m} \tag{3.6}$$

$$\dot{v} = pw - ru + g\sin(\psi_v)\cos(\theta_v) + \frac{F_y}{m} \tag{3.7}$$

$$\dot{w} = qu - pv - g\sin(\theta_v) + \frac{F_z}{m} \tag{3.8}$$

$$\dot{p} = c_1 pq - c_2 qr + c_3 L + c_4 N \tag{3.9}$$

$$\dot{q} = c_5 pr - c_6(p^2 - r^2) + c_7 M \tag{3.10}$$

$$\dot{r} = c_8 pq - c_9 qr - c_{10} L + c_{11} N \tag{3.11}$$

$$\dot{\phi}_v = \frac{\cos\psi_v}{\cos\theta_v} p - \frac{\sin\psi_v}{\cos\theta_v} q \tag{3.12}$$

$$\dot{\theta}_v = p\sin\psi_v + q\cos\psi_v \tag{3.13}$$

$$\dot{\psi}_v = p\frac{\cos\psi_v \sin\theta_v}{\cos\theta_v} + q\frac{\sin\psi_v \sin\theta_v}{\cos\theta_v} + r \tag{3.14}$$

where

$$c_1 = \frac{I_{xz}(I_{xx} - I_{yy} + I_{zz})}{I_{xx}I_{zz} - I_{xz}^2} \tag{3.15}$$

$$c_2 = \frac{I_{zz}(I_{zz} - I_{yy}) + I_{xz}^2}{I_{xx}I_{zz} - I_{xz}^2} \tag{3.16}$$

$$c_3 = \frac{I_{zz}}{I_{xx}I_{zz} - I_{xz}^2} \tag{3.17}$$

$$c_4 = \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} \tag{3.18}$$

$$c_5 = \frac{(I_{zz} - I_{xx})}{I_{yy}} \tag{3.19}$$

$$c_6 = \frac{I_{xz}}{I_{yy}} \tag{3.20}$$

$$c_7 = \frac{1}{I_{yy}} \tag{3.21}$$

$$c_8 = \frac{I_{xx}(I_{xx} - I_{yy}) + I_{xz}^2}{I_{xx}I_{zz} - I_{xz}^2} \tag{3.22}$$

$$c_9 = \frac{I_{xz}(I_{xx} - I_{yy} + I_{zz})}{I_{xx}I_{zz} - I_{xz}^2} \tag{3.23}$$

$$c_{10} = \frac{I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} \tag{3.24}$$

$$c_{11} = \frac{I_{xx}}{I_{xx}I_{zz} - I_{xz}^2}. \tag{3.25}$$

In order to linearize the equations of motion in thrust-borne flight around a given vertical velocity u_1, substitutions are made to the aircraft states:

$$u \rightarrow u_1 + \Delta u \tag{3.26}$$

$$v \rightarrow \Delta v \tag{3.27}$$

$$w \rightarrow \Delta w \tag{3.28}$$

$$p \rightarrow \Delta p \tag{3.29}$$

$$q \rightarrow \Delta q \tag{3.30}$$

$$r \rightarrow \Delta r \tag{3.31}$$

$$\phi_v \rightarrow \Delta\phi_v \tag{3.32}$$

$$\theta_v \rightarrow \Delta\theta_v \tag{3.33}$$

$$\psi_v \rightarrow \Delta\psi_v. \tag{3.34}$$

After applying standard small angle assumptions and discarding second order terms, the equations of motion (3.6)-(3.14) of the YAK-54 then become:

$$\dot{u} = -g + \frac{F_x}{m} \tag{3.35}$$

$$\dot{v} = -r\ u_1 + g\psi_v + \frac{F_y}{m} \tag{3.36}$$

$$\dot{w} = q\ u_1 - g\theta_v + \frac{F_z}{m} \tag{3.37}$$

$$\dot{p} = c_3 L + c_4 N = L/I_{xx} \tag{3.38}$$

$$\dot{q} = c_7 M = M/I_{yy} \tag{3.39}$$

$$\dot{r} = c_{10} L + c_{11} N = N/I_{zz} \tag{3.40}$$

$$\dot{\phi}_v = p \tag{3.41}$$

$$\dot{\theta}_v = q \tag{3.42}$$

$$\dot{\psi}_v = r \tag{3.43}$$

where instead of Δp, we have written p. Additionally, the product of inertia I_{xz} is taken to be zero as it is assumed to be significantly smaller than the moments of inertia [15].

As the force and moment terms are functions of the states and inputs, a first order Taylor expansion is used. For example, the Taylor expansion of the forces in the body-axis x direction is given by

$$\begin{aligned} F_x &= F_{x_1} + F_{x_u}\Delta u + F_{x_v}\Delta v + F_{x_w}\Delta w + F_{x_p}\Delta p + F_{x_q}\Delta q + F_{x_r}\Delta r \\ &+ F_{x_{\delta_T}}\Delta\delta_T + F_{x_{\delta_e}}\Delta\delta_e + F_{x_{\delta_r}}\Delta\delta_r + F_{x_{\delta_a}}\Delta\delta_a, \end{aligned} \tag{3.44}$$

where the second subscript denotes the partial derivative with respect to that variable ($F_{x_u} = \frac{\partial F_x}{\partial u}$) [6]. In order for $\dot{u}$ to limit to zero in Equation 3.35, the zero-order term of Equation 3.44 must be equal to mg, while the zero-order terms of $F_y,\ F_z,\ L,\ M,$ and N are all zero.

Following the analysis of [15], many of the partial derivatives of each Taylor expansion can reasonably be expected to be close to zero for the thrust-borne flight regime, so the LTI system is given by

$$\dot{x} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\phi}_v \\ \dot{\theta}_v \\ \dot{\psi}_v \\ \dot{P}_N \\ \dot{P}_E \\ \dot{h} \end{bmatrix} = \begin{bmatrix} F_{x_u}/m & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & F_{y_v}/m & 0 & 0 & 0 & F_{y_r}/m - u_1 & 0 & 0 & g & 0 & 0 & 0 \\ 0 & 0 & F_{z_w}/m & 0 & F_{z_q}/m - u_1 & 0 & 0 & -g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_p/I_{xx} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M_w/I_{yy} & 0 & M_q/I_{yy} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_v/I_{zz} & 0 & 0 & 0 & N_r/I_{zz} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -u_1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & u_1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} x$$

$$+ \begin{bmatrix} F_{x_{\delta_T}}/m & 0 & 0 & 0 \\ 0 & 0 & 0 & F_{y_{\delta_r}}/m \\ 0 & F_{z_{\delta_e}}/m & 0 & 0 \\ 0 & 0 & L_{\delta_a}/I_{xx} & 0 \\ 0 & M_{\delta_e}/I_{yy} & 0 & 0 \\ 0 & 0 & 0 & N_{\delta_r}/I_{zz} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} u, \tag{3.45}$$

where the navigation equations are included, assuming that the trim condition is with the body-axis z direction facing north, so that the vertical Euler angles are zero.

With Equation 3.45 in the form of Equation 2.5, the LTI system is transformed in order to reorder the states and partition the dynamics from one large system into four smaller decoupled systems. The transformation matrix

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.46}$$

is used to form the new state-space system

$$\dot{z} = \tilde{A}z + \tilde{B}u \tag{3.47}$$

where $z = T^{-1}x$, $\tilde{A} = T^{-1}AT$ and $\tilde{B} = T^{-1}B$. This state transformation admits the following decoupled subsystems.

- The axial subsystem with $x_a = [u\ h]^T$ and $u_a = \delta_T$ is given by:

$$\dot{x}_a = \begin{bmatrix} F_{x_u}/m & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ h \end{bmatrix} + \begin{bmatrix} F_{x_{\delta_T}}/m \\ 0 \end{bmatrix} \delta_T. \tag{3.48}$$

- The roll subsystem with $x_r = [p\ \phi_v]^T$ and $u_r = \delta_a$ is given by:

$$\dot{x}_r = \begin{bmatrix} L_p/I_{xx} & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi_v \end{bmatrix} + \begin{bmatrix} L_{\delta_a}/I_{xx} \\ 0 \end{bmatrix} \delta_a. \tag{3.49}$$

- The longitudinal subsystem with $x_{long} = [w\ q\ \theta_v\ P_N]^T$ and $u_{long} = \delta_e$ is given by:

$$\dot{x}_{long} = \begin{bmatrix} F_{z_w}/m & F_{z_q}/m - u_1 & -g & 0 \\ M_w/I_{yy} & M_q/I_{yy} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -u_1 & 0 \end{bmatrix} \begin{bmatrix} w \\ q \\ \theta_v \\ P_N \end{bmatrix} + \begin{bmatrix} F_{z_{\delta_e}}/m \\ M_{\delta_e}/I_{yy} \\ 0 \\ 0 \end{bmatrix} \delta_e. \tag{3.50}$$

- The lateral subsystem with $x_{lat} = [v\ r\ \psi_v\ P_E]^T$ and $u_{lat} = \delta_r$ is given by:

$$\dot{x}_{lat} = \begin{bmatrix} F_{y_v}/m & F_{y_r}/m - u_1 & g & 0 \\ N_v/I_{zz} & N_r/I_{zz} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & u_1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi_v \\ P_E \end{bmatrix} + \begin{bmatrix} F_{y_{\delta_r}}/m \\ N_{\delta_r}/I_{zz} \\ 0 \\ 0 \end{bmatrix} \delta_r. \tag{3.51}$$

For the aircraft in a vertical position with the belly pointing north (body-axis z direction) with zero vertical velocity u_1, the trim inputs are

$$u_0 = [\ 0.7396 \quad 0 \quad -0.0961 \quad 0\ ]^T, \tag{3.52}$$

which have the expected form of a non-zero throttle input to counteract weight and a non-zero aileron input to counteract the rotational effects of the motor and the prop wash.

3.3.2 Aerovel Flexrotor

The linearization of the Flexrotor in thrust-borne flight is performed by Aerovel's comprehensive software suite developed for the Flexrotor. The nonlinear dynamics prescribed by (2.30) - (2.33) are then expressed as an LTI system in the form of (2.5). As with the YAK-54, a state transformation (3.46) is applied to diagonalize the system and decouple it into subsystems.

- The axial subsystem with $x_a = [u\ h]^T$ and $u_a = [P_0\ R_0]^T$ is given by:

$$\dot{x}_a = \begin{bmatrix} -0.41 & 0 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} u \\ h \end{bmatrix} + \begin{bmatrix} 0.54 & -1.6 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ R_0 \end{bmatrix}. \tag{3.53}$$

- The roll subsystem with $x_r = [p\ \phi_v]^T$ and $u_r = dp'c$ is given by:

$$\dot{x}_r = \begin{bmatrix} -1.4 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi_v \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} dp'c. \tag{3.54}$$

- The longitudinal subsystem with $x_{long} = [w\ q\ \theta_v\ P_N]^T$ and $u_{long} = dq'c$ is given by:

$$\dot{x}_{long} = \begin{bmatrix} -0.47 & 0.05 & -9.81 & 0 \\ 0.32 & -2.27 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ q \\ \theta_v \\ P_N \end{bmatrix} + \begin{bmatrix} -0.1 \\ 1 \\ 0 \\ 0 \end{bmatrix} dq'c. \tag{3.55}$$

- The lateral subsystem with $x_{lat} = [v\ r\ \psi_v\ P_E]^T$ and $u_{lat} = dr'c$ is given by:

$$\dot{x}_{lat} = \begin{bmatrix} -0.06 & -0.22 & 9.81 & 0 \\ 1.54 & -2.31 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi_v \\ P_E \end{bmatrix} + \begin{bmatrix} 0.18 \\ 1 \\ 0 \\ 0 \end{bmatrix} dr'c. \tag{3.56}$$

Due to the use of the diagonalized control inputs, all three controls for the axial, longitudinal, and lateral subsystems of the Flexrotor must be mapped back to the physical controls by using the prescribed mixing matrix for the linearization. For thrust-borne flight, the physical controls are given by

$$\begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \\ R_{cycy} \\ R_{cycz} \\ P1 \\ P2 \end{bmatrix} = \begin{bmatrix} 0.012 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0.008 & -0.014 \\ 0 & 0.008 & 0.014 \\ 11.2 & 0.168 & 0.293 \\ 11.2 & 0.167 & 0.293 \end{bmatrix} \begin{bmatrix} dp'c \\ dq'c \\ dr'c \end{bmatrix}. \tag{3.57}$$

3.4 Linear Controllers

In thrust-borne flight, LQ control techniques are developed and then extended to include the effects of uncertainty in the plant and measurements. The effects of wind are also considered.

3.4.1 YAK-54

In thrust-borne flight, and assuming that the vehicle is in hover with no upward velocity component u_1, four velocity controllers are used to control the YAK-54. The controllers are:

- U-velocity controller that adjusts vertical velocity with the throttle. This controller corresponds to the axial subsystem.

- Vertical roll rate controller which uses the ailerons to trace either a commanded roll rate or a commanded orientation (ϕ_v) about the body-axis x direction. The vertical roll rate controller corresponds to the roll subsystem.

- W-velocity controller which uses the elevator to adjust the velocity in the body-axis z direction. This controller corresponds to the longitudinal coordinate system.

- V-velocity controller uses the rudder to adjust the velocity in the body-axis y direction. And the V-velocity controller corresponds to the lateral subsystem.

Looking first at the U-velocity controller, the subsystem (3.48) becomes

$$\begin{bmatrix} \dot{u} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} -0.038 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ h \end{bmatrix} + \begin{bmatrix} 25.8 \\ 0 \end{bmatrix} \delta_T \tag{3.58}$$

when numerical values are determined about the linearization point. Observing that $\dot{h} = u$, an LQR controller is developed that also incorporates an integral term, thus ensuring zero-steady state error properties [6], as the height can be thought of as the integral of the u-velocity component. For the U-velocity controller, weights were chosen to heavily penalize throttle changes and deviations in u-velocity, while placing less emphasis on aircraft height. The weight matrices used to solve the Riccati Equation (2.8) and that penalize state errors and control displacements are

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R = 100, \tag{3.59}$$

which lead to a gain of

$$K_{ax} = [\ 0.327 \quad 0.1\ ]. \tag{3.60}$$

The throttle control input can be thought of as

$$\delta_T = -0.327(u - u_{command}) - 0.1 \int (u - u_{command}) dt. \tag{3.61}$$

In addition, a saturation block is used to prevent the throttle setting from dropping too low while in thrust-borne flight. The saturation block is set at 50%, and is necessary to ensure sufficient airflow and dynamic pressure over the control surfaces due to prop-wash. A block diagram of the throttle controller is shown in Figure 3.2.

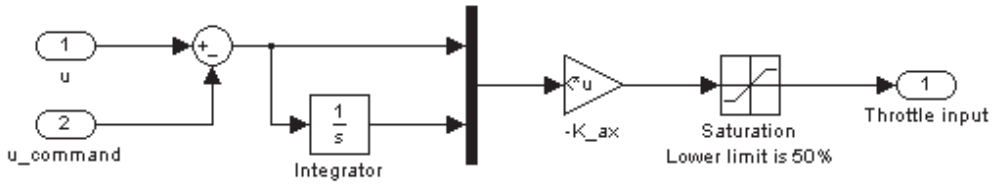

Figure 3.2: Block diagram of the vertical velocity throttle controller.

The roll subsystem (Equation 3.49), controlled via the roll rate controller has weight matrices of

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}, \quad R = 1, \tag{3.62}$$

which give a gain matrix of

$$K_r = \begin{bmatrix} -1.0424 & -3.6123 \end{bmatrix} \tag{3.63}$$

when solving the Riccati Equation that is used to calculate the aileron input. The weights for adjusting the aileron penalize deviations in vertical roll angle, ϕ_v, the most. The aileron input is then given by

$$\delta_a = 1.0424p + 3.6123(\phi_v - \phi_{v,command}). \tag{3.64}$$

The longitudinal subsystem (3.50) and the lateral subsystem (3.51) both make use of the hybrid LQR/integral approach used in the development of the vertical velocity controller. Weights were chosen to more heavily penalize deviations of both w and θ_v, while also penalizing large deflections of the elevator. For the longitudinal system, the weight matrices are

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R = 10, \tag{3.65}$$

and the gain matrix is

$$K_{long} = [\ 0.578 \quad -0.446 \quad -3.776 \quad 0.316\ ]. \tag{3.66}$$

The elevator control input is then

$$\delta_e = -0.578(w - w_{command}) + 0.446q + 3.776\theta_v - 0.316\int(w - w_{command})dt. \tag{3.67}$$

Similarly for the lateral subsystem and the V-velocity controller, though the chosen weights penalize deviations in v, ψ_v, and δ_r:

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad R = 100 \tag{3.68}$$

$$K_{lat} = [\ -0.478 \quad -0.447 \quad -3.782 \quad -0.1\ ] \tag{3.69}$$

$$\delta_r = 0.478(v - v_{command}) + 0.447r + 3.782\psi_v + 0.1\int(v - v_{command})dt. \tag{3.70}$$

For thrust-borne flight, the YAK-54 control structure is shown in Figure 3.3.

For control of the YAK-54 in wing-borne level flight, proportional integral (PI) controllers are used to control the rudder, aileron, elevator and throttle. In maintaining level flight, the rudder is used to eliminate side slip of the aircraft. Side slip is given by

$$\beta = \sin^{-1}(v/V_a), \tag{3.71}$$

where V_a is the overall velocity vector of the aircraft. With all values measured in degrees, the rudder control is

$$\delta_r = 0.5\beta + -0.05\int\beta dt. \tag{3.72}$$

The aileron is controlled by a wing-leveler that tracks a commanded bank angle, ϕ. For wings-level flight, the controller tracks a zero reference value, but can also be

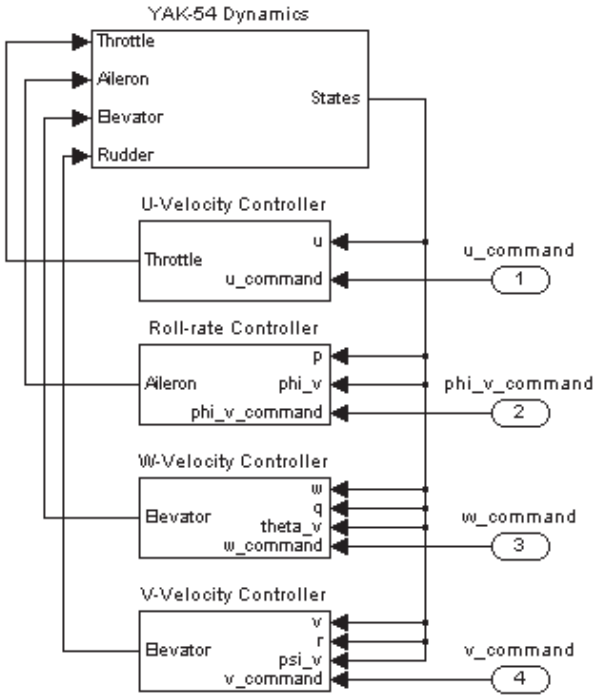

Figure 3.3: Block diagram of the control structure of the YAK-54 in thrust-borne flight.

used to track bank angles for turning flight. The aileron control signal is

$$\delta_a = 3(\phi - \phi_{command}) + 0.75 \int (\phi - \phi_{command}) dt. \tag{3.73}$$

For small disturbances from trimmed flight, the elevator is used to control the measured altitude of the YAK-54 by observing the altitude and tracking a reference altitude. Again using a PI control structure, the altitude controller gives an elevator control of

$$\delta_e = 0.5(h - h_{command}) - 0.045 \int (h - h_{command}) dt. \tag{3.74}$$

Finally, the throttle is used to control the airspeed V_a of the aircraft for small disturbances and changes from trimmed level flight. The magnitude of the airspeed

is

$$V_a = (u^2 + v^2 + w^2)^{(1/2)}, \tag{3.75}$$

which is used to find the throttle control signal

$$\delta_T = -0.5(V_a - V_{a,\ command}) - 0.05 \int (V_a - V_{a,\ command})dt. \tag{3.76}$$

3.4.2 Aerovel Flexrotor

The development of the Flexrotor's controllers for thrust-borne flight closely follows the approach used for the YAK-54; however, different weights are used in solving the Riccati Equation. The controllers for each subsystem are of the form $u = -Kx$:

- The axial subsystem has weights

$$Q = \begin{bmatrix} 100 & 0 \\ 0 & 0.01 \end{bmatrix}, \quad R = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \tag{3.77}$$

  which admit a gain matrix of

$$K_{ax} = \begin{bmatrix} 0.93 & -0.01 \\ -2.78 & 0.03 \end{bmatrix}. \tag{3.78}$$

- The roll subsystem has weights

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 100 \end{bmatrix}, \quad R = 10, \tag{3.79}$$

  which admit a gain matrix of

$$K_{roll} = \begin{bmatrix} 1.65 & 3.16 \end{bmatrix}. \tag{3.80}$$

- The longitudinal subsystem has weights

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad R = 1000, \tag{3.81}$$

which admit a gain matrix of

$$K_{long} = \begin{bmatrix} 0.27 & 1.08 & 1.97 & -0.03 \end{bmatrix}. \tag{3.82}$$

- The lateral subsystem has weights

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \quad R = 100, \tag{3.83}$$

which admit a gain matrix of

$$K_{lat} = \begin{bmatrix} 4.52 & 4.03 & 20.8 & 0.316 \end{bmatrix}. \tag{3.84}$$

3.5 Open Loop Controllers

In order to develop open loop controllers for transition between wing-borne flight and thrust-borne flight, a few details must to be considered. First, the aircraft design plays a large role in choosing a desired trajectory. The YAK-54 has a much higher thrust-to-weight ration than the Flexrotor so it can accelerate faster and to higher speeds in the vertical direction, allowing a greater range of maneuvers. The Flexrotor, with its focus on long range and endurance is less agile, so specific care needs to be taken to develop transition maneuvers suitable for the aircraft. Second, the cost of transitions needs to be considered so that the maneuvers can be improved upon by a reinforcement or apprenticeship learning algorithm. Possible cost functions include minimizing time to transition, fuel or battery power consumed, or altitude lost during the transition. Finally, some hard constraints might exist, such as on altitude if operating in a confined area or a minimum altitude for the Flexrotor's thrust-borne launch/recovery.

3.5.1 YAK-54

With the YAK-54's high thrust-to-weight ratio, many possibilities exist for transitions to level flight. These possibilities include climbing in an accelerating arc to sustained wing-borne flight that would be accomplished with the throttle and a slow elevator ramp, while the ailerons and rudder are used to keep the wings level. A much faster maneuver could be a simultaneous power and pitch rate reduction to induce departure from thrust-borne flight, then use the elevator to recover into wing-borne flight.

For transition to hover from wings-level flight, a full throttle climb combined with a slow elevator ramp to put the YAK-54 into an accelerating vertical climb could be used. The aircraft would then reduce u velocity until stationary hover is reached; however, this maneuver would be an inefficient maneuver given its power and altitude consumption. A faster transition could be made by rapid pitch-up past the stall regime, using the aircraft's power to avoid departure from controlled flight and reach a hover configuration. This approach would require much less altitude, time, and power consumption.

3.5.2 Aerovel Flexrotor

At a disadvantage power wise when compared to the YAK-54, multiple choices are nevertheless present for transition to wing-borne flight for the Flexrotor. The first, a hammerhead transition [19] involves a slow climb in the vertical direction until sufficient altitude is gained. Then, using the rudder and the rotor cyclic in the y-body axis direction, a fast yaw would be commanded, resulting in the Flexrotor in a dive with its nose to the ground. A simple pull-up maneuver with the elevator would conclude the transition to level flight. Alternately, a twist-and-slew transition [19] would use the rotor cyclic to translate the aircraft in the y-body axis direction. As the aircraft gains speed, a twist performed with ailerons, rudder, and cyclic would rotate the Flexrotor into a wings-level configuration, at which point the wing-borne

control laws would dictate control inputs.

The Flexrotor's transition to hovering flight from wings-level flight could be accomplished in much the same manner as the YAK-54. By using the elevator, as well as the rotor cyclic in the z-body axis direction, a quick pitch up followed by an increase in throttle hover could be reached.

Ultimately, for both aircraft, it is desirable to have a human pilot demonstrate the transition maneuvers, while recording his/her commanded control inputs. These inputs should then be used in simulation with the apprenticeship-learning algorithm to determine a near-optimal transition for the aircraft.

3.6 General Control Framework

While control laws are specific to a vehicle, the ideas and approach used in this thesis combine to form an appropriate general framework for control of UAVs capable of sustained thrust-borne flight. The proposed framework can be separated into three categories.

3.6.1 Wing-Borne Flight

Control of the aircraft in the traditional flight modes of wing-borne flight should be executed according to the well-documented strategies in [30] and [25], and that are used in UAVs currently in operation. Feedback laws such as pitch attitude hold, altitude hold, bank angle hold, and heading hold can be used here.

3.6.2 Thrust-Borne Flight

For control of the agile UAV in thrust-borne flight, the approach taken by this thesis is recommended. By linearizing the aircraft dynamics about a hover trim point and subsequently decoupling the system, control laws for controlling $u,\ v,\ w$, and ϕ_v can be generated with LQ techniques, and if required, control gains can subsequently be tweaked to obtain desired controller performance characteristics.

3.6.3 Transitions

Either using a software simulation, a hardware simulation in the loop (HiL), or the UAV with a human pilot, successful open loop transitions to both thrust-borne and wing-borne flight should be captured. While these open loop control sequences will work to navigate between the two flight regimes, by applying the MDP formulation to the system and operating on the transitions with the apprenticeship learning algorithm, the control sequences' optimality can be improved.

Chapter 4
SIMULATION RESULTS

The motivating problem of controlling an agile autonomous aircraft capable of sustained thrust-borne flight is demonstrated here in simulation using the various controllers of the preceding chapter. Simulations for both the YAK-54 and the Aerovel Flexrotor are presented.

The chapter is composed of several sections, each focusing on demonstrating different controllers. In the first section, the thrust-borne flight controllers for both the YAK-54 and the Flexrotor will demonstrate the performance of each controller. Section 4.2 presents simulations for the wing-borne flight controllers of the YAK-54. Next, in Section 4.3, transition controllers are simulated for the YAK-54.

All simulations were done in `MATLAB` with `ode45` on an `Intel Pentium` equipped computer in the Autonomous Flight Systems Lab at the University of Washington.

4.1 Thrust-Borne Flight Controllers

The thrust-borne flight controllers using the LQ approach and controllers of Section 3.4 were used to simulate the dynamics of the agile aircraft systems.

4.1.1 YAK-54

The simulations for the YAK-54 were initialized from a trimmed position in thrust-borne flight at a location of [0 0 200] in the NED frame.

Figure 4.1 shows the throttle response to a commanded 2 meter per second step input in u velocity. The throttle does saturate to max throttle levels, and then returns to a setting slightly higher than the trim setting. The saturation does not

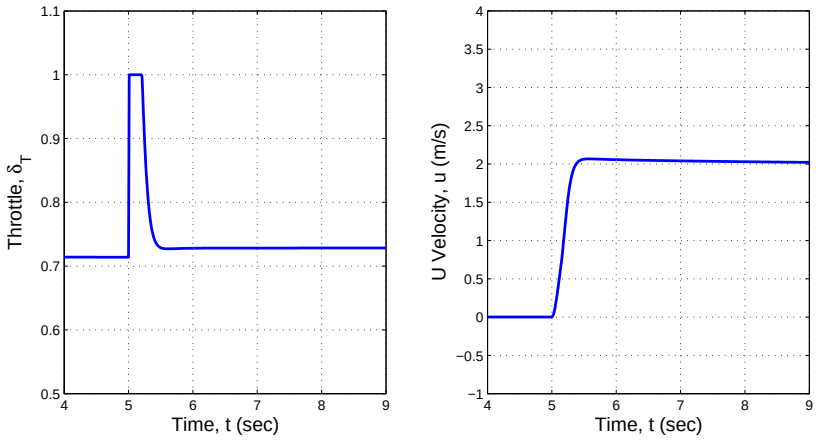

Figure 4.1: Nonlinear U-velocity controller response to a 2 m/s step input in u velocity for the YAK-54 demonstrating a fast response with a 5% overshoot.

cause concern, as it is for a short duration of less than half a second. Additionally, increasing vertical speed is a "maximum performance" type of maneuver, and while the YAK-54 has a sufficiently high thrust-to-weight ratio, accelerating in the vertical direction requires significant thrust. The response in u velocity is nicely shaped, with a 5% overshoot and fast rise and settling times.

Figure 4.2 depicts the response of the vertical roll rate controller for the YAK-54. At five seconds, a 30 degree step input in commanded ϕ_v occurred. As with the throttle step response, the aileron fully saturates for a short period of time to its maximum deflection of -35 degrees before settling back down to the trim setting necessary to counteract the rotational effects of the motor and prop wash. The response in vertical roll angle demonstrates no oscillations as 30 degrees is reached.

Figure 4.3 displays the elevator response of a -3 m/s step input in w velocity for the W-velocity controller. Following a quick deflection, the elevator settles to a value of 10 degrees, necessary for sustained -3 m/s translation in the body z-axis. The controller must overcome the inertial and wind resistance of the wing before settling

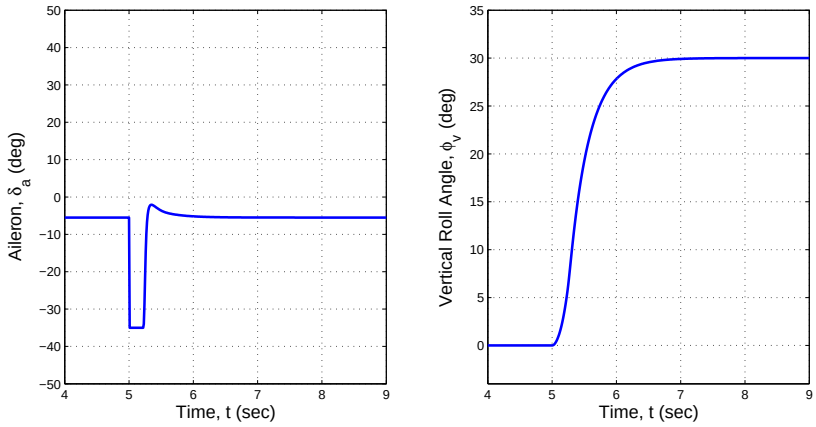

Figure 4.2: Nonlinear vertical roll rate controller response to a 30 degrees step input in commanded vertical roll angle ϕ_v for the YAK-54. The control signal is characterized by a fast response with no measurable oscillations.

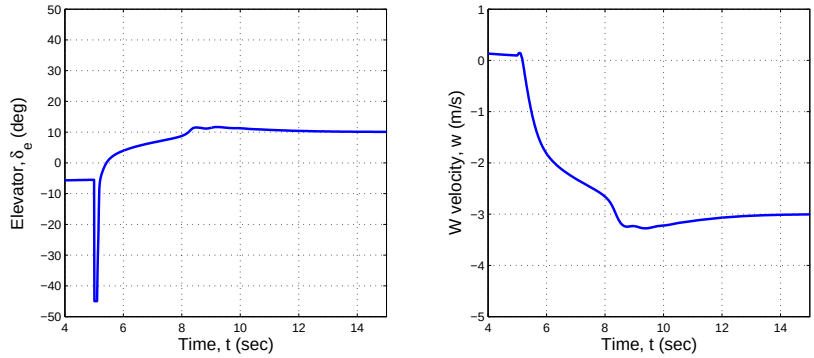

Figure 4.3: Nonlinear W-velocity controller response to a -3 m/s step input in w velocity for the YAK-54 with a 10% overshoot.

to the commanded value with a 10% overshoot.

Figure 4.4 shows the response of the YAK-54's V-velocity controller to a commanded step input of 3 m/s in v velocity at 5 seconds. The control signal dictates a fast rudder doublet before reaching a trimmed value of 11 degrees of deflection. The

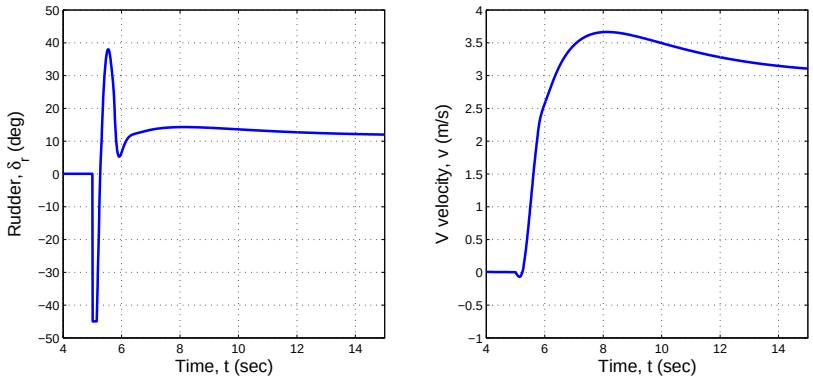

Figure 4.4: Nonlinear V-velocity controller response to a 3 m/s step input in v velocity for the YAK-54 with a 21% overshoot.

doublet is necessary to initiate the tracking of the reference velocity. The response in v velocity has a 21% overshoot and the slowest settling time of the four controllers for the YAK-54.

While each of the four controllers can track its commanded reference signal, all four must also be able to track multiple commanded values simultaneously. Figure 4.5 shows the input traces for one such simulation and Figure 4.6 shows the corresponding control traces for the same simulation. As before, the YAK-54 was initialized at [0 0 200] in the NED frame, and at five seconds, step inputs were commanded. The multiple step inputs were -2 m/s in v velocity, 2 m/s in w velocity, and 15 degrees in vertical roll angle, ϕ_v. The vehicle does successfully track the commanded inputs, and its position moves to the northwest, ending at -25 meters east and 13 meters north at the end of the 15 second simulation. The YAK-54 also loses 3.5 meters of altitude in this simulation. The U-velocity controller continues to track the commanded value of 0 m/s, so the YAK-54 gradually drops in altitude as its overall velocity vector, $V_a = [u\ v\ w]^T$, points below the horizon.

The control traces of Figure 4.6 depict the necessary control surface deflections

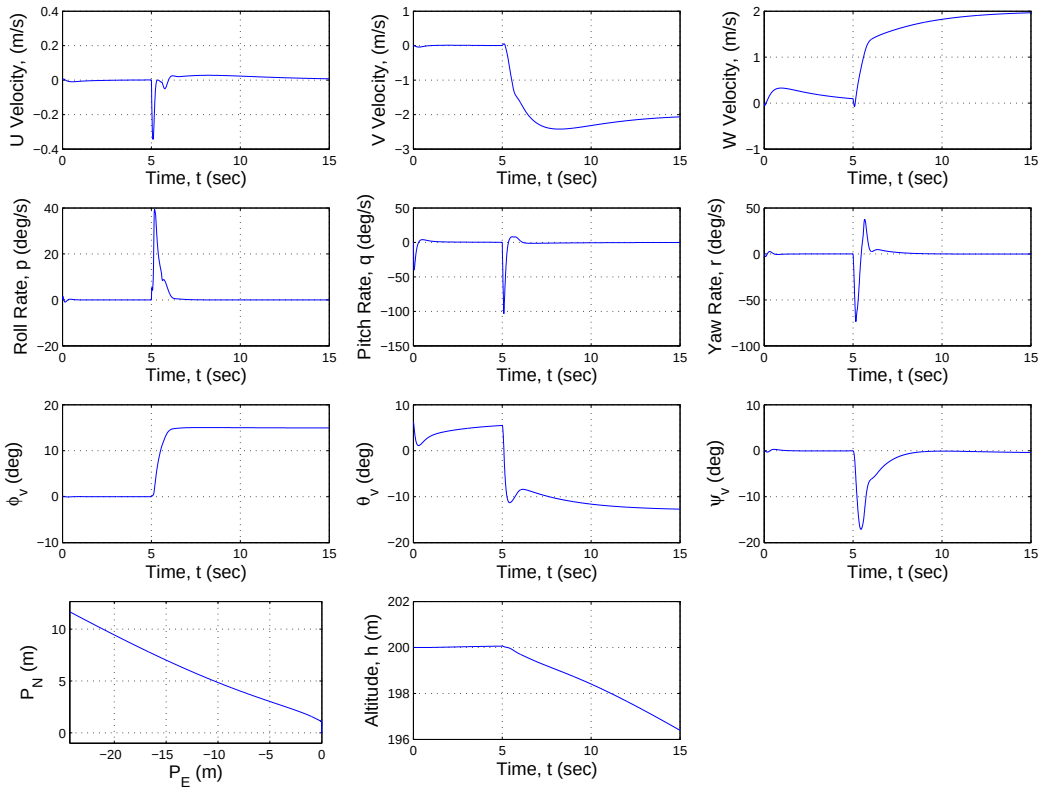

Figure 4.5: State traces for the YAK-54 in thrust-borne flight showing aircraft response to three commanded step inputs occurring at five seconds. The step inputs are -2 m/s in v velocity, 2 m/s in w velocity, and 15 degrees in vertical roll angle. The simulation was initialized at [0 0 200] in the NED frame.

to track the commanded step inputs, as well as settling to new trim values. After the step inputs, the elevator reaches a new trim state of -15 degrees, after a positive deflection from its previous state of -7 degrees. The aileron affects the vertical roll rate necessary to track the input, and then returns to its trim value of just over -5 degrees of deflection. The small difference in the trim values of the aileron setting is due to the v and w components of aircraft velocity affecting the tendency to rotate the YAK-54 about its x-body axis. The rudder, initially trimmed at zero degrees,

reaches a new trim state of -9 degrees. Finally, the throttle setting only drops slightly as the U-velocity controller continues to track a zero u velocity.

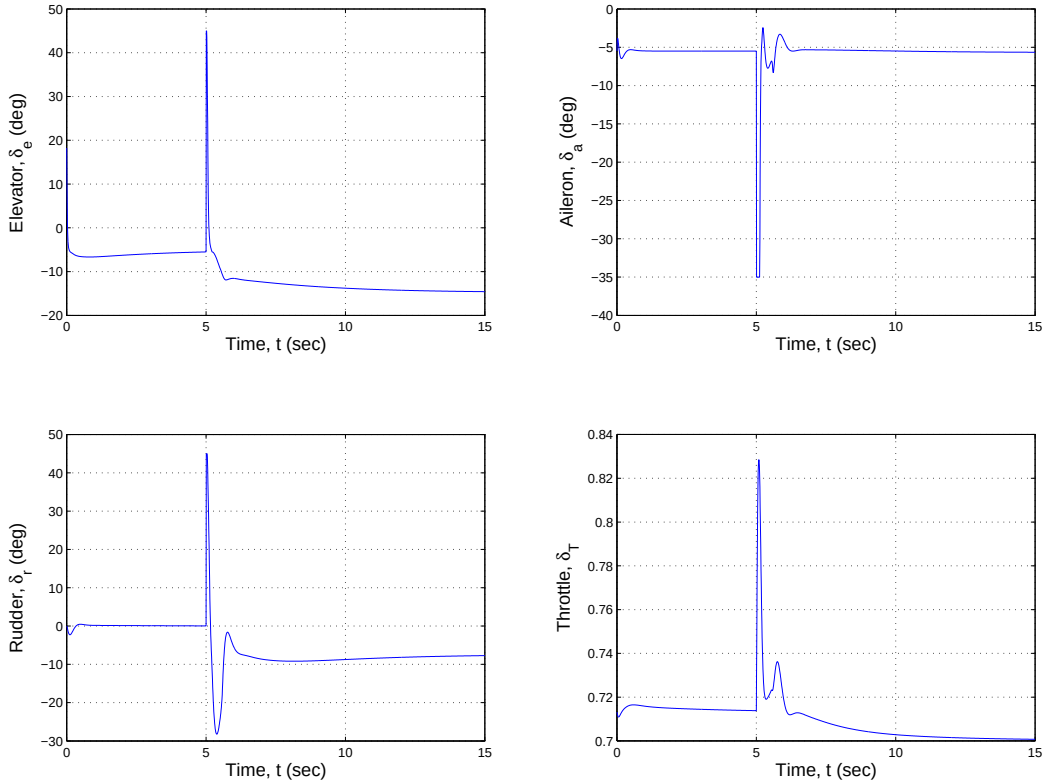

Figure 4.6: Control traces for the YAK-54 in thrust-borne flight necessary to track three commanded step inputs occurring at five seconds and corresponding to the state traces in Figure 4.5. The step inputs are -2 m/s in v velocity, 2 m/s in w velocity, and 15 degrees in vertical roll angle. The simulation was initialized at [0 0 200] in the NED frame.

4.1.2 Aerovel Flexrotor

The simulations for the Aerovel Flexrotor and its four thrust-borne flight controllers developed in Section 3.4 used the linear system, as the non-linear dynamics were not

accessible in the Matlab/Simulink environment. While the controllers were developed for the diagonalized and decoupled linear system, the full linear system was used in simulation. Additionally, rather than the diagonalized control inputs $dp'c$, $dq'c$, and $dr'c$, the mixing matrix was used to map to the physical control most closely associated with each controller.

Figure 4.7 demonstrates the Flexrotor's response to a 2 m/s step input in vertical velocity. The U-velocity controller does not exhibit any overshoots, while exhibiting a fast response. As the Flexrotor has a smaller thrust-to-weight ratio compared to the YAK-54, accelerating in the vertical direction is a difficult maneuver, and the response shown is too fast for the physical system.

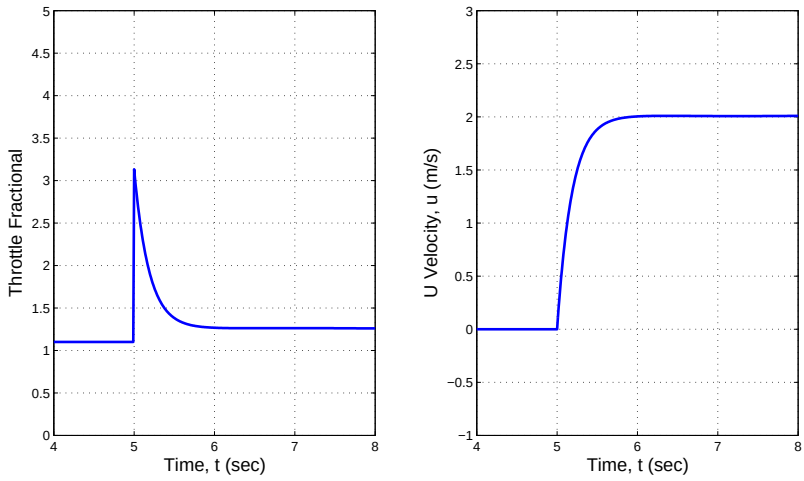

Figure 4.7: Linear U-velocity controller response for the Aerovel Flexrotor to a 2 m/s step input in commanded u velocity. The response does not exhibit any overshoot.

Figure 4.8 shows the tip thruster response to a 30 degree step input in commanded vertical roll angle. The vertical roll rate controller commands slightly over 18 watts of power for the thrusters to track the step input, out of an available 75 watts. The response in vertical roll angle does not have any overshoot and settles to the

commanded value in 7.5 seconds.

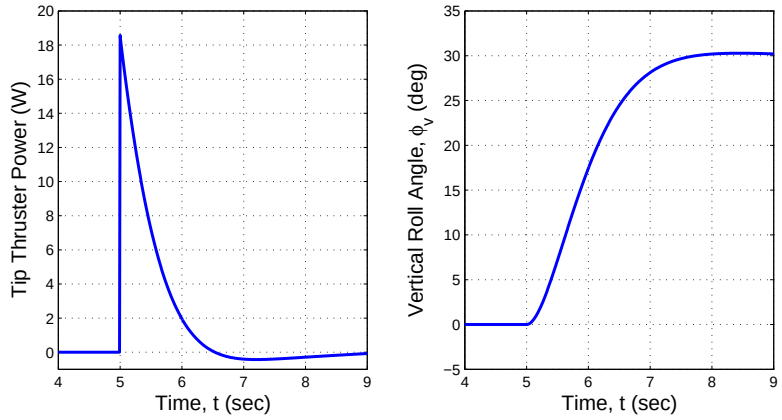

Figure 4.8: Linear vertical roll rate controller response for the Flexrotor to a 30 degrees step input in commanded vertical roll rate ϕ_v.

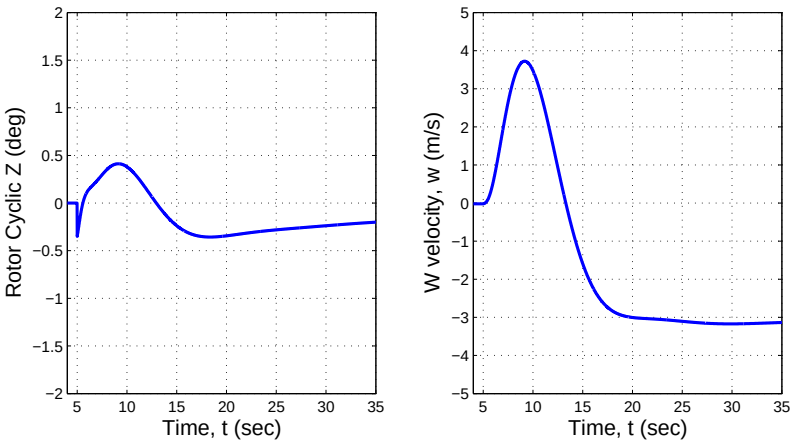

Figure 4.9: Linear W-velocity controller response for the Flexrotor to a $-3\ m/s$ step input in w velocity.

Figure 4.9 demonstrates the response of the hybrid rotor/propeller of the Flexrotor to a commanded step input in w velocity. The W-velocity controller commands small

changes for the cyclic in the z-body axis direction to effect the necessary change in velocity. To effect the desired response, the controller must first reach positive w velocity before it is able to settle to the commanded velocity at 18 seconds.

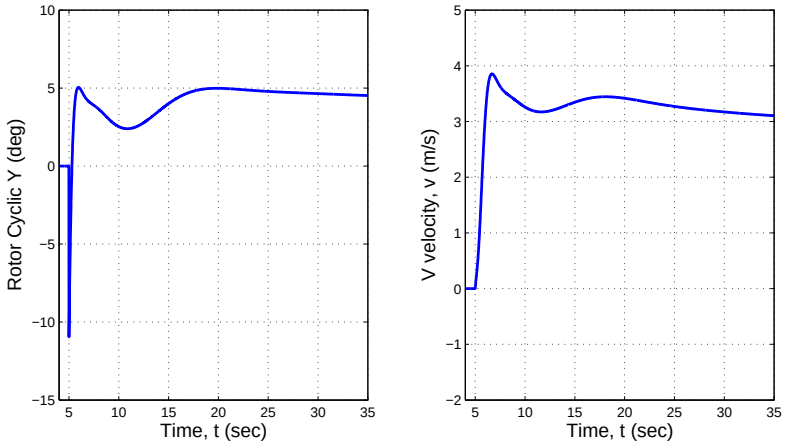

Figure 4.10: Linear V-velocity controller response for the Flexrotor to a 3 m/s step input in v velocity with a 21% overshoot.

Figure 4.10 depicts a much faster response via the cyclic in the y-body axis direction to a commanded step input in v velocity. The response of the V-velocity controller has a 30% overshoot, and three oscillations before settling to the commanded value of 3 m/s.

4.2 Wing-Borne Flight Controllers

The response of the wing-borne level flight controllers for the YAK-54 is shown in Figures 4.11 and 4.12. This full-state nonlinear simulation was initialized at [0 0 200] in the NED frame, and depicts the aircraft response as the PI controllers reach a trimmed state.

The state traces show small oscillations in the aircraft velocities, orientation, and altitude. Additionally, the YAK-54 tracks slightly the west of north in the simulation,

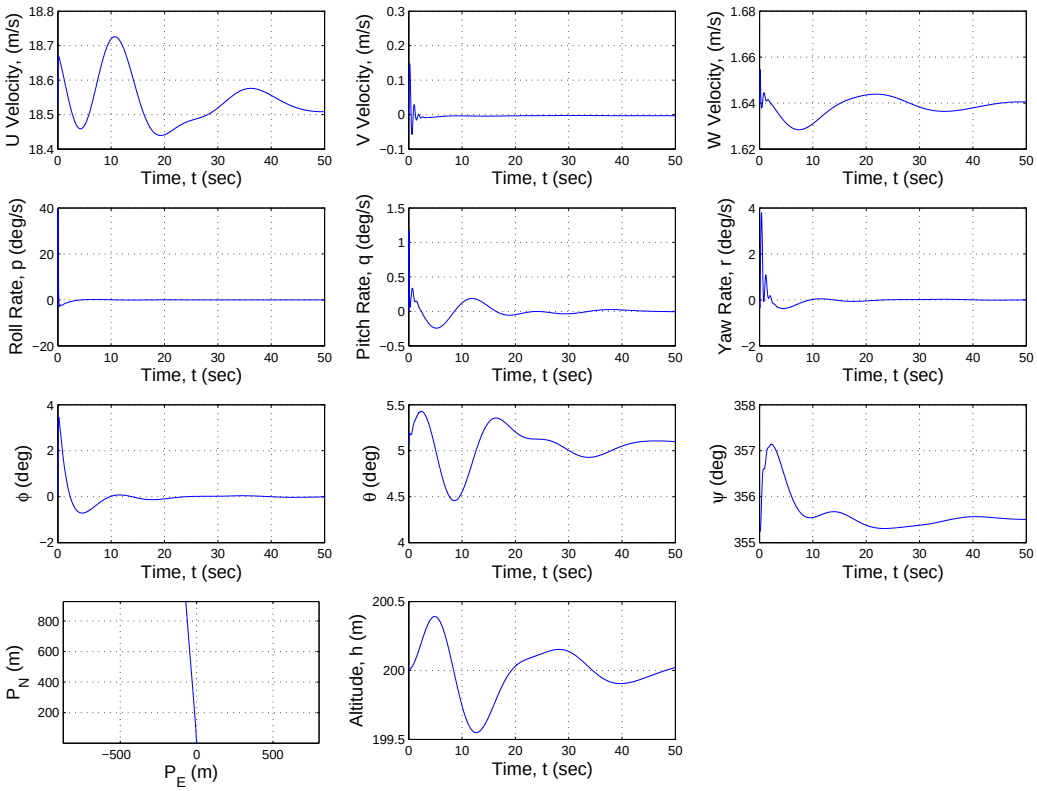

Figure 4.11: State traces for the YAK-54 in wing-borne flight showing aircraft response as the wing-borne flight controllers settle the aircraft into a wings-level flight trimmed condition. The simulation was initialized at [0 0 200] in the NED frame.

following a heading angle of 355.5 degrees. This heading change results from the increase in bank angle after the simulation begins, and the heading angle settles to steady state as the bank angle returns to zero degrees. The coupling between pitch angle and altitude can also be seen, as the oscillations are quite similar as the signals settle to steady-state.

The control traces of Figure 4.12 demonstrate oscillations that are small in magnitude as well. The elevator deflection varies between -6.38 and -6.51 degrees over 50 seconds, while the aileron varies between 0.05 and -0.14 degrees. The changes in

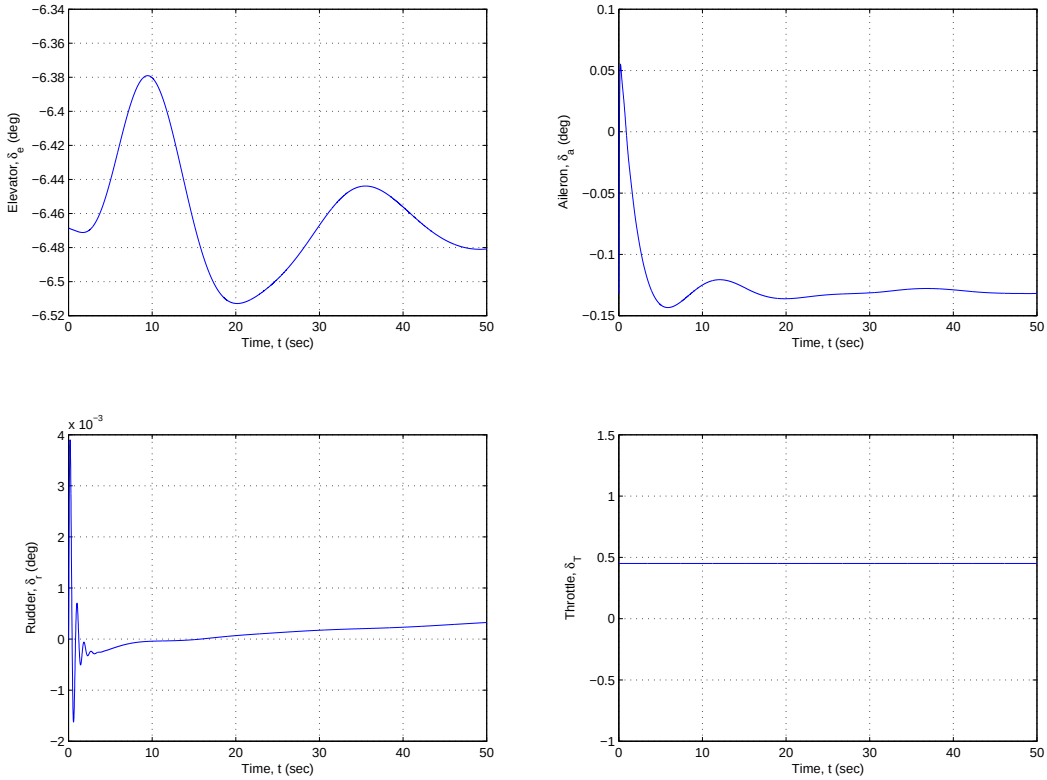

Figure 4.12: Control traces for the YAK-54 in wing-borne flight corresponding to the state traces in Figure 4.11. The simulation was initialized at [0 0 200] in the NED frame.

rudder and throttle are negligible.

Chapter 5
CONCLUSION

5.1 Summary

This thesis has addressed the problem of autonomous control of the class of agile aircraft capable of performing sustained thrust-borne flight by looking specifically at two different platforms, the University of Washington YAK-54 and the Aerovel Flexrotor. The approach taken was built upon new and existing examinations of thrust-borne control. First, the problem of system dynamics and modeling was addressed. Then, the systems were linearized and decoupled so that linear controllers could be developed for thrust-borne flight. For the YAK-54, level flight was also considered. Finally, open loop control methods were examined for both aircraft. Simulation results were shown to support the developed controllers.

These results present a new approach to the control of vehicles capable of autonomous hover, in addition to examining a radical new concept in the Flexrotor. Prior work was varied and many different approaches and techniques discussed. This thesis aimed to examine previous work and validate previous techniques while proposing a general framework for problems of this nature.

5.2 Future Work

The work in this thesis is limited in scope as no tests involving hardware-in-the-loop or actual flight test were performed. While the work presented here is a necessary step towards implementing and testing these controllers on the YAK-54, much work remains to be done. For example, the wing-borne flight controllers need more development and a guidance and navigation scheme for level flight must be considered.

Additionally, the YAK-54 must be fully instrumented and prepared for autonomous flight (see Figure 5.1), with all controllers finalized, including the near-optimal transition sequences found via apprenticeship learning.

Figure 5.1: The University of Washington YAK-54 agile aircraft.

In order to progress to flight test, the suggested approach is to develop a tether system to test the hover controllers prior to untethered autonomous flight. Additionally, a fail-safe switch to return control of the YAK-54 to a human pilot should be considered. Successful flight testing of the YAK-54 using apprenticeship learning sequences would represent the first instance of the algorithm being used for autonomous aircraft aerobatics, though the apprenticeship learning approach has been shown to work for an autonomous helicopter.

Another direction for future work would be to implement a target tracking ability that would require transition between flight regimes. An extension could also be the use of outer loop search algorithms, such as the one presented in [13] and [14]. Further application of autonomous hover control techniques to specific problems such as search, constricted space maneuvers, or object avoidance would validate the approach discussed in this thesis.

BIBLIOGRAPHY

[1] Pieter Abbeel. *Apprenticeship Learning and Reinforcement Learning with Application to Robotic Control*. PhD thesis, Stanford, 2008.

[2] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y. Ng. An application of reinforcement learning to aerobatic helicopter flight.

[3] Pieter Abbeel, Varun Ganapathi, and Andrew Y. Ng. Learning vehicular dynamics, with application to modeling helicopters.

[4] Pieter Abbeel and Andrew Y. Ng. Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

[5] Panos J. Antsaklis and Anthony N. Michel. *A Linear Systems Primer*. Birkhauser, 2007.

[6] Pedro Castillo, Rogelio Lozano, and Alejandor E. Dzul. *Modeling and Control of Mini-Flying Machines*. Springer-Verlag, 2005.

[7] Peter Dorato, Chaouki T. Abdallah, and Vito Cerone. *Linear Quadratic Control, An Introduction*. Krieger Publishing Company, 2000.

[8] Adrian Frank, James McGrew, Mario Valenti, Daniel Levine, and Jonathan P. How. Hover, transition, and level flight control design for a single-propeller indoor airplane. Technical report, Massachusetts Institute of Technology, 2007.

[9] William E. Green and Paul Y. Oh. A mav that flies like an airplane and hovers like a helicopter. In *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2005.

[10] Eric N. Johnson, Michael A. Turbe, Allen D. Wu, Suresh K. Kannan, and James C. Neidhoefer. Flight test results of autonomous fixed-wing uav transitions to and from stationary hover. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.

[11] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time.

[12] Kyle Krogh. Linear analysis of research civil aircraft model, 2007.

[13] Christopher W. Lum. *Coordinated Searching and Target Identification Using Teams of Autonomous Agents*. PhD thesis, University of Washington, Seattle, WA, March 2009.

[14] Christopher W. Lum and Juris Vagners. A modular algorithm for exhaustive map searching using occupancy based maps. In *To appear in Proceedings of the 2009 Infotech@Aerospace Conference*, Seattle, WA, April 2009.

[15] Wolfram Martens. Hover dynamics, July 2008.

[16] Barnes W. McCormick. *Aerodynamics of V/STOL Flight*. Dover Publications, 1999.

[17] Tad McGeer. Clothesline model, October 2008.

[18] Tad McGeer. Personal correspondence, 2008.

[19] Tad McGeer and Andy von Flotow. Flexrotor: Miniature robotic aircraft with long range, long endurance, and vtol, October 2007. First semi-annual progress report.

[20] Tad McGeer and Andy von Flotow. Flexrotor: Miniature robotic aircraft with long range, long endurance, and vtol, April 2008. Second semi-annual progress report.

[21] Roberto Naldi, Ricadro G. Sanfelice, Michelangelo Graziano, and Emilio Frazzoli. Robust hybrid control algorithms for aggressive multi-point hovering of airplanes.

[22] NationMaster. Website, http://www.nationmaster.com/encyclopedia/tailsitter.

[23] Stuart Russel and Perter Norvig. *Artificial Intelligence, A Modern Approach*. Prentice Hall, 1995.

[24] RealFlight R/C Flight Simulator. Website, http://realflight.com/.

[25] Brian L. Stevens and Frank L. Lewis. *Aircraft Control and Simulation*. John Wiley and Sons, 2nd edition, 2003.

[26] H. Stone. Modeling, simulation and control design for the t-wing tail-sitter uav. In *4th Australian Pacific Vertiflite Conference on Helicopter Technology*, Melbourne, July 2003.

[27] H. Stone and K.C. Wong. Preliminary design of a tandem-wing tail sitter uav using multi-disciplinary design optimization.

[28] Defense Update. Website, http://www.defense-update.com/.

[29] Wikipedia. Website, http://en.wikipedia.org/wiki/yakovlev_yak-54.

[30] Thomas R. Yechout. *Introduction to Aircraft Flight Mechanics: Performance, Static Stability, Dynamic Stability, and Classical Feedback Control*. American Institute of Aeronautics and Astronautics, 2003.