Control of Rotorcraft at Low Speed

Kristoffer Raul Gauksheim

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

Doctor of Philosophy

University of Washington

2021

Reading Committee:

Juris Vagners, Chair

Brian T McGeer

Mehran Mesbahi

Program Authorized to Offer Degree:
Aeronautics & Astronautics

University of Washington

Abstract

Control of Rotorcraft at Low Speed

Kristoffer Raul Gauksheim

Chair of the Supervisory Committee:
Juris Vagners
Aeronautics and Astronautics

Nonlinearities, uncertainties, and disturbances challenge automatic control systems. For rotorcraft at low speed, nonlinearities can include reversals in control trim and effectiveness gradients with airspeed. These phenomena are uncertain as is the three-dimensional airflow field on which they depend. Furthermore, wind gusts and aircraft motion directly affect airflow. In this dissertation, a nonuniform rotor induced velocity model is adapted and applied, resulting in control gradient reversals and qualitative agreement with flight data. There has been little previous mention in the literature of cyclic effectiveness gradient reversal. Trim feedforward control and model reference adaptive control are developed and tested in simulation. Ultimately, incremental nonlinear dynamic inversion (INDI) control is applied in simulation and flight of a low disk loading tailsitter with a 2 m diameter rotor. The induced velocity model is used in simulation, providing consistency in controller behavior between simulation and flight. A significant result is that attitude control performance with the INDI controller is markedly better than with the existing controller. The INDI controller promises to make landing onto a small deck reliable in winds and gusts extending across the flight envelope, whereas the existing adaptive controller performs marginally in moderate but gusty wind.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

ACKNOWLEDGMENTS

Many thanks to Juris Vagners for advising me through my time in graduate school. I am very grateful to Tad McGeer for his guidance and support on this research and throughout my career. Thank you to Mehran Mesbahi for review of this work, teaching, and inspiration. Thanks also to the other faculty that I had the pleasure of learning from, especially those who served on my supervisory and examination committees: Behçet Açıkmeşe, Howard Jay Chizeck, Antonino Ferrante, Eli Livne, and Kristi Morgansen.

I would like to thank the staff of the William E. Boeing Department of Aeronautics & Astronautics, the Autonomous Flight Systems Laboratory, and the fellow students that I collaborated with and learned from.

Thank you to Aerovel Corporation for the time, funding, and other resources that made this work possible. I also wish to acknowledge those who provided advice and assistance, particularly Rutger Engbersen, Andy von Flotow, Christopher Lum, and Mathieu Mercadal.

Finally, special thanks to my family for their support and inspiration, especially Amy, my partner in engineering and life.

DEDICATION

To my family.

Chapter 1
INTRODUCTION

1.1 Research Problem

Rotorcraft such as helicopters are used because of their ability to fly efficiently at low airspeed, enabling takeoff, landing, and other operations in small and unprepared spaces. There is current interest in new rotorcraft applications and configurations, e.g. the NASA Mars Helicopter [30] and the U.S. Army Future Tactical Unmanned Aircraft System [4]. This dissertation addresses control system design for the following problems that sometimes occur with rotorcraft at low airspeed:

1. Nonlinearities
   (a) Strong nonlinear dependence of aircraft loads (i.e. forces and moments) on airspeed, including control trim gradient reversal
   (b) Nonlinear dependence of control effectiveness on airspeed, including gradient reversal
2. Uncertainties in
   (a) Model: Uncertainty in aircraft loads and control effectiveness as functions of airspeed and other variables
   (b) State: Uncertainty in the three-dimensional airflow field on which aircraft loads and control effectiveness depend
3. Disturbances: Wind and aircraft motion directly affect airflow and therefore aircraft loads and control effectiveness

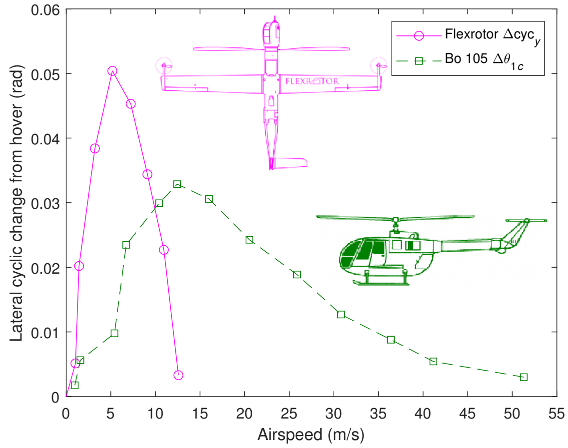


Figure 1.1: Lateral cyclic control angles in level trimmed flight for Flexrotor tailsitter [52] and Bo 105 helicopter [60] flight tests. Flexrotor data are for translation to the left (with 1.85 m diameter rotor, 16 kg, 560 m altitude, and legs closed); cyclic control angles for translation to the right are similar (with opposite sign). Drawings are from [1] and [12].

Examples of problem 1a (strong nonlinear dependence of aircraft loads on airspeed, including control gradient reversal) is shown in Figure 1.1 for an MBB Bo 105 helicopter (9.82 m main rotor diameter, 2200 kg) and an Aerovel Flexrotor tailsitter (1.85 m main rotor diameter, 16 kg). Lateral cyclic trim for Flexrotor is more sensitive to airspeed changes than for the Bo 105 (e.g. peak change from hover around 5 m/s and return to 10% of the peak change from hover at <13 m/s for the Flexrotor compared to 12 m/s and >41 m/s, respectively, for the Bo 105). This indicates why measures not necessary for a helicopter like the Bo 105 might be necessary for an aircraft like Flexrotor. Note that rotor induced velocity for the Bo 105 is approximately twice that for Flexrotor (i.e. using momentum theory [42]). The cyclic control amplitudes of Figure 1.1 are significant, e.g. for Flexrotor a 0.01 rad cyclic step from hover trim can result in a peak pitch acceleration of $0.6\,\mathrm{rad/s^2}$ or more [49].

Unlike the trim reversal of problem 1a—which is known to helicopter researchers [60, 40, 42] and pilots [23]—the control-effectiveness reversal of problem 1b is little discussed in the

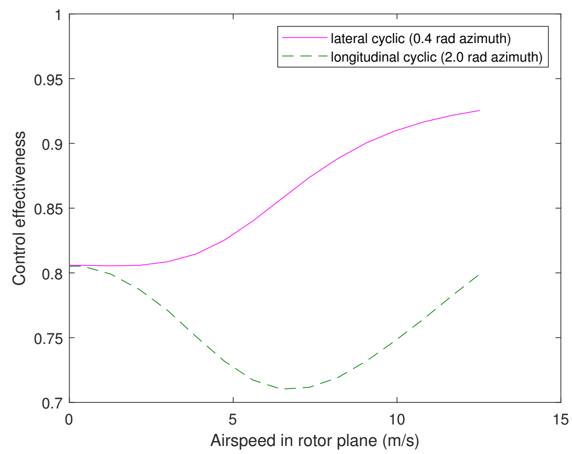


Figure 1.2: Flexrotor cyclic control effectiveness (moments at rotor hub) with Pitt-Peters [64] rotor induced velocity model (see Modeling chapter). Control effectiveness is relative to that with a blade-element model. The calculations are with 1.85 m diameter rotor, 16 kg, 560 m altitude, and legs closed. Note that at zero airspeed (i.e. hover), effectiveness is 0.81. This is qualitatively consistent with ground apparatus measurement of 0.53 ± 0.03 (at 89% rotor speed) [49].

literature. Nevertheless, it can be predicted with well-established inflow (i.e. rotor induced velocity) theory, as shown in Figure 1.2. Independently, control effectiveness qualitatively similar to that in Figure 1.2 was hypothesized for Flexrotor as an explanation for problematic and even dangerous oscillations that were observed in flight but not in simulation [48].

An example of uncertainty in trim (part of problem 2a) is shown in Figure 1.3. Simulation is qualitatively correct but underestimates lateral cyclic despite including an induced velocity model with azimuthal variation (the Glauert trapezoidal model) [60]. Such discrepancies can have significant consequences. For example, the first Sikorsky XH-59A Advancing Blade Concept coaxial helicopter was damaged in 1973 due to insufficient cyclic authority at low speed [67]. Recently, vacuum chamber measurements indicated that Mars Helicopter moments due to edgewise flow were significantly larger than modeled, with the discrepancy

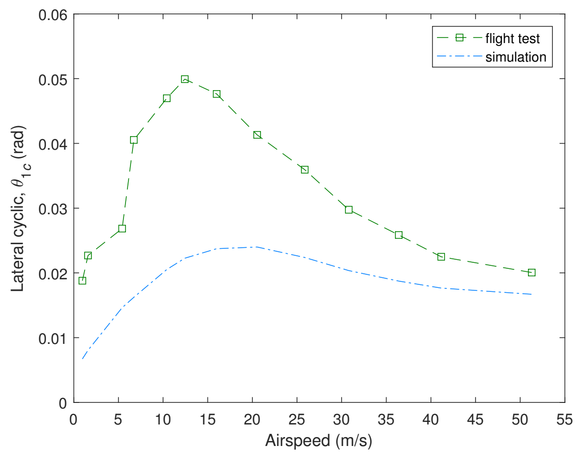


Figure 1.3: Lateral cyclic control angles in level trimmed flight for Bo 105 helicopter flight test and *Helisim* simulation (partial reproduction of Fig. 4.10 in [60]).

then resolved by amplifying induced velocity gradient sensitivity to edgewise flow by approximately 2.5 [30].

For problem 2b, uncertainty in the airflow field, pitot-static sensors are ineffective at low speed [22]. While there are three-dimensional anemometers that have been mounted on miniature rotorcraft [5], the associated weight and drag might not be acceptable for aircraft such as Flexrotor. Furthermore, it is unclear whether measurement at a single location is sufficient for accurately calculating loads and control effectiveness. The airflow field is affected not only by the rotor wake but also airframe components including, for Flexrotor, a high aspect ratio wing.

To motivate problem 3 (wind and aircraft motion directly affecting airflow)—as well as the entire research problem—consider the practical and pressing requirement to land on a small boat in strong, gusty wind and stormy seas (Figure 1.4). In addition to wind and boat air wake, motion of the rotorcraft as it tracks the rotating and translating boat directly affects three-dimensional airflow and therefore aircraft loads and control effectiveness.

Figure 1.4: Westland Lynx helicopter (12.80 m main rotor diameter [60]) preparing to land on a ship in difficult conditions [16]. Video of this landing is at https://youtu.be/bC2XIGMI2kM. Smaller aircraft have the potential to operate from smaller boats, which typically have more-challenging motion and more-confined spaces than larger ships [9, 24].

1.2 Rotorcraft Control Background

Rotorcraft at low speed have instabilities, inter-axis coupling, and high-order dynamics [77]. Furthermore, as observed by Faulkner and Buchner [22], "...in hover [and] low speed...only small changes in airspeed are rapidly followed by substantial changes in control sensitivities, couplings and trim positions." Pilots train to adapt to these challenges (e.g. Figure 1.4). This includes the lateral control gradient reversal of Figure 1.1, known to pilots as *transverse flow effect* [23].

For automatic control of rotorcraft, classical architectures such as proportional-integral-derivative (PID) and explicit model-following have traditionally been used [77]. Explicit model-following, which uses a command model and lower-order inverse plant feedforward, is commonly used in modern rotorcraft [77]. PID controllers are in use on helicopters including the unmanned MQ-8B Fire Scout [17] and are used as examples and for performance comparisons [77, 72, 70].

Control techniques that recently have been applied to rotorcraft include H_∞ [28], μ

synthesis [28], backstepping [65], sliding mode control [31], model predictive control [29], and nonlinear dynamic inversion (NDI) [73].

Geluardi et al. [28] evaluated H_∞ and μ synthesis control methods with a linear model of the Robinson R44 helicopter (10.06 m main rotor diameter, 1015 kg) in hover [27]. Both approaches provided good performance, and μ synthesis stabilized against model parametric uncertainties. However, airspeed-dependent nonlinearities were not considered and the μ synthesis controller is high order and thus harder to implement and more computationally intensive [28].

Raj et al. [65] used backstepping to design a robust attitude-tracking controller. However, this controller requires flap angle feedback. To avoid this requirement, a "structure-preserving, passively robust controller" was developed and tested in flight of an Align Trex 700 electric helicopter (1.5 m main rotor diameter, 6 kg). Wind was not explicitly considered, and the tested maneuvers did not significantly vary trim airspeed (assuming that wind was light).

Halbe and Oza [31] developed a continuous second-order sliding mode controller. Controller performance and robustness were evaluated with a linear model of the Bo 105 in hover augmented with uncertainties and disturbances. Controls were changes from trim and none of the tested uncertainties or disturbances were dependent on airspeed. The second-order sliding mode controller demonstrated better performance than a linear quadratic controller.

Greer and Sultan [29] used high-fidelity helicopter and ship motion models to evaluate model predictive control for landing of a SA 330 Puma helicopter (15.0 m main rotor diameter, 5805 kg) on a ship. The helicopter model included 44 states and Pitt-Peters linear static induced velocity. For controller design, the model was linearized and discretized. Computation time was significant: approximately 90 minutes on a consumer computer for a 3.75 s landing maneuver simulation.

Soneson et al. [73] tested NDI controllers in simulation of the GENHEL-PSU model of a UH-60 helicopter (16.36 m main rotor diameter [35]). The controller used a six-degree-of-freedom nonlinear dynamic model. Stability derivatives, control derivatives, and trim states were scheduled with airspeed. Controller tracking performance was good in testing with the full simulation model including higher-order dynamics and turbulence.

Continuing with dynamic inversion control and adding adaptive control to handle uncertainty, model reference adaptive control (MRAC) based on dynamic inversion has been tested on over 1000 flights with many miniature aircraft [74]. In addition, this approach has been deployed in guided munitions, perhaps the only adaptive flight control system to reach production [58]. For helicopters, this approach has been successful in outdoor flight for rotational and translational control [38].

Incremental nonlinear dynamic inversion (INDI) replaces much of the state model of NDI with sensor measurements, increasing robustness [72]. INDI controllers can be simple (conceptually and computationally) and effective to control rotational and translational acceleration [76]. INDI rotational and translational controllers for quadrotors have demonstrated robustness to large disturbances including wind [76, 71, 72]. In addition to quadrotors, INDI controllers have been developed for low disk loading rotors with collective and cyclic (e.g. conventional helicopters) [61, 15, 70, 69].

Simplício et al. [69] applied INDI to helicopter control and tested the resulting controllers with a Bo 105 model. Pavel et al. [61] applied INDI to an AH-64 Apache helicopter (14.63 m main rotor diameter [57]) model, demonstrating an improvement in handling qualities in piloted simulation. Slinger [70] demonstrated INDI attitude control in flight tests of the Walkera Genius CP V2 helicopter (0.24 m main rotor diameter, 0.05 kg). Unlike typical INDI attitude control implementations—including the one in this dissertation—Slinger used rotational rates instead of rotational accelerations. De Wagter et al. [15] applied INDI to position control of the DelftaCopter tailsitter (1.00 m main rotor diameter, 4.3 kg) but used a PID controller for attitude control.

This dissertation might describe the first flights of INDI rotational acceleration control for a rotorcraft with cyclic control (e.g. not a quadrotor [76, 71, 72] or rotational rate control of a 0.05 kg helicopter [70]).

1.3 Dissertation Overview

In this dissertation, nonuniform induced velocity is modeled to gain understanding into problems 1a and 1b, to provide an effective environment for controller testing, and for possible use in control. The resulting cyclic effectiveness gradient reversal with airspeed has been little discussed in the literature. Modeling results are compared to trim data for the Bo 105 helicopter and the low disk loading Flexrotor tailsitter. Different control approaches are developed and applied to the problem. The resulting controllers are tested in simulation and ultimately in flight of Flexrotor. The induced velocity model is used in simulation, providing consistency in controller behavior between simulation and flight.

Contributions and results include

- Modeling (Modeling chapter)
  - Rotor induced velocity modeling
    - Pitt-Peters effects are applied, introducing nonmonotonic variations of cyclic trim and effectiveness with airspeed which qualitatively match flight data (problems 1a and 1b). Trim reversal is well known but there is little mention in the literature of effectiveness gradient reversal.
    - A new algorithm is developed and used to apply the Pitt-Peters effects. This algorithm can be useful for rotor models that cannot directly use nonuniform induced velocity (e.g. a rotor performance lookup table) or when computational efficiency is important (e.g. for online use or simulation at rates faster than real time).
  - Lift deficiency functions for rotor moments
    - A correction is made to lift deficiency functions for moments, revealing nonmonotonic control effectiveness vs airspeed (problem 1b).
    - The corrected lift deficiency functions are consistent with control effectiveness variations using the Pitt-Peters algorithm, providing a simpler effectiveness

calculation.

- Trim feedforward and three-dimensional airspeed measurement (Trim Feedforward chapter)

  – Poor tilt regulation with the existing Flexrotor controller is almost entirely corrected in simulation by adding feedforward of estimated steady-state control as a function of three-dimensional airspeed. This demonstrates that poor regulation is due to trim variations with airspeed (problems 1a and 1b).

  – Simulation indicates that performance with trim scale errors (problem 2a) might be acceptable (e.g. by attenuating the nominal feedforward signal). However, performance degrades with small airspeed errors (problem 2b), likely requiring accurate three-dimensional airspeed sensors. A survey of such sensors reveals promising candidates, but weight, drag, complexity, and robustness in service are not acceptable for all aircraft. Furthermore, candidate sensors would not measure potentially important variations in flow across the rotor and wing.

  – The use of a three-dimensional airspeed sensor allows trim feedforward also to provide wind disturbance attenuation (problem 3).

  – Algorithms are developed for online trim estimation—including "local" forgetting of old data—reducing model uncertainty (problem 2a). Use of the online trim estimate results in an adaptive controller.

- Model reference adaptive control (MRAC) based on dynamic inversion (Adaptive Control chapter)

  – MRAC in simulation results in moderate improvement in regulation, but stability and performance depend on learning rates. This controller is capable of learning highly variable state dynamics and control effectiveness (problems 1a, 1b, and 2a), but remains vulnerable to airspeed estimation errors and disturbances (problems 2b and 3).

- An "actuator" model (i.e. a model of sensor response to control) is developed and incorporated, improving stability by modeling dynamics (e.g. time delays and rate limits) that cannot be canceled by the adaptive element. This model is also used with the INDI controller.

- Incremental nonlinear dynamic inversion (INDI) control (Incremental Control chapter)

  - INDI rotational control is applied and tested in simulation and flight on Flexrotor. Sometimes poor tilt regulation with the existing Flexrotor controller is largely corrected even though the INDI controller does not use a rotor model, trim model, or airspeed estimate. The INDI controller (an incremental controller) quickly corrects for steady state error (e.g. due to problems 1a and 1b) without the negative effects of integral action. The INDI controller (a sensor-based controller) handles uncertainty (problems 2a and 2b) and disturbances (problem 3) using acceleration measurements rather than model knowledge (e.g. in NDI) or learning (e.g. in adaptive control).
  - INDI translational control is extended to allow the controller to vary for different body tilt axes (e.g. to handle low-drag and high-drag axes on a helicopter) and tested in flight.
  - Simulations, flight test, and analysis demonstrate graceful degradation of INDI controllers with heavy sensor filtering, retaining useful performance and robustness even with high vibration on inertial sensors.
  - The INDI controller promises to make Flexrotor landing onto a small deck reliable in winds and gusts extending across the flight envelope, whereas the existing Flexrotor adaptive controller performs marginally in moderate but gusty wind.

Chapter 2

MODELING

Useful aircraft and environment models are crucial for achieving the research objectives, e.g. the development of control techniques for landing and takeoff that are robust to disturbances such as gusty wind and base motion. First, models are used for testing control algorithms in simulation. Second, models are used online by control algorithms. The phenomena to be modeled include substantial variations in cyclic authority, and nonlinearities in trim, over the 3D velocity envelope in thrust-borne flight. For Flexrotor, to date these have been very roughly described by a non-physical ad hoc treatment. Improved fidelity requires a model which captures the underlying physical mechanism, which turns out to be variation in induced velocity over the rotor disc. At the same time the model must be practical for fast online computation: i.e. much simpler than a free-wake vortex-filament treatment. A model developed by Pitt and Peters [64] turns out to do the job nicely.

In this present modeling work, the most significant insight was to use a simple rotor induced velocity model to address long-standing aircraft modeling deficiencies. Original contributions include the efficient implementation of the Pitt-Peters model as linearized perturbations, a modification of lift deficiency functions (revealing non-monotonic variations in control authority with airspeed), and comparisons to flight data for rotors of this type.

The induced velocity modeling and lift deficiency functions described in this chapter are generally useful for any aircraft with a helicopter-like rotor, as well as diverse configurations including tiltrotors and quadrotors [59, 13]. Aircraft having low disc loading are particularly relevant, since the lower the disc loading, the smaller the airspeed scale over which control peculiarities occur. While the algorithm and equations provided here can be used with varied rotorcraft—especially where computational efficiency is important—this chapter also provides an illustration of the importance of modeling fidelity at least at the level of the Pitt-Peters model.

2.1 *Flexrotor*

The rotorcraft example used in much of this research is the Aerovel Flexrotor (Figure 2.1). Flexrotor is a miniature (1.85 or 2.20 m diameter proprotor) tailsitter capable of both helicopter-like thrust-borne flight and airplane-like wing-borne flight [1]. The main rotor has collective and cyclic controls like a conventional helicopter. Electrically-driven wingtip thrusters play the role of a helicopter tail rotor and the rear fuselage opens into legs for VTOL. Note that compared to full-size helicopters Flexrotor disc loading is low: 86 N/m^2 at maximum launch weight (with 1.85 m diameter rotor) [1] and 41–63 N/m^2 for the flight data presented in this chapter [55, 52, 53]. For comparison, the Robinson R22 light helicopter has a disc loading of 85 N/m^2 at *empty* weight (72 N/m^2 for the Robinson R66) [66]. Low disc loading is the basis of long endurance: Flexrotor has demonstrated flight endurance of 32 hours [2], exceeding the advertised endurance of similarly sized *fixed-wing* unmanned aerial vehicles (UAVs) [36, 75]. But a corollary of low disc loading is that some peculiarities of thrust-borne flight are manifest at particularly low speed, and loom large in normal operations.

2.2 *Aircraft Model*

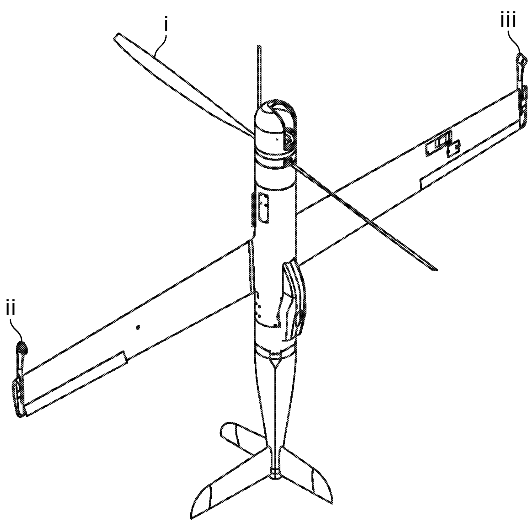
The Aerovel nonlinear, physics-based simulation [50] was designed for sufficient fidelity in both wing-borne and thrust-borne flight. The simulation models have been validated and refined through flight, ground, and wind-tunnel testing.

2.2.1 Rotor Model

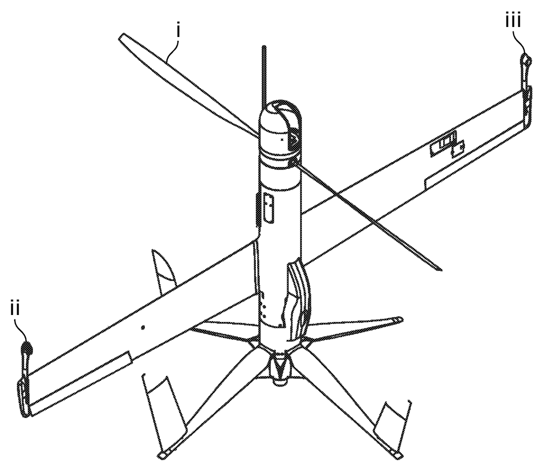
Included in the Aerovel simulation is a blade element aeroelastic rotor model. The aeroelastic rotor solution is calculated for axisymmetric (i.e. symmetric around the rotor spin axis) con-

(a) Thrust-borne flight with legs open for takeoff or landing [1].

(b) Wing-borne flight [1].



(c) Thrust-borne flight attitude with (i) main [prop]rotor and (ii), (iii) wingtip thrusters [21].



(d) With legs open for takeoff or landing [21].

Figure 2.1: Aerovel Flexrotor.

ditions, as a function of *axial advance ratio*[1], rotor hub collective pitch, and rotor rotational speed. Cyclic variation around the rotor spin axis (i.e. hub cyclic pitch, in-plane velocity, or in-plane rotation rate) is calculated by linear perturbation from axisymmetric trim. Measurements have verified that, *other things being equal*, in-plane loading is proportional to cyclic pitch [49].

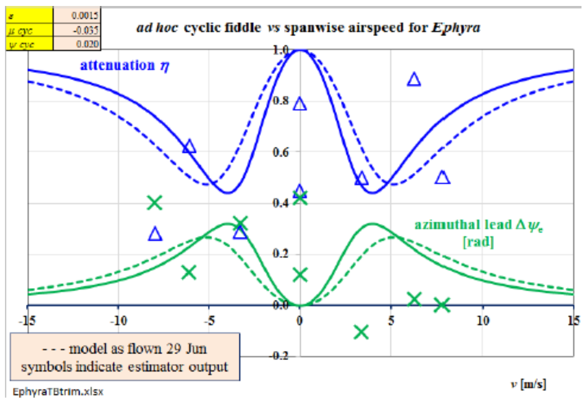
Ad Hoc Model

The Aerovel simulation has an *ad hoc* model that "fiddles" loading due to in-plane perturbations—e.g. cyclic—by scaling the perturbation load, and rotating it in rotor azimuth, as a function of in-plane speed (Figure 2.2). The ad hoc model parameters were selected to fit flight data, more or less [47]. The ad hoc model is unsatisfactory in that it does not entirely reproduce phenomena of interest (e.g. oscillations that occur when regulating position (Figure 2.3) and trim reversal as shown in Figures 2.7–2.10) and relies on heuristic fits. The inability to reproduce phenomena of interest hinders development since control techniques that work well in simulation have turned out to be less than satisfactory in flight.

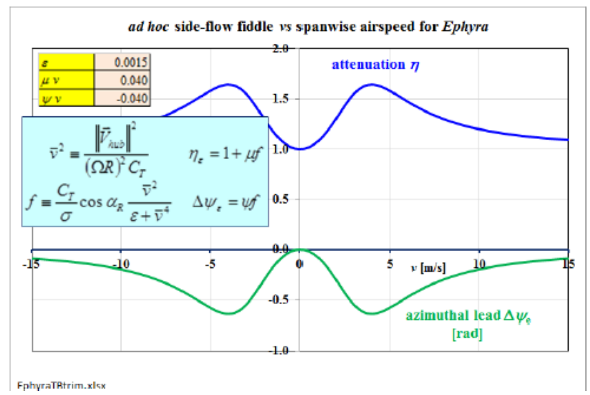
2.3 Rotor Induced Velocity

A rotor in stationary hover has zero in-plane force and torque. Now consider the effect of asymmetry, caused by cyclic pitch, in-plane translation, or in-plane rotation. If the in-plane speed is small compared with tip speed then a linearized treatment might seem reasonable: thus, calculate the variation in angle of attack across the rotor disc by elementary geometry, and integrate over the disc to get blade loads. But this turns out to give a result that is badly wrong even for in-plane flow small compared with tip speed. In order to get the right answer one must account for arising nonuniformity in induced velocity through the rotor.

[1] The *axial advance ratio* is a ratio of axial velocity (i.e. in the rotor spin axis direction, the x axis in Figure 2.4) to rotor tip speed. This is referred to as advance ratio J for airplane propellers and sometimes as climb ratio λ for helicopter rotors. There is typically a difference in scaling, i.e. $J = \pi\lambda$. Helicopter rotor advance ratio μ is referred to here as *in-plane advance ratio* (i.e. for speed in the rotor plane, the y-z plane in Figure 2.4).



(a) Cyclic pitch response attenuation and azimuthal lead. Note that the response is attenuated relative to the response at zero speed.



(b) In-plane speed response attenuation and azimuthal lead. Note that the response is amplified relative to the response at zero speed.

Figure 2.2: Aerovel simulation ad hoc model for Flexrotor *Ephyra* as a function of airspeed spanwise to the wing v ($\approx v_{\text{hub}}$ in Figure 2.4) [47]. The ad hoc model scales and rotates the rotor responses relative to the aeroelastic model (Section 2.2.1).

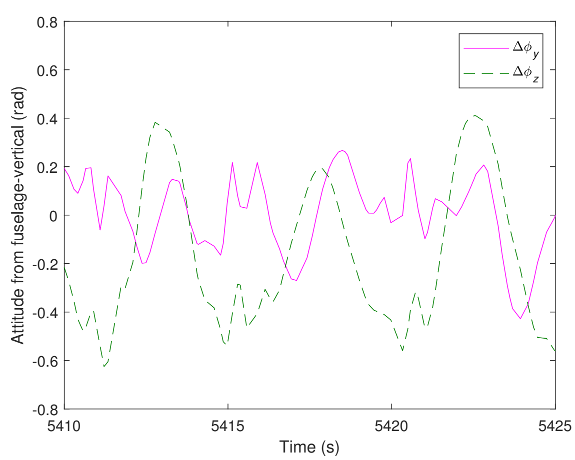


Figure 2.3: Attitude limit cycle for an Aerovel Flexrotor (at 17.3 kg and legs open) when regulating position in 4–7 m/s wind [55].

A *dynamic inflow* model is appropriate. It calculates the induced velocity distribution from a finite number of ordinary differential equations in time [62]. The Pitt-Peters model [64] showed good correlation with helicopter-rotor forward flight (i.e. in-plane airspeed) data [10, 26], the first dynamic inflow model to do so [62]. Later developments extended the Pitt-Peters model and added wake curvature, off-rotor flow field, and swirl velocity calculations [62].

The Pitt-Peters physics is derived from potential flow theory (based on the circular wing theory of Kinner) and is consistent with quasi-steady numerical values for a vortex lattice wake [62]. In hover, the time delays are identical to those calculated for the apparent mass and inertia of an impermeable disc being accelerated in still air. Furthermore, the quasi-steady hover case reduces to that calculated from momentum theory with separate momentum tubes for each side of the rotor disc.

For high-fidelity calculation of rotor aerodynamics, a vortex free-wake model is the method of choice when computational load is not a constraint. But for whole-aircraft dynamical simulation, and for online control calculations, the computational load would be impractically heavy, and the fidelity unnecessarily high. We need a model which is computationally practical and good enough to capture the phenomena underlying Figure 2.3.

2.4 Pitt-Peters Induced Velocity

The Pitt-Peters model described here is primarily the nonlinear[2] version from Peters and HaQuang [63], rather than the common perturbation version without hub motion.

2.4.1 Geometry and Mass Flow Definitions

As defined by Peters and HaQuang [63], μ_1, μ_2, and μ_3 are "forward", "sideward", and "vertical" flight velocities, respectively. All velocities are normalized on ΩR, where Ω is rotor spin speed and R is rotor blade radius. For Flexrotor, $\mu_1 = -v_{\mathrm{hub}}/\Omega R$ (e.g. airspeed spanwise to

[2] Nonlinear in thrust coefficient C_T and uniform variation in induced velocity ν_o but linear in moments. A fully nonlinear version of the Pitt-Peters model—which can be used when the net flow *direction* is not uniform across the rotor disc—is available in Hong [33].

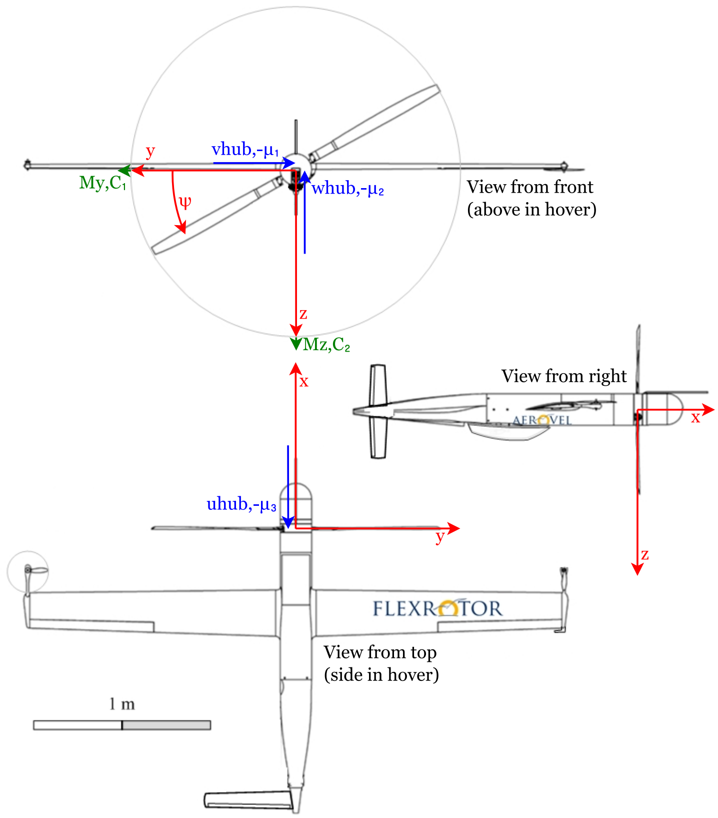


Figure 2.4: Flexrotor [50] and Pitt-Peters [63] coordinate systems. For example, a moment about the Flexrotor y axis is M_y, or coefficient C_1 after normalization for Pitt-Peters. An airspeed at the rotor hub from the y direction is *positive* v_{hub} but *negative* μ_1 after normalization for Pitt-Peters. Rotor azimuth $\psi = 0$ when aligned with the y-axis. The three-view drawing is from [1].

the wing), $\mu_2 = -w_{\text{hub}}/\Omega R$ (e.g. airspeed broadside to the wing), and $\mu_3 = -u_{\text{hub}}/\Omega R$ (e.g. airspeed parallel to the rotor spin axis) are defined here (Figures 2.4 and 2.5).

Continuing with Peters and HaQuang [63], forward velocity in the *wind-axis* system (i.e. in-plane axes oriented with airspeed) is

$$\mu = \sqrt{\mu_1^2 + \mu_2^2}$$

and total inflow through the rotor is

$$\lambda = -\mu_3 + \lambda_m \tag{2.1}$$

where λ_m is the momentum-theory induced velocity due to rotor thrust, defined below in Equations (2.12) and (2.14). Total resultant flow through the rotor is defined as

$$V_T = \sqrt{\lambda^2 + \mu^2} \tag{2.2}$$

and the mass-flow parameter due to cyclic disturbances is

$$V = \frac{\mu^2 + (2\lambda_m - \mu_3)(\lambda_m - \mu_3)}{V_T} \tag{2.3}$$

The wake angle with respect to the rotor disc is

$$\alpha = \tan^{-1}\left(\frac{|\lambda_m - \mu_3|}{\mu}\right) \tag{2.4}$$

The wind-axis azimuth angle $\bar{\psi}$ with respect to the rotor azimuth angle ψ (Figure 2.5) is

$$\psi = \Delta + \bar{\psi}$$

where

$$\sin\Delta = \frac{\mu_2}{\mu}$$
$$\cos\Delta = \frac{\mu_1}{\mu}$$

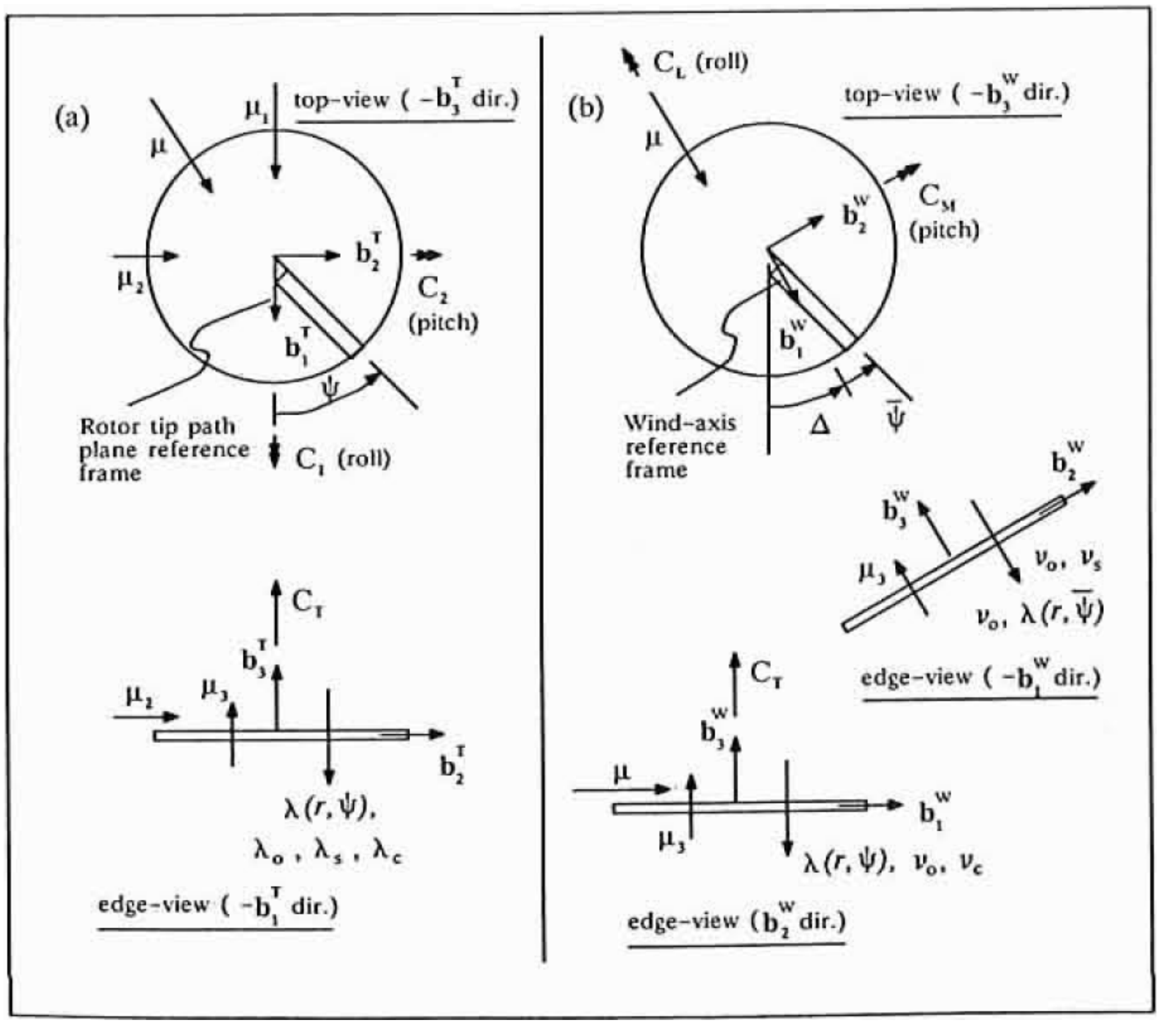


Figure 2.5: Pitt-Peters wind-axis and rotor tip path plane reference frames (Figure 2 in [63]).

2.4.2 Nonlinear Version of Pitt-Peters

Wind-axis coordinates

As defined by Peters and HaQuang [63], induced velocity in wind-axis coordinates is

$$\lambda\left(r,\bar{\psi}\right) = \nu_o + \nu_s \frac{r}{R}\sin\bar{\psi} + \nu_c \frac{r}{R}\cos\bar{\psi} \tag{2.5}$$

where r is the radial coordinate and ν_o, ν_s, and ν_c are the uniform, lateral, and longitudinal variations in induced velocity, respectively. The governing differential equation set allows calculation from a load vector $\boldsymbol{C}$, geometry, and inflow variables:

$$[M]\begin{bmatrix}\dot{\nu}_o\\ \dot{\nu}_s\\ \dot{\nu}_c\end{bmatrix} + [L]_{\mathrm{nl}}^{-1}\begin{bmatrix}\nu_o\\ \nu_s\\ \nu_c\end{bmatrix} = \begin{bmatrix}C_T\\ -C_L\\ -C_M\end{bmatrix}_{\mathrm{aero}} \tag{2.6}$$

where $[M]$ is the matrix of apparent mass terms; $[L]_{\mathrm{nl}}$ is the nonlinear version of the inflow gains matrix; and C_T, C_L, and C_M are the instantaneous rotor thrust, and roll and pitching moment coefficients, respectively, in the wind-axis system. The aero subscript indicates that only aerodynamic contributions are considered (i.e. inertial terms are omitted). Note that time is normalized on rotor speed, i.e. $\tau = \Omega t = \psi$. The mass matrix is

$$[M] = \begin{bmatrix}\frac{8}{3\pi}* & 0 & 0\\ 0 & \frac{16}{45\pi} & 0\\ 0 & 0 & \frac{16}{45\pi}\end{bmatrix} \tag{2.7}$$

The * indicates that $M[1,1] = 128/75\pi$ for twisted rotors. However, Chen [10] states that studies found that $M[1,1] = 8/3\pi$ correlates better with twisted-blade flight data. Note that Flexrotor, being required to perform from hover to high axial advance ratio, has much more blade twist than a helicopter, e.g. 0.5 rad of built-in twist for the standard 1.85 m diameter rotor [3]. The gains matrix is

$$[L]_{\mathrm{nl}} = [L]\,[V]^{-1}$$

where, with α from Equation (2.4)

$$[L] = \begin{bmatrix} \frac{1}{2} & 0 & -\frac{15\pi}{64}\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}} \\ 0 & \frac{4}{1+\sin\alpha} & 0 \\ \frac{15\pi}{64}\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}} & 0 & \frac{4\sin\alpha}{1+\sin\alpha} \end{bmatrix} \tag{2.8}$$

and

$$[V] = \begin{bmatrix} V_T & 0 & 0 \\ 0 & V & 0 \\ 0 & 0 & V \end{bmatrix} \tag{2.9}$$

The mass-flow parameter matrix $[V]$ denotes a weighted downstream velocity.

Rotor coordinates

As defined by Peters and HaQuang [63], the transformation from rotor coordinates to wind axes is

$$[T] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Delta & \sin\Delta \\ 0 & -\sin\Delta & \cos\Delta \end{bmatrix}$$

so

$$\begin{bmatrix} C_T \\ -C_L \\ -C_M \end{bmatrix} = [T] \begin{bmatrix} C_T \\ C_1 \\ -C_2 \end{bmatrix} = [T] \begin{bmatrix} \frac{T}{\rho\pi\Omega^2 R^4} \\ \frac{M_y}{\rho\pi\Omega^2 R^5} \\ -\frac{M_z}{\rho\pi\Omega^2 R^5} \end{bmatrix}_{\text{Flexrotor}}$$

where we have recast the load vector into the thrust and moment coefficients of interest for Flexrotor (thrust is in the x-axis direction in Figure 2.4). ρ is air density. Following Equation (2.5), induced inflow in rotor coordinates is

$$\lambda(r,\psi) = \lambda_o + \lambda_s \frac{r}{R}\sin\psi + \lambda_c \frac{r}{R}\cos\psi \tag{2.10}$$

where, continuing with Peters and HaQuang [63],

$$\begin{bmatrix} \nu_o \\ \nu_s \\ \nu_c \end{bmatrix} = [T] \begin{bmatrix} \lambda_o \\ \lambda_s \\ \lambda_c \end{bmatrix}$$

The differential equation in rotor coordinates (like Equation (2.6) in wind-axis coordinates) is then

$$[M]\begin{bmatrix}\dot{\lambda}_o\\ \dot{\lambda}_s\\ \dot{\lambda}_c\end{bmatrix} + [\hat{L}]^{-1}\begin{bmatrix}\lambda_o\\ \lambda_s\\ \lambda_c\end{bmatrix} = \begin{bmatrix}C_T\\ C_1\\ -C_2\end{bmatrix}_{\text{aero}} \tag{2.11}$$

$$[\hat{L}]^{-1} = [V]\,[T]^T\,[L]^{-1}\,[T] = [V]\,[\mathscr{L}]^{-1}$$

$$[\mathscr{L}] = \begin{bmatrix}1/2 & -B\sin\Delta & -B\cos\Delta\\ B\sin\Delta & E\cos^2\Delta + D\sin^2\Delta & (D-E)\sin\Delta\cos\Delta\\ B\cos\Delta & (D-E)\cos\Delta\sin\Delta & E\sin^2\Delta + D\cos^2\Delta\end{bmatrix}$$

$$E = \frac{4}{1+\sin\alpha}, \qquad (E-D) = 4\frac{1-\sin\alpha}{1+\sin\alpha}$$

$$D = \frac{4\sin\alpha}{1+\sin\alpha}, \qquad B = \frac{15\pi}{64}\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}}$$

$$\lambda_m = \frac{1}{2}\begin{bmatrix}1\\ 0\\ 0\end{bmatrix}^T [\mathscr{L}]^{-1}\begin{bmatrix}\lambda_o\\ \lambda_s\\ \lambda_c\end{bmatrix} \tag{2.12}$$

Static (Quasi-Steady)

The apparent-mass term in Equation (2.11) is significant only at short timescale: according to Gaonkar and Peters [26], only when cyclic loading varies faster than about 1/5 of rotor frequency. For Flexrotor while thrust-borne this is 3 Hz or so. Our interest is in slower rigid-body motion, so we can safely drop the apparent-mass terms in favor of a quasi-steady approximation

$$\begin{bmatrix}\lambda_o\\ \lambda_s\\ \lambda_c\end{bmatrix} = [\hat{L}]\begin{bmatrix}C_T\\ C_1\\ -C_2\end{bmatrix}_{\text{aero}} = [\mathscr{L}]\,[V]^{-1}\begin{bmatrix}C_T\\ C_1\\ -C_2\end{bmatrix}_{\text{aero}} \tag{2.13}$$

Setting moments $C_1 = C_2 = 0$ results in the Peters and HaQuang [63] static equations which lead to a first-order nonuniform induced-velocity distribution. Thus (using an overline for steady values):

$$\overline{\lambda}_m = \frac{C_T}{2\overline{V}_T} = \overline{\lambda}_o \tag{2.14}$$

$$\overline{\lambda}_s = 2B\left(\sin\overline{\Delta}\right)\overline{\lambda}_m \tag{2.15}$$

$$\overline{\lambda}_c = 2B\left(\cos\overline{\Delta}\right)\overline{\lambda}_m \tag{2.16}$$

For $\Delta = 0$, i.e. wind axes (without overlines),

$$\lambda_o = \frac{C_T}{2V_T}$$

$$\lambda_s = 0$$

$$\lambda_c = \frac{C_T}{2V_T}\frac{15\pi}{32}\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}}$$

Using Equation (2.10),

$$\lambda\left(r,\psi\right) = \frac{C_T}{2V_T}\left(1 + \frac{15\pi}{32}\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}}\frac{r}{R}\cos\psi\right) = \lambda_o\left(1 + k_x\frac{r}{R}\cos\psi\right) \tag{2.17}$$

This nonuniform induced velocity model $\lambda_o\left(1 + k_x\left(r/R\right)\cos\psi\right)$, first suggested by Glauert in 1926, is known as (static) linear "longitudinal" or first harmonic inflow [42].

2.4.3 Implementation

We have now to apply the inflow model for use by an onboard computer or simulator. For Flexrotor, rotor behavior is precalculated and stored in lookup tables as functions of axial advance ratio, collective, and rotor speed. Stored parameters include coefficients for linearized calculation of in-plane loads (for hub cyclic pitch, in-plane velocity, and in-plane rotation rate) [3, 50]:

$$\left[\begin{array}{cccccccccccc} \frac{\partial F_p}{\partial \mathrm{cyc}} & \frac{\partial F_n}{\partial \mathrm{cyc}} & \frac{\partial M_p}{\partial \mathrm{cyc}} & \frac{\partial M_n}{\partial \mathrm{cyc}} & \frac{\partial F_p}{\partial \mathrm{vel}} & \frac{\partial F_n}{\partial \mathrm{vel}} & \frac{\partial M_p}{\partial \mathrm{vel}} & \frac{\partial M_n}{\partial \mathrm{vel}} & \frac{\partial F_p}{\partial \mathrm{rot}} & \frac{\partial F_n}{\partial \mathrm{rot}} & \frac{\partial M_p}{\partial \mathrm{rot}} & \frac{\partial M_n}{\partial \mathrm{rot}} \end{array}\right] \tag{2.18}$$

where a p subscript indicates in-phase with the perturbation and an n subscript indicates quadrature, e.g. $\partial F_n/\partial \mathrm{vel}$ is the quadrature force due to in-plane velocity. The coefficients are normalized on air density, rotor spin speed, and rotor blade radius.

The remainder of this subsection details the new implementation of the quasi-steady Pitt-Peters model using linearized perturbations.

Two additional sets of coefficients were added for aerodynamic moments and aerodynamic rotations:

$$\boldsymbol{C}_{\mathrm{aero}} = \begin{bmatrix} \frac{\partial M_p}{\partial \mathrm{cyc}} & \frac{\partial M_n}{\partial \mathrm{cyc}} & \frac{\partial M_p}{\partial \mathrm{vel}} & \frac{\partial M_n}{\partial \mathrm{vel}} & \frac{\partial M_p}{\partial \mathrm{rot}} & \frac{\partial M_n}{\partial \mathrm{rot}} \end{bmatrix}_{\mathrm{aero}}$$

$$\boldsymbol{C}_{\mathrm{aeroRot}} = \begin{bmatrix} \frac{\partial F_p}{\partial \mathrm{aeroRot}} & \frac{\partial F_n}{\partial \mathrm{aeroRot}} & \frac{\partial M_p}{\partial \mathrm{aeroRot}} & \frac{\partial M_n}{\partial \mathrm{aeroRot}} \end{bmatrix}$$

As before, the aero subscript indicates that only aerodynamic contributions are included (i.e. inertial load terms are omitted). aeroRot indicates an *aerodynamic rotation*, i.e. a rotation (about an in-plane axis) of the air through the rotor plane rather than a rotation of the rotor plane in inertial space. These coefficients are also normalized.

The linearized aerodynamic moment coefficients can be calculated using the terms from $\boldsymbol{C}_{\mathrm{aero}}$ and current perturbations as

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} \pm\frac{\partial M_p}{\partial \mathrm{cyc}} & -\frac{\partial M_n}{\partial \mathrm{cyc}} & \pm\frac{\partial M_p}{\partial \mathrm{vel}} & -\frac{\partial M_n}{\partial \mathrm{vel}} & \frac{\partial M_p}{\partial \mathrm{rot}} & \mp\frac{\partial M_n}{\partial \mathrm{rot}} \\ \frac{\partial M_n}{\partial \mathrm{cyc}} & \pm\frac{\partial M_p}{\partial \mathrm{cyc}} & \frac{\partial M_n}{\partial \mathrm{vel}} & \pm\frac{\partial M_p}{\partial \mathrm{vel}} & \pm\frac{\partial M_n}{\partial \mathrm{rot}} & \frac{\partial M_p}{\partial \mathrm{rot}} \end{bmatrix}_{\mathrm{aero}} \begin{bmatrix} \mathrm{cyc}_y \\ \mathrm{cyc}_z \\ \mathrm{vel}_y \\ \mathrm{vel}_z \\ \mathrm{rot}_y \\ \mathrm{rot}_z \end{bmatrix} \tag{2.19}$$

The signs depend on rotor spin direction [50] and all terms have been normalized. (Flexrotor's spin is left-handed, the engine's conventional right-handed rotation being reversed by a reduction gearbox.) The linearized in-plane force and moment coefficients due to the nonuniform induced velocity components can be calculated using the terms from $\boldsymbol{C}_{\text{aeroRot}}$ as

$$\begin{bmatrix} F_y \\ F_z \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} \pm\frac{\partial F_p}{\partial \text{aeroRot}} & -\frac{\partial F_n}{\partial \text{aeroRot}} \\ \frac{\partial F_n}{\partial \text{aeroRot}} & \pm\frac{\partial F_p}{\partial \text{aeroRot}} \\ \frac{\partial M_p}{\partial \text{aeroRot}} & \mp\frac{\partial M_n}{\partial \text{aeroRot}} \\ \pm\frac{\partial M_n}{\partial \text{aeroRot}} & \frac{\partial M_p}{\partial \text{aeroRot}} \end{bmatrix} \begin{bmatrix} \lambda_s \\ -\lambda_c \end{bmatrix} \tag{2.20}$$

where again all terms are normalized. The use of *aerodynamic-rotation* perturbations follows from the fact that the nonuniform terms in models of this form (e.g. Equation (2.10)) are equivalent to rotating inflow, a relationship noted by Dreier [18].

Pitt-Peters effects are introduced with the algorithm

1. Calculate linearized aerodynamic moment coefficients C_1 and C_2 from Equation (2.19).

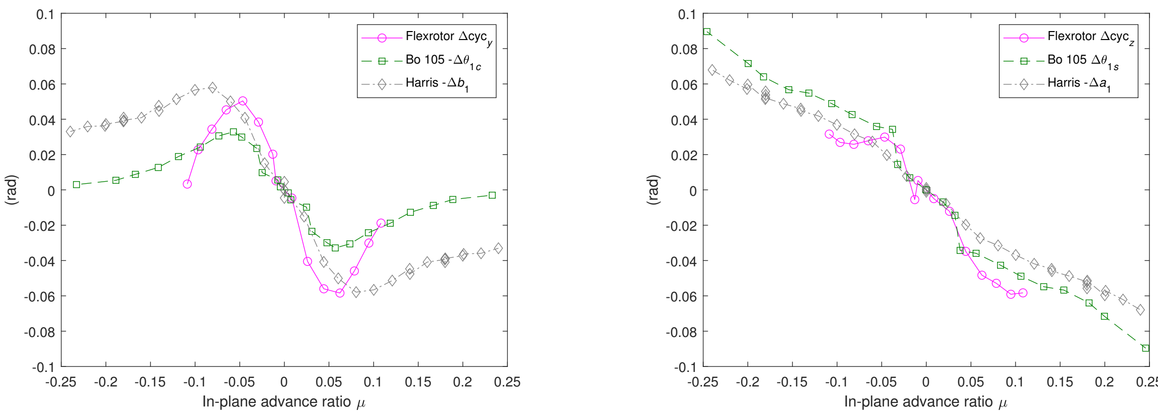
2. Calculate azimuthally nonuniform induced velocity components λ_s and λ_c using C_1, C_2, and other states in the quasi-steady Pitt-Peters Equation (2.13).

3. Calculate linearized in-plane force and moment coefficients using λ_s and λ_c in Equation (2.20).

4. Superpose these forces and moments due to the Pitt-Peters induced velocity components on those calculated directly from the hub cyclic pitch, in-plane velocity, and in-plane rotation rate coefficients (Equation (2.18)).

The second step is where nonlinear effects are introduced by otherwise linear calculations (e.g. linear loading with respect to cyclic and rotation rate). In Equation (2.13) the nonlinear dependence on in-plane speed primarily comes from the mass-flow parameter V (Equation (2.3)). This entire algorithm can be used with any system where aerodynamic moment and aerodynamic rotation coefficients can be determined, e.g. from a blade element model.

This algorithm has been implemented and tested in the Aerovel simulation. This elementary implementation turns out to be effective as described in the following subsections (2.4.4 and 2.4.5). We will first discuss trim and then turn to dynamic behavior.



(a) Lateral. (b) Longitudinal.

Figure 2.6: Cyclic pitch trim (Flexrotor and Bo 105) and flapping (Harris) angles. Data are changes from hover values. Flexrotor data are as in Figure 2.7, e.g. velocity spanwise to the wing. Bo 105 and Harris data for forward flight have also been mirrored here to negative in-plane advance ratio so that they can be compared to the mostly symmetric Flexrotor data.

2.4.4 Variation in Cyclic Pitch Trim

Figure 2.6 shows that cyclic trim is a strongly nonlinear function of in-plane advance ratio μ. Flight data for two aircraft with hingeless rotors are shown: Flexrotor [52] and the DLR research Bo 105 helicopter [60]. For "lateral" cyclic, the magnitudes of Flexrotor cyc_y and Bo 105 $-\theta_{1c}$ peak near $\mu = 0.05$ to 0.06. These curves are qualitatively similar, although the narrower curve and factor of two difference in speed scaling (e.g. $\mu = 0.1$ is 11 m/s for Flexrotor and 22 m/s for the Bo 105) indicate that trim for Flexrotor is significantly more sensitive to speed changes than for the Bo 105. In general, the cosine component of induced velocity peaks at a higher flight speed for a higher thrust coefficient [10]. For these data, Flexrotor's disc loading is about one fifth that of the Bo 105, and the airspeed scale for control nonlinearities is consequently compressed. This illustrates the particularly high sensitivity of lightly-loaded rotors to the effects discussed. For "longitudinal" cyclic, Flexrotor cyc_z and Bo 105 θ_{1s} are qualitatively similar, although flatter for Flexrotor at higher advance ratios.

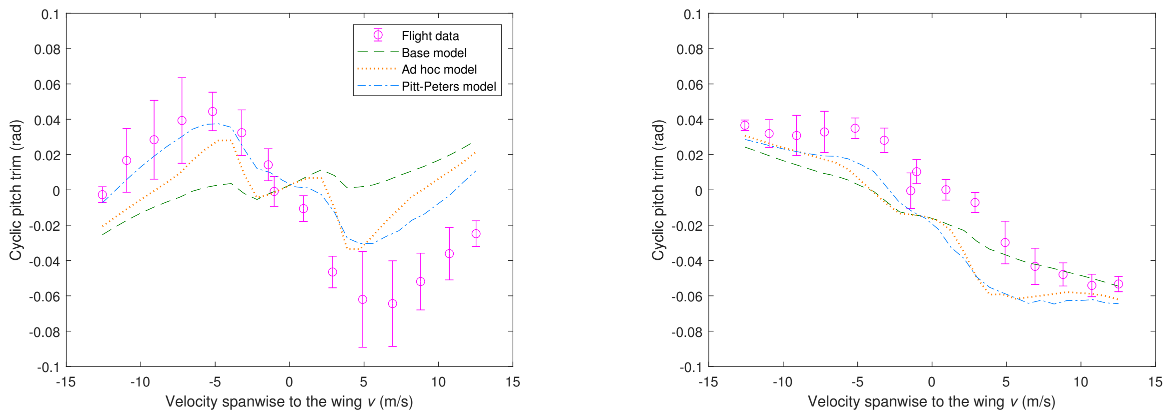
Also shown in Figure 2.6 are articulated blade flapping angles (for constant thrust, angle-of-attack, and cyclic) from a tunnel test by Harris [32]. The lateral flapping peaks at $\mu = 0.08$. These data have been described as "...for the definitive assessment of rotor wake models at low speed...." [39]

As shown in Figures 2.7–2.10, Flexrotor trim correlation with flight data is substantially improved both qualitatively and quantitatively with the Pitt-Peters model relative to the models previously used. The Pitt-Peters model is implemented as described in Section 2.4.3. Four configurations were tested: *short* and *long* rotor blades, each with legs closed and open. The improvement from the ad hoc model to the Pitt-Peters model is impressive since the ad hoc model was developed and adjusted using earlier Flexrotor test data while the Pitt-Peters model is derived from first principles. Remaining discrepancies between flight and model could be in the rotor model or other aircraft component models (e.g. wing, tail, fuselage, or rotor wake-airframe interaction).

Figure 2.7 shows cyclic trim for a Flexrotor, with 1.85 m diameter rotor and legs closed, as a function of velocity spanwise to the wing v ($\approx v_{\text{hub}}$ in Figure 2.4). The three rotor models shown are the *base* aeroelastic model (Section 2.2.1), the base model with the *ad hoc* model (Ad Hoc Model subsection), and the base model with the *Pitt-Peters* model. Trim is calculated for the entire aircraft model, so trim is not proportional to speed even with the base rotor model. The Pitt-Peters model is implemented as described in Section 2.4.3.

Figure 2.7 shows that for y-axis cyclic, correlation with the flight data is significantly improved both qualitatively and quantitatively with the Pitt-Peters model compared to the base and ad hoc models. For z-axis cyclic, the Pitt-Peters model is similar to the ad hoc model. Results are similar for trim with legs open (Figure 2.8).

As indicated in Figure 2.8, the aerodynamic-rotation coefficients used in this implementation of the Pitt-Peters model are close to optimal for these short-blade flight data, with only small improvements in fidelity available by adjusting them. In contrast, there is a large



(a) Lateral cyc_y.   (b) Longitudinal cyc_z.

Figure 2.7: Trim for a Flexrotor with 1.85 m diameter rotor and legs closed (at 15.8 kg and 560 m altitude). The one standard deviation error bars are each calculated over 10-40 seconds of flight data [52].

improvement in fidelity for long blades when the aerodynamic-rotation coefficients are scaled by 2.0 and rotated by -0.6 rad (Figures 2.9 and 2.10). This indicates a substantial modeling error. Aerodynamically-identical but elastically-different long blades performed differently in flight, indicating that discrepancies may be in the elastic model [56]. Figure 2.11 shows trim with the Pitt-Peters model at higher altitude and for airspeed broadside rather than spanwise to the wing.

2.4.5 Variation in Cyclic Pitch Authority

For the Pitt-Peters model, the following can be derived from Equation (2.6):

$$\frac{\partial \nu_s}{\partial C_L} = \frac{-4}{(1+\sin\alpha)\,V} \tag{2.21}$$

$$\frac{\partial \nu_c}{\partial C_M} = \frac{-4\sin\alpha}{(1+\sin\alpha)\,V} = \sin\alpha \frac{\partial \nu_s}{\partial C_L} \tag{2.22}$$

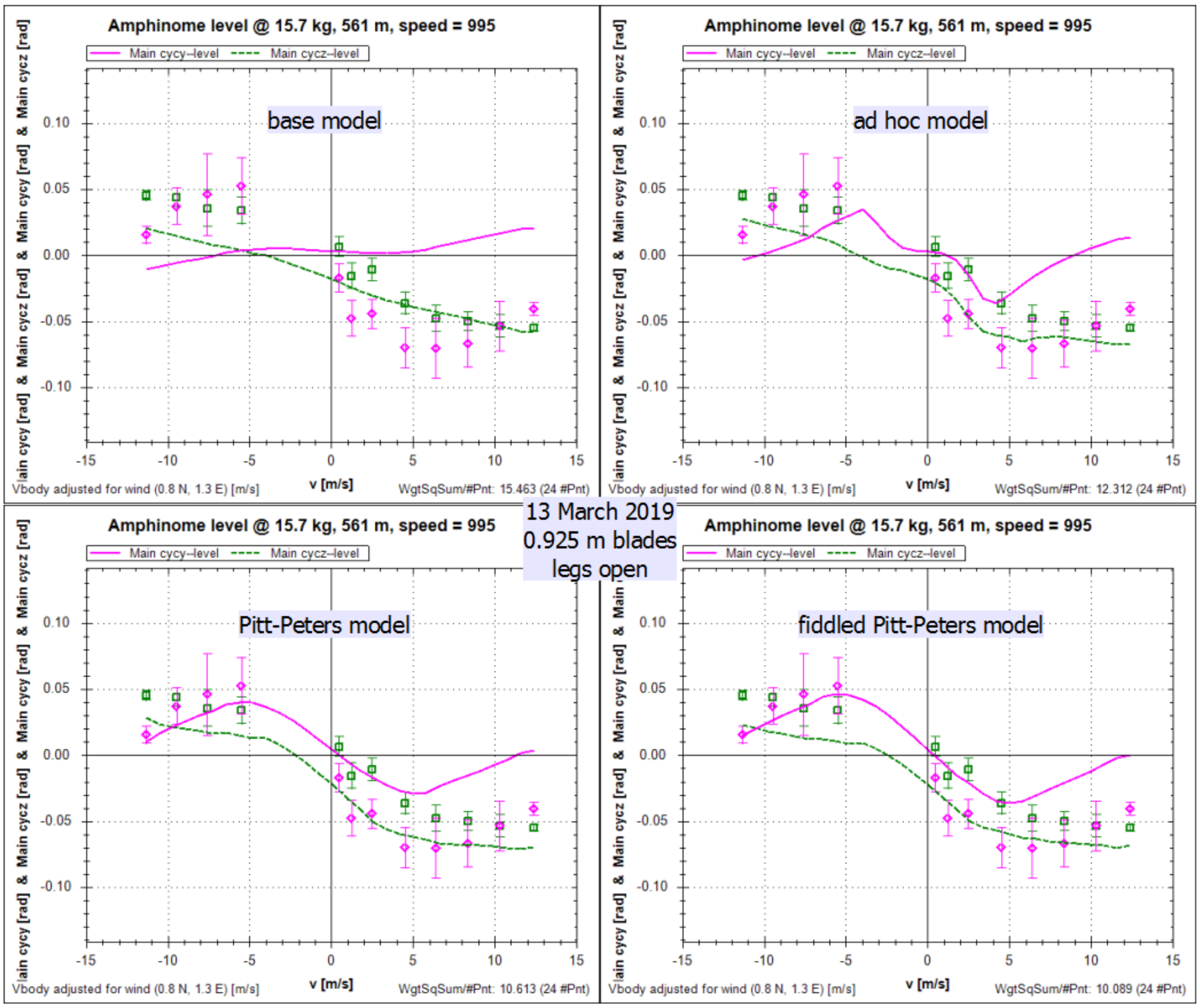


Figure 2.8: Trim as in Figure 2.7 (Flexrotor *Amphinome* with short blades (0.925 m radius)) except here legs are open.

As in Figure 2.7, y-axis cyclic correlation with the flight data is significantly improved both qualitatively and quantitatively for the Pitt-Peters model compared to the base and ad hoc models. For z-axis cyclic, the Pitt-Peters model is again similar to the ad hoc model. There is a small improvement in correlation to flight data when the Pitt-Peters model is fiddled (aerodynamic-rotation coefficients scaled by 1.1 and rotated by -0.1 rad).

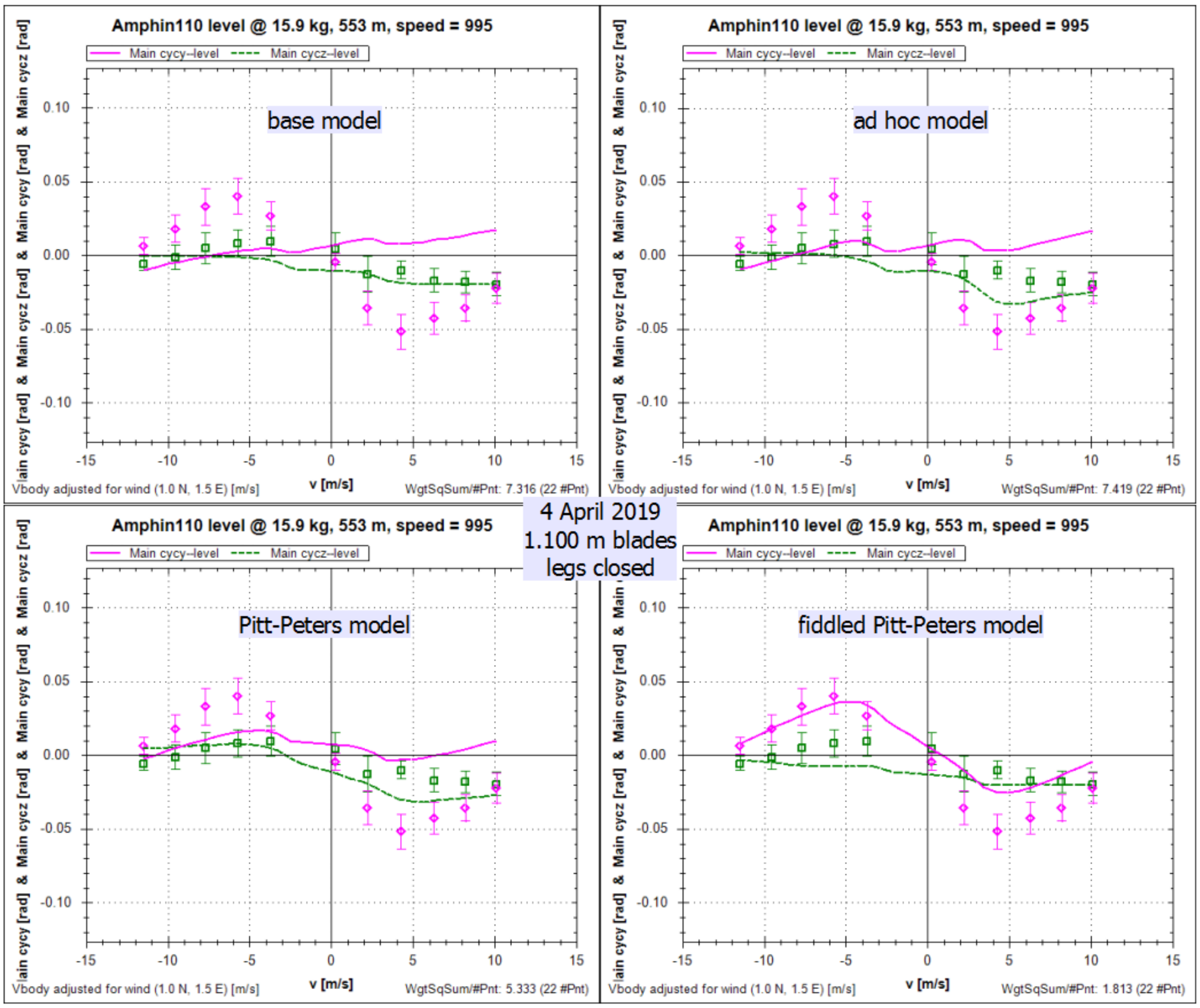


Figure 2.9: Trim as in Figure 2.7 (Flexrotor *Amphinome* with legs closed) except here rotor blades are long (1.100 m radius).

For y-axis cyclic, correlation with flight data [53] is improved both qualitatively and quantitatively for the Pitt-Peters model compared to the base and ad hoc models. For z-axis cyclic, the Pitt-Peters model is similar to the ad hoc model. There is a significant improvement in correlation to flight data when the Pitt-Peters model is fiddled by a large amount (i.e. aerodynamic-rotation coefficients scaled by 2.0 and rotated by -0.6 rad).

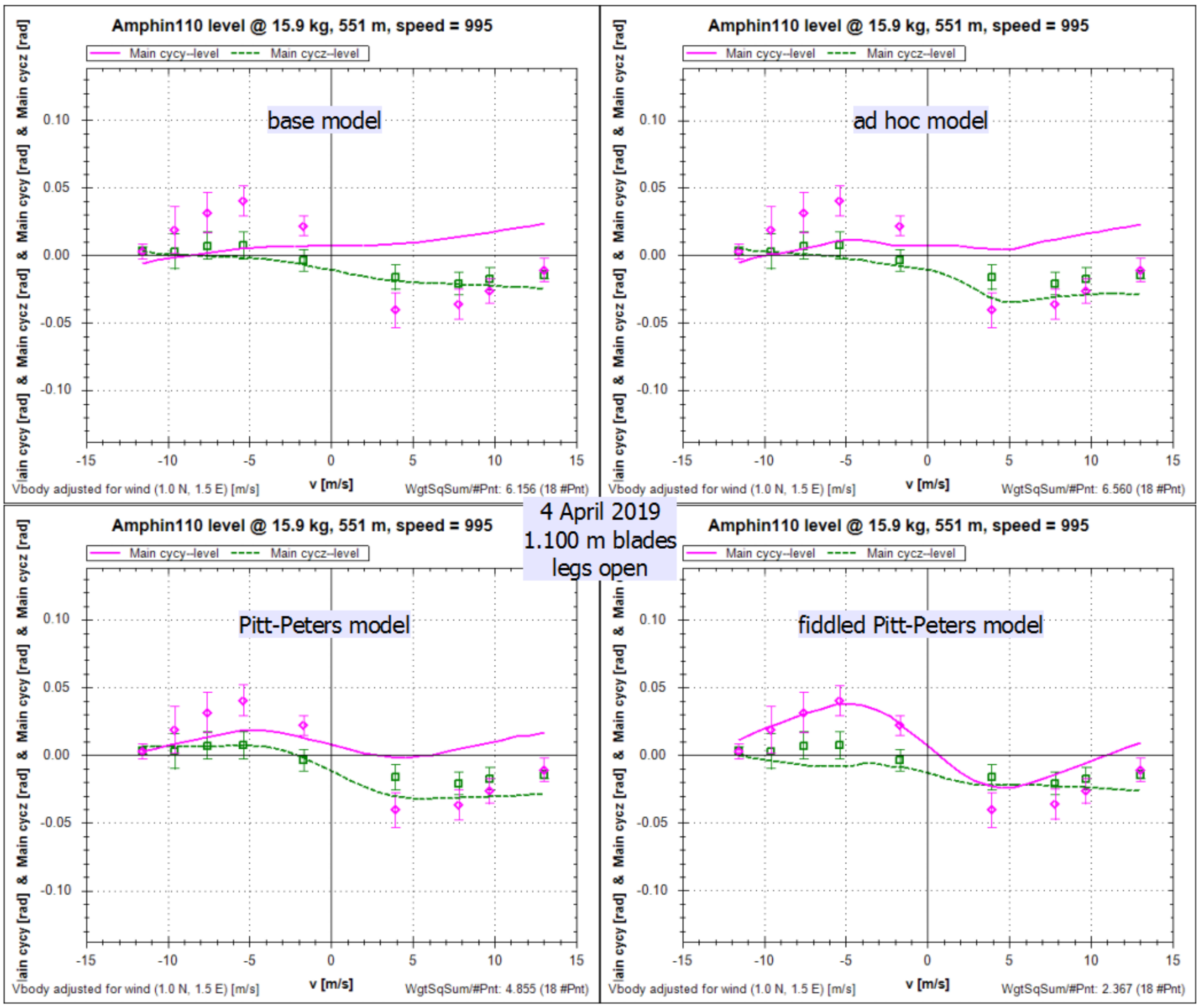


Figure 2.10: Trim as in Figure 2.9 (Flexrotor *Amphinome* with long blades (1.100 m radius)) except here legs are open.

As in Figure 2.9, y-axis cyclic correlation with the flight data is improved both qualitatively and quantitatively for the Pitt-Peters model compared to the base and ad hoc models. For z-axis cyclic, the Pitt-Peters model is a small improvement compared to the ad hoc model. There is a significant improvement in correlation to flight data when the Pitt-Peters model is fiddled (aerodynamic-rotation coefficients scaled by 2.0 and rotated by -0.6 rad).

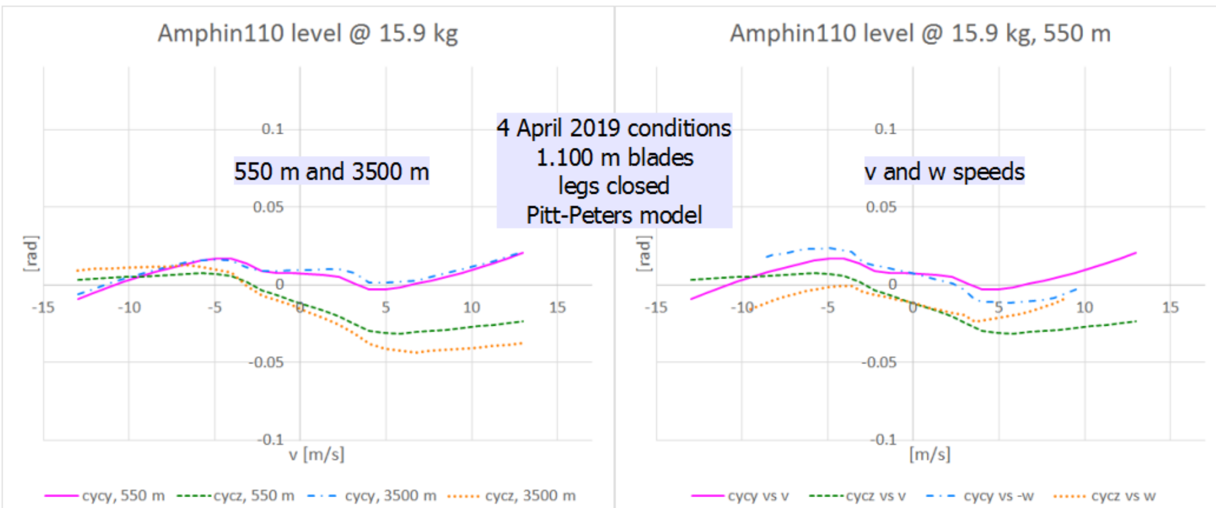


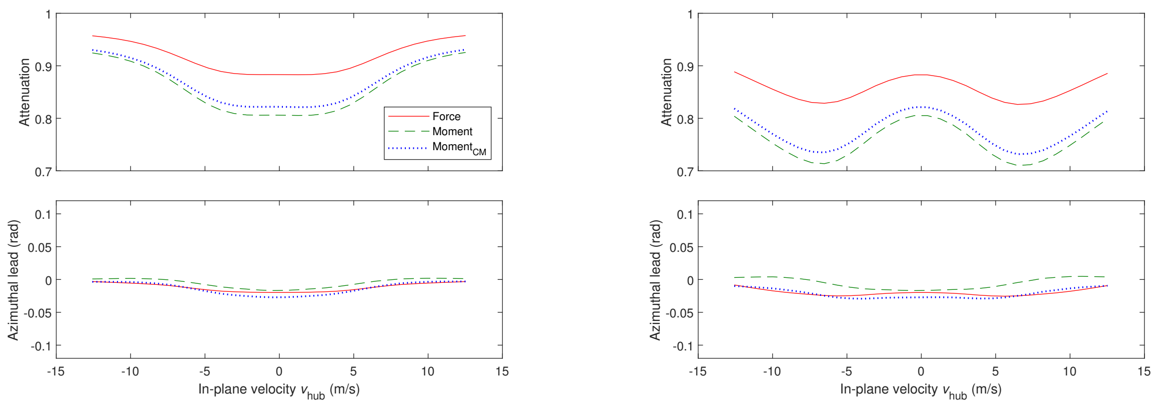
Figure 2.11: Trim as in Figure 2.9 (Flexrotor *Amphinome* with long blades (1.1 m radius) and legs closed). Both figures here have the Pitt-Peters model in solid magenta and dashed green as in Figure 2.9.

The figure at left also has trim with the Pitt-Peters model at higher altitude (3500 m). The figure at right also has trim (with the Pitt-Peters model) as a function of velocity broadside to the wing w ($\approx w_{\mathrm{hub}}$ in Figure 2.4).

which are the partial derivatives of the nonuniform induced velocity components with respect to in-plane aerodynamic moments. Consider an increment in hub cyclic pitch. This increment produces an aerodynamic moment which in turn changes induced velocity components (Equations (2.21) and (2.22)). These changes in induced velocity reduce control authority relative to that indicated by a blade-element calculation. Equations (2.21) and (2.22) show that cyclic authority varies with airspeed (α and V) and axis of control application (sine and cosine terms). Note that the ad hoc model (Figure 2.2a) for adjusting authority does not even attempt to account for this directional dependence.

Figures 2.12–2.15 show the Pitt-Peters model (Section 2.4.3) effects on control authority at various trim conditions. Scaling and rotation for some directions of control azimuths (i.e. rotor azimuth for maximum cyclic) (Figures 2.13–2.15) are qualitatively similar to the ad hoc fiddling (Figure 2.2a), but with smaller amplitudes. Effects at other azimuths are qualitatively different (including amplification relative to hover at some azimuths). Cyclic authority variation with azimuth could partially explain the poor correlation between the ad hoc model and parameter estimator outputs shown in Figure 2.2a. In addition, aircraft motion in response to cyclic causes additional Pitt-Peters effects (e.g. aerodynamic moments from rotation rate) which change apparent cyclic authority. In fact, nonuniform induced velocity due to rotation rate increases rotation damping [62], decreasing apparent cyclic authority.

Figure 2.12 shows cyclic authority varying with in-plane airspeed, for a Flexrotor with 1.85 m diameter rotor and legs closed, as modeled by Pitt-Peters (Section 2.4.3). Cyclic authority scale and rotation (azimuthal lead) are shown at two directions of control application. The azimuths are approximately those for the two mode shapes: 0.4 rad for the cosine mode (Equation (2.22)) and 2.0 rad for the sine mode (Equation (2.21)). The resulting in-plane forces and moments are shown, as well as moments at the aircraft center of mass (CM). For the cosine mode (Figure 2.12a) there is amplification relative to hover beyond $v_{\mathrm{hub}} \approx 3\,\mathrm{m/s}$ and for the sine mode (Figure 2.12b), there is attenuation relative to hover up to $v_{\mathrm{hub}} \approx 12\,\mathrm{m/s}$.



(a) 0.4 rad azimuth.

(b) 2.0 rad azimuth.

Figure 2.12: Cyclic authority varying with in-plane airspeed as modeled by Pitt-Peters. Moment$_{\mathrm{CM}}$ indicates moments at the aircraft center of mass (CM) for combined rotor forces and moments. The conditions are those for Figure 2.7.

The Pitt-Peters model causes loading in response to cyclic to vary with trim condition, in both magnitude and direction around the rotor azimuth. Thus, cyclic peaking at 2 rad rotor azimuth Figure 2.12b) generates only 0.9 of the control moment at 7 m/s lateral speed as it does in hover, on an axis rotated by less than a degree relative to that in hover.

Figure 2.13 shows cyclic authority at four directions of control application. Scale and rotation are shown for moments at the aircraft CM. The four azimuths shown are approximately those for the two mode shapes as in Figure 2.12 as well as those for minimum and maximum azimuthal lead.

Figures 2.14 and 2.15 show increased cyclic authority effects with aerodynamic-rotation coefficient fiddles. An alternative is to instead fiddle the aerodynamic moment coefficients $\partial M_p/\partial \mathrm{cyc}$ and $\partial M_n/\partial \mathrm{cyc}$, which has a smaller effect on trim. Figure 2.16 shows the authority effects at higher altitude and for airspeed broadside rather than spanwise to the wing.

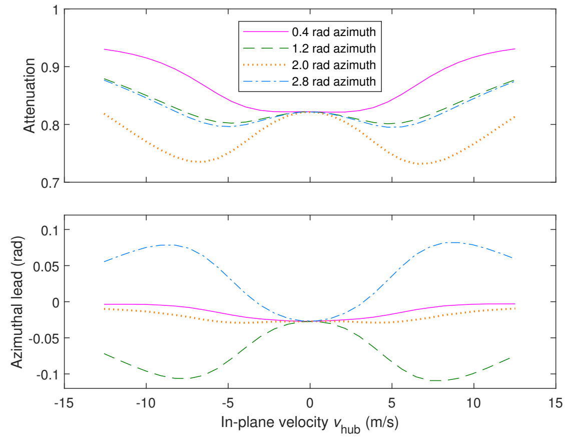


Figure 2.13: Cyclic authority (moments at aircraft CM) at four directions of control application. The conditions are those for Figure 2.7.

2.5 *Modeling of Variable Control Authority by Lift Deficiency Functions*

Lift deficiency functions are an alternative for modeling variable control authority. W. Johnson [40] defined the following dynamic inflow lift deficiency functions for rotor thrust and in-plane moments, e.g. thrust coefficient $C_T = C'_{[\text{thrust}]}\,(C_T)_{\text{no wake-induced velocities}}$, using the Pitt-Peters model [64].

$$C'_{[\text{thrust}]} = \frac{1}{1 + \frac{\sigma a}{8V_{\text{eff}}}} \tag{2.23}$$

where σ is rotor solidity, a is blade section two-dimensional lift-curve slope, and

$$V_{\text{eff[thrust]}} = \frac{\mu^2 + \lambda\left(\lambda + \lambda_i\right)}{\sqrt{\mu^2 + \lambda^2}} \tag{2.24}$$

where λ_i is equivalent to λ_m as defined earlier. Note that Equation 2.24 is equivalent to V as defined earlier (Equation 2.3). For moments, W. Johnson [40] defined

$$C'_{[\text{pitch moment}]} = \frac{1}{1 + \frac{\sigma a}{8V_{\text{eff}}}\frac{2\cos\chi}{1+\cos\chi}} \tag{2.25}$$

$$C'_{[\text{roll moment}]} = \frac{1}{1 + \frac{\sigma a}{8V_{\text{eff}}}\frac{2}{1+\cos\chi}} \tag{2.26}$$

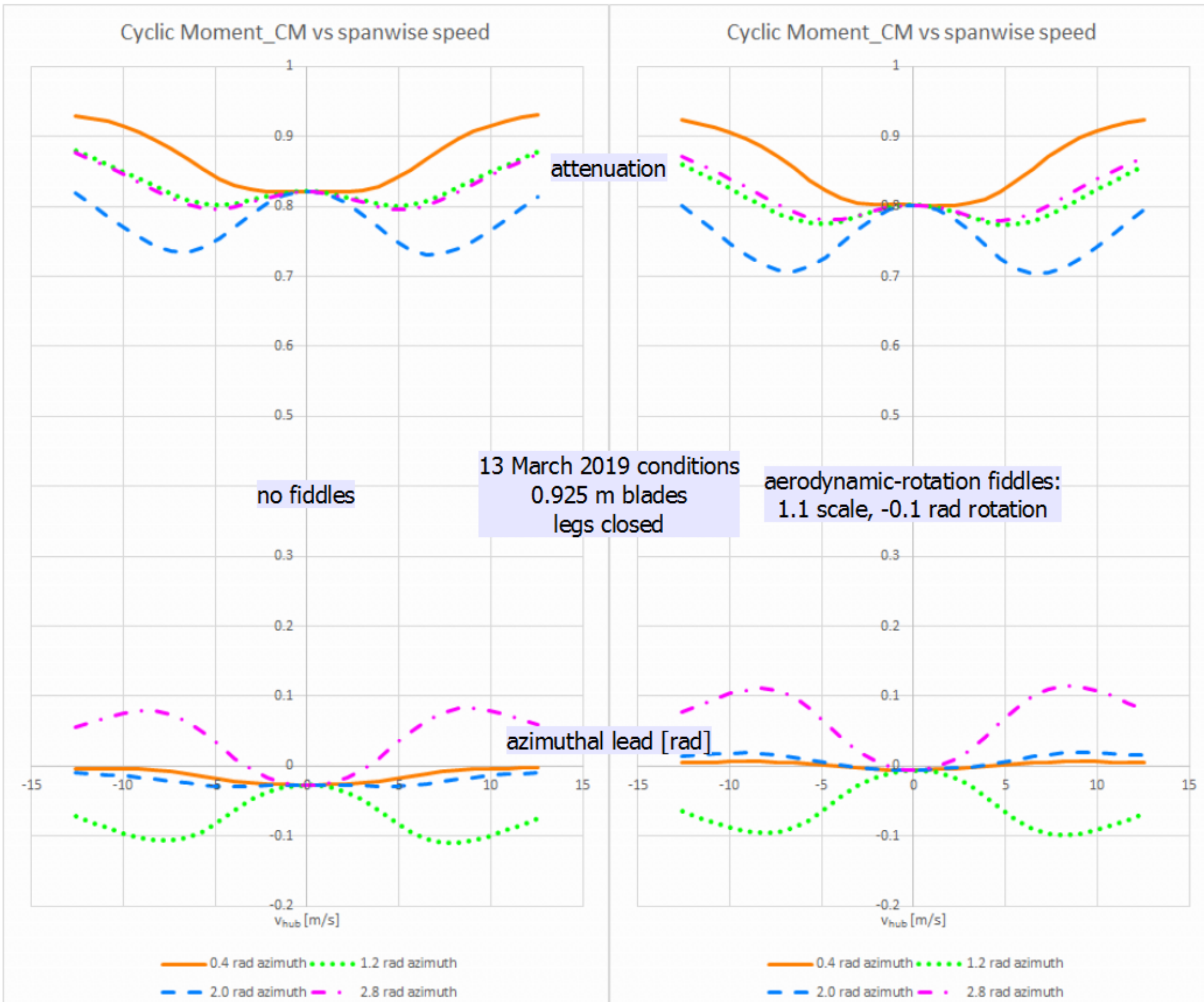


Figure 2.14: Cyclic authority changes as in Figure 2.13 (e.g. short blades (0.925 m radius)), except here both figures are for legs closed.

The figure at left is the same as in Figure 2.13, and the figure at right has aerodynamic-rotation coefficients scaled by 1.1 and rotated by -0.1 rad as for trim in Figure 2.8. The aerodynamic-rotation coefficient scaling increases the attenuation effect.

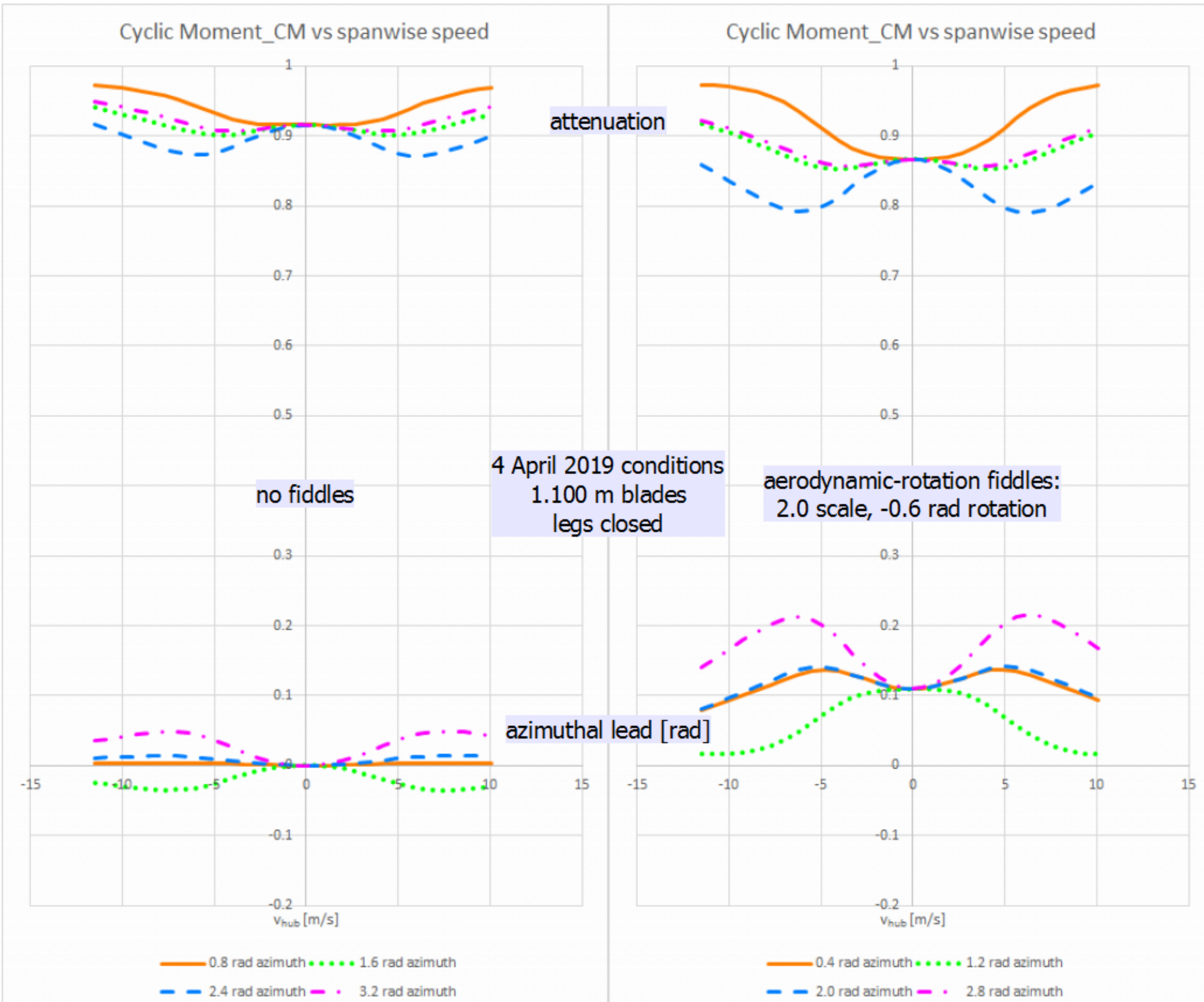


Figure 2.15: Cyclic authority changes as in Figure 2.14 (e.g. legs closed), except here blades are long (1.100 m radius) and conditions are as on 4 April 2019 (Figure 2.9).

The figure at left shows smaller effects than for short blades (Figure 2.14). The figure at right has aerodynamic-rotation coefficients scaled by 2.0 and rotated by -0.6 rad as for trim in Figure 2.9. The aerodynamic-rotation scaling increases the attenuation and azimuthal lead effects.

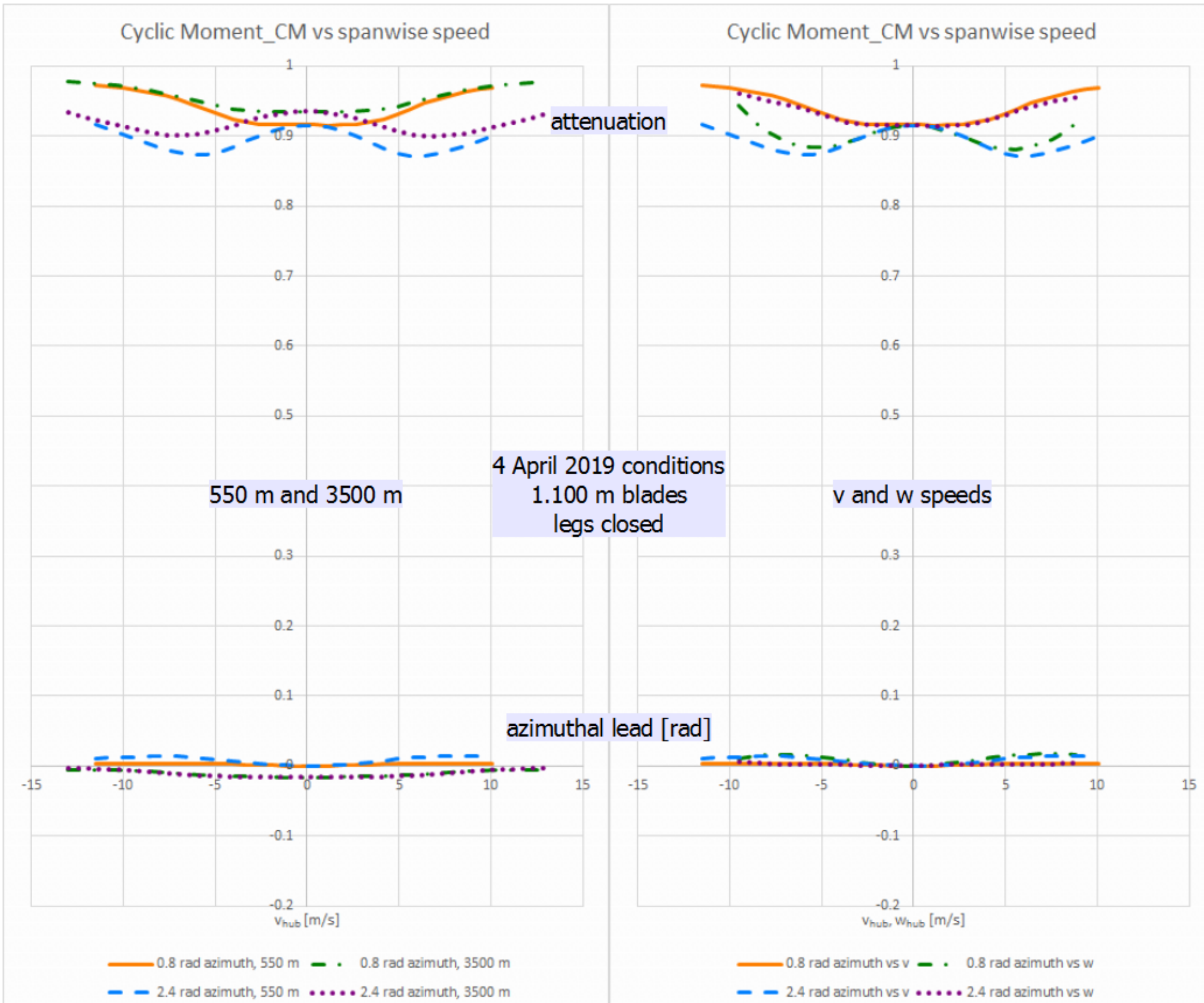


Figure 2.16: Cyclic authority changes as in Figure 2.15 (e.g. long blades (1.100 m radius) and legs closed). Both figures here have the Pitt-Peters model (without aerodynamic-rotation fiddles) mode shapes in solid orange and dashed azure as in Figure 2.15.
The figure at left also has the authority changes at higher altitude (3500 m), where the attenuation effect is reduced. The figure at right also has the authority changes as a function of velocity broadside to the wing w ($\approx w_{\text{hub}}$ in Figure 2.4), where the mode shape azimuths have rotated by a quarter turn since the wind axes have rotated by that amount.

where the "roll" axis is aligned with in-plane airspeed, χ is the wake skew angle (0 for axial flow and $\pi/2$ rad for edgewise flow) and

$$V_{\text{eff[moments]}} = \sqrt{\mu^2 + \lambda^2} \tag{2.27}$$

Equation 2.27 is equivalent to V_T as defined earlier (Equation 2.2). W. Johnson [40] noted that "...[t]ypical values of these lift deficiency functions are $C' \cong 0.8$ for forward flight; $C' \cong .7$ for thrust changes in hover; and $C' \cong 0.5$ for moment changes in hover."

Using $\cos\chi = \sin\alpha$, Equations 2.25 and 2.26 are

$$C'_{\text{[pitch moment]}} = \frac{1}{1 + \frac{\sigma a}{8V_{\text{eff}}}\frac{2\sin\alpha}{1+\sin\alpha}} \tag{2.28}$$

$$C'_{\text{[roll moment]}} = \frac{1}{1 + \frac{\sigma a}{8V_{\text{eff}}}\frac{2}{1+\sin\alpha}} \tag{2.29}$$

W. Johnson [40] defined $V_{\text{eff}} = V_T$ for these functions but $V_{\text{eff}} = V$ for consistency with Peters et al. [63, 25] (see Equation 2.9). The use of V rather than V_T for V_{eff} can also be seen in Equations 2.21 and 2.22, as well as the relationship to Equations 2.28 and 2.29.

Figure 2.17 shows cyclic moment authority as modeled by Pitt-Peters, as in Figure 2.12, and by the lift deficiency functions. The results are similar. $V_{\text{eff}} = V$ has been used in lift deficiency Equations 2.28 and 2.29 for consistency with Peters et al. If $V_{\text{eff}} = V_T$ is instead used (as specified by W. Johnson [40]), cyclic moment authority increases monotonically from hover. An examination of such competing mass-flow parameters is in Gaonkar and Peters [25].

2.6 Conclusions

Figure 2.18 shows a simulation of aircraft position regulation with the Pitt-Peters model: a limit cycle as in flight (Figure 2.3). While amplitudes are larger in simulation, particularly about the y axis, the cycles are similar (e.g. 4.5 s period about the z axis).

Limit cycling in flight (e.g. Figure 2.3) was not understood for some time as it was not reproducible with a rotor model in which in-plane loads (due to cyclic, velocity, and

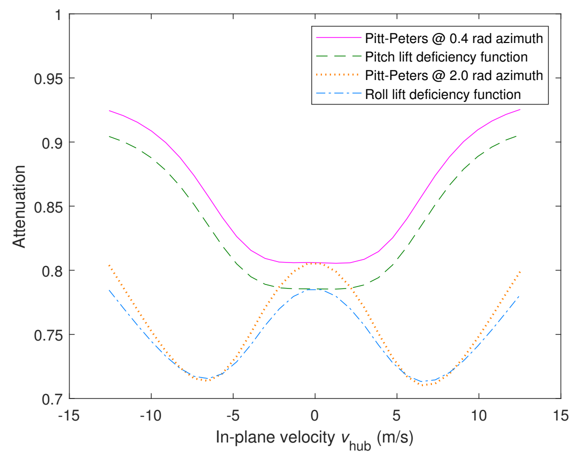


Figure 2.17: Cyclic moment authority at two azimuths as modeled by Pitt-Peters and lift deficiency functions. The conditions are those for Figure 2.7.

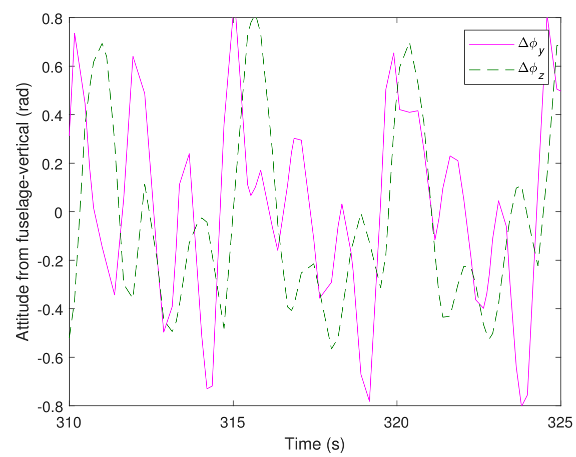


Figure 2.18: Attitude limit cycle for a Flexrotor simulation with the Pitt-Peters model (aircraft 17.3 kg and legs open) in 3 m/s wind. The cycle is similar to flight (Figure 2.3). See Figure 2.4 for y and z axis definitions.

rotation) were linearized relative to axial flow. It was eventually realized that loads must be quite nonlinear functions of in-plane perturbations. Furthermore, it appeared that in-plane flow was not only introducing unexpected loads, but also causing cyclic authority to vary: the moment generated by a given cyclic input was evidently different when translating laterally at just a few m/s than when in stationary hover. Hence the autopilot had to adapt, but in order to devise appropriate adaptation it was necessary first to understand and to reproduce the behavior observed in flight. In this chapter, the ultimately successful result of the modeling effort is presented.

An efficient implementation of the Pitt-Peters nonuniform inflow model was shown to be effective, augmenting a real-time simulation and significantly improving fidelity for a low disc loading tailsitter. The non-monotonic changes in control trim and authority over a small speed range show the danger in simply performing system identification or linearization at hover and other widely spaced speeds. Flight controllers must be robust to these changes. In practice this can mean adaptive or robust control since airspeed is difficult to measure or estimate at low speed using commonly available sensors.

Chapter 3

TRIM FEEDFORWARD

Section 2.4.4 (e.g. Figure 2.6) showed that trim is a strongly nonlinear function of airspeed. Particularly for the low disc loading Flexrotor, the nonlinearity occurs over a small airspeed scale. This affects translational control and gust response. This chapter examines feedforward of trim control to improve command response and gust attenuation, including consideration of modeling and airspeed estimation errors. Furthermore, adaptive feedforward using online trim estimation is developed.

If knowledge of model and state is good enough, then feedforward can improve on feedback. This applies to the addition of both estimated steady-state control and control for producing desired dynamics. For rotorcraft, there is notoriously high uncertainty in measuring low airspeeds. Therefore, feedforward is investigated here both with and without accurate airspeed estimates.

3.1 Flexrotor Control Architecture

The *standard* Flexrotor control architecture is shown in Figure 3.1. The focus in the figure and in this research is on the inner controllers, i.e. Translation and Rotation. Guidance and Outer Controllers includes guidance (e.g. flight plan or loiter), flight management (e.g. takeoff, landing, lost link, or transition between thrust-borne and wing-borne flight), and commands from human operators.

In Figure 3.1, Guidance and Outer Controllers provide reference commands to the Translation and Rotation Controllers. For example, position $\boldsymbol{\Delta r}_{\text{ref}}$ or velocity $\boldsymbol{v}_{\text{ref}}$ from a guidance controller, attitude $\boldsymbol{\Delta\phi}_{\text{ref}}$ from keyboard or joystick command (e.g. during lost GPS), or rotation rate $\boldsymbol{\omega}_{\text{ref}}$ from the thrust-borne to wing-borne flight transition controller.

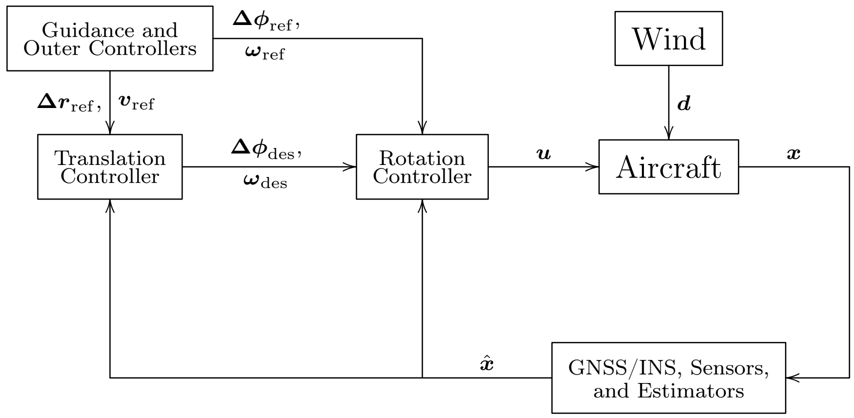


Figure 3.1: Conceptual diagram of Flexrotor control architecture, focused on inner controllers.

The Translation Controller outputs translational controls (not shown) and desired rotation commands $\boldsymbol{\Delta\phi}_{\mathrm{des}}, \boldsymbol{\omega}_{\mathrm{des}}$ for the Rotation Controller. Rotational controls $\boldsymbol{u}$ from the Rotation Controller are output to Aircraft actuators. The Aircraft response is disturbed by Wind $\boldsymbol{d}$ and the aircraft state $\boldsymbol{x}$ is estimated by GNSS/INS, Sensors, and Estimators. The Translation and Rotation Controllers use aircraft state estimate $\hat{\boldsymbol{x}}$.

The standard rotation controller for thrust-borne flight is a multiple-input, multiple-output PID controller that outputs a desired acceleration vector. Control mixing is then used to map from accelerations to actuators, i.e. primarily heading to wingtip thrusters and $\Delta\phi_y, \Delta\phi_z$ to rotor cyclic. The controller gains are set using pole placement for specified natural frequencies and damping ratios, with the addition of specified integral time constants. There is additional logic, e.g. tilt/velocity/cyclic compensation for “heading” rate (“heading” here is used intuitively rather than formally). The controller runs at 50 Hz.

3.1.1 Cyclic Authority Compensation

The standard Flexrotor control system has logic that scales and rotates (in rotor azimuth) cyclic commands to compensate for estimated variations in rotor response. The cyclic AUTHORITY COMP modes are (1) OFF, (2) FLOW MODEL, and (3) the default ADAPTIVE [3]. The OFF mode does not adjust the cyclic commands at this step. The FLOW MODEL mode compensates for cyclic authority (i.e. effectiveness) varying according to the ad hoc model (Section 2.2.1). The ad hoc model effects are averaged over $\pm 3\,\mathrm{m/s}$ relative to the current airspeed estimate, to protect against airspeed estimate errors.

The ADAPTIVE mode uses FLOW MODEL azimuth rotations but estimates current scale from observed aircraft dynamics. Cyclic scale is decreased in response to a “fast” “overcontrol” oscillation and increased in response to a “slow” “undercontrol” oscillation [46]. These characteristic oscillations are further described in Section 4.1.

3.2 Equilibrium and Trim

Consider the system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(t, \boldsymbol{x}, \boldsymbol{u}\right)$$

with time t, state $\boldsymbol{x}$ (derivative $\dot{\boldsymbol{x}}$), and input $\boldsymbol{u}$. Following Khalil [41], to stabilize the system with respect to an arbitrary point $\boldsymbol{x}_{\mathrm{ss}}$ the steady-state $\boldsymbol{u}_{\mathrm{ss}}$ is used to maintain equilibrium

$$\boldsymbol{0} = \boldsymbol{f}\left(t, \boldsymbol{x}_{\mathrm{ss}}, \boldsymbol{u}_{\mathrm{ss}}\right), \quad \forall t \geq 0$$

The change of variables

$$\boldsymbol{x}_\delta = \boldsymbol{x} - \boldsymbol{x}_{\mathrm{ss}}, \qquad \boldsymbol{u}_\delta = \boldsymbol{u} - \boldsymbol{u}_{\mathrm{ss}}$$

results in

$$\dot{\boldsymbol{x}}_\delta = \boldsymbol{f}\left(t, \boldsymbol{x}_{\mathrm{ss}} + \boldsymbol{x}_\delta, \boldsymbol{u}_{\mathrm{ss}} + \boldsymbol{u}_\delta\right) \overset{\mathrm{def}}{=} \boldsymbol{f}_\delta\left(t, \boldsymbol{x}_\delta, \boldsymbol{u}_\delta\right)$$

Continuing with Khalil [41], control can now be designed for $\boldsymbol{u}_\delta$ using $\dot{\boldsymbol{x}}_\delta = \boldsymbol{f}_\delta\left(t, \boldsymbol{x}_\delta, \boldsymbol{u}_\delta\right)$. The overall control is

$$\boldsymbol{u} = \boldsymbol{u}_\delta + \boldsymbol{u}_{\mathrm{ss}} \tag{3.1}$$

which includes the feedforward component $\boldsymbol{u}_{\text{ss}}$.

3.2.1 Aircraft Trim

For aircraft, a steady-state $(\boldsymbol{x}_{\text{ss}}, \boldsymbol{u}_{\text{ss}})$ is known as a *trim* condition [74]. For example, E. N. Johnson and S. K. Kannan [38] derived the following model for the attitude dynamics of a helicopter by linearizing around hover and neglecting coupling between attitude and translational dynamics:

$$\dot{\boldsymbol{\omega}}_B = \hat{\boldsymbol{A}}_1 \boldsymbol{\omega}_B + \hat{\boldsymbol{A}}_2 \boldsymbol{v}_B + \hat{\boldsymbol{B}} \left(\boldsymbol{\delta}_{m_{\text{des}}} - \boldsymbol{\delta}_{m_{\text{trim}}} \right) \tag{3.2}$$

with body angular velocity vector $\boldsymbol{\omega}_B$, body velocity vector $\boldsymbol{v}_B$, and moment control vector $\boldsymbol{\delta}_m$. Note the explicit use of the trim control vector $\boldsymbol{\delta}_{m_{\text{trim}}}$. For desired angular acceleration $\dot{\boldsymbol{\omega}}_B = \boldsymbol{\alpha}_{\text{des}}$, E. N. Johnson and S. K. Kannan [38] invert $\hat{\boldsymbol{B}}$ for desired controls

$$\boldsymbol{\delta}_{m_{\text{des}}} = \hat{\boldsymbol{B}}^{-1} \left(\boldsymbol{\alpha}_{\text{des}} - \hat{\boldsymbol{A}}_1 \boldsymbol{\omega}_B - \hat{\boldsymbol{A}}_2 \boldsymbol{v}_B \right) + \boldsymbol{\delta}_{m_{\text{trim}}} \tag{3.3}$$

which is in the same form as Equation (3.1), including feedforward trim control $\boldsymbol{\delta}_{m_{\text{trim}}}$.

3.2.2 Dynamic Trim

Some systems can be separated into fast and slow dynamics. For aircraft this is typically the fast attitude dynamics and slow translational dynamics. The fast states (e.g. rotation) can reach a quasi-steady-state *dynamic trim* while the slow states (e.g. translation) are still changing [34].

3.3 Feedforward Control

Returning to Figure 2.6, rotor cyclic trim is a strongly nonlinear function of airspeed in the plane of the rotor disc. For typical rotorcraft there is a low-drag in-rotor-plane axis (e.g. longitudinal for helicopters and spanwise to wing y for Flexrotor) and a high-drag in-rotor-plane axis (e.g. lateral for helicopters and broadside to the wing z for Flexrotor). Here the focus will be on airspeed in the low-drag axis, e.g. v for Flexrotor (Figure 2.4). The

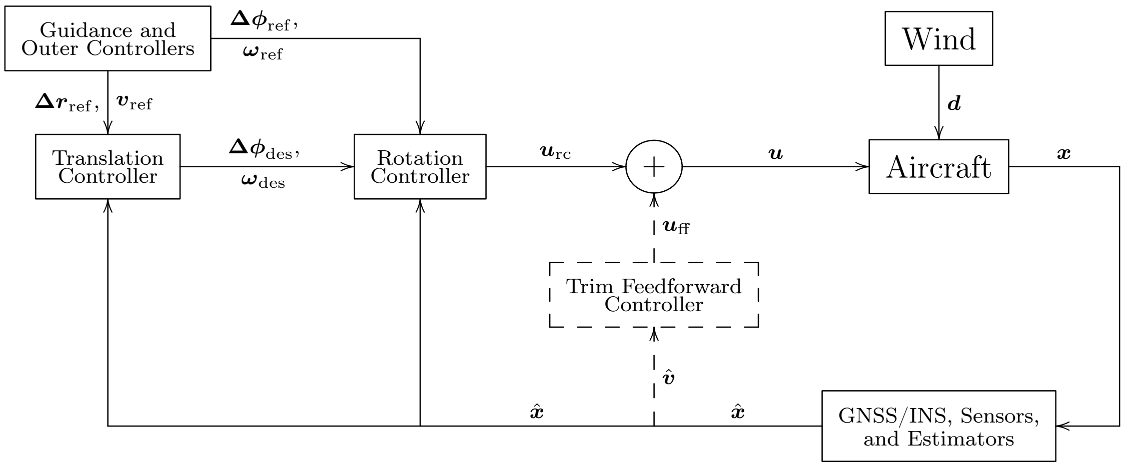


Figure 3.2: Conceptual diagram of Flexrotor control architecture with trim feedforward.

Flexrotor autopilot typically keeps this axis aligned with airspeed, and off-axis airspeed is further limited by high drag. It is therefore a reasonable first approximation to focus on the cyclic trim map as a function of airspeed spanwise to the wing v (with w, altitude rate, and turn rate zero), i.e.

$$\begin{bmatrix} \mathrm{cyc}_y \\ \mathrm{cyc}_z \end{bmatrix}_{\mathrm{trim}} = \boldsymbol{T}\left(v\right) \tag{3.4}$$

Note that the $\boldsymbol{T}$ function can be used in the dynamic trim sense, i.e. when the fast attitude dynamics are in quasi-steady-state but the slow dynamics (including v) are still changing.

As in Equations (3.1) and (3.3), feedforward control can be implemented by adding $\boldsymbol{T}\left(v\right)$ to the existing controller output. Figure 3.2 shows the *standard* Flexrotor control architecture (as in Figure 3.1) with the addition of a Trim Feedforward Controller. Rotational controls $\boldsymbol{u}_{\mathrm{rc}}$ from the Rotation Controller are summed with feedforward $\boldsymbol{u}_{\mathrm{ff}}$ and output to Aircraft actuators. The velocity estimate $\hat{\boldsymbol{v}}$, a subset of state estimate $\hat{\boldsymbol{x}}$, is used for trim calculation by the Trim Feedforward Controller.

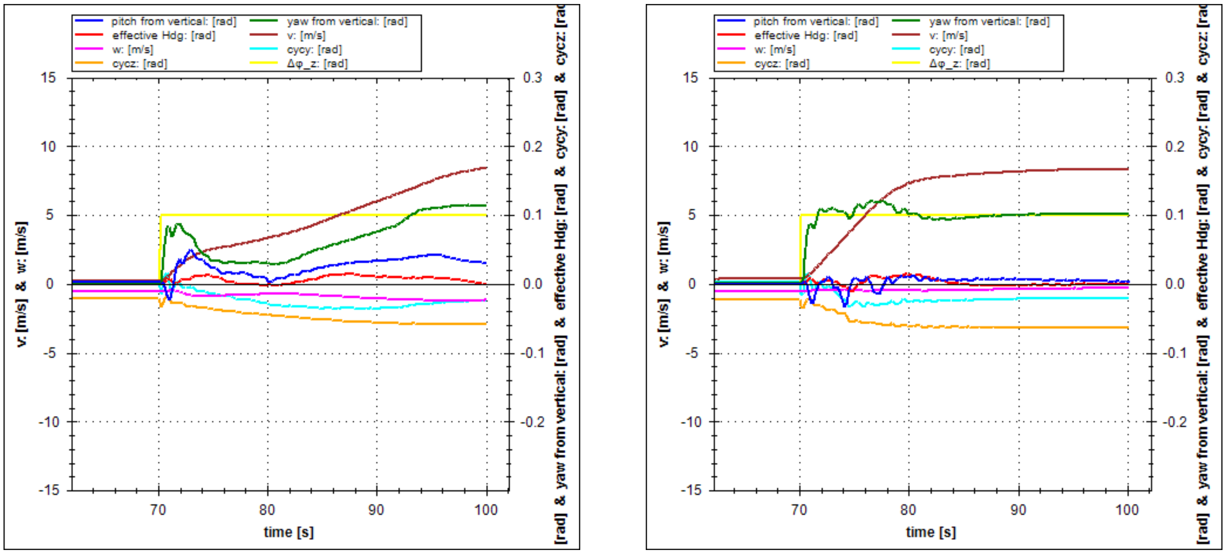
The results for trim feedforward $\boldsymbol{u}_{\mathrm{ff}} = \boldsymbol{T}\left(v\right)$ are striking. Figure 3.3 shows the response in simulation for a yaw ($\Delta\phi_z$) command step in zero wind. $\Delta\phi_y$ and $\Delta\phi_z$ are incremental rotations from fuselage vertical about the y and z axes, respectively (Figure 2.4), e.g. $\Delta\phi_y = 0$ and $\Delta\phi_z = 0$ is fuselage vertical and near hover in zero wind. In Figure 3.3 the $\Delta\phi_z$ command steps from 0 to 0.1 rad while $\Delta\phi_y$ and heading targets remain at zero. In Figure 3.3a for the standard controller (Section 3.1), the initial $\Delta\phi_z$ on-axis response is relatively good (to ˜80% of target) but is followed by a large droop to ˜30% of target and then slow correction. The $\Delta\phi_y$ off-axis response has a large peak (˜50% of on-axis target) followed by a large (˜40%), slow response and correction. In Figure 3.3b with the addition of trim feedforward, the $\Delta\phi_z$ on-axis response settles to within ˜20% quickly, and the off-axis responses (both $\Delta\phi_y$ and heading) are relatively small. This demonstrates the high sensitivity to trim changes over a small airspeed scale shown in Section 2.4.4. These simulations are with the Pitt-Peters model, and Figure 2.7 shows that cyc_y trim changes in flight are larger than this model. $\boldsymbol{T}(v)$ in these simulations is nearly perfect with "simulation truth" v and lookup table $\boldsymbol{T}(v)$ using simulation trim points with 1 m/s spacing and linear interpolation.

Figure 3.4 shows the response to a spanwise groundspeed step command from 0 (position regulation) to 8 m/s. The conditions are as for Figure 3.3 except off-axis regulation is of ground-speed w instead of tilt $\Delta\phi_y$. With the addition of trim (Figure 3.4b), the v on-axis response is almost twice as fast (4.4 s versus 8.2 s 0-100% rise time) and has less overshoot, with $\Delta\phi_z$ on-axis tilt smoother. The off-axis responses (w and heading) are initially larger.

Adding the calculated trim as a steady bias in the otherwise unmodified controller results in remarkable performance improvements for both tilt and velocity steps. A large nonlinearity is effectively removed, unburdening the standard controller and providing it with a more linear system to control (note the large change in cyc_y and cyc_z trim compared to the transient control). Section 3.3.3 will consider robustness to trim map and velocity errors.

3.3.1 Integral Control

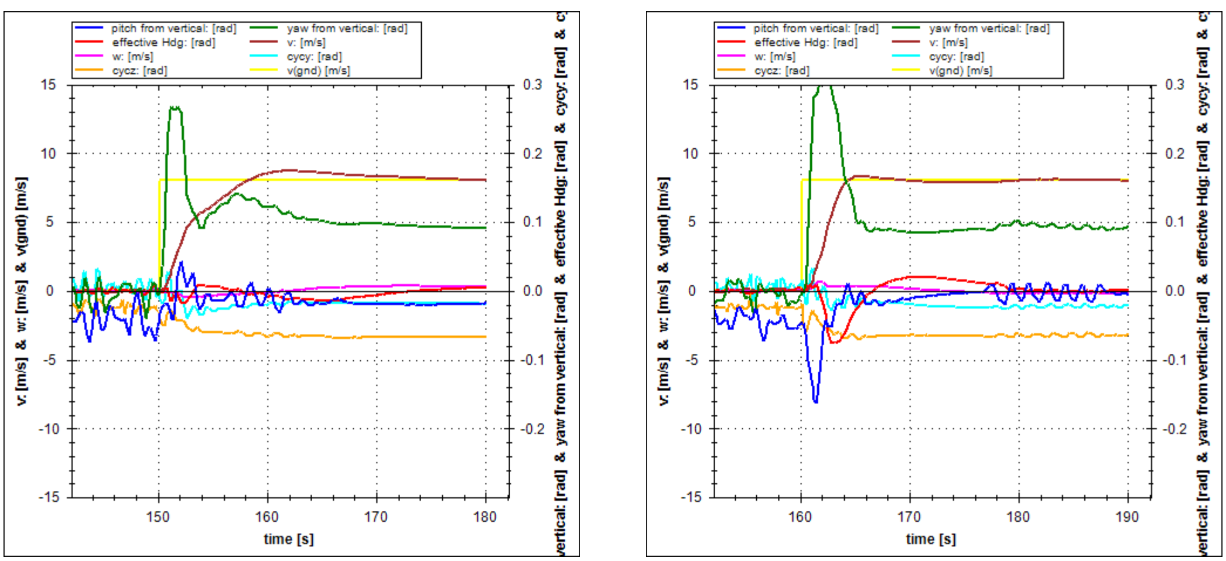
Instead of (or in addition to) trim feedforward, integral control is commonly used. As Khalil [41] noted, integral action "...is the only way to achieve asymptotic regulation in the



(a) Standard controller.

(b) With trim feedforward.

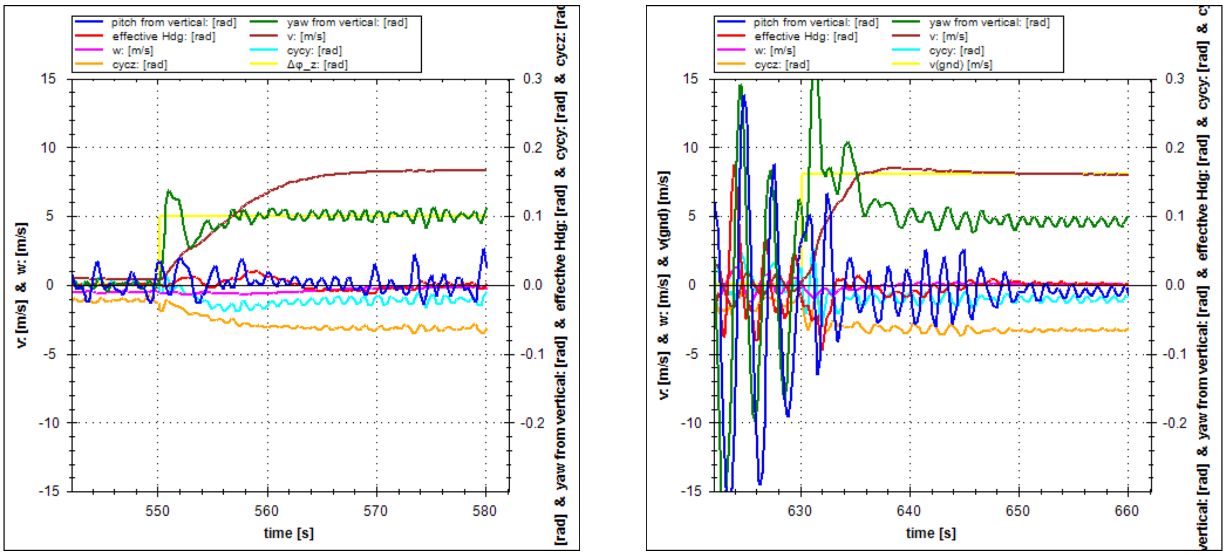
Figure 3.3: Tilt attitude command steps in simulation.



(a) Standard controller.

(b) With trim feedforward.

Figure 3.4: Velocity command steps in simulation.



(a) Tilt attitude step. (b) Velocity step.

Figure 3.5: Command steps with high integral gain in simulation.

presence of parametric uncertainties." The controllers used in Figures 3.3 and 3.4 do use integral control although it is weak enough that trim feedforward makes a large performance improvement. Figure 3.5 shows the responses to tilt attitude and velocity command steps with the integral gain increased to ten times its original value. While the oscillations show that the gain is now too high, performance is still not as good as with trim feedforward (Figures 3.3b and 3.4b). With high integral gain, the tilt attitude response has a short undershoot, and the velocity response is slower than with trim feedforward (5.5 s versus 4.4 s 0-100% rise time). Thus, the performance improvement of trim feedforward cannot easily be achieved by a simple increase in integral gain.

3.3.2 Wind Gust Disturbance Attenuation

In addition to improving command response, $\boldsymbol{T}(\boldsymbol{v})$ feedforward can improve gust rejection if airspeed is sufficiently well known. For a miniature helicopter, Danapalasingam, la Cour-

Harbo, and Bisgaard [14] added trim feedforward to a linear quadratic regulator (LQR) in order to assist the feedback loop in rejecting wind disturbances. After testing position-hold in simulation—with perfect wind measurement—they concluded that this approach was feasible, with a 17% decrease in primary axis (wind and longitudinal axis) root-mean-square position (21% decrease peak-to-peak). Note that unlike in the present work, Danapalasingam, la Cour-Harbo, and Bisgaard [14] subtract $\boldsymbol{T}\left(\boldsymbol{v}_{\text{inertial}}\right)$[1] from the feedforward term, leaving the LQR to provide a $\boldsymbol{T}\left(\boldsymbol{v}_{\text{inertial}}\right)$ equivalent.

3.3.3 Trim Map and Airspeed Errors

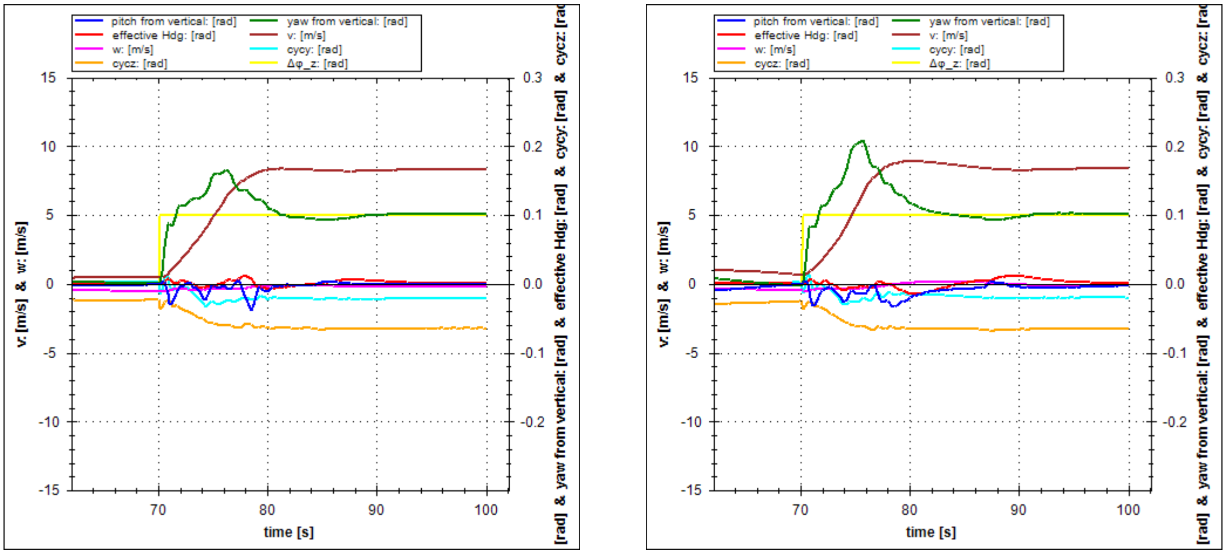
Errors in trim map $\boldsymbol{T}(v)$ are now considered. Figure 3.6 shows the responses to $\Delta\phi_z$ command steps with feedforward as in Figure 3.3b, but here amplified using scale factor s, i.e. $s\boldsymbol{T}(v)$. Compared to no feedforward (Figure 3.3a), the off-axis responses are improved but the on-axis responses have large overshoot. If overshoot is not considered worse than undershoot, the on-axis $s = 1.25$ response (Figure 3.6a) is better than without trim feedforward.

The responses for attenuated trim feedforward are shown in Figure 3.7. $s = 0.50$ (Figure 3.7a) is a notable improvement over zero trim feedforward (Figure 3.3a) and $s = 0.80$ (Figure 3.7b) is not much worse than full trim ($s = 1.00$, Figure 3.4b) and perhaps better if overshoot is considered worse than undershoot.

Next, the responses are shown in Figure 3.8 for v offset errors, i.e. $\boldsymbol{T}(v + c)$. Compared to no trim feedforward (Figure 3.3a), the off-axis responses are improved but the on-axis responses now have large overshoot. If overshoot is not considered worse than undershoot, the on-axis $c = 1\,\text{m/s}$ response (Figure 3.8a) is better than without trim feedforward. 1 m/s is not a large error, so trim feedforward is sensitive to v errors.

For practical trim feedforward, the trim map and airspeed are replaced with estimates $\hat{\boldsymbol{T}}(\hat{v})$. The effect of feedforward using estimate $\hat{\boldsymbol{T}}(\hat{v})$ can be examined by comparing to true $\boldsymbol{T}(v)$. Assuming that steady-state offset is handled by a mechanism such as integral action,

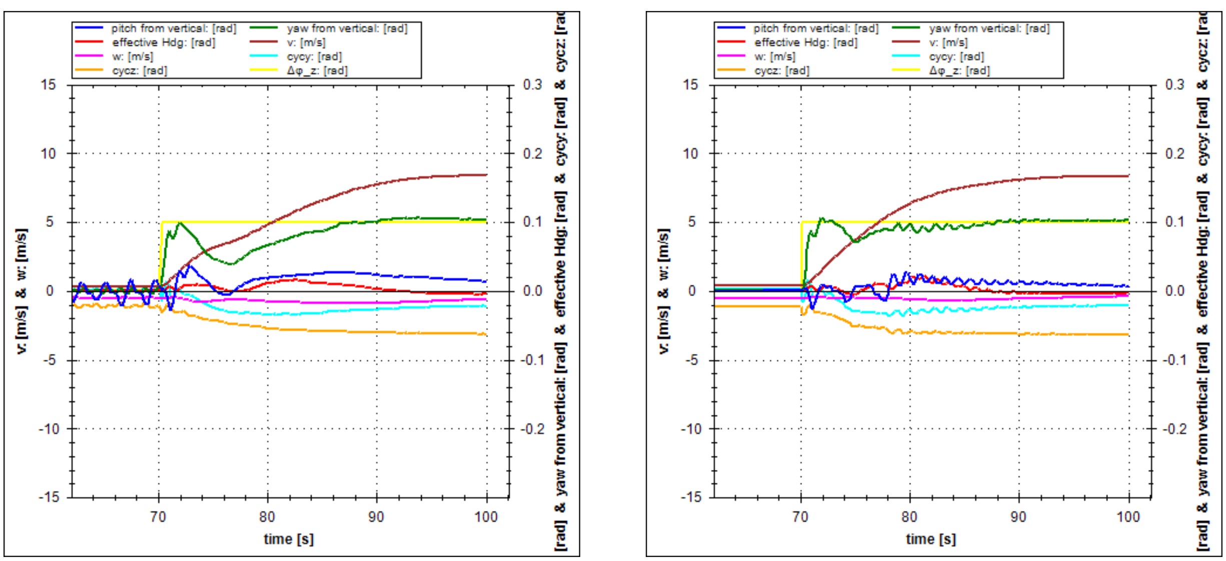
[1] Using notation from this dissertation.



(a) Trim feedforward scaled by $s = 1.25$.

(b) Trim feedforward scaled by $s = 1.50$.

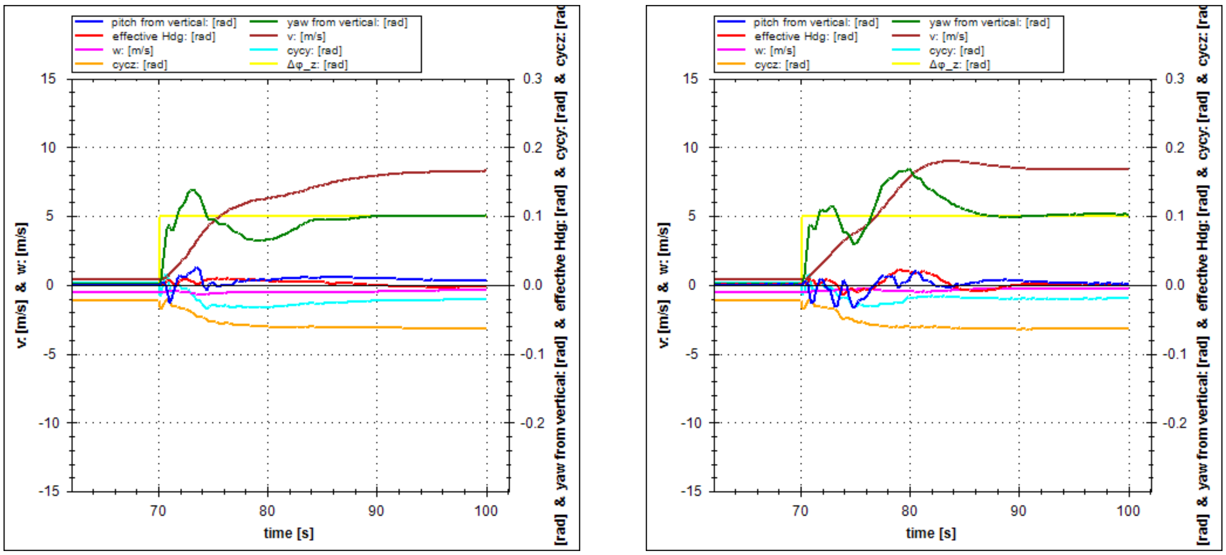
Figure 3.6: Tilt attitude command steps with amplified trim feedforward in simulation.



(a) Trim feedforward scaled by $s = 0.50$.

(b) Trim feedforward scaled by $s = 0.80$.

Figure 3.7: Tilt attitude command steps with attenuated trim feedforward in simulation.



(a) Trim feedforward with v offset error $c = 1\,\mathrm{m/s}$.

(b) Trim feedforward with v offset error $c = -1\,\mathrm{m/s}$.

Figure 3.8: Tilt attitude command steps with trim feedforward and velocity offset error in simulation.

the relevant comparison is $\mathrm{d}\hat{\boldsymbol{T}}(\hat{v})/\mathrm{d}\hat{v}$ with $\mathrm{d}\boldsymbol{T}(v)/\mathrm{d}v$ for points of interest in the map. The various cases are shown in Table 3.1. In general, the case can vary across the map and with time (as $\hat{v}$ error changes).

To handle errors in $\hat{\boldsymbol{T}}$ and $\hat{v}$, a scale parameter $0 < s < 1$ can be used, i.e. $s\hat{\boldsymbol{T}}(\hat{v})$. s is selected to balance performance and robustness to errors. This follows an approach used in industry, where s may be considered a gain with typical values from 0.5 to 0.8 [20]. The examples in Figures 3.3–3.8 indicate the potential for performance improvement while avoiding performance impairment and stability loss. Note that s scaling does not eliminate the possibility of a sign mismatch in trim slope (case 2) if there is a trim slope reversal and large enough error in $\hat{v}$, but it can be used to limit the magnitude of the resulting performance impairment.

There are alternatives to s scaling, including averaging over a velocity range or modifying

Table 3.1: Trim feedforward effects with errors.

Case		Effect compared to no trim feedforward
1	$\frac{\mathrm{d}\hat{\boldsymbol{T}}(\hat{v})}{\mathrm{d}\hat{v}} = 0$	Equivalent to no feedforward
2	$\operatorname{sgn}\left(\frac{\mathrm{d}\hat{\boldsymbol{T}}(\hat{v})}{\mathrm{d}\hat{v}}\right) \neq \operatorname{sgn}\left(\frac{\mathrm{d}\boldsymbol{T}(v)}{\mathrm{d}v}\right)$	Performance impaired, stability lost for large $\left\lvert\frac{\mathrm{d}\hat{\boldsymbol{T}}(\hat{v})}{\mathrm{d}\hat{v}}\right\rvert$
3	$\operatorname{sgn}\left(\frac{\mathrm{d}\hat{\boldsymbol{T}}(\hat{v})}{\mathrm{d}\hat{v}}\right) = \operatorname{sgn}\left(\frac{\mathrm{d}\boldsymbol{T}(v)}{\mathrm{d}v}\right)$	
3a	$\left\lvert\frac{\mathrm{d}\hat{\boldsymbol{T}}(\hat{v})}{\mathrm{d}\hat{v}}\right\rvert \leq \left\lvert\frac{\mathrm{d}\boldsymbol{T}(v)}{\mathrm{d}v}\right\rvert$	Performance improved
3b	$\left\lvert\frac{\mathrm{d}\boldsymbol{T}(v)}{\mathrm{d}v}\right\rvert < \left\lvert\frac{\mathrm{d}\hat{\boldsymbol{T}}(\hat{v})}{\mathrm{d}\hat{v}}\right\rvert \leq \left\lvert\frac{\mathrm{d}\boldsymbol{T}(v)}{\mathrm{d}v}\right\rvert + \epsilon$	Performance improved (except more overshoot)
3c	$\left\lvert\frac{\mathrm{d}\boldsymbol{T}(v)}{\mathrm{d}v}\right\rvert + \epsilon < \left\lvert\frac{\mathrm{d}\hat{\boldsymbol{T}}(\hat{v})}{\mathrm{d}\hat{v}}\right\rvert$	Performance impaired, stability lost for large $\left\lvert\frac{\mathrm{d}\hat{\boldsymbol{T}}(\hat{v})}{\mathrm{d}\hat{v}}\right\rvert$

$\hat{\boldsymbol{T}}$ to minimize performance or stability impairment. For example, with a known bound on $\hat{v}$ error the map could be flattened ($\mathrm{d}\hat{\boldsymbol{T}}(\hat{v})/\mathrm{d}\hat{v} = 0$) within the range of uncertainty, eliminating the possibility of sign mismatch in trim slope (case 2).

3.3.4 Airspeed and Trim Map Measurement and Estimation

Conventional pitot-static sensors, as used for airplanes and rotorcraft at high airspeed, are ineffective at low speed [22]. Furthermore, for a tailsitter like Flexrotor the pitot tube is in the wrong orientation for measuring horizontal airspeed in thrust-borne flight. For rotorcraft, airspeed measurement is also made difficult by interference from rotor wake. Nevertheless, low airspeed measurements remain valuable, and various scalar and vector sensors and estimation techniques have been developed over the years [22, 19].

Ultrasonic anemometers have been used widely in ground-based weather stations, and more recently on miniature rotorcraft for atmospheric research [5]. These sensors are small and accurate. For example, the Anemoment TriSonica Mini is a 0.09 x 0.09 x 0.05 m, 0.05

kg three-dimensional (direction in one dimension limited to $\pm 0.5\,\mathrm{rad}$) ultrasonic anemometer [43]. Accuracy is listed at $\pm 0.3\,\mathrm{m/s}$ or better up to 30 m/s of wind speed with output at up to 10 Hz. While accuracy on rotorcraft can be achieved by mounting well above or upwind of the airframe and rotor(s) [5], even these small sensors are a significant source of drag for miniature aircraft at high speed.

Without a low airspeed sensor, the airspeed vector is estimated. GNSS[2]/INS, as commonly found on rotorcraft, provides accurate attitude and ground (inertial) speed. The missing components are wind speeds. For Flexrotor in thrust-borne flight, wind is estimated onboard using a drag model and measurements of tilt and acceleration, which is error-prone and unsuited to rapid gust response [3]. Wind estimation errors dominate thrust-borne airspeed estimate errors. The wind estimate is aided by measurements from a ground-based anemometer when near primary takeoff and landing sites.

The trim map can be measured (and for Flexrotor has been) in flight tests. The map is measured for different configurations, e.g. Flexrotor legs closed and open, and possibly parameterized, e.g. on thrust coefficient. Due to the typical wind estimation errors, these flight tests should be conducted when the wind is steady and well known (usually light wind). This constraint could be relaxed by use of an onboard wind sensor, even if such a sensor is not installed for other flights. Online estimation and use of a trim map will be examined in Section 3.4.

3.4 Trim Online Estimation and Adaptive Control

Instead of using a trim map developed from prior data and analysis, it can be advantageous to estimate the map in flight. The primary potential advantage is handling apparent trim changes (with aircraft, state, or time) that are not handled by previously-determined trim map(s) and algorithms. These changes include known and unknown configuration changes (e.g. payload change or airframe damage) or sensor biases (e.g. airspeed offset). Poten-

[2]e.g GPS

tial disadvantages of online estimation are insufficient accuracy, robustness, or algorithm efficiency.

3.4.1 Trim Point Estimation

In the next section, a curve will be fit to trim points. First, individual trim points will be estimated. To determine trim points, the simplest and most reliable option is to only use filtered data from steady-state flight as is done in flight test to obtain trim data. However, this would discard much or even all the data from a mission-oriented flight. Another option is to weight the points, de-weighting those with more significant dynamics. Here we will correct for dynamics and provide weightings since the corrections are only approximate.

The following linear model of the attitude dynamics is used, which is similar to Equation (3.2).

$$\dot{\boldsymbol{\omega}} = \boldsymbol{A}\boldsymbol{\omega} + \boldsymbol{B}\boldsymbol{u}_\delta \tag{3.5}$$

with body-axis rotation rate vector $\boldsymbol{\omega}$ and the rotational control vector change from trim $\boldsymbol{u}_\delta$ (i.e. $\boldsymbol{u}_\delta = \boldsymbol{u} - \boldsymbol{u}_{\text{trim}}$). For non-turning trim flight all three terms are zero, i.e. $\dot{\boldsymbol{\omega}} = \boldsymbol{\omega} = \boldsymbol{u}_\delta = 0$. Inverting $\boldsymbol{B}$,

$$\boldsymbol{u}_\delta = \boldsymbol{B}^{-1}\left(\dot{\boldsymbol{\omega}} - \boldsymbol{A}\boldsymbol{\omega}\right) \tag{3.6}$$

Equation (3.6) is used to estimate $\hat{\boldsymbol{u}}_\delta$, with the trim point estimate then

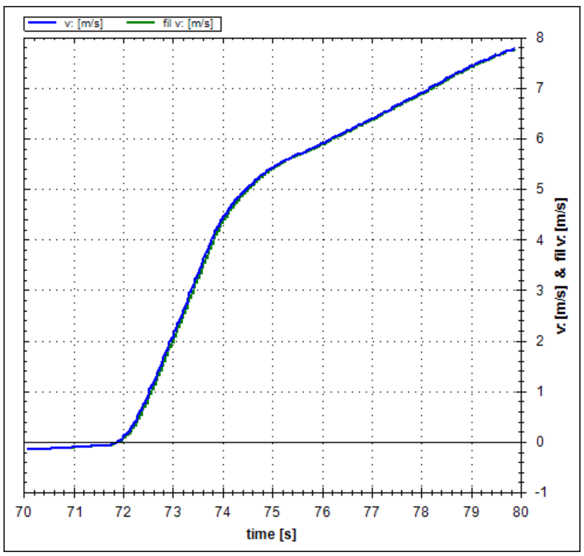
$$\hat{\boldsymbol{u}}_{\text{trim}} = \boldsymbol{u}\left(t - \tau\right) - \hat{\boldsymbol{u}}_\delta \tag{3.7}$$

$\boldsymbol{A}$ and $\boldsymbol{B}$ are from an aircraft model[3], $\boldsymbol{\omega}$ is from INS, $\dot{\boldsymbol{\omega}}$ is a finite difference of $\boldsymbol{\omega}$, and $\boldsymbol{u}$ is from a history buffer with τ the response lag, i.e. the time from control command to $\boldsymbol{\omega}$ response. v, $\boldsymbol{\omega}$, $\dot{\boldsymbol{\omega}}$, and $\boldsymbol{u}$ are all filtered with a time constant long enough for noise and control chatter but short enough not to filter velocity (airspeed) dynamics.

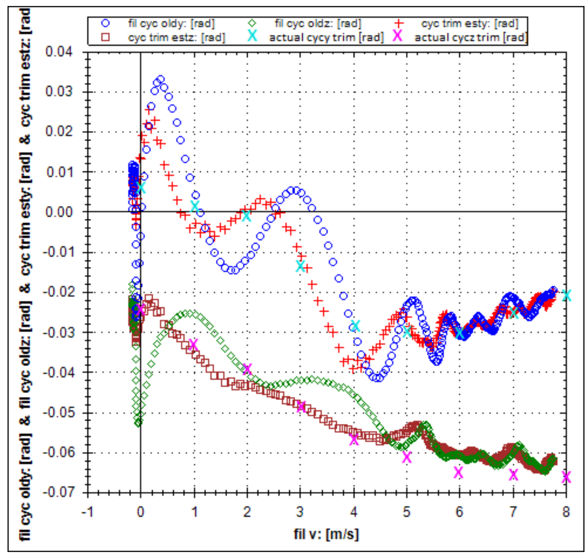
Figure 3.9 shows trim estimation using Equation (3.7) during a velocity step (as in Figure 3.4a). Figure 3.9b shows that trim estimate $\hat{\boldsymbol{u}}_{\text{trim}}$ significantly reduces the dynamics in $\boldsymbol{u}\left(t - \tau\right)$, more closely approximating actual trim $\boldsymbol{u}_{\text{trim}}$.

---

[3]The model could itself be estimated online.



(a) Velocity v.



(b) Cyclic: filtered and lagged $\boldsymbol{u}\,(t-\tau)$ (blue circles and green diamonds), trim estimate $\hat{\boldsymbol{u}}_{\text{trim}}$ (red '+'s and brown squares), and simulation truth trim $\boldsymbol{u}_{\text{trim}}$ (cyan 'x's and magenta 'x's).

Figure 3.9: Trim estimation from velocity step in simulation.

Figure 3.10 shows data from the almost ten minutes of flight used to create the trim points in Figure 2.7. Here, v and $\boldsymbol{u}(t-\tau)$ are filtered and $\hat{\boldsymbol{u}}_{\text{trim}}$ calculated using Equation (3.7) as described above. Since these data are from flight test to gather trim points, the data should be relatively clean although there are dynamics present, especially during the transitions between test points. Figure 3.10 also shows cubic polynomial least squares fits to the data. The coefficients of determination R^2 increase from 0.67 and 0.83 for the fits to $\text{cyc}_y(t-\tau)$ and $\text{cyc}_z(t-\tau)$, respectively, to 0.85 and 0.89 for the fits to $\hat{\boldsymbol{u}}_{\text{trim}}$, indicating a decrease in scatter and probably an increase in accuracy. For fits to $\hat{\boldsymbol{u}}_{\text{trim},\boldsymbol{A\omega}=0}$, i.e. $\hat{\boldsymbol{u}}_{\text{trim}}$ calculated with $\boldsymbol{A\omega}=0$, R^2 is 0.80 and 0.87, respectively. This indicates that the $\boldsymbol{A\omega}$ term does have a significant effect for these data.

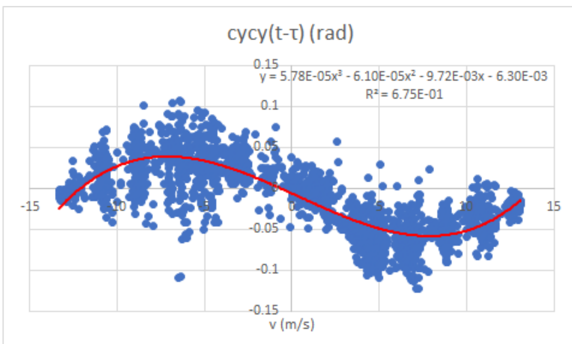
3.4.2 Trim Curve Fitting

As shown in the previous section, Equation (3.7) can be used to generate trim point estimates from flight data. The next step is to fit a curve to these points. A well-established approach to fitting a nonlinear curve is to use a polynomial least-squares fit. Figure 3.11 shows polynomial fits to the flight trim data of Figure 2.7. A cubic polynomial is sufficient to qualitatively model the cyc_y trim slope reversals (Figure 3.11a), and a quintic polynomial better identifies the airspeeds at which these reversals occur (Figure 3.11b).

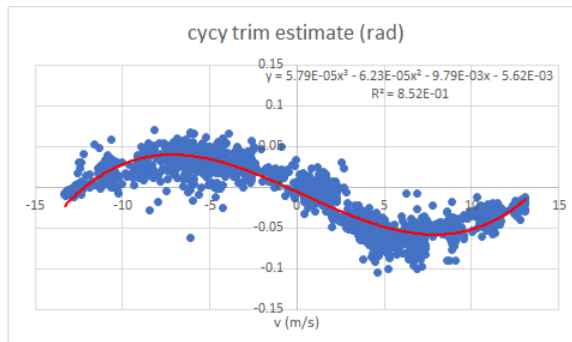
When fitting to trim point estimates from Equation (3.7) using least squares, a weight should be used for each point, i.e. weighted least squares. This allows points with significant dynamics or off-axis rates to be de-weighted. The weight used is

$$w = \frac{1}{\underbrace{\dot{\omega}_y^2 + \dot{\omega}_z^2 + w_1\dot{\omega}_x^2}_{\text{rotational accelerations}} + \underbrace{w_2\left(\omega_y^2 + \omega_z^2\right) + w_3\omega_x^2}_{\text{rotation rates}} + \underbrace{w_4 w^2 + w_5\dot{h}^2}_{\text{off-axis speed and altitude rate}} + w_6} \tag{3.8}$$

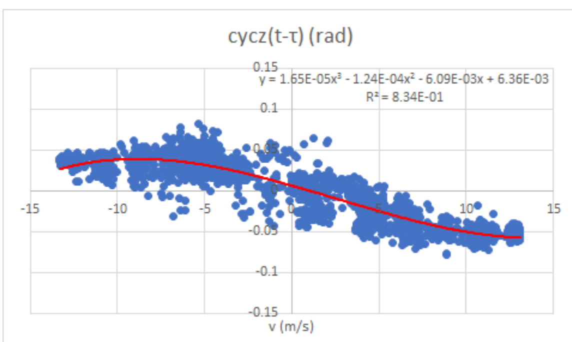
with component weights $w_1, \ldots, w_6$. At a true steady-state trim (with w, altitude rate, and turn rate zero), only the w_6 term is non-zero. For online curve fitting, appropriately weighted points from previous flight test and analysis can be used to initialize the system.



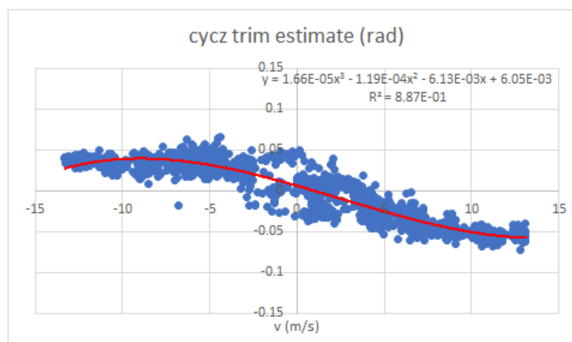
(a) Filtered and lagged $\mathrm{cyc}_y\,(t-\tau)$.



(b) Trim estimate $\hat{\mathrm{cyc}}_{y_{\mathrm{trim}}}$.

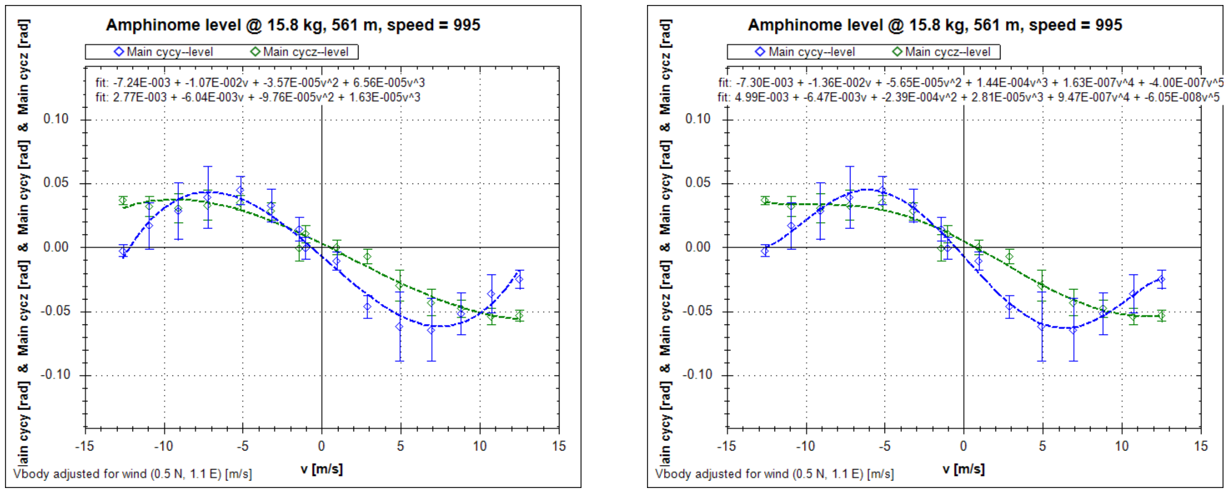


(c) Filtered and lagged $\mathrm{cyc}_z\,(t-\tau)$.



(d) Trim estimate $\hat{\mathrm{cyc}}_{z_{\mathrm{trim}}}$.

Figure 3.10: Cyclic trim data from flight.



(a) Degree 3 (cubic) polynomial fits.

(b) Degree 5 (quintic) polynomial fits.

Figure 3.11: Polynomial least squares fits to trim flight data.

For fitting of nonlinear curves, there are alternatives to global polynomials that instead use local basis functions. One option that is suitable for this application is the spline. A spline is constructed piecewise from polynomials, with the pieces joined at *knots* with certain continuity requirements (e.g. matching value and derivatives) [44]. Compared to global polynomials, splines use local polynomials, usually of low degree. Two approaches for constructing a spline approximation to data are the *least-squares spline approximation* and the *smoothing spline*. Least-squares spline approximation requires the knot locations to be specified while a smoothing spline has a knot at each data point and requires specification of a smoothing parameter to weight fit versus smoothness. For a least-squares spline approximation of the flight trim data of Figure 2.7, candidate parameters are cubic polynomials on the intervals $\ldots, [-9,-3], [-3,3], [3,9], \ldots$ m/s. These intervals, of diameter 6 m/s, provide flexibility to fit the cyc_y trim slope reversals, approximately center intervals on hover and the reversals, and avoid a large number of intervals with associated overfitting.

Figure 3.12 shows least-squares approximations to flight trim data using cubic splines. These splines were fit using the MATLAB SPAP2 function. Splines with three, five, and seven pieces are shown. Uniformly-spaced intervals between -15 to 15 m/s were used, e.g. $[-15, -5], [-5, 5], [5, 15]$ m/s for the three-piece spline. A two-piece spline (not shown) was similar to a global cubic polynomial fit, e.g. errors in cyc_y slope reversal airspeeds. In Figure 3.12, the seven-segment splines appear to be overfit, curving tightly near certain end points. The desirable number of cubic spline pieces for these data is perhaps three to five, with even the five-piece splines appearing to curve more tightly than desired in places.

Figure 3.13 shows cubic smoothing spline approximations to flight trim data. These splines were fit using the MATLAB CSAPS function. Splines for different smoothing parameters p are shown. The smoothing parameter allows the desired amount of smoothing to be selected, but significantly different values of p are needed as data point spacing changes (e.g. from the "few" to the "many" data sets) to avoid underfitting or overfitting the data. Therefore, the smoothing parameter must be chosen carefully as the characteristics of the data change.

3.4.3 Recursive Least Squares with Local Forgetting

The recursive least squares algorithm is well established. Following McGeer [45], for $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{u}$ with $\boldsymbol{y}$ of length N, $\boldsymbol{u}$ of length M, and $N > M$, the weighted least squares solution minimizes

$$J = \sum_{n=1}^{N} w_n^2 \left(y_n - \boldsymbol{A}_n \boldsymbol{u}\right)^2 = \left(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{u}\right)^T \boldsymbol{W}^2 \left(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{u}\right)$$

with weights w_n. The solution is

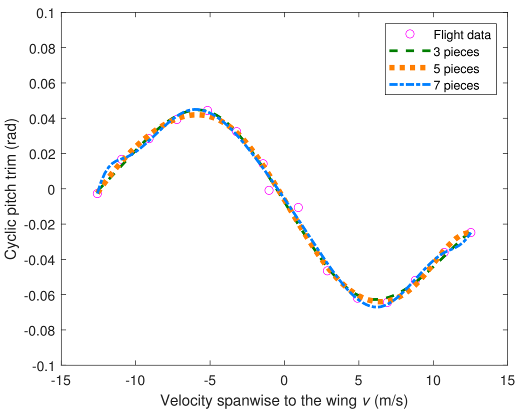
$$\boldsymbol{u} = \left(\boldsymbol{A}^T \boldsymbol{W}^2 \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{W}^2 \boldsymbol{y} \tag{3.9}$$

Continuing with McGeer [45], in the recursive case

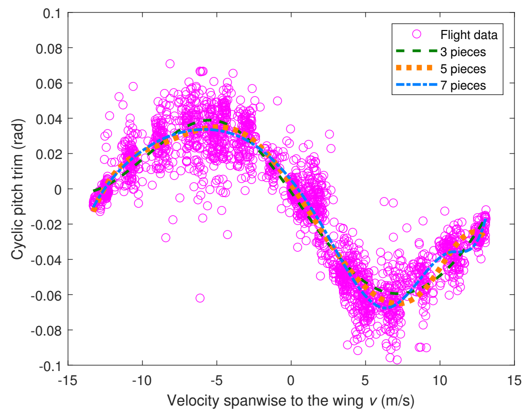
$$\begin{aligned} J &= w_{N+1}^2 \left(y_{N+1} - \boldsymbol{a}_{N+1}\boldsymbol{u}\right)^2 + \sum_{n=1}^{N} w_n^2 \left(y_n - \boldsymbol{A}_n \boldsymbol{u}\right)^2 \\ &= w_{N+1}^2 \left(y_{N+1} - \boldsymbol{a}_{N+1}\boldsymbol{u}\right)^2 + \left(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{u}\right)^T \boldsymbol{W}^2 \left(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{u}\right) \end{aligned}$$

with solution

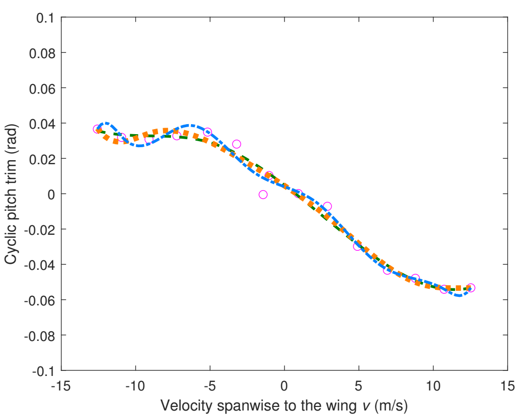
$$\boldsymbol{u} = \left(\boldsymbol{A}^T \boldsymbol{W}^2 \boldsymbol{A} + w_{N+1}^2 \boldsymbol{a}_{N+1}^T \boldsymbol{a}_{N+1}\right)^{-1} \left(\boldsymbol{A}^T \boldsymbol{W}^2 \boldsymbol{y} + w_{N+1}^2 \boldsymbol{a}_{N+1}^T y_{N+1}\right) \tag{3.10}$$



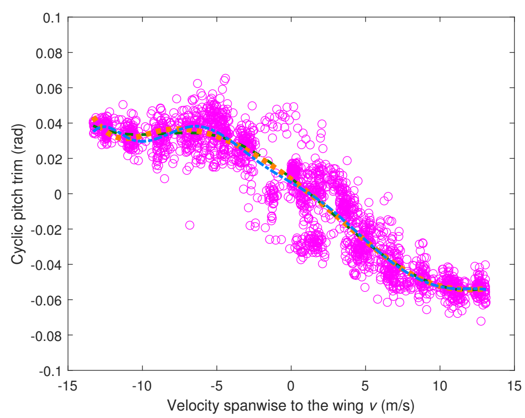
(a) cyc_y for the few points of Figure 2.7.



(b) cyc_y for the many points of Figure 3.10.

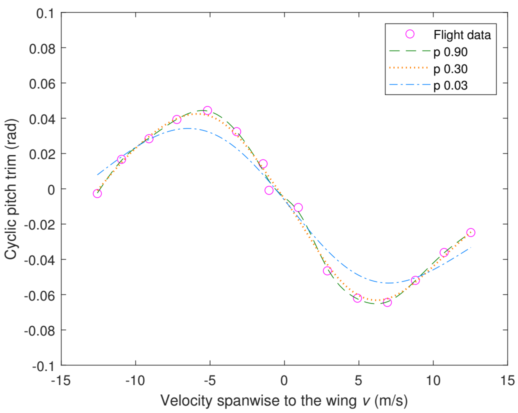


(c) cyc_z for the few points of Figure 2.7.

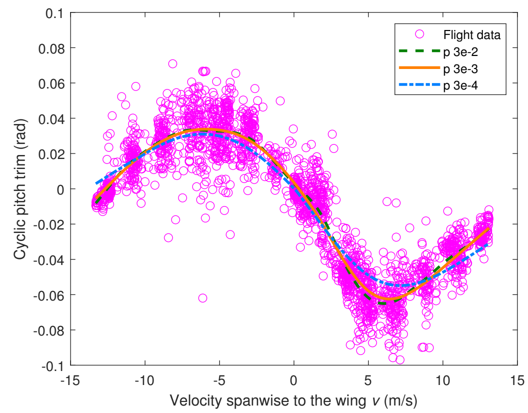


(d) cyc_z for the many points of Figure 3.10.

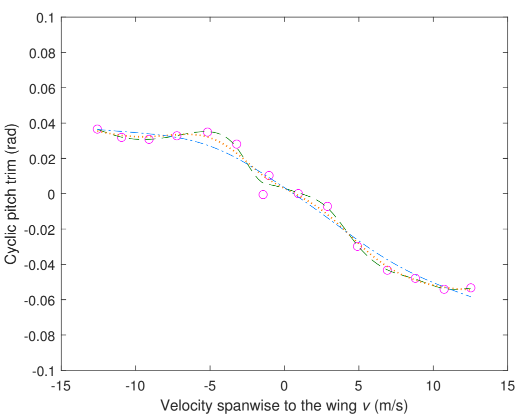
Figure 3.12: Least-squares cubic spline approximations to flight trim data.



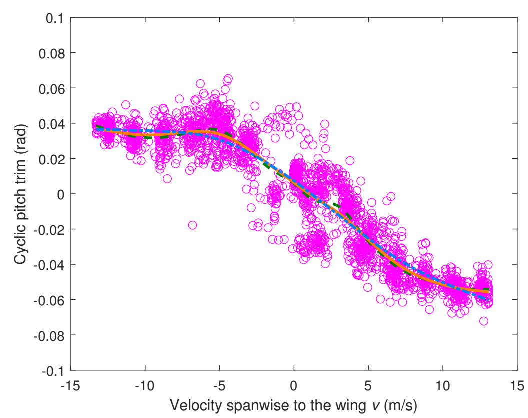
(a) cyc_y for the few points of Figure 2.7.



(b) cyc_y for the many points of Figure 3.10.



(c) cyc_z for the few points of Figure 2.7.



(d) cyc_z for the many points of Figure 3.10.

Figure 3.13: Cubic smoothing spline approximations to flight trim data.

With *forgetting factor* $0 < \lambda \leq 1$

$$\boldsymbol{A}^T\boldsymbol{W}^2\boldsymbol{A}\big|_{N+1} = \lambda\ \boldsymbol{A}^T\boldsymbol{W}^2\boldsymbol{A}\big|_N + w_{N+1}^2\boldsymbol{a}_{N+1}^T\boldsymbol{a}_{N+1} \tag{3.11}$$

$$\boldsymbol{A}^T\boldsymbol{W}^2\boldsymbol{y}\big|_{N+1} = \lambda\ \boldsymbol{A}^T\boldsymbol{W}^2\boldsymbol{y}\big|_N + w_{N+1}^2\boldsymbol{a}_{N+1}^T y_{N+1} \tag{3.12}$$

$0 < \lambda < 1$ results in *exponential forgetting* [68] which is used when the $\boldsymbol{u}$ of interest may vary with time (i.e. for a time-varying system or for short-term learning of different portions of the state space [74]), and to keep the matrix from growing unbounded.

Short-term learning such as exponential forgetting recursive least squares has difficulties if the data are not persistently exciting, a well-known problem in estimation and adaptive control [37]. One solution is local structure and learning. For recursive least squares, *directional forgetting* algorithms have been developed to only forget in the direction where new data are coming from [8].

For the present online trim estimation problem, information should be forgotten since the configuration, conditions, or sensor biases could change. However, it is typical to have only recently experienced a small subset of airspeeds (i.e. the input is not uniformly exciting), so a local forgetting algorithm is developed here to retain useful information at not-recently-visited airspeeds.

To enable local forgetting, define

$$\boldsymbol{A}^T\boldsymbol{W}^2\boldsymbol{A} = \sum_{b=1}^{B}\boldsymbol{A}_b^T\boldsymbol{W}_b^2\boldsymbol{A}_b \tag{3.13}$$

$$\boldsymbol{A}^T\boldsymbol{W}^2\boldsymbol{y} = \sum_{b=1}^{B}\boldsymbol{A}_b^T\boldsymbol{W}_b^2\boldsymbol{y}_b \tag{3.14}$$

with $1 \leq B \leq N$. Then from Equation (3.9),

$$\boldsymbol{u} = \left(\sum_{b=1}^{B}\boldsymbol{A}_b^T\boldsymbol{W}_b^2\boldsymbol{A}_b\right)^{-1}\sum_{b=1}^{B}\boldsymbol{A}_b^T\boldsymbol{W}_b^2\boldsymbol{y}_b \tag{3.15}$$

From Equations (3.11) and (3.12), the one submatrix $\boldsymbol{A}_b^T\boldsymbol{W}_b^2\boldsymbol{A}_b$ and one vector $\boldsymbol{A}_b^T\boldsymbol{W}_b^2\boldsymbol{y}_b$

associated with the $N+1$ data are updated with

$$\boldsymbol{A}_b^T \boldsymbol{W}_b^2 \boldsymbol{A}_b\big|_{N+1} = \left(1 - w_{N+1}^2\right) \boldsymbol{A}_b^T \boldsymbol{W}_b^2 \boldsymbol{A}_b\big|_N + w_{N+1}^2 \boldsymbol{a}_{N+1}^T \boldsymbol{a}_{N+1} \tag{3.16}$$

$$\boldsymbol{A}_b^T \boldsymbol{W}_b^2 \boldsymbol{y}_b\big|_{N+1} = \left(1 - w_{N+1}^2\right) \boldsymbol{A}_b^T \boldsymbol{W}_b^2 \boldsymbol{y}_b\big|_N + w_{N+1}^2 \boldsymbol{a}_{N+1}^T y_{N+1} \tag{3.17}$$

where $\lambda = 1 - w_{N+1}^2$ for $0 \leq w_{N+1}^2 < 1$. With this formula, forgetting only occurs locally and only in the appropriate amount given the weight of the new data, in order to keep the updates from over- or under-weighting the updated submatrix relative to the others. The w^2 formula should be scaled as necessary, e.g. to keep typical $w^2 \approx 0.01$ or maximum $w^2 \approx 0.1$. This algorithm is essentially the directional forgetting algorithm of Shorten, Schütte, and Fagan [68], but here the weight on new data is less than one and the forgetting factor is calculated from the new data weight.

For some applications, this algorithm relaxes the persistent excitation requirement. For example, extended time at one trim condition is acceptable for the present trim estimation problem if the submatrices are assigned to tightly-spaced airspeed intervals. One submatrix could converge to a point without significantly degrading the overall curve fit. Note that the additional computations required for this algorithm compared to exponential forgetting least squares are matrix additions (the number scaling with B), and logic to determine which submatrix and vector to update.

As an example, consider division of the input space into uniform segments between $x_{\min}$ and $x_{\max}$ with update submatrix and vector assignment based on the second column of $\boldsymbol{A}$ (e.g. x for polynomial regression). Then, b for the matrix and vector to be updated is

$$b = \text{floor}\left(B \frac{\boldsymbol{a}_2 - x_{\min}}{x_{\max} - x_{\min}}\right) + 1 \tag{3.18}$$

3.4.4 Trim Feedforward Adaptive Control

The results from the previous sections are combined into trim feedforward adaptive control as follows. The estimation algorithm is

1. Initialize trim map estimate $\hat{\boldsymbol{T}}(v)$ and least squares estimator to a polynomial fit to previous flight data, e.g. as in Figure 3.11b.

2. During thrust-borne flight, estimate trim points $\hat{\boldsymbol{u}}_{\text{trim}}$ using Equation (3.7).

3. Use $\hat{\boldsymbol{u}}_{\text{trim}}$ to update recursive least squares with local forgetting (Equations (3.16) and (3.17) and weights from Equation (3.8)).

4. Update trim map estimate $\hat{\boldsymbol{T}}(v)$ to a polynomial fit from recursive least squares with local forgetting using Equation (3.15).

5. Continue with step 2.

For control, the current scaled trim map is added to the output of the rotation controller, i.e. $\boldsymbol{u} = \boldsymbol{u}_{\text{rc}} + s\hat{\boldsymbol{T}}(\hat{v})$ (Figure 3.2). The control is adaptive since the trim estimate used for control is updated online.

3.5 Conclusions

Trim feedforward can provide substantial performance improvements when small changes in a trim variable result in large changes in control trim. This is the case for Flexrotor cyclic trim as a function of horizontal airspeed in thrust-borne flight. Bounded trim function scale errors can be handled by attenuating feedforward control values. However, trim function argument errors (e.g. airspeed for the Flexrotor example) are not as easily managed.

For low airspeed trim feedforward input, accurate measurements are needed. Sensors capable of such measurements are available but are not always practical due to weight, drag, and physical robustness limitations. Furthermore, measurement at one or a few points might not measure potentially important variations in flow across the rotor and wing. If airspeed sensors are used, changes due to wind gusts are also directly measured, with trim feedforward then promising gust response attenuation.

The trim function can be estimated online, accounting for trim changes and function input errors (e.g. an input shift). For fitting a trim function to data, local forgetting is a practical (i.e. computationally tractable) extension to recursive least squares, relaxing the requirement for persistent excitation across the domain.

Chapter 4

ADAPTIVE CONTROL

This chapter primarily develops adaptive control techniques for improved performance and robustness. The techniques were tested with the Flexrotor low disc loading tailsitter in simulation, resulting in moderate performance improvements that are dependent on learning rates. First, Section 4.1 explores two types of position-regulation oscillations observed with Flexrotor, examples of behaviors to be corrected with improved control. Next, an approach to neural network adaptive control is described after first explaining why this approach was selected for use here (Section 4.2). Section 4.3 presents and tests adaptive rotational control.

As noted in the Introduction, typical rotorcraft at low speed have instabilities, inter-axis coupling, and high-order dynamics that make the control problem difficult [77]. Furthermore, modeling rotorcraft flight dynamics from first principles usually results in limited accuracy [77]. For example, the algorithm described in the Modeling chapter resulted in a nonlinear trim model that is qualitatively correct but still has significant quantitative differences from flight data (Figure 2.7). System identification (e.g. parameter estimation from flight data) can result in more accurate models [77]. Nevertheless, accurate three-dimensional low airspeed measurements are rarely available for rotorcraft, resulting in large uncertainty about key aircraft states even with an accurate model. Rotorcraft dynamics have additional uncertainty from disturbances such as wind gusts.

Adaptive control can improve performance in the presence of uncertainty, accommodating rather than only tolerating the uncertainty [58]. This motivates the use of adaptive control here. Adaptive feedforward control was developed in the Trim Feedforward chapter, addressing uncertainties in trim. In this chapter, adaptive control is developed for uncertain or time-varying dynamics and trim.

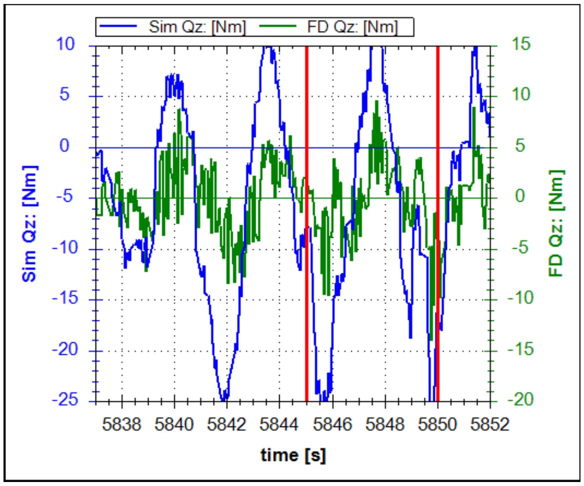
4.1 Control Oscillations

When regulating position (e.g. relative to a fixed or moving point for landing), Flexrotor dynamics anomalies have resulted in tilt attitude oscillations (e.g. Figure 2.3). The characteristic period is usually that of an *undercontrol* (e.g. 3–4 s) or *overcontrol* (e.g. 1 s) oscillation [46]. These oscillations are examples of behaviors to be corrected with adaptive control. Investigation and correction of these anomalies has been hindered by difficulty in consistently reproducing the oscillations in flight under apparently similar conditions and limited ability to produce the oscillations at all in simulation.

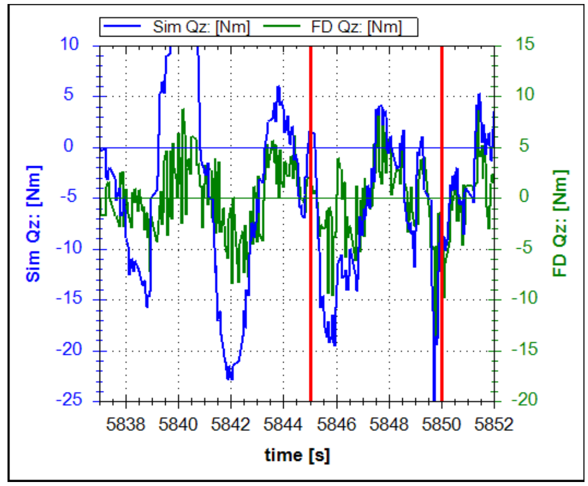
4.1.1 Undercontrol Oscillations

The undercontrol oscillation was named when analysis indicated that the oscillation was due to low control authority. However, the oscillation amplitude might be large enough that the variation is in loading, e.g. the loading indicated by the control trim of Figure 2.7.

To explore the hypothesis of oscillation due to loading variation with airspeed, Figure 4.1 shows torques during part of a Flexrotor undercontrol oscillation when regulating position in flight [51]. The torques are about the aircraft z axis (Figure 2.4), i.e. the yaw tilt axis. In addition to yaw tilt, the oscillation also had significant pitch tilt, heading, and altitude. For the z axis, "Sim Q_z" is torque as calculated from reported state by the Flexrotor simulation model (with the Pitt-Peters model of Section 2.4.3). "FD Q_z" is torque calculated from a finite difference of reported INS rotation rate. Figure 4.1a shows a significant difference between model torque ("Sim Q_z") and that calculated from actual aircraft motion ("FD Q_z"). This discrepancy is a possible explanation for the oscillation and led to the hypothesis that cyclic authority varies with in-plane airspeed. In Figure 4.1b an additional 5 N m per m/s of velocity spanwise to the wing v (Figure 2.4) has been added to the simulation model "Sim Q_z". There is a notable reduction in the difference from "FD Q_z" during the period where velocity broadside to the wing w varies little compared to velocity spanwise v (5845–5850 s). The rate 5 N m per m/s was selected to approximately minimize the torque differences in this period.



(a) Standard model (with Pitt-Peters).



(b) Model with additional 5 N m per m/s of velocity spanwise to the wing v.

Figure 4.1: Model and finite difference yaw tilt torques for a Flexrotor during an *undercontrol* oscillation in flight [51]. w velocity varies little compared to v velocity in the period marked in red (5845–5850 s). See Figure 2.4 for z, v, and w geometry.

5 N m is about 0.02 rad of cyclic which is at least plausible as a mismatch between model and reality (e.g. by considering Figure 2.7). This is promising for explaining undercontrol oscillations as a speed-dependent loading.

However, the other discrepancies in Figure 4.1 were not easily reduced by adding a torque proportional to broadside airspeed w. Furthermore, a separate position-regulation oscillation [54]—mainly in pitch—was examined. For that oscillation, the addition of torque proportional to airspeed broadside to the wing w did not easily reduce the discrepancy between simulation and finite-difference pitch torques.

Regardless of the causes of undercontrol oscillations, control should be robust to the uncertainties in dynamics and trim that can lead to these and other behaviors.

4.1.2 Overcontrol Oscillations

Figure 2.13 shows Flexrotor control authority—as calculated with the Pitt-Peters model of Section 2.4.3—from hover to 13 m/s airspeed spanwise to the wing (covering much of the thrust-borne flight envelope). Over this domain, minimum authority scale is 0.73 (at 7 m/s and 2.0 rad azimuth), and maximum authority is 0.93 (at 13 m/s and 0.4 rad azimuth). This is a ratio of 1.3, which is less than 3 dB (and less than 2 dB from the hover value of 0.82). This moderate variation falls within the range handled by standard gain margins, e.g. 3–6 dB [77]. However, it is not necessarily sufficient to solely rely on gain margin given performance requirements, system characteristics, and model uncertainties. The authority variations as shown in Figure 2.13 can be used as a model for control. In addition, control improvements to address undercontrol oscillations could allow for low-enough gains to also avoid overcontrol oscillations despite remaining uncertainty or variation in control authority with airspeed.

4.2 Neural Network Adaptive Control

Before considering the adaptive control approaches used in this chapter, note that the standard Flexrotor controller (Section 3.1) uses adaptive control in the form of adaptive cyclic authority compensation as described in Section 3.1.1.

The adaptive control algorithms described in this chapter start with the methods developed at the Georgia Institute of Technology (Georgia Tech) by A. J. Calise, R. T. Rysdyk, E. N. Johnson, et al. [7, 38]. This approach is verified in the sense that it has been tested on more than 1000 flights with many different types of miniature aircraft [74]. In addition, this approach has been deployed in guided munition production systems, perhaps the only adaptive flight control system to reach production [58]. The processing requirements for these methods are relatively low, e.g. processing requirements were dominated by extended Kalman filter state estimation calculations on a miniature helicopter with adaptive control for translation and rotation [74].

For rotorcraft, the Georgia Tech approach has been successfully tested in flight for rotational and translational control [38]. As will be shown in more detail, the Georgia Tech approach uses adaptation to correct for dynamic inversion model errors [38]. The use of this architecture for rotorcraft can be motivated as follows. To start, the *explicit model-following architecture* is widely used for production rotorcraft, allowing independent design of command and feedback responses [77]. Compared to typical explicit model-following with a lower-order inverse model, performance can be improved by using a higher-order inverse model, e.g. in a dynamic inversion controller [77] For a linear single-input single-output system the dynamic inversion architecture is identical to explicit model-following [77]. Dynamic inversion, which uses feedback to replace plant dynamics with desired dynamics (i.e. feedback linearization), has been successfully used in flight control on the Lockheed Martin F-35 airplane [74, 77]. However, dynamic inversion requires an accurate model—or adaptation for model errors, the adaptive approach being described here. Further details on neural network adaptive control as described here are in B. L. Stevens, F. L. Lewis, and E. N. Johnson [74] and other references [7, 38, 58].

4.2.1 Model Reference Adaptive Control Based on Dynamic Inversion

Figure 4.2 shows the use of adaptation to correct model error in a dynamic inversion controller as in the Georgia Tech approach [74]. The adaptive element uses the error in the nominal inverse to provide a correction, i.e. $\boldsymbol{\nu}_{\mathrm{ad}}$ is the model error in the nominal controller design. Determining $\boldsymbol{\nu}_{\mathrm{ad}}$ to correct for the model error as a function of state and input is a nonlinear curve fitting problem [74]. The selected methods for this curve fitting problem are a single hidden layer artificial neural network and a parametric neural network. Other options include radial basis function neural networks.

4.2.2 Single Hidden Layer Neural Network

Figure 4.3 shows a single hidden layer neural (SHL) network, i.e. there is a single layer of neurons between the input and output layers. $\bar{\boldsymbol{x}} \in \Re^{(n_1+1)}$ is the vector of inputs, i.e.

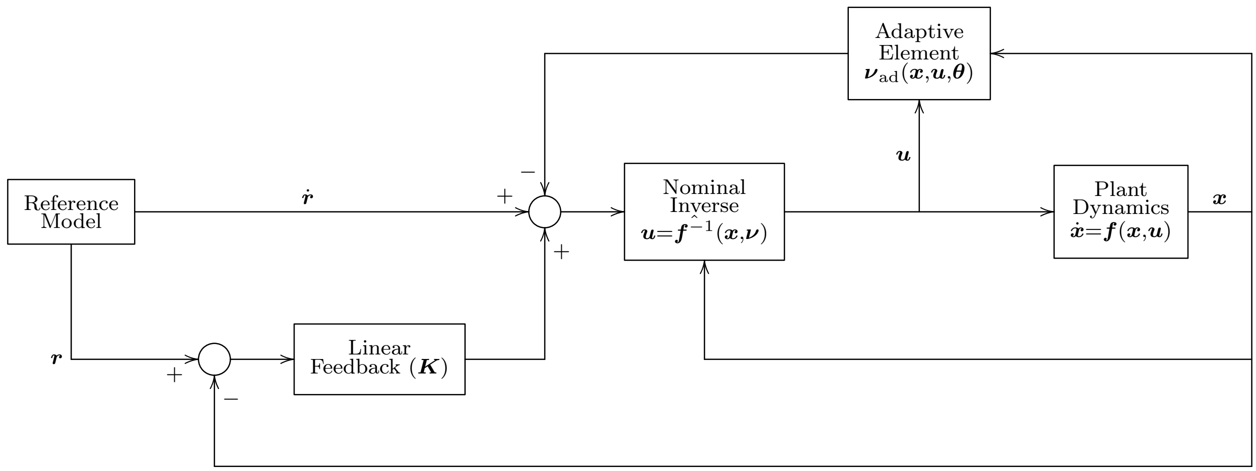


Figure 4.2: Model reference adaptive control (MRAC) based on dynamic inversion (reproduction of Figure 9.2-1 in [74]). $\boldsymbol{\theta}$ are internal parameters of the adaptive element.

$\bar{\boldsymbol{x}}^T = \begin{bmatrix} b_v & \boldsymbol{x}_{\text{in}}^T \end{bmatrix}$ where b_v is the inner layer bias and $\boldsymbol{x}_{\text{in}} \in \Re^{n_1}$ in this application is a vector of system states and inputs. $\boldsymbol{\nu}_{\text{ad}} \in \Re^{n_3}$ is the vector of outputs, $\boldsymbol{V} \in \Re^{(n_1+1)\times n_2}$ are the input weights, $\boldsymbol{W} \in \Re^{(n_2+1)\times n_3}$ are the output weights, and b_w is the outer layer bias.

The neural network operates as follows. First, the squashing function is defined as [38, 74]

$$\sigma_j\left(z_j\right) = \frac{1}{1 + e^{-a_j z_j}}$$

for $j = 1, \ldots, n_2$ with activation potentials

$$a_j = \tan\left(\arctan\left(a_{\min}\right) + \left[\arctan\left(a_{\max}\right) - \arctan\left(a_{\min}\right)\right]\frac{j-1}{n_2-1}\right)$$

for specified minimum and maximum values, e.g. $a_{\min} = 0.01$ and $a_{\max} = 10$. Next, define [38]

$$\boldsymbol{\sigma}\left(\boldsymbol{z}\right) = \begin{bmatrix} b_w \\ \sigma_1\left(z_1\right) \\ \vdots \\ \sigma_{n_2}\left(z_{n_2}\right) \end{bmatrix}$$

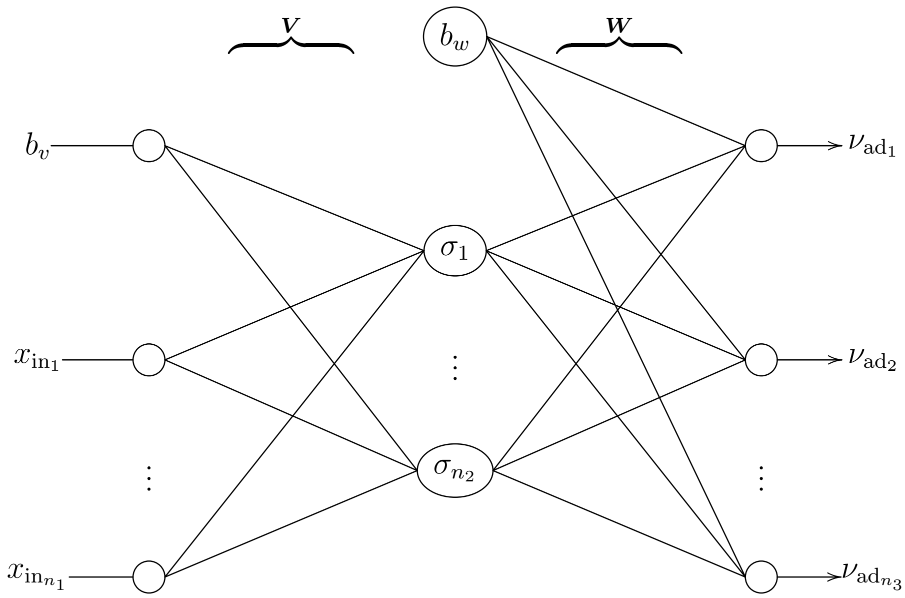


Figure 4.3: Single hidden layer neural network (reproduction of Figure 2 in [38]).

and the neural network is then calculated as

$$\boldsymbol{\nu}_{\text{ad}} = \boldsymbol{W}^T \boldsymbol{\sigma}\left(\boldsymbol{V}^T \bar{\boldsymbol{x}}\right)$$

This neural network is nonparametric in the sense that the functional form of the model error is not explicitly specified [74]. A universal approximation theorem states that fit error for this neural network can be bounded within a compact set of states and inputs for any chosen error bound [74]. As part of this, a sufficient number of hidden layer neurons are needed to perform the curve fit to the desired accuracy. For rotorcraft combined rotational and translational control, $n_2 = 5$ hidden layer neurons have been successfully used [38, 74]. The neural network can be trained on model error using [38]

$$\dot{\boldsymbol{W}} = -\left[\left(\boldsymbol{\sigma} - \acute{\boldsymbol{\sigma}} \boldsymbol{V}^T \bar{\boldsymbol{x}}\right) \boldsymbol{s}^T + \lambda \left\|\boldsymbol{e}\right\| \boldsymbol{W}\right] \boldsymbol{\Gamma}_{\boldsymbol{W}} \tag{4.1}$$

$$\dot{\boldsymbol{V}} = -\boldsymbol{\Gamma}_{\boldsymbol{V}} \left[\bar{\boldsymbol{x}} \boldsymbol{s}^T \boldsymbol{W}^T \acute{\boldsymbol{\sigma}} + \lambda \left\|\boldsymbol{e}\right\| \boldsymbol{V}\right] \tag{4.2}$$

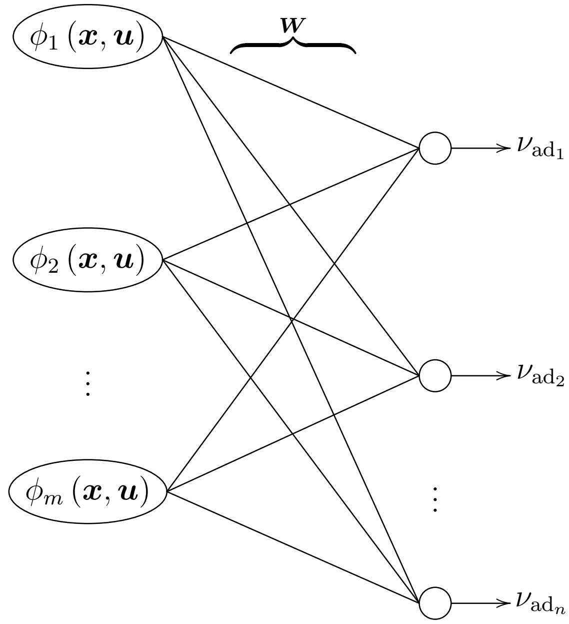


Figure 4.4: Parametric neural network (based on Figure 9.3-1 in [74]).

$\boldsymbol{\Gamma_W}$ and $\boldsymbol{\Gamma_V}$ are diagonal matrices of learning rates, matrix $\acute{\boldsymbol{\sigma}}$ is the gradient of $\boldsymbol{\sigma}$, *e-modification* scalar $\lambda > 0$ adds damping to bound adaptive parameters [58], $\boldsymbol{e}$ is the tracking error, and $\boldsymbol{s}$ is filtered tracking error. As expected, $\dot{\boldsymbol{W}}$ and $\dot{\boldsymbol{V}}$ are zero when tracking error $\boldsymbol{e}$ (and therefore $\boldsymbol{s}$) is zero. Since this learning is short-term, the controller must adapt to new or not-recently-visited portions of the state space [74]. Adaptation must occur quickly enough for this to be effective.

4.2.3 Parametric Neural Network

Figure 4.4 shows a parametric neural network, i.e. a neural network using m known basis functions $\boldsymbol{\Phi}(\boldsymbol{x},\boldsymbol{u}) = [\phi_1(\boldsymbol{x},\boldsymbol{u}), \phi_2(\boldsymbol{x},\boldsymbol{u}), \ldots, \phi_m(\boldsymbol{x},\boldsymbol{u})]^T$ with unknown parameters $\boldsymbol{W} \in \Re^{m\times n}$.

The adaptive signal is [11]

$$\boldsymbol{\nu}_{\mathrm{ad}} = \boldsymbol{W}^T \boldsymbol{\Phi}\left(\boldsymbol{x}, \boldsymbol{u}\right)$$

Outside of neural network classification this is known as a linear-in-the-parameters static parametric model (SPM) [37]. Adaptation is restricted to weighting of the specified basis functions. Compared to the SHL neural network of the previous section this could increase robustness since adaptation is restricted to selected basis functions. However, this approach decreases robustness to phenomena not handled by the specified basis functions.

The neural network can be trained on model error using [11]

$$\dot{\boldsymbol{W}} = -\boldsymbol{\Phi}\left(\boldsymbol{x}, \boldsymbol{u}\right) \boldsymbol{s}^T \boldsymbol{\Gamma}_{\boldsymbol{W}}$$

$\boldsymbol{\Gamma}_{\boldsymbol{W}}$ is a positive definite matrix of learning rates and $\boldsymbol{s}$ is filtered tracking error. An *e-modification* term can be added, resulting in

$$\dot{\boldsymbol{W}} = -\left[\boldsymbol{\Phi}\left(\boldsymbol{x}, \boldsymbol{u}\right) \boldsymbol{s}^T + \lambda \left\|\boldsymbol{e}\right\| \boldsymbol{W}\right] \boldsymbol{\Gamma}_{\boldsymbol{W}}$$

e-modification scalar $\lambda > 0$ adds damping to bound adaptive parameters [58] and $\boldsymbol{e}$ is the tracking error.

4.2.4 Pseudocontrol Hedging

This section describes pseudocontrol hedging (PCH), a method used with dynamic inversion adaptive control to address issues such as input saturation, input rate saturation, and cascaded adaptive systems [74]. PCH allows adaptation to correct for some system characteristics while not adapting to others. For example, a designer can choose not to adapt to input saturation or to responses of subsystems that have their own adaptation [74]. For saturation, PCH is used to avoid adapting to the effects of saturation while continuing adaptation even when there is saturation [74]. Here, PCH for input saturation is defined.

As shown in Figure 4.5, the hedge signal $\boldsymbol{\nu}_h = \boldsymbol{\nu} - \hat{\boldsymbol{f}}\left(\boldsymbol{x}, \boldsymbol{u}\right)$, the difference between commanded pseudocontrol $\boldsymbol{\nu}$ and predicted pseudocontrol, is used by the reference model.

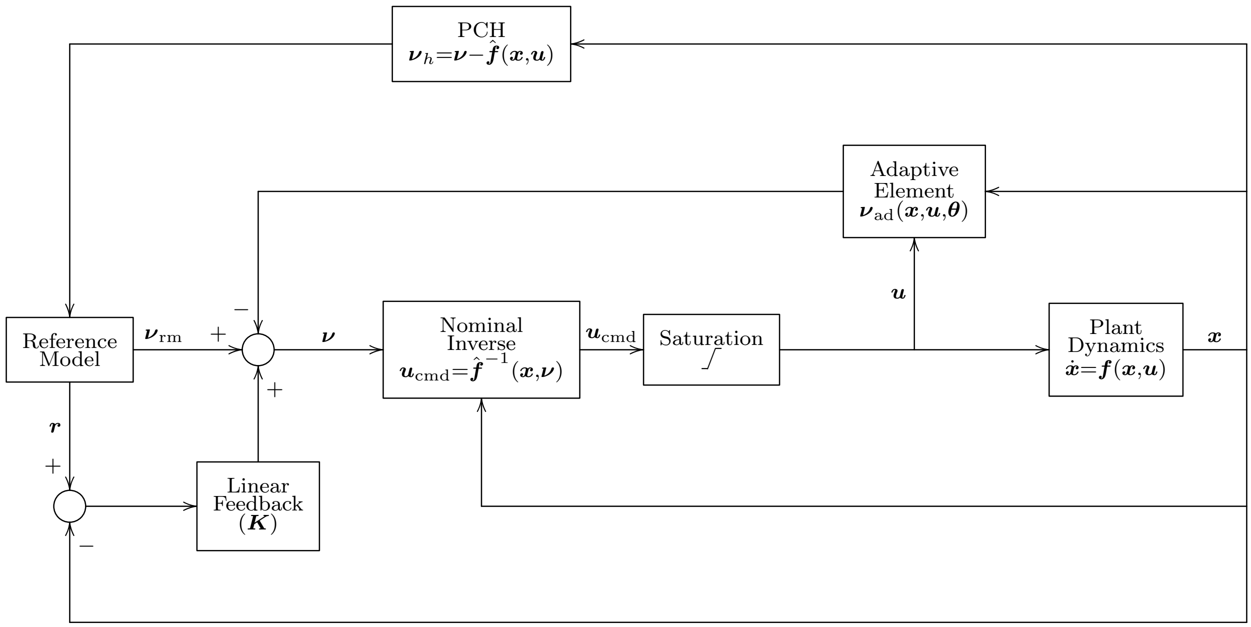


Figure 4.5: MRAC based on dynamic inversion with pseudocontrol hedging (PCH) (reproduction of Figure 9.4-1 in [74]), i.e. Figure 4.2 with the addition of PCH and Saturation blocks.

$\boldsymbol{u}$ is the saturated input to the plant. If the reference model without PCH was

$$\dot{\boldsymbol{r}} = \boldsymbol{f}_{\text{rm}}(\boldsymbol{r})$$

then the reference model with PCH is [74]

$$\dot{\boldsymbol{r}} = \boldsymbol{f}_{\text{rm}}(\boldsymbol{r}) - \boldsymbol{\nu}_h \tag{4.3}$$

$$\boldsymbol{\nu}_{\text{rm}} = \boldsymbol{f}_{\text{rm}}(\boldsymbol{r}) \tag{4.4}$$

Note that the hedge signal $\boldsymbol{\nu}_h$ does not affect reference model outputs $\boldsymbol{r}$ and $\boldsymbol{\nu}_{\text{rm}}$ directly, but rather through $\dot{\boldsymbol{r}}$.

4.2.5 Actuator Modeling

In this section, the saturation example of the PCH section is generalized to controller output filtering and actuator modeling, measurement, and estimation. Three actuator position types are used. First, $\boldsymbol{u}_{\text{cmd}}$ is the vector of actuator commands from the nominal inverse. Second, $\boldsymbol{u}$ is the vector of controller outputs actually sent to the actuators, e.g. after applying position and rate limits to $\boldsymbol{u}_{\text{cmd}}$. Finally, $\hat{\boldsymbol{u}}$ is an estimate of current actuator positions. $\hat{\boldsymbol{u}}$ is used for PCH, and by the adaptive element for both current input and learning. Figure 4.6 shows a generic controller Output Filter replacing the Saturation block of Figure 4.5 and the addition of $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{u}}$ estimates. The Actuators Estimator can be based on outputs, models, or measurements; e.g. (1) controller output $\boldsymbol{u}$, (2) an actuator model updated by $\boldsymbol{u}$, or (3) measurements of actual actuator positions.

A key point for actuators estimate $\hat{\boldsymbol{u}}$ is that model errors that affect the dynamics of the actuators—such as time constants and rate limits—cannot be canceled by the neural network [38]. It is therefore important that these time constants and rate limits are estimated accurately if actuator measurements are unavailable. One way to do this is to artificially limit actuator rates (in the controller output filter) to conservative values and then assume that the actuators accurately track these conservative commands (i.e. $\hat{\boldsymbol{u}} = \boldsymbol{u}$) [38]. Position limits $\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}$ and rate limits $\dot{\boldsymbol{u}}_{\min}, \dot{\boldsymbol{u}}_{\max}$ can be implemented as [38]

$$\boldsymbol{u}\left[k+1\right] = \mathbf{sat}\left[\boldsymbol{u}\left[k\right] + \mathbf{sat}\left(\mathbf{sat}\left(\boldsymbol{u}_{\text{cmd}}, \boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}\right) - \boldsymbol{u}\left[k\right], \Delta T\dot{\boldsymbol{u}}_{\min}, \Delta T\dot{\boldsymbol{u}}_{\max}\right), \boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}\right]$$

where $\boldsymbol{u}$ controller output $\boldsymbol{u}\left[k+1\right]$ is calculated from $\boldsymbol{u}_{\text{cmd}}$ using the limits and the previous output $\boldsymbol{u}\left[k\right]$ from ΔT time earlier.

Conservative rate limits with $\hat{\boldsymbol{u}} = \boldsymbol{u}$ as just described can sacrifice performance or even cause instability (due to restrictive limits or $\hat{\boldsymbol{u}}, \boldsymbol{u}$ mismatch such as time delay). Here, dynamics and time delay modeling are examined for accurate $\hat{\boldsymbol{u}}$ estimation in the presence of time delay or absence of conservative $\boldsymbol{u}$ rate limits.

There are many potential sources of time delay in the control loop, e.g. computation, communication, actuator dynamics, sensors, and filters. The various time delays can be

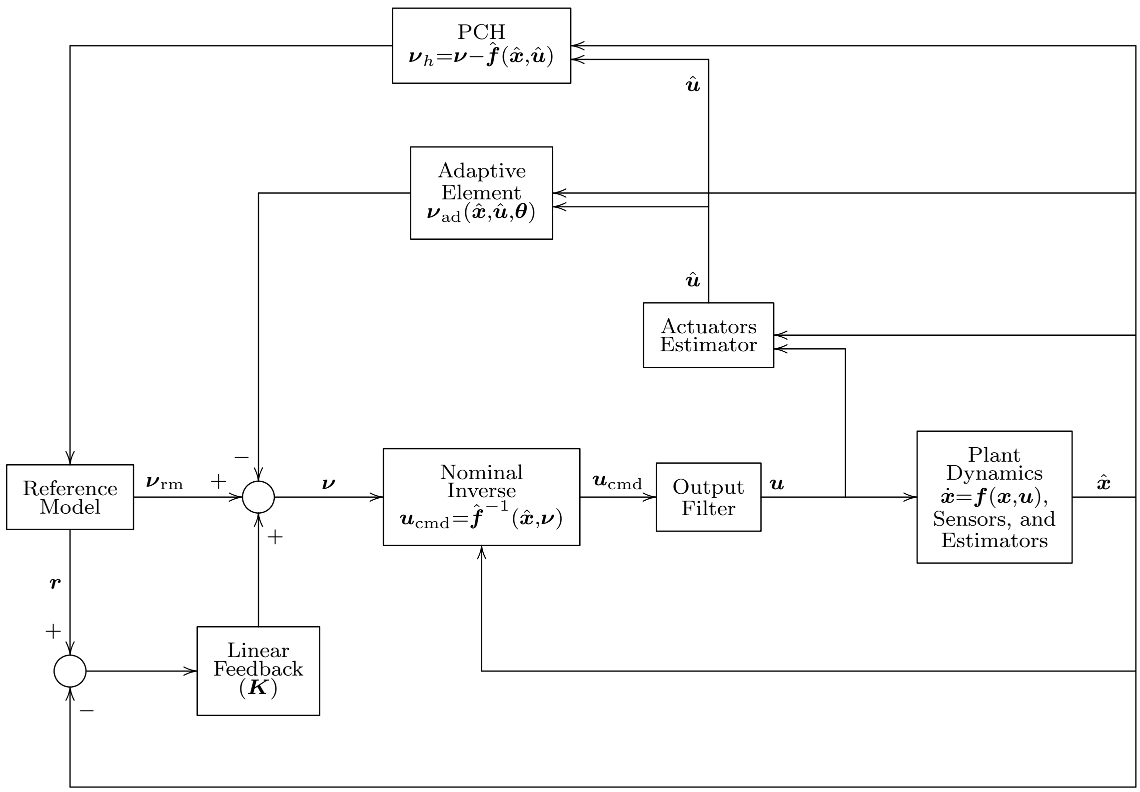


Figure 4.6: MRAC based on dynamic inversion with PCH, controller output filtering, actuator estimation, and state estimation; i.e. Figure 4.5 with modification of the saturation and plant dynamics blocks, addition of Actuators Estimator block, and explicit control feedback to the PCH block. The Plant Dynamics block is expanded to include sensors and estimators with state estimate output $\hat{\boldsymbol{x}}$.

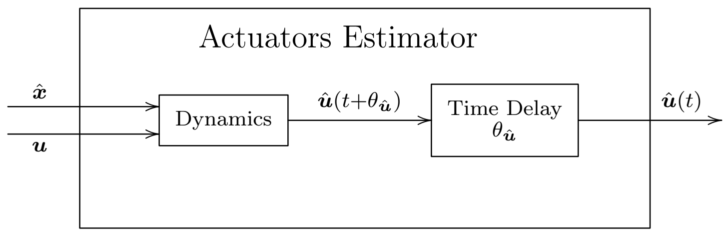


Figure 4.7: Actuators estimator with dynamics and time delay.

combined into one value with the goal of matching the time of the $\hat{\boldsymbol{u}}$ estimate to that of the $\hat{\boldsymbol{x}}$ estimate [74]. Note that not all the time delay is necessarily from the actual actuators, but is still appropriately modeled in the Actuators Estimator. Similarly, it is appropriate to include dynamics not only from the actuators themselves but also other dynamics that appear to delay or limit actuation. One example of such higher-order dynamics is rotor flapping in response to cyclic pitch control, which typically introduces a lag of 70–100 ms for full-scale rotorcraft [77] (i.e. the effective frequency of rotor response is comparable to the spin frequency). The total time delay and dynamics can be measured from controller output $\boldsymbol{u}$ and state estimate $\hat{\boldsymbol{x}}$ response, e.g. the rotation rate $\hat{\boldsymbol{\omega}}$ response to a controller output $\boldsymbol{u}$ step [74].

Figure 4.7 shows an implementation of the Actuators Estimator block of Figure 4.6 with dynamics and delay. One option is to model all dynamics, including pure time delays, in the Dynamics block, with time delay $\theta_{\hat{\boldsymbol{u}}}$ set to zero. Alternatively, all dynamics could be approximated with the time delay $\theta_{\hat{\boldsymbol{u}}}$, leaving the Dynamics block unused. Actuator dynamics can be modeled with a low-order system, e.g. a second-order model (with position and rate limits if not handled by the Output Filter block) [77, 18]. In a digital system with time delay much larger than step time ($\theta_{\hat{\boldsymbol{u}}} \gg \Delta T$), the time delay is easily implemented as a history buffer.

4.3 Adaptive Rotational Control

In this section, adaptive rotational control is developed for rotorcraft and tested on Flexrotor. The controller structure is as described in the previous section, i.e. model reference adaptive control based on dynamic inversion with neural network adaptation and pseudocontrol hedging (Figure 4.6). The adaptive rotation controller is shown in Figure 4.8. Compared to Figure 4.6, (1) the Reference Model Equations (4.3) and (4.4) are added (including dependence on attitude $\boldsymbol{\Delta\phi}$ and rotation rate $\boldsymbol{\omega}$ commands $\boldsymbol{c}^T = \left[\begin{array}{cccc} \boldsymbol{\Delta\phi}_{\text{ref}}^T & \boldsymbol{\Delta\phi}_{\text{des}}^T & \boldsymbol{\omega}_{\text{ref}}^T & \boldsymbol{\omega}_{\text{des}}^T \end{array}\right]$ as in Figure 3.1), (2) the adaptive element is specifically named as a Neural Network, and (3) the plant dynamics block is specifically named as aircraft dynamics.

The reference model on each axis is a second-order system with a specified natural frequency ω_n and damping ratio ζ [74]:

$$\ddot{r} = -K_D\dot{r} - K_P r \tag{4.5}$$

where $K_D = 2\zeta\omega_n$ and $K_P = \omega_n^2$. Rewriting as

$$\ddot{r} = K_D\left(-\frac{K_P}{K_D}r - \dot{r}\right)$$

$v_{\text{Unlimited}} = -\left(K_P/K_D\right) r$ can be seen as the angular velocity to eliminate attitude error. When $v_{\text{Unlimited}}$ exceeds limit v_{Limited}, the reference dynamics are updated with [74]

$$\ddot{r} = K_D\left(v_{\text{Limited}} - \dot{r}\right)$$

in order to limit angular velocity for large attitude errors. With non-zero commands and switching notation to explicit attitude $\boldsymbol{\Delta\phi}$ and rotation rate $\boldsymbol{\omega}$, the reference model dynamics can be written as [38]

$$\begin{aligned}
\dot{\boldsymbol{\omega}}_r &= \boldsymbol{f}_{\text{rm}}\left(\boldsymbol{\Delta\phi}_r, \boldsymbol{\omega}_r, \boldsymbol{\Delta\phi}_c, \boldsymbol{\omega}_c\right) - \boldsymbol{\nu}_h \\
\boldsymbol{\nu}_{\text{rm}} &= \boldsymbol{f}_{\text{rm}}\left(\boldsymbol{\Delta\phi}_r, \boldsymbol{\omega}_r, \boldsymbol{\Delta\phi}_c, \boldsymbol{\omega}_c\right) = \boldsymbol{K}_D\left[\boldsymbol{\omega}_c - \boldsymbol{\omega}_r + \textbf{sat}\left(\boldsymbol{K}_D^{-1}\boldsymbol{K}_P\left(\boldsymbol{\Delta\phi}_c - \boldsymbol{\Delta\phi}_r\right), \boldsymbol{\omega}_{\text{lim}}\right)\right]
\end{aligned}$$

where c subscripts are commands, r subscripts are reference model, $\boldsymbol{K}_P \in \Re^{3\times3}$ and $\boldsymbol{K}_D \in \Re^{3\times3}$ diagonal matrices, and **sat** is the saturation function with limits $\boldsymbol{\omega}_{\text{lim}}$. Note that

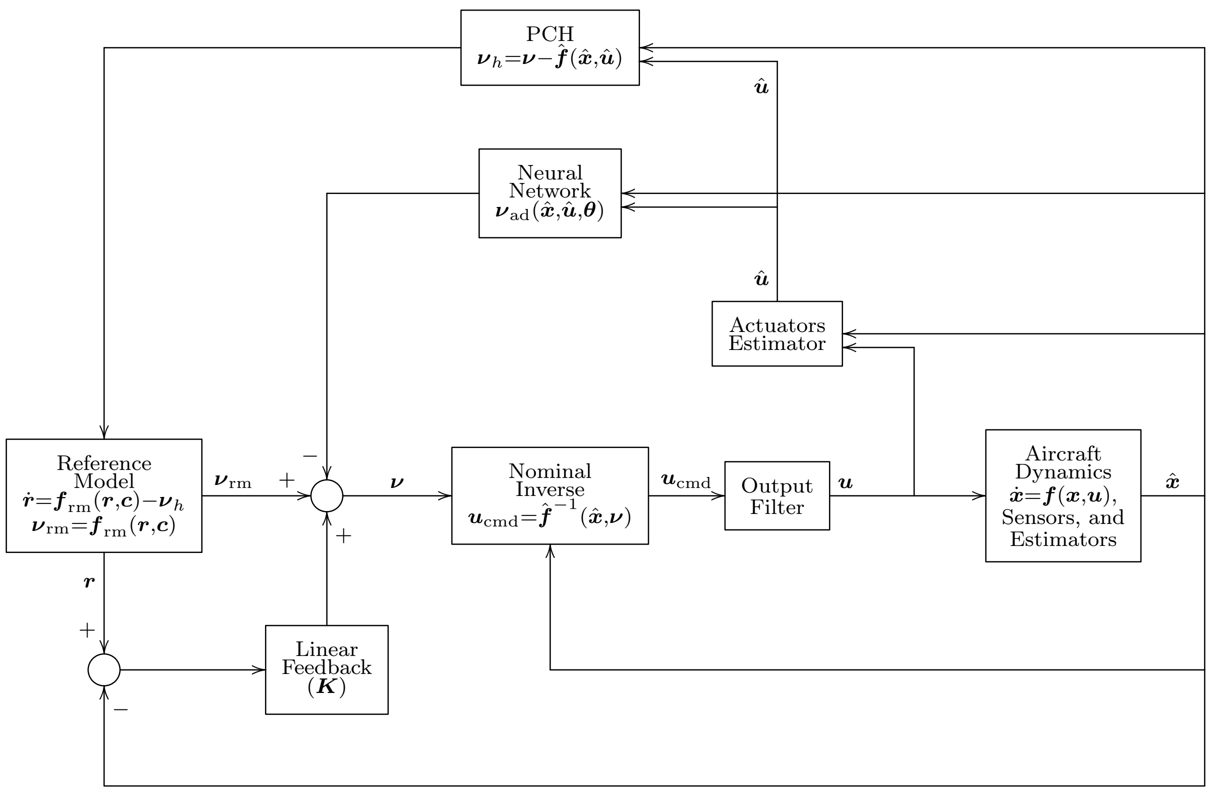


Figure 4.8: Adaptive rotation controller, i.e. the structure of Figure 4.6 for aircraft rotation within the architecture of Figure 3.1.

attitude $\boldsymbol{\Delta\phi} \in \Re^3$ can be replaced with attitude quaternions, using a function that returns an error angle vector with three components given two quaternions [38].

The reference model tracking error vector is defined as

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{\Delta\phi}_r - \hat{\boldsymbol{\Delta\phi}} \\ \boldsymbol{\omega}_r - \hat{\boldsymbol{\omega}} \end{bmatrix}$$

where aircraft state estimates are shown with a hat ˆ.

Linear feedback is then defined as [74]

$$\boldsymbol{\nu}_{\text{fb}} = \boldsymbol{K}\boldsymbol{e} = \begin{bmatrix} \boldsymbol{K}_P & \boldsymbol{K}_D \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta\phi}_r - \hat{\boldsymbol{\Delta\phi}} \\ \boldsymbol{\omega}_r - \hat{\boldsymbol{\omega}} \end{bmatrix}$$

with $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ with elements for specified natural frequencies and damping ratios as in the reference model, i.e. $K_D = 2\zeta\omega_n$ and $K_P = \omega_n^2$. This is referred to as proportional-derivative (PD) feedback [38], in this case with feedback proportional to attitude error and to approximately the derivative of attitude error for small angles, i.e. $\mathrm{d}\boldsymbol{\Delta\phi}/\mathrm{d}t = \boldsymbol{\omega}$ for small $\boldsymbol{\Delta\phi}$. Note that there is no integral feedback. In this system integral action is provided by the neural network adaptation.

Gains, i.e. natural frequencies and damping ratios, must be selected for the reference model and for feedback on each axis. B. L. Stevens, F. L. Lewis, and E. N. Johnson [74] chose identical gains for the reference model and feedback, noting that in some applications the feedback gains could be chosen somewhat faster so that reference model dynamics dominate. For explicit-model following (and dynamic inversion) on rotorcraft, the reference model gains are typically selected for desired response to commands, while the feedback gains are set for stability and performance requirements, e.g. reference model tracking, gust rejection, and robust stability and performance margins [77].

Next, an approximate model $\hat{\boldsymbol{f}}$ is needed for the nominal inverse and PCH. Starting with the linear model $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$, there are design choices for each item, i.e. $\boldsymbol{A}, \boldsymbol{x}, \boldsymbol{B}, \boldsymbol{u}$. The model used here is [38] (Equation (3.2))

$$\hat{\boldsymbol{f}}\left(\boldsymbol{x}, \boldsymbol{u}\right) = \boldsymbol{\nu} = \dot{\boldsymbol{\omega}} = \hat{\boldsymbol{A}}_1\boldsymbol{\omega} + \hat{\boldsymbol{A}}_2\boldsymbol{v}_B + \hat{\boldsymbol{B}}\left(\boldsymbol{u} - \boldsymbol{u}_{\text{trim}}\right) \tag{4.6}$$

where pseudocontrol $\boldsymbol{\nu} \in \Re^3$ is the desired rotational acceleration vector, $\boldsymbol{v}_B \in \Re^3$ is the body velocity vector, $\boldsymbol{u} \in \Re^3$ is the rotational control vector, and $\boldsymbol{u}_{\text{trim}} \in \Re^3$ is control trim. With $\hat{\boldsymbol{A}}_2$ zero, this is the same model used earlier for trim point estimation, i.e. Equation (3.5). As described in Section 4.2.1, errors in the model structure (e.g. missing state variables and nonlinearities) and parameters (e.g. matrix elements) should be corrected by the adaptive element (i.e. the neural network). Equation (4.6) is the model used in B. L. Stevens, F. L. Lewis, and E. N. Johnson [74], except there rate damping $\hat{\boldsymbol{A}}_1$ and control effectiveness $\hat{\boldsymbol{B}}$ are specified as diagonal and $\hat{\boldsymbol{A}}_2$ is zero. Here general matrices are retained. For the Flexrotor example—which has significant y-z coupling—at least the following block diagonal elements of $\hat{\boldsymbol{A}}_1$ should be non-zero:

$$\hat{\boldsymbol{A}}_1 = \begin{bmatrix} a_{xx} & 0 & 0 \\ 0 & a_{yy} & a_{yz} \\ 0 & a_{zy} & a_{zz} \end{bmatrix}$$

Returning to Equation (4.6), $\hat{\boldsymbol{B}}$ is selected such that it is invertible, then

$$\hat{\boldsymbol{f}}^{-1}(\boldsymbol{x}, \boldsymbol{\nu}) = \boldsymbol{u} = \hat{\boldsymbol{B}}^{-1}\left(\boldsymbol{\nu} - \hat{\boldsymbol{A}}_1\boldsymbol{\omega} - \hat{\boldsymbol{A}}_2\boldsymbol{v}_B\right) + \boldsymbol{u}_{\text{trim}}$$

For the single hidden layer neural network (Section 4.2.2), parameters are defined as in E. N. Johnson and S. K. Kannan [38], i.e.

$$\boldsymbol{x}_{\text{in}} = \begin{bmatrix} \boldsymbol{v}_B \\ \boldsymbol{\omega} \\ \hat{\boldsymbol{\nu}} \end{bmatrix}$$

with body axis velocities $\boldsymbol{v}_B$, body axis rotation rates $\boldsymbol{\omega}$, and estimated pseudocontrols $\hat{\boldsymbol{\nu}} = \hat{\boldsymbol{f}}(\hat{\boldsymbol{x}}, \boldsymbol{u})$. There are three neural network outputs (i.e. $n_3 = 3$) $\boldsymbol{\nu}_{\text{ad}}$, the model error in the rotational acceleration vector. The dependence of rotorcraft dynamics on $\boldsymbol{v}_B$ was discussed in the Modeling chapter and it is therefore important to include as a neural network input. The filtered tracking error, used in $\dot{\boldsymbol{W}}$ and $\dot{\boldsymbol{V}}$ Equations (4.1) and (4.2), is [38]

$$
\begin{aligned}
\boldsymbol{s} &= \left(\boldsymbol{e}^T \boldsymbol{P} \boldsymbol{B}_{\text{ted}}\right)^T \\
&= \left(\begin{bmatrix} \boldsymbol{\Delta\phi}_r - \hat{\boldsymbol{\Delta\phi}} \\ \boldsymbol{\omega}_r - \hat{\boldsymbol{\omega}} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{K}_P^2 + \frac{1}{2}\boldsymbol{K}_P\boldsymbol{K}_D^2 & \frac{1}{2}\boldsymbol{K}_P\boldsymbol{K}_D \\ \frac{1}{2}\boldsymbol{K}_P\boldsymbol{K}_D & \boldsymbol{K}_P \end{bmatrix} \frac{1}{\frac{1}{4}n_2 + b_w^2} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \right)^T \\
&= \frac{1}{\frac{1}{4}n_2 + b_w^2} \begin{bmatrix} \frac{1}{2}\boldsymbol{K}_P\boldsymbol{K}_D & \boldsymbol{K}_P \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta\phi}_r - \hat{\boldsymbol{\Delta\phi}} \\ \boldsymbol{\omega}_r - \hat{\boldsymbol{\omega}} \end{bmatrix}
\end{aligned}
$$

where $\boldsymbol{P}$ is from a Lyapunov function and $\boldsymbol{B}_{\text{ted}}$ is from the tracking error dynamics.

For the linear-in-the-parameters static parametric model (Section 4.2.3), the basis functions are a bias and estimated pseudocontrols, i.e.

$$
\boldsymbol{\Phi} = \begin{bmatrix} 1 \\ \hat{\boldsymbol{\nu}} \end{bmatrix}
$$

4.3.1 Flexrotor Implementation

The MRAC controller described in this section (Figure 4.8) was applied to Flexrotor and tested in simulation. A first-order low-pass filter on the adaptive element output $\boldsymbol{\nu}_{\text{ad}}$ was also used for some testing (not shown in Figure 4.8). The actuator models are first-order plus time delay.

Except where specified otherwise, controller parameters for Flexrotor simulation testing are as in Table 4.1 and this paragraph. Reference model angular velocity limits for attitude errors are all $2\,\text{rad/s}$. Neural network biases are $b_v = b_w = 0.5$. The number of hidden layer neurons is $n_2 = 5$. Neural network activation potentials are $a_{\min} = 0.01$ and $a_{\max} = 10$. Output filter position and rate limits are effectively disabled for all axes: $u_{\min} = -100\,\text{rad/s}^2$, $u_{\max} = 100\,\text{rad/s}^2$, $\dot{u}_{\min} = -1000\,\text{rad/s}^3$, $\dot{u}_{\max} = 1000\,\text{rad/s}^3$. The actuator model time constants $\tau_{\hat{\boldsymbol{u}}}$ and a time delays $\theta_{\hat{\boldsymbol{u}}}$ were determined by a least-squares fit of a first-order plus time delay model for each axis to the software simulation $\dot{\boldsymbol{\omega}}$ (i.e. finite-differenced rotational acceleration) response to a step from trim rotational control $\boldsymbol{u}$ command.

Table 4.1: Adaptive rotation controller parameters for Flexrotor testing.

Category	Parameter	Value
Reference model and feedback PD gains	natural frequencies $\omega_{n_y}, \omega_{n_z}$	$2.8\,\mathrm{rad/s}$
	damping ratios ζ_y, ζ_z	1.0
Approximate model $\hat{\boldsymbol{f}}$	rotation rate dynamics $\hat{\boldsymbol{A}}_1$	$\mathbf{0}$
	velocity dynamics $\hat{\boldsymbol{A}}_2$	$\mathbf{0}$
	control effectiveness $\hat{\boldsymbol{B}}$	identity matrix $\boldsymbol{I}$
	control trim $\boldsymbol{u}_{\mathrm{trim}}$	hover values
Adaptation	inner layer learning rate $\boldsymbol{\Gamma_V}$	5.0
	outer layer learning rate $\boldsymbol{\Gamma_W}$	1.0
	e-modification λ	0.1
Adaptive element output filter	time constant $\tau_{\boldsymbol{\nu}_{\mathrm{ad}}}$	$0\,\mathrm{s}$
Control model $\hat{\boldsymbol{u}}$	time constant $\tau_{\hat{\boldsymbol{u}}_{y,z}}$	$0.05\,\mathrm{s}$
	time delay $\theta_{\hat{\boldsymbol{u}}_{y,z}}$	$0.14\,\mathrm{s}$

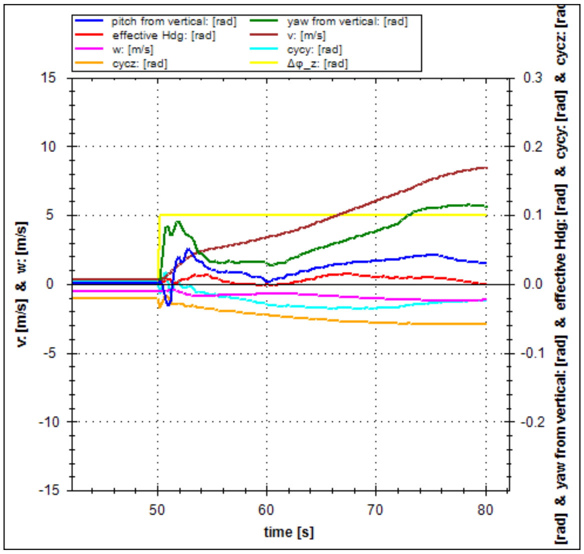
Note that for the approximate model $\hat{\boldsymbol{f}}$ parameters of Table 4.1 (i.e. $\hat{\boldsymbol{A}}_1 = \mathbf{0}$, $\hat{\boldsymbol{A}}_2 = \mathbf{0}$, $\hat{\boldsymbol{B}} = \boldsymbol{I}$), the model reduces to

$$\hat{\boldsymbol{f}}\left(\boldsymbol{u}\right) = \boldsymbol{\nu} = \boldsymbol{u} - \boldsymbol{u}_{\mathrm{trim}}$$

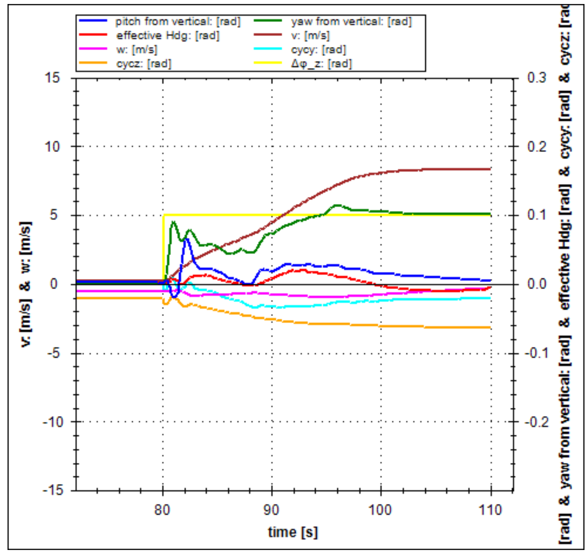
$\hat{\boldsymbol{B}} = \boldsymbol{I}$ since outputs from this controller are desired rotational accelerations that are sent to a rotational control mixing module (Section 3.1). In particular, $\hat{\boldsymbol{A}}_2 = \mathbf{0}$ seems appropriate since the velocity dynamics are strongly nonlinear.

4.3.2 Simulation Results

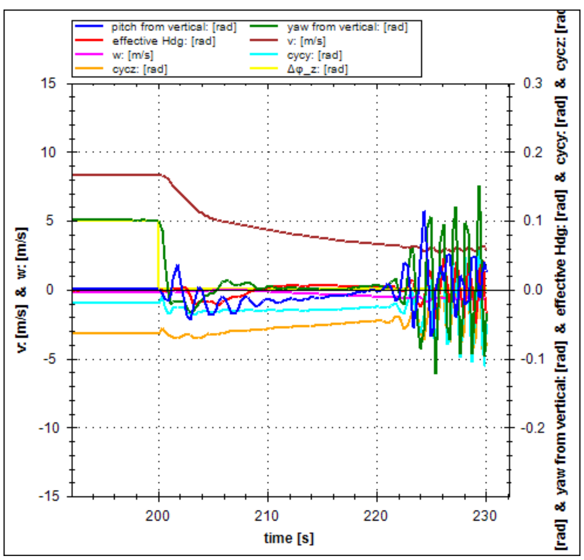
For all the tests in this section, roll was controlled by the standard controller, i.e. the MRAC controller was only used for tilt (pitch and yaw) control. For a yaw ($\Delta\phi_z$) command step from fuselage vertical (i.e. $\Delta\phi_z = 0$) to $\Delta\phi_z = 0.1\,\mathrm{rad}$ in zero wind, regulation is better



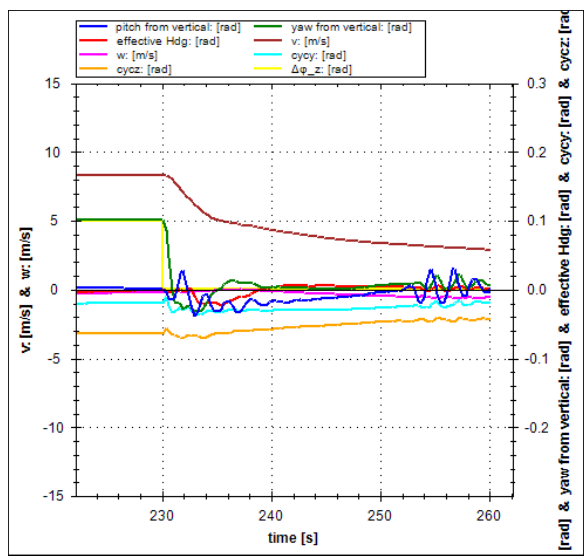
(a) Standard controller.



(b) First step with MRAC controller.



(c) After second step with MRAC controller.



(d) After second step with MRAC controller and actuator model time delay $\theta_{\hat{\boldsymbol{u}}_{y,z}} = 0.14\,\mathrm{s}$.

Figure 4.9: Tilt attitude command steps in simulation with standard and MRAC controllers. Approximate model $\hat{\boldsymbol{f}}$ rotation rate dynamics matrix $\hat{\boldsymbol{A}}_1$ is set to hover values, adaptation learning rates are $\boldsymbol{\Gamma_V} = 0.5$ and $\boldsymbol{\Gamma_W} = 0.1$, and actuator model parameters are time constant $\tau_{\hat{\boldsymbol{u}}_{y,z}} = 0$ and time delay $\theta_{\hat{\boldsymbol{u}}_{y,z}} = 0$. Simulations are of Flexrotor *Amphinome* (1.85 m diameter rotor) at 17 kg and with the Pitt-Peters model.

with the MRAC controller described in this section than with the standard controller. For example, the step with the standard controller in Figure 4.9a has a 20% settling time of 20.2 s compared to 10.4 s for the MRAC controller in Figure 4.9b.

Stability was substantially improved by the addition of first-order plus time delay actuator models (the Actuators Estimator block of Figure 4.8). For example, without the actuator model (i.e. $\hat{\boldsymbol{u}} = \boldsymbol{u}$), learning rates of $\boldsymbol{\Gamma_V} = 1.0$ and $\boldsymbol{\Gamma_W} = 0.2$ (i.e. one fifth of the values in Table 4.1) resulted in a loss of control and aircraft crash after the controller was enabled in hover. As another example, Figure 4.9c shows an oscillation after two consecutive 0.0 to 0.1 to 0.0 rad yaw command steps with $\hat{\boldsymbol{u}} = \boldsymbol{u}$. The oscillation is significantly reduced after repeating the sequence with an actuator model time delay of $\theta_{\hat{\boldsymbol{u}}_{y,z}} = 0.14\,\mathrm{s}$ (Figure 4.9d).

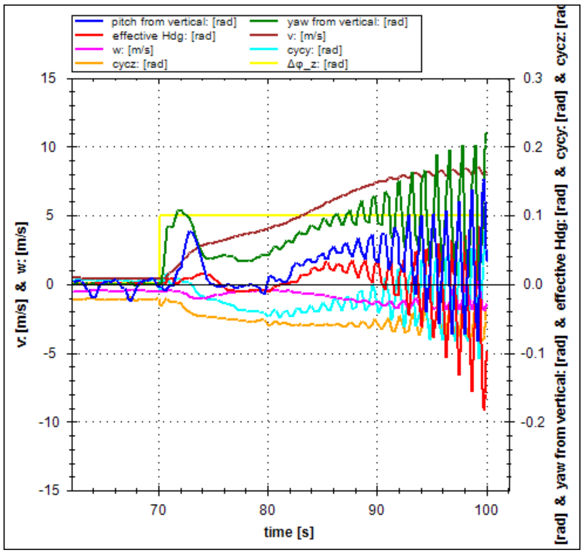
Stability was further improved in tests where the first-order low-pass filter on the adaptive element output $\boldsymbol{\nu}_{\mathrm{ad}}$ was used. Most tests were conducted with this filter disabled, as indicated in Table 4.1 (i.e. filter time constant of zero).

Performance did not substantially improve in tests when approximate model dynamics matrices $\hat{\boldsymbol{A}}_1$ or $\hat{\boldsymbol{A}}_2$ were set to hover model values. Therefore, $\hat{\boldsymbol{A}}_1$ and $\hat{\boldsymbol{A}}_2$ were set to zero for most tests (as indicated in Table 4.1), resulting in a simplified approximate model $\hat{\boldsymbol{f}}$. This also avoids feeding state oscillations into the reference model through PCH.

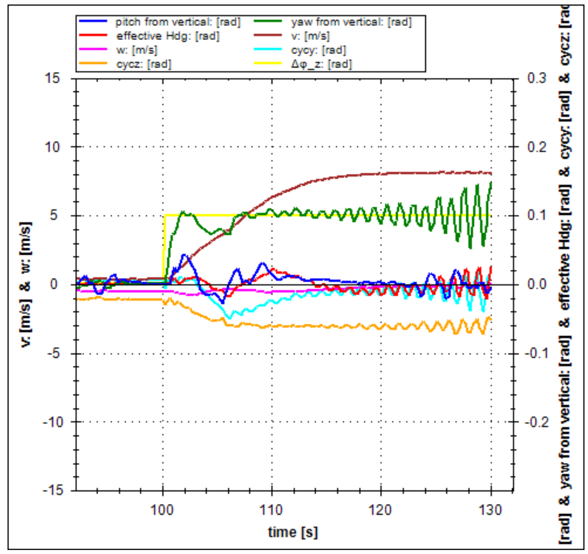
The remaining tests, i.e. Figures 4.10 and 4.11, were conducted with $\boldsymbol{u}$ instead of $\hat{\boldsymbol{u}}$ as the current input to the adaptive element ($\hat{\boldsymbol{u}}$ was still used for PCH and adaptive element learning).

Figure 4.10a shows a yaw command step with the ad hoc model (Section 2.2.1) instead of Pitt-Peters. There is a large-amplitude oscillation for simulation in these conditions and a 20% settling time is 13.8 s if the oscillation is excluded. With the MRAC controller (Figure 4.10b), oscillation amplitude is substantially reduced and 20% settling time is 6.2 s (excluding the oscillation). As shown in Figure 4.10c, a second consecutive yaw command step (i.e. the full sequence is 0.0 to 0.1 to 0.0 to 0.1 rad) results in a further reduction in oscillation amplitude (and 20% settling time is again 6.2 s).

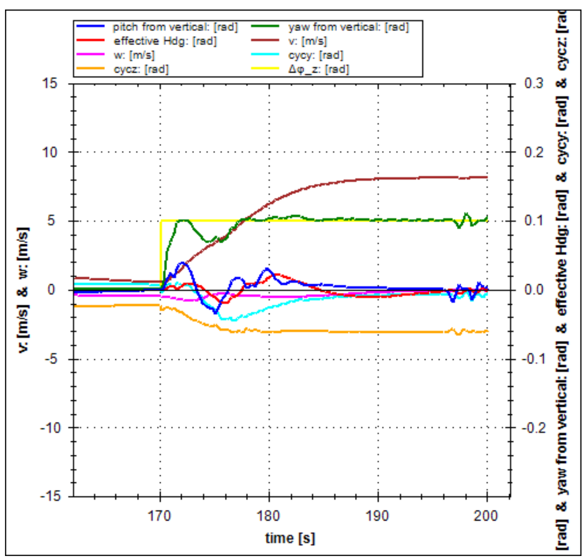
Figure 4.11 shows yaw command steps with the MRAC controller using a static para-



(a) Standard controller.



(b) First step with MRAC controller.



(c) Second step with MRAC controller.

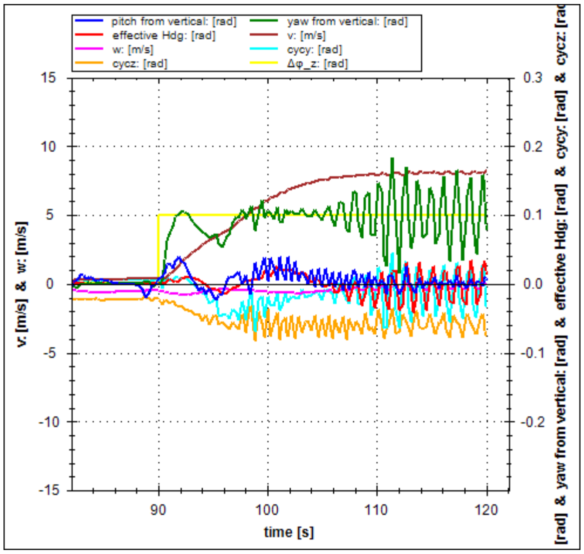
Figure 4.10: Tilt attitude command steps in simulation with standard and MRAC controllers. Simulations are of Flexrotor *Amphinome* (1.85 m diameter rotor) at 17 kg.

metric model (SPM) adaptive element instead of a single hidden layer neural network. The oscillation in cyclic control and attitude during and after the first step (Figure 4.11a) is reduced on the second (Figure 4.11b). However, a reduction in simulator cyclic effectiveness from the nominal 1.0 to 0.8 followed by a third step results in a loss of control (Figure 4.11c).

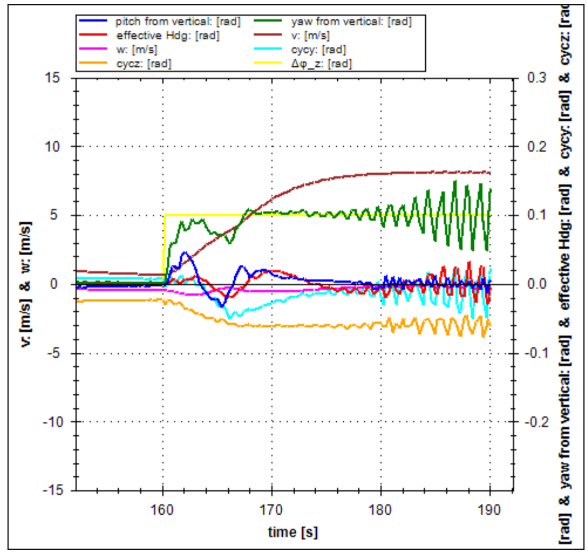
4.4 Conclusions

The application of the MRAC attitude controller to Flexrotor resulted in improved tilt regulation performance relative to the standard controller (e.g. Figure 4.10). However, the higher learning rates that quickly correct for large modeling errors also risk loss of stability (e.g. Figure 4.11). The addition of an actuator model was key for improving stability by modeling dynamics (e.g. time delays) that cannot be canceled by the adaptive element.

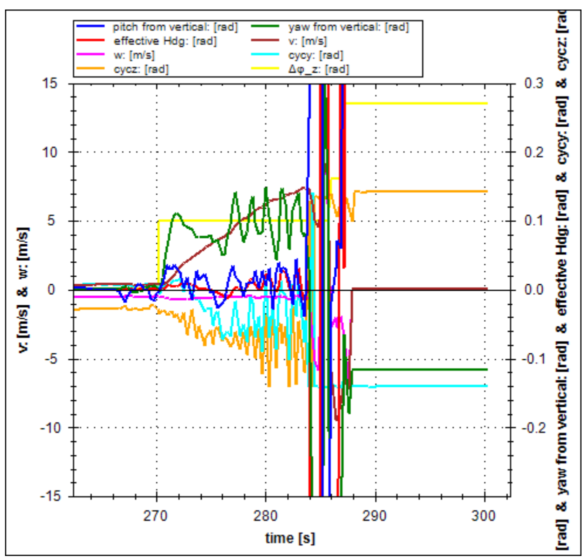
Adaptive element changes could be made to avoid destabilizing behaviors such as the learning of fast dynamics. Options include adaptive element input or output filtering, changes in adaptive element inputs, and different adaptation approaches (e.g. the use of radial basis functions). Nevertheless, for this research project the INDI approach of the next chapter was pursued due to advantages in simplicity, performance, and apparent robustness.



(a) First step with MRAC controller using SPM.



(b) Second step with MRAC controller using SPM.



(c) Third step with MRAC controller using SPM, with simulator cyclic effectiveness reduced to 0.8 from the nominal 1.0. The result is a loss of control and aircraft crash.

Figure 4.11: Tilt attitude command steps in simulation with MRAC controller using a static parametric model (SPM) adaptive element instead of a single hidden layer neural network. Simulations are of Flexrotor *Amphinome* (1.85 m diameter rotor) at 17 kg.

Chapter 5

INCREMENTAL CONTROL

Incremental nonlinear dynamic inversion (INDI) addresses the problems of interest in this dissertation. For nonlinear dependence of aircraft loads on airspeed, incremental control corrects steady-state errors without the negative effects of integral action [76]. For uncertainties, a sensor-based acceleration estimate is used to directly respond to observed dynamics [72]. For disturbances, acceleration control directly responds to disturbances as soon as they are measured. Finally, the control law is robust to nonlinear and uncertain control effectiveness since control increments correct for acceleration errors [61].

Concerns with INDI for rotorcraft control include vulnerability to vibration and other sources of sensor noise [61]. A second concern is actuator model errors when actuator measurements are not used. Finally, INDI theory relies on a time scale separation that might not exist with long filter time constants [72, 61]. These concerns will be addressed in this chapter for the present application.

Compared to the adaptive dynamic inversion controller of the Adaptive Control chapter, the INDI controllers in this chapter are simpler and avoid the practical and certification concerns of adaptation, e.g. proof that learning will always be correct or recoverable [69]. For good performance, an adaptive dynamic inversion controller must learn the model well whereas an INDI controller avoids most of the need for a model. This difference also indicates better disturbance rejection with the INDI controller, i.e. acceleration feedback is faster than learning.

5.1 Rotational INDI Control

The following angular acceleration control law was developed primarily from those used with quadrotors in Smeur et al. (2016) [72] and Tal and Karaman [76]. The objective was to formulate a simple INDI control law that could be evaluated without additions that might be unnecessary for certain applications, e.g. online control effectiveness estimation.

The INDI control law for angular acceleration is

$$\boldsymbol{\mu}_c = \boldsymbol{\mu}_f + \boldsymbol{G}^{-1}\left(\dot{\boldsymbol{\omega}}_c - \dot{\boldsymbol{\omega}}_f\right) \tag{5.1}$$

$\boldsymbol{\mu}_c$ is the commanded rotational control vector. $\boldsymbol{\mu}_f$ is a filtered estimate of the rotational control vector (e.g. actuator positions), $\boldsymbol{G}^{-1}$ is the inverse of the rotational control effectiveness matrix, $\dot{\boldsymbol{\omega}}_c$ is the commanded angular acceleration vector, and $\dot{\boldsymbol{\omega}}_f$ is a filtered estimate (e.g. measurement) of the angular acceleration vector. The estimators and filters should be such that $\boldsymbol{\mu}_f$ and $\dot{\boldsymbol{\omega}}_f$ are at the same point in time [72].

Equation (5.1) is close to those in the literature. Compared to Smeur et al. (2016) [72], terms dependent on input or actuator rate have been removed. Compared to Tal and Karaman [76], a general control effectiveness inverse matrix is used instead of a moment of inertia matrix.

Note that Equation (5.1) is an incremental control law because current control $\boldsymbol{\mu}_c$ is calculated as an increment $\boldsymbol{G}^{-1}\left(\dot{\boldsymbol{\omega}}_c - \dot{\boldsymbol{\omega}}_f\right)$ to a previous control $\boldsymbol{\mu}_f$. Also, despite the derivation using nonlinear dynamic inversion theory (e.g. as in [72]), this particular control law (Equation (5.1)) is linear.

5.1.1 Flexrotor Implementation

For the implementation on Flexrotor, $\boldsymbol{\mu}_c$ is an angular acceleration vector that is sent to a rotational control mixing module (Section 3.1). The nominal case is $\boldsymbol{G} = \boldsymbol{G}^{-1} = \boldsymbol{I}$ with identity matrix $\boldsymbol{I}$. $\boldsymbol{G}$ is allowed to be a diagonal matrix with positive elements (i.e. $\boldsymbol{G} = \mathbf{diag}\left(g_x, g_y, g_z\right)$), which allows scaling relative to the nominal control mixing (and $\boldsymbol{G}$ is always invertible).

$\dot{\boldsymbol{\omega}}_c$ is the output of a proportional-derivative (PD) controller with linear feedback on attitude and rotation rate errors. The PD gains are set for a specified natural frequency ω_n and damping ratio ζ on each axis. Note that there is no need for integral action (e.g. a PID controller) with this incremental control law for angular acceleration.

$\dot{\boldsymbol{\omega}}_f$ is calculated as a finite difference approximation of the derivative of filtered rotation rate from an inertial measurement unit (IMU) $\boldsymbol{\omega}$. Filtering is required due to vibration, noise, and other IMU and finite difference errors. Filtering occurs both before and after the finite difference.

Unlike motor speed on some quadrotors [72, 76], actuator position measurements are not available. Without direct measurements available, rotational control vector $\boldsymbol{\mu}_f$ is a model-based estimate. Consider the use of main rotor cyclic: $\boldsymbol{\mu}_c$ commands are mixed to cyclic and other controls, passed through compensation and other logic, and output to servos. The servos move with load-dependent dynamics, driving a swashplate and rotor blade roots. The rotor responds with its own dynamics, producing loads that affect the aircraft motion measured by the IMU. Note that this process includes time delays as well as dynamics that might be approximated by time delay or other low-order models. Simple model options for this process include first-order or second-order models with time delay, rate limits, and/or position limits (see Section 4.2.5). Component models (e.g. servo, swashplate, and rotor) could also be used. Since $\boldsymbol{\mu}_f$ should be at the same point in time as $\dot{\boldsymbol{\omega}}_f$, it is appropriate and convenient to select and fit a model using measurements of the response of IMU $\boldsymbol{\omega}$ (e.g. $\dot{\boldsymbol{\omega}}$ from finite difference) to $\boldsymbol{\mu}_c$. For this implementation, the $\hat{\boldsymbol{\mu}}$ models are first-order (with time constant $\tau_{\hat{\boldsymbol{\mu}}}$) plus time delay $\theta_{\hat{\boldsymbol{\mu}}}$.

Since $\boldsymbol{\mu}_f$ and $\dot{\boldsymbol{\omega}}_f$ should be synchronous, the same low-pass filter is applied to both. Filters such as first-order and second-order (e.g. second-order Butterworth) could be used. For this implementation, the low-pass filters are first-order with time constant τ_f.

The controller runs at 50 Hz, the same as the standard rotation controller (Section 3.1).

5.1.2 Performance Analysis

Insight can be gained into controller performance with errors and disturbances by considering various scenarios. From steady state (i.e. $\dot{\boldsymbol{\omega}}_c = \dot{\boldsymbol{\omega}}_f$ and $\boldsymbol{\mu}_c = \boldsymbol{\mu}_f$), consider a change $\boldsymbol{\delta}$ in commanded angular acceleration $\dot{\boldsymbol{\omega}}_c$ resulting in a change $\boldsymbol{G}^{-1}\boldsymbol{\delta}$ in $\boldsymbol{\mu}_c$. On subsequent controller steps, filtered angular acceleration estimate changes $\Delta\dot{\boldsymbol{\omega}}_f$ result in $-\boldsymbol{G}^{-1}\Delta\dot{\boldsymbol{\omega}}_f$ changes in $\boldsymbol{\mu}_c$. In the nominal case, $\Delta\boldsymbol{\mu}_f = \Delta\boldsymbol{G}^{-1}\dot{\boldsymbol{\omega}}_f$, i.e. $\boldsymbol{\mu}_c$ remains constant as acceleration steps are balanced by control steps. In general, $\boldsymbol{\mu}_c$ will increment at each controller step to correct mismatches due to disturbances, nonlinearities, and measurement, estimation, and controller parameter errors.

With a long filter time constant, the controller acts like a PD attitude controller at short time scales with the filtered terms ($\boldsymbol{\mu}_f$ and $\dot{\boldsymbol{\omega}}_f$) acting as slowly-changing bias or trim terms that remove steady-state error. In this way, the controller gracefully degrades to a PID-like controller if a long filter time constant is used (e.g. due to high vibration or for initial testing). While the time scale separation principle [72] may be violated—i.e. the changes in actuator moments may no longer dominate the moment changes due to variations in body rotation rates or velocities—PID-like performance and stability can be retained.

Next, actuator model $\hat{\boldsymbol{\mu}}$ time errors are considered. A formal analysis of actuator model and control effectiveness errors, including root loci, is in Binz and Moormann [6]. If $\hat{\boldsymbol{\mu}}$ is faster than actual $\boldsymbol{\mu}$, $\boldsymbol{\mu}_f$ will update faster than acceleration $\dot{\boldsymbol{\omega}}_f$, and control command $\boldsymbol{\mu}_c$ will be incremented (and later decremented to reach steady state). This reduces damping [6]. With everything else nominal, for a change $\boldsymbol{\delta}$ from steady state in commanded angular acceleration $\dot{\boldsymbol{\omega}}_c$, the worst case is a $2\boldsymbol{G}^{-1}\boldsymbol{\delta}$ change in $\boldsymbol{\mu}_c$, i.e. an initial change of $\boldsymbol{G}^{-1}\boldsymbol{\delta}$ followed by an increase to $2\boldsymbol{G}^{-1}\boldsymbol{\delta}$ as $\boldsymbol{\mu}_f$ completes its response and finally a return to $\boldsymbol{G}^{-1}\boldsymbol{\delta}$ as acceleration $\dot{\boldsymbol{\omega}}_f$ responds.

If $\hat{\boldsymbol{\mu}}$ is slower than actual $\boldsymbol{\mu}$, $\boldsymbol{\mu}_f$ will update slower than acceleration $\dot{\boldsymbol{\omega}}_f$, and control command $\boldsymbol{\mu}_c$ will be decremented (and later incremented to reach steady state). This increases damping (but slows the response) [6], indicating that it is safer to model the actuator as slower instead of faster than the actual actuator [70]. With everything else nominal, for a change $\boldsymbol{\delta}$ from steady state in commanded angular acceleration $\dot{\boldsymbol{\omega}}_c$, the worst case is an initial change of $\boldsymbol{G}^{-1}\boldsymbol{\delta}$ followed by a decrease to the previous steady-state value as acceleration $\dot{\boldsymbol{\omega}}_f$ completes its response and finally a return to $\boldsymbol{G}^{-1}\boldsymbol{\delta}$ as $\boldsymbol{\mu}_f$ responds.

If control effectiveness $\boldsymbol{G}$ is too small/large (i.e. $\boldsymbol{G}^{-1}$ too large/small), then the initial change in $\boldsymbol{\mu}_c$ will be too large/small and will be decremented/incremented as $\boldsymbol{\mu}_f$ and $\dot{\boldsymbol{\omega}}_f$ respond. The former case reduces damping and the latter case increases damping but the response is slowed [6].

There is no windup when commands are saturated at a known limit, e.g.

$$\boldsymbol{\mu}_c = \mathbf{sat}\left(\boldsymbol{\mu}_f + \boldsymbol{G}^{-1}\left(\dot{\boldsymbol{\omega}}_c - \dot{\boldsymbol{\omega}}_f\right), \boldsymbol{\mu}_{\text{lim}}\right)$$

where **sat** is the saturation function with limits $\boldsymbol{\mu}_{\text{lim}}$. Steady state when saturated at the limit is $\boldsymbol{\mu}_c = \boldsymbol{\mu}_{\text{lim}}$, $\boldsymbol{\mu}_f = \boldsymbol{\mu}_{\text{lim}}$, with each subsequent controller step capable of remaining saturated or moving away from the limit depending on $\dot{\boldsymbol{\omega}}_c - \dot{\boldsymbol{\omega}}_f$.

5.1.3 Controller Tuning Guidelines

Tischler et al. [77] recommended controller tuning using multi-objective parametric optimization to meet specifications and requirements. Nevertheless, heuristic guidelines are provided here. The attitude controller PD gains should be set for good performance and stability (e.g. using standard eigenvalue analysis), with the presence of the INDI acceleration controller allowing the attitude and rate gains to be relaxed relative to a controller without acceleration feedback. Control effectiveness $\boldsymbol{G}$ should be set accurately, erring to larger values for initial tests [6] or if the correct values are uncertain. Similarly, actuator model $\hat{\boldsymbol{\mu}}$ should be set accurately, erring to slower values for initial tests [6] or if the correct values are uncertain. The filter time constant should be set as short as possible for performance and to correct for control effectiveness errors but set as long as necessary for $\dot{\boldsymbol{\omega}}_f$ noise [6], $\boldsymbol{\mu}_f$ errors, and actuator activity.

5.2 Translational INDI Control

The following translational acceleration control law was developed primarily from those used with quadrotors in Smeur et al. (2018) [71] and Tal and Karaman [76]. The objective was to formulate a simple INDI control law that could be evaluated without additions that might be unnecessary for certain applications.

The INDI control law for translational acceleration is

$$\boldsymbol{T}_c = \boldsymbol{T}_f + m\boldsymbol{E}^{-1}\left(\boldsymbol{a}_c - \boldsymbol{a}_f\right) \tag{5.2}$$

All vectors are in an earth-fixed frame. $\boldsymbol{T}_c$ is the commanded thrust (control) vector, $\boldsymbol{T}_f$ is a filtered estimate of the thrust vector (i.e. actuator positions), m is aircraft mass, $\boldsymbol{E}^{-1}$ is the inverse of the acceleration control effectiveness matrix (nominally the identity matrix), $\boldsymbol{a}_c$ is the commanded acceleration vector, and $\boldsymbol{a}_f$ is a filtered estimate (e.g. measurement) of the acceleration vector. $\boldsymbol{a}_f$ is calculated by rotating the body axis accelerometer measurements to the North East Down (NED) frame, adding the gravity vector, and then filtering. The estimators and filters should be such that $\boldsymbol{T}_f$ and $\boldsymbol{a}_f$ are at the same point in time [71].

Equation (5.2) is close to those in the literature. Compared to Smeur et al. (2018) [71] and Tal and Karaman [76], a control effectiveness inverse matrix has been added.

Attitude and scalar thrust commands can be calculated from $\boldsymbol{T}_c$ using nonlinear inversion as

$$\begin{aligned} T_c &= \|\boldsymbol{T}_c\|_2 \\ \left(\Delta\phi_z\right)_c &= \arcsin\left(\frac{\sin\left(\Delta\phi_x\right)\left(\boldsymbol{T}_c\right)_N + \cos\left(\Delta\phi_x\right)\left(\boldsymbol{T}_c\right)_E}{T_c}\right) \\ \left(\Delta\phi_y\right)_c &= \arcsin\left(\frac{-\cos\left(\Delta\phi_x\right)\left(\boldsymbol{T}_c\right)_N + \sin\left(\Delta\phi_x\right)\left(\boldsymbol{T}_c\right)_E}{T_c}\right) \end{aligned}$$

$\boldsymbol{\Delta\phi}$ are incremental rotations about each body axis from x axis up, y axis east, and z axis north. Calculations for quaternions [76] and Euler angles [71] are also available.

For the case where $\boldsymbol{T}_c$ is rotated so that the horizontal components are aligned with the body y and z axes then

$$(\Delta\phi_z)_c = \arcsin\left(\frac{(\boldsymbol{T}_c)_y}{T_c}\right)$$
$$(\Delta\phi_y)_c = \arcsin\left(\frac{-(\boldsymbol{T}_c)_z}{T_c}\right)$$

5.2.1 Horizontal Control

The INDI control law (Equation (5.2)) can be modified for the case where only horizontal and not vertical acceleration is being controlled by this law. One option is to assume that vertical acceleration error is zero, i.e.

$$(a_D)_{\text{err}} = (a_D)_c - (a_D)_f = 0 \tag{5.3}$$

This is an assumption of good vertical regulation or slowly-changing vertical dynamics. An alternative assumption is zero vertical acceleration, i.e. $(a_D)_c = 0$. Other assumptions can be made with additional information, e.g. current vertical acceleration controller command or powerplant dynamics.

5.2.2 Body-Related Axes

Both Smeur et al. (2018) [71] and Tal and Karaman [76] used an NED frame for INDI acceleration control. This is an appropriate choice for earth-fixed position, velocity, and acceleration references as well as for exogenous disturbances such as wind (and gravity). Also, position and velocity data from sensors such as GPS are in an earth-fixed frame. Furthermore, the quadrotors used in Smeur et al. (2018) [71] and Tal and Karaman [76] have similar dynamics in various horizontal directions, e.g. similar roll and pitch dynamics.

In contrast, there are arguments for calculations in a body frame. The thrust vector is typically aligned with, or at least defined relative to, a body frame. Aircraft dynamics can vary significantly in different horizontal directions, e.g. rotorcraft such as typical helicopters—or tailsitters—with low-drag and high-drag horizontal axes. These asymmetries can benefit from different gains or even controller structures. Accelerometers are mounted in the body frame as are vision sensors that can be used for position or velocity references or feedback. These factors can lead to body-fixed commands such as high forward speed with small side speed or slip.

With consistent transformations, any convenient frame or set of frames can be used. However, errors can be introduced with inconsistent transformations, e.g. using rotation matrices from different points in time.

One body-related frame is *yawed NED*. Define the yawed NED frame as the NED frame rotated by the aircraft yaw angle about the Down (local vertical) axis, i.e. using rotation matrix

$$\boldsymbol{R}_{\text{NED to yawed NED}} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

5.2.3 Flexrotor Implementation

For the implementation on Flexrotor, the yawed NED frame is used, allowing gains to vary for different horizontal body-aligned axes.

$\boldsymbol{a}_c$ is the output of a proportional-derivative (PD) controller with linear feedback on position and velocity errors. The PD gains are set for a specified natural frequency and damping ratio on each axis. Note that there is no need for integral action (e.g. PID controller) with this incremental control law for acceleration.

The thrust estimate is first calculated so that the vertical component equals aircraft weight. Thrust axis acceleration as measured by the IMU is then added if and only if this will increase the thrust estimate. This increase-only condition is used to avoid “overcontrol” due to too low of a thrust estimate.

Since $\boldsymbol{T}_f$ and $\boldsymbol{a}_f$ should be synchronous, the same low-pass filter is applied to both. For this implementation, the low-pass filters are first-order with time constant τ_f.

Vertical acceleration error is assumed to be zero (Equation (5.3)) and only the controller tilt attitude command outputs are used, leaving the vertical dimension to be handled by a

Table 5.1: Rotational INDI controller parameters for simulation testing.

Category	Parameter	Value
PD gains	natural frequencies $\omega_{n_x}, \omega_{n_y}, \omega_{n_z}$	$3.0\,\mathrm{rad/s}$
	damping ratios $\zeta_x, \zeta_y, \zeta_z$	1.0
Control effectiveness	g_x, g_y, g_z	1.0
Acceleration and control model filters	time constant τ_f	$0.06\,\mathrm{s}$
Control model $\hat{\boldsymbol{\mu}}$	roll time constant $\tau_{\hat{\boldsymbol{\mu}}_x}$	$0.05\,\mathrm{s}$
	roll time delay $\theta_{\hat{\boldsymbol{\mu}}_x}$	$0.09\,\mathrm{s}$
	pitch-yaw time constant $\tau_{\hat{\boldsymbol{\mu}}_{y,z}}$	$0.05\,\mathrm{s}$
	pitch-yaw time delay $\theta_{\hat{\boldsymbol{\mu}}_{y,z}}$	$0.13\,\mathrm{s}$

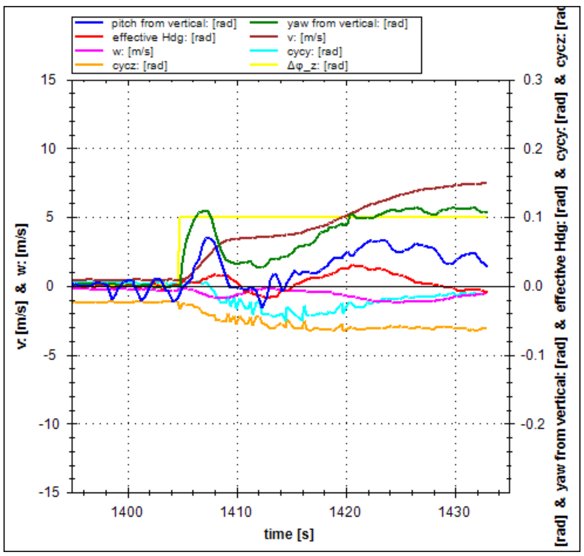
separate altitude or total-energy controller using engine power as the actuator.

5.3 Simulation Results

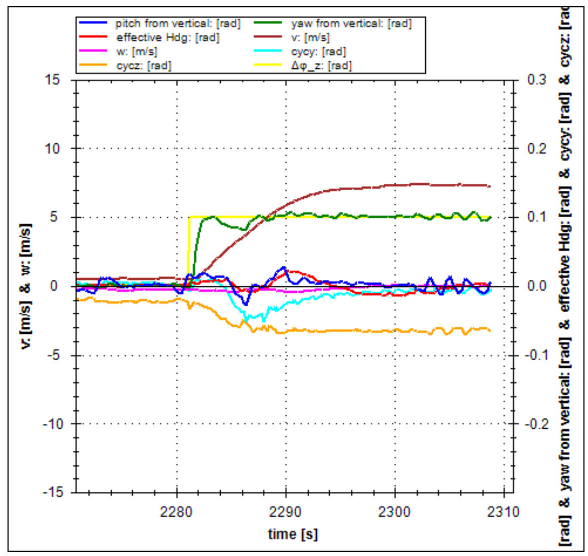
Numerical time simulation is used for testing since strong nonlinearities are a key system characteristic. Two simulation types are used: numerical *software simulation* and hardware-in-the-loop *HIL simulation*. Both simulation types have separate modules for the environment, aircraft dynamics, and onboard software. The same flight computer and other avionics that are used on an actual aircraft are used in HIL simulation, connecting to the simulator computer to exchange simulated sensor signals, actuator measurements, and other data. In software simulation, the onboard software runs on the simulator computer instead of the flight computer and the necessary avionics components are simulated.

5.3.1 Rotational INDI Simulation Results

Except where specified otherwise, simulation tests of both the standard and INDI controllers are with legs closed and cyclic AUTHORITY COMP set to FLOW MODEL (Section 3.1.1).



(a) Standard controller.



(b) INDI controller for pitch and yaw (standard controller for roll).

Figure 5.1: HIL simulation yaw tilt command steps from hover for standard and INDI rotation controllers. The yellow trace ($\Delta\phi_z$) is the yaw tilt command, green is yaw from vertical and blue is pitch from vertical. Simulations are of Flexrotor *Amphinome* (1.85 m diameter rotor) at 16.8 kg. INDI actuator model parameters are $\tau_{\hat{\boldsymbol{\mu}}_{y,z}} = 0.06\,\text{s}, \theta_{\hat{\boldsymbol{\mu}}_{y,z}} = 0.11\,\text{s}$.

Except where specified otherwise, INDI controller parameters for simulation testing are as in Table 5.1. The actuator model time constants $\tau_{\hat{\boldsymbol{\mu}}}$ and time delays $\theta_{\hat{\boldsymbol{\mu}}}$ were determined by a least-squares fit of a first-order plus time delay model for each axis to the software simulation $\dot{\boldsymbol{\omega}}$ (i.e. finite-differenced rotational acceleration) response to a step from trim rotational control command $\boldsymbol{\mu}_c$.

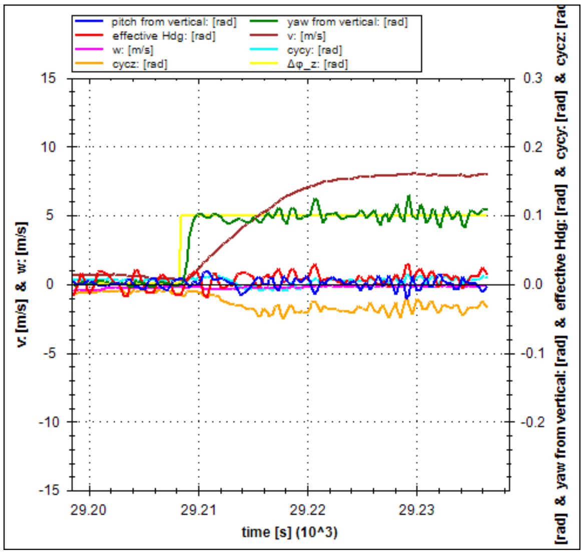
Figure 5.1 demonstrates improved on-axis and off-axis attitude regulation with the INDI controller compared to with the standard controller (Section 3.1). Starting from hover in zero wind with fuselage vertical (i.e. $\Delta\phi_y = 0$ and $\Delta\phi_z = 0$) and heading zero, the yaw tilt $\Delta\phi_z$ command is stepped from 0 to 0.1 rad while $\Delta\phi_y$ and heading targets remain at zero. With the standard controller (Figure 5.1a), yaw reaches the target quickly with 10% overshoot, but then decreases to 0.3 of target and is not back to 0.8 of target until 13 s after the command step. Off-axis pitch tilt $\Delta\phi_y$ reaches 0.7 of the step amplitude and is still more than 20% off the zero target 30 s after the yaw command step. With the INDI controller (Figure 5.1b), yaw tilt reaches the target quickly with no overshoot, then remains within 20% of target. Off-axis pitch tilt remains at or below 0.3 of on-axis target (i.e. close to the desired zero).

Figures 5.2–5.4 explore sensitivity to INDI controller parameter variations. Figure 5.2 indicates low sensitivity to filter time constant and that the INDI controller regulates attitude better than the standard controller, even with a long $\tau_f = 0.96\,\mathrm{s}$.

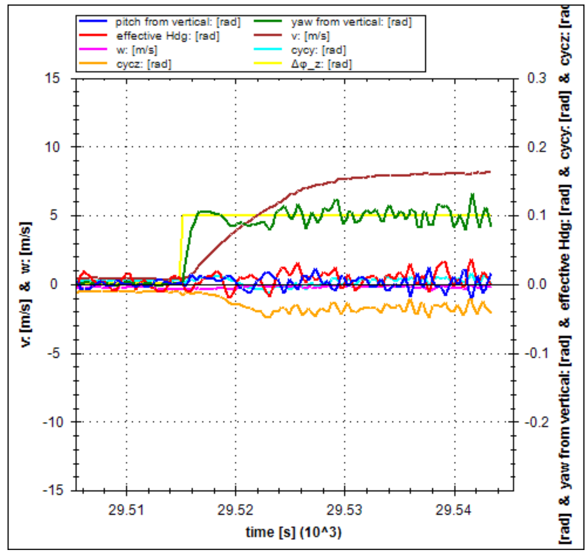
Figure 5.3 indicates low sensitivity to longer actuator model time constants and an "over-control" oscillation for a time constant that is shorter than the actual actuator time constant (Figure 5.3b).

Figure 5.4 indicates that decreasing control effectiveness from the nominal $g = 1.0$ improves low-frequency attitude regulation at the cost of increased actuator activity. Conversely, increasing control effectiveness degrades attitude regulation, resulting in an "under-control" oscillation for $g = 1.5$ (Figure 5.4c).

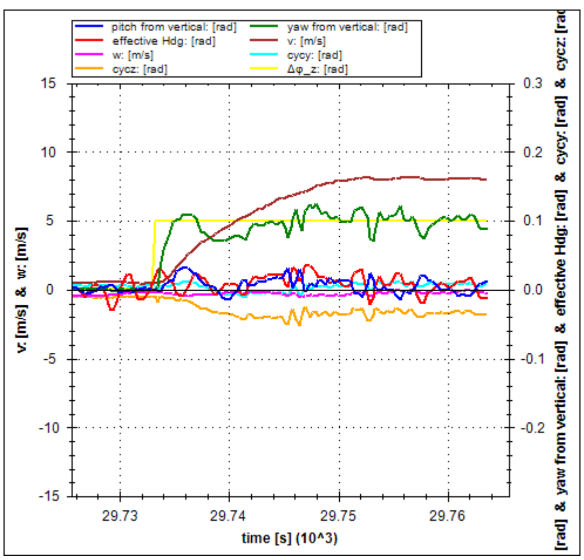
Vibration or other rotation rate noise affects both the standard and INDI rotation controllers through rotation rate (and attitude) feedback. Rotation rate noise also affects the



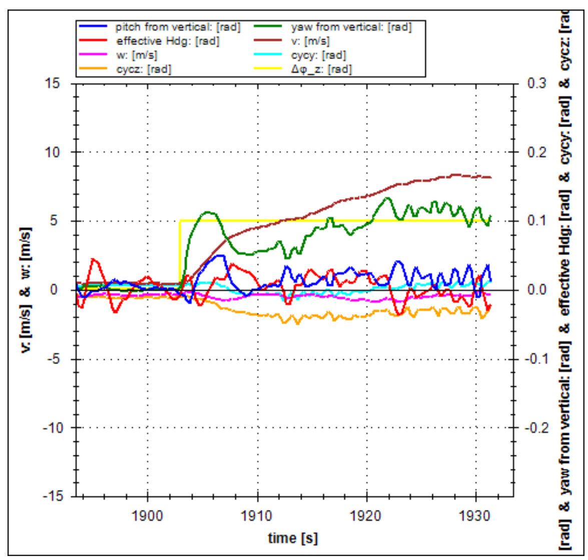
(a) INDI controller with nominal $\tau_f = 0.06\,\mathrm{s}$.



(b) INDI controller with $\tau_f = 0.24\,\mathrm{s}$.

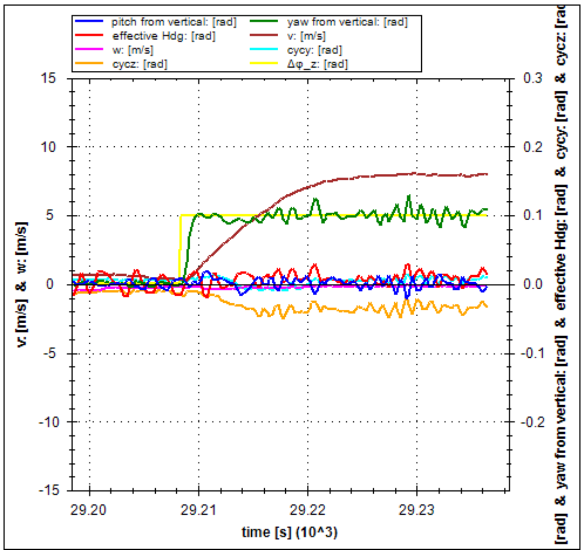


(c) INDI controller with $\tau_f = 0.96\,\mathrm{s}$.

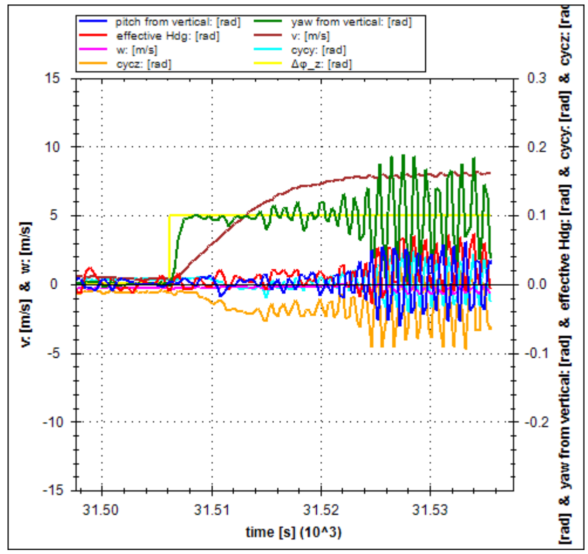


(d) Standard controller.

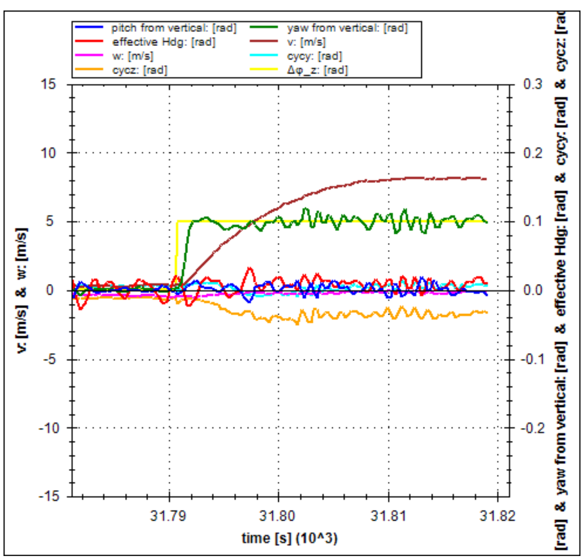
Figure 5.2: HIL simulation yaw tilt command steps from hover for various INDI rotation controller filter time constants τ_f. Simulations are of Flexrotor *Doto* (2.20 m diameter rotor) at 20 kg.



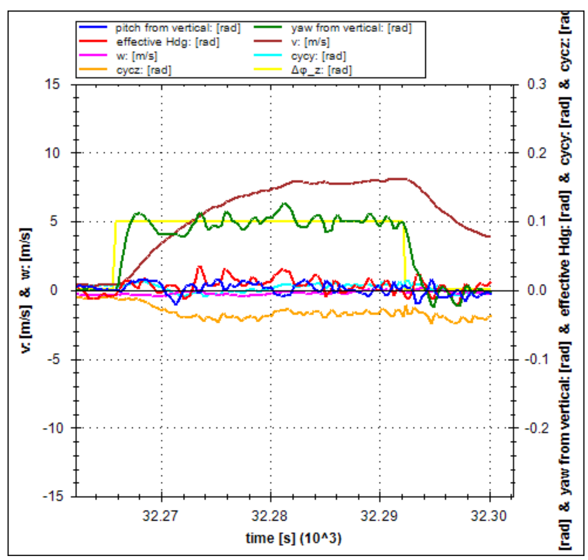
(a) INDI controller with nominal $\tau_{\hat{\mu}} = 0.05\,\text{s}$.



(b) INDI controller with $\tau_{\hat{\mu}} = 0.03\,\text{s}$.

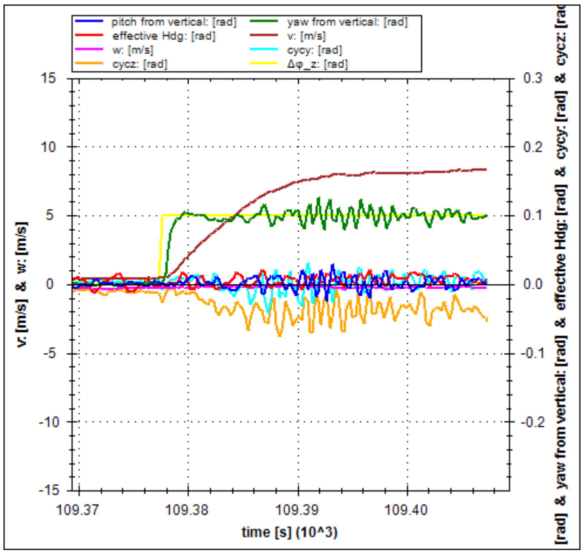


(c) INDI controller with $\tau_{\hat{\mu}} = 0.10\,\text{s}$.

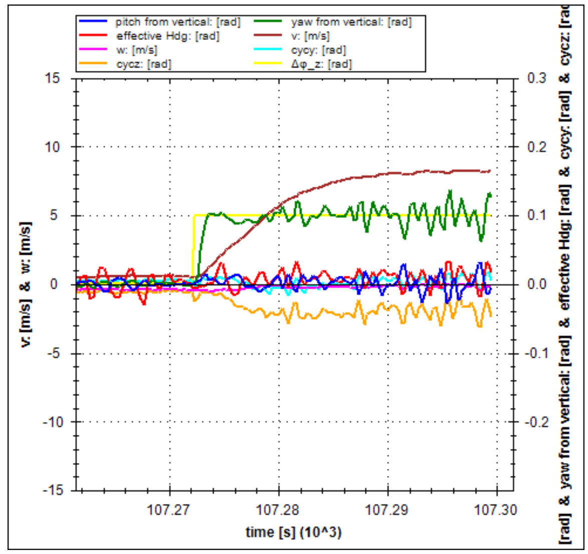


(d) INDI controller with $\tau_{\hat{\mu}} = 0.40\,\text{s}$.

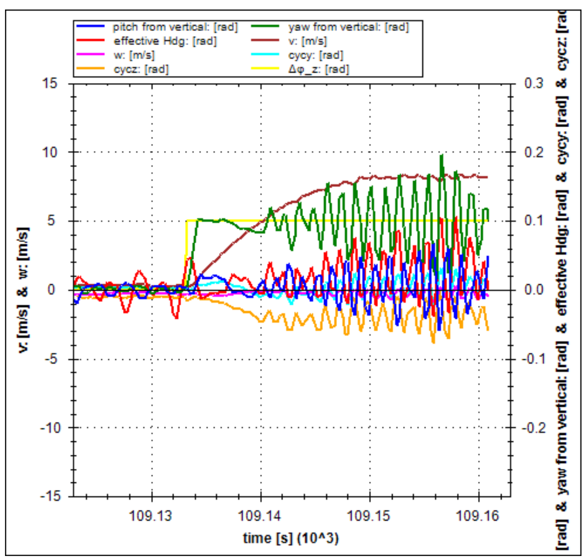
Figure 5.3: HIL simulation yaw tilt command steps from hover for various INDI rotation controller actuator time constants $\tau_{\hat{\mu}}$. Simulations are of Flexrotor *Doto* (2.20 m diameter rotor) at 20 kg.



(a) INDI controller with $g = 0.67$.

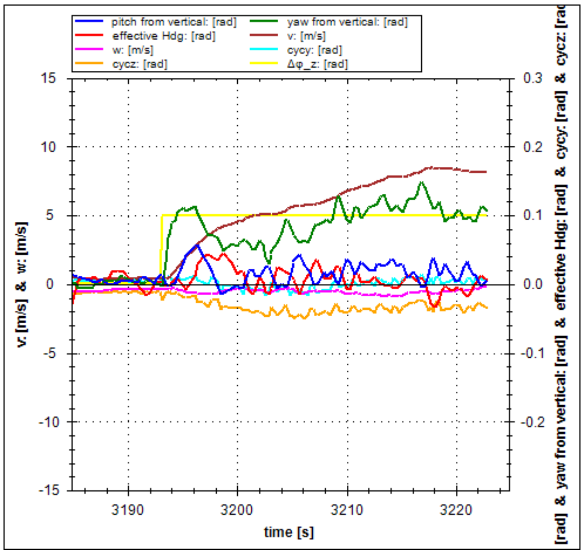


(b) INDI controller with nominal $g = 1.0$.

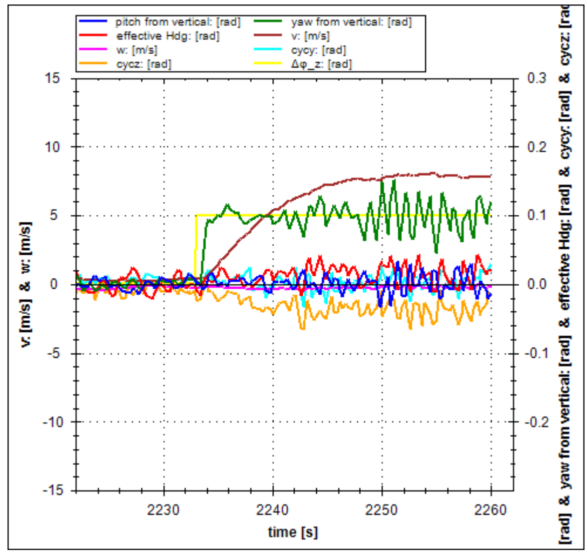


(c) INDI controller with $g = 1.5$.

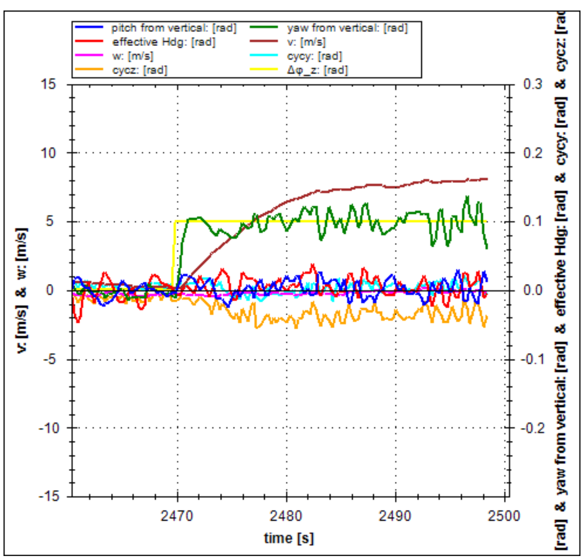
Figure 5.4: HIL simulation yaw tilt command steps from hover for various INDI rotation controller control effectiveness values g. The cyan and orange traces are cyclic commands. Simulations are of Flexrotor *Doto* (2.20 m diameter rotor) at 20 kg.



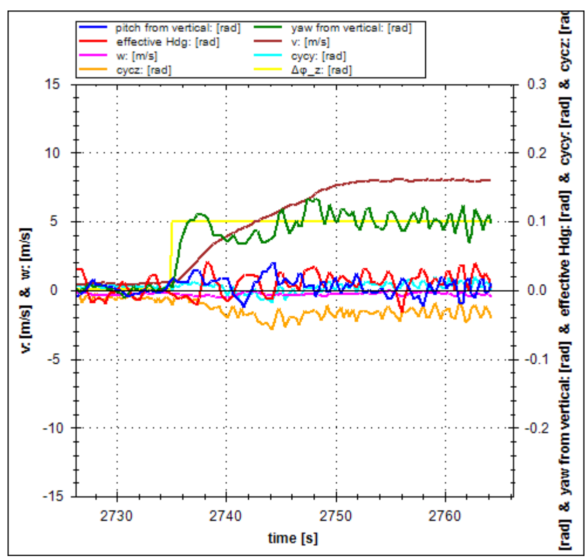
(a) Standard controller.



(b) INDI controller with nominal $\tau_f = 0.06\,\mathrm{s}$.

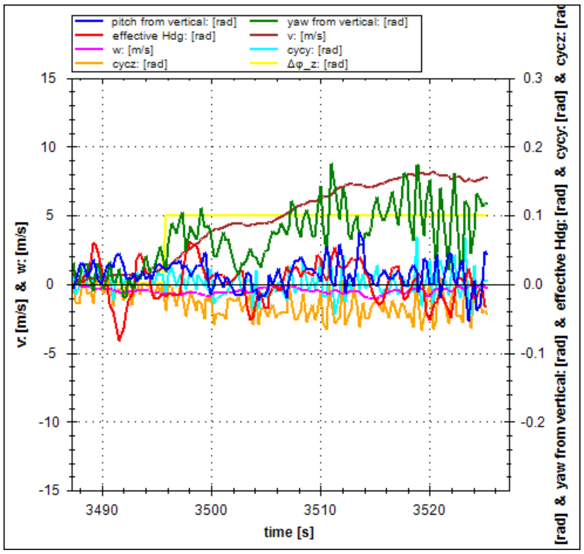


(c) INDI controller with $\tau_f = 0.24\,\mathrm{s}$.

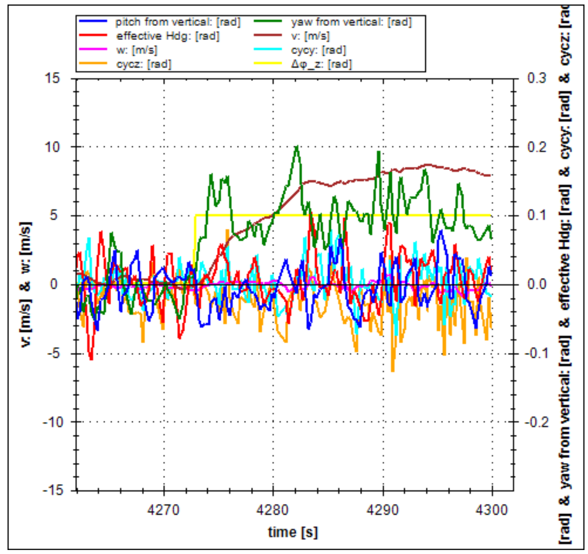


(d) INDI controller with $\tau_f = 0.96\,\mathrm{s}$.

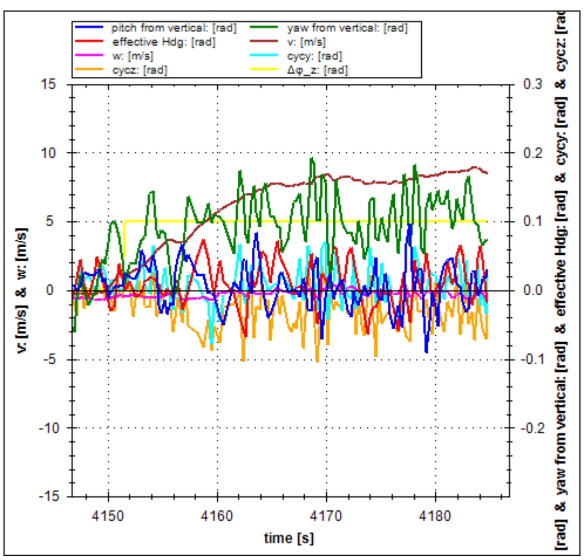
Figure 5.5: HIL simulation yaw tilt command steps from hover with 10 times the default simulator rotation rate noise amplitude (i.e. 0.040 rad/s noise amplitude and 0.02 s period). Simulations are of Flexrotor *Doto* (2.20 m diameter rotor) at 20 kg.



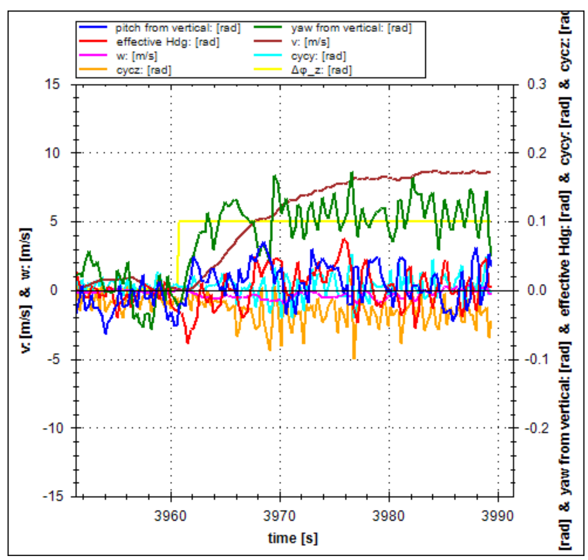
(a) Standard controller.



(b) INDI controller with $\tau_f = 0.12\,\mathrm{s}$.



(c) INDI controller with $\tau_f = 0.24\,\mathrm{s}$.



(d) INDI controller with $\tau_f = 0.96\,\mathrm{s}$.

Figure 5.6: HIL simulation yaw tilt command steps from hover with 50 times the default simulator rotation rate noise amplitude (i.e. 0.200 rad/s noise amplitude and 0.02 s period). Simulations are of Flexrotor *Doto* (2.20 m diameter rotor) at 20 kg.

Table 5.2: Rotational INDI controller parameters for flight testing.

Category	Parameter	Value
PD gains	natural frequencies $\omega_{n_x}, \omega_{n_y}, \omega_{n_z}$	$3.0\,\mathrm{rad/s}$
	damping ratios $\zeta_x, \zeta_y, \zeta_z$	1.0
Control effectiveness	g_x, g_y, g_z	1.0
Acceleration and control model filters	time constant τ_f	$0.24\,\mathrm{s}$
Control model $\hat{\boldsymbol{\mu}}$	roll time constant $\tau_{\hat{\boldsymbol{\mu}}_x}$	$0.10\,\mathrm{s}$
	roll time delay $\theta_{\hat{\boldsymbol{\mu}}_x}$	$0.09\,\mathrm{s}$
	pitch-yaw time constant $\tau_{\hat{\boldsymbol{\mu}}_{y,z}}$	$0.10\,\mathrm{s}$
	pitch-yaw time delay $\theta_{\hat{\boldsymbol{\mu}}_{y,z}}$	$0.13\,\mathrm{s}$

INDI controller through finite-differenced rotational acceleration. Figures 5.5 and 5.6 indicate that even for high noise levels, an INDI controller filter time constant can be selected for improved regulation and a moderate increase in actuator activity compared to the standard controller.

5.4 *Flight Results*

The Flexrotor rotational INDI controller was tested on four days of flights, with the fourth also testing the translational INDI controller. Except where specified otherwise, all flight tests of both the standard (Section 3.1) and INDI controllers were with 2.20 m diameter rotor, legs closed, and cyclic AUTHORITY COMP set to FLOW MODEL (Section 3.1.1).

Except where specified otherwise, INDI rotation controller parameters for flight testing are as in Table 5.2. The filter time constant τ_f and actuator model time constants $\tau_{\hat{\boldsymbol{\mu}}}$ are conservative (i.e. large) values.

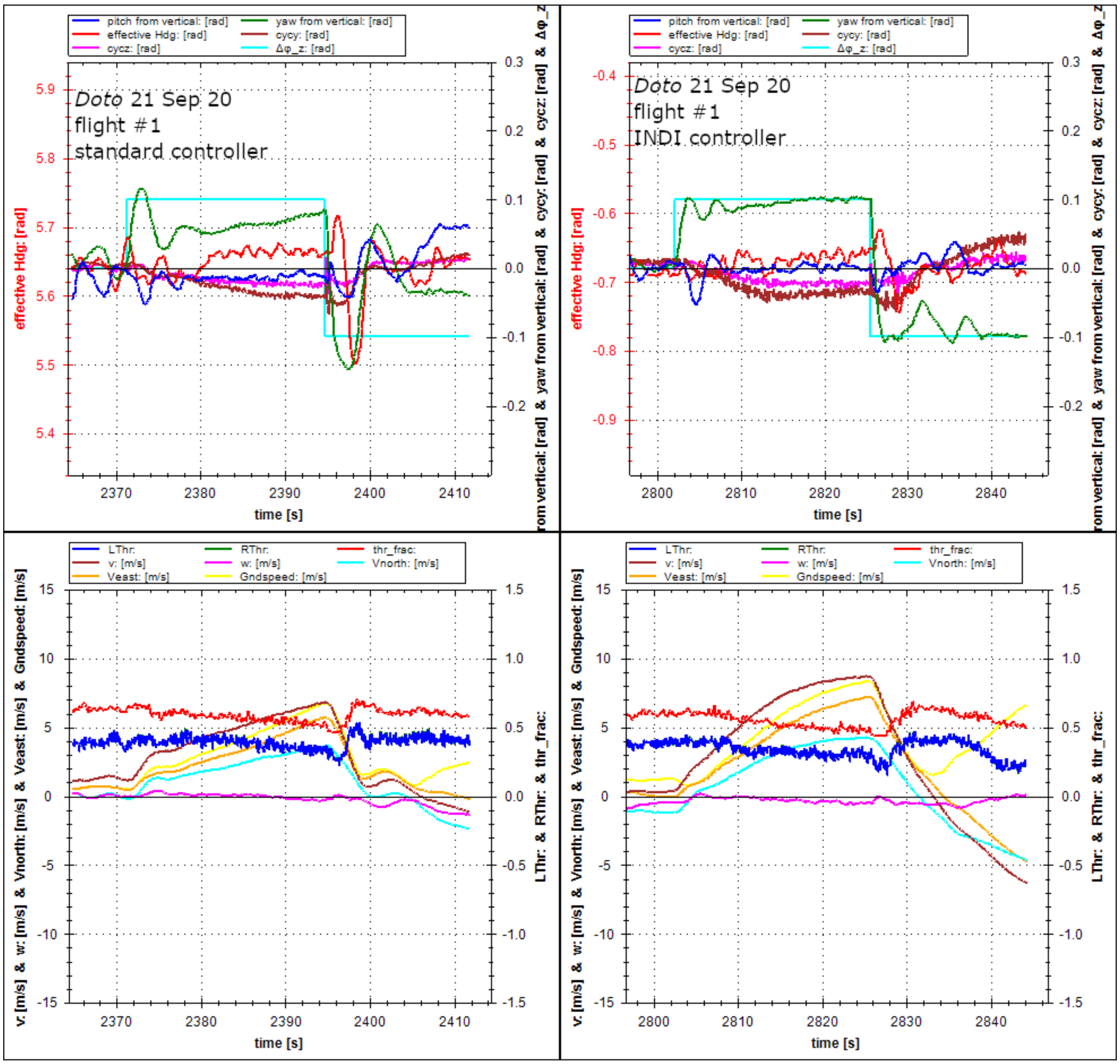


Figure 5.7: Flight yaw tilt command steps from fuselage vertical for standard (left) and INDI rotation controllers (right). In the upper plots, the cyan trace ($\Delta\phi_z$) is the yaw tilt command, green is yaw from vertical, blue is pitch from vertical, and red is heading. The upper-plot brown and magenta traces are cyclic commands.

5.4.1 Rotational INDI Flight Results

Attitude regulation was significantly better with the INDI controller than with the standard controller (Figure 5.7). Actuator activity was generally higher with the INDI controller, but gains can be adjusted to trade actuator activity with performance (see Figures 5.13 and 5.14). For Flexrotor, there are INDI gains that result in an acceptable amount of actuator activity while simultaneously achieving significantly better regulation than with the standard controller.

Consistent with simulation (Figure 5.1), Figure 5.7 demonstrates improved on-axis and off-axis attitude regulation with the INDI controller compared to with the standard controller. Starting from fuselage vertical (i.e. $\Delta\phi_y = 0$ and $\Delta\phi_z = 0$), the yaw tilt $\Delta\phi_z$ command is stepped from 0 to 0.1 rad and then to -0.1 rad while $\Delta\phi_y$ and heading targets remain unchanged. With the standard controller, yaw reaches the 0.1 rad target quickly with some overshoot, but then decreases to a fraction of target and is not back to target 20 s after the command step. After the yaw command is stepped to -0.1 rad, yaw reaches the target with overshoot but then reverses back to positive tilt before partially correcting toward target. Off-axis pitch tilt $\Delta\phi_y$ and heading also have large deviations after the 0.1 to -0.1 rad yaw command step. Both on-axis and off-axis regulation are substantially improved with the INDI controller. There is an increase in cyclic command activity with the INDI controller. These types of differences between controllers were also seen for pitch tilt steps (not shown).

Figure 5.8 shows heading command steps for standard and INDI rotation controllers. The settling time is much faster with the INDI controller than with the standard controller. Overshoot is larger with the INDI controller for a +1.8 rad heading step but smaller for +0.2 and -0.2 rad steps. Heading regulation was also better with the INDI controller than with the standard controller for yaw tilt (Figure 5.7) and spanwise airspeed (Figure 5.9) command steps.

Figure 5.9 shows faster velocity regulation with the INDI rotation controller than with

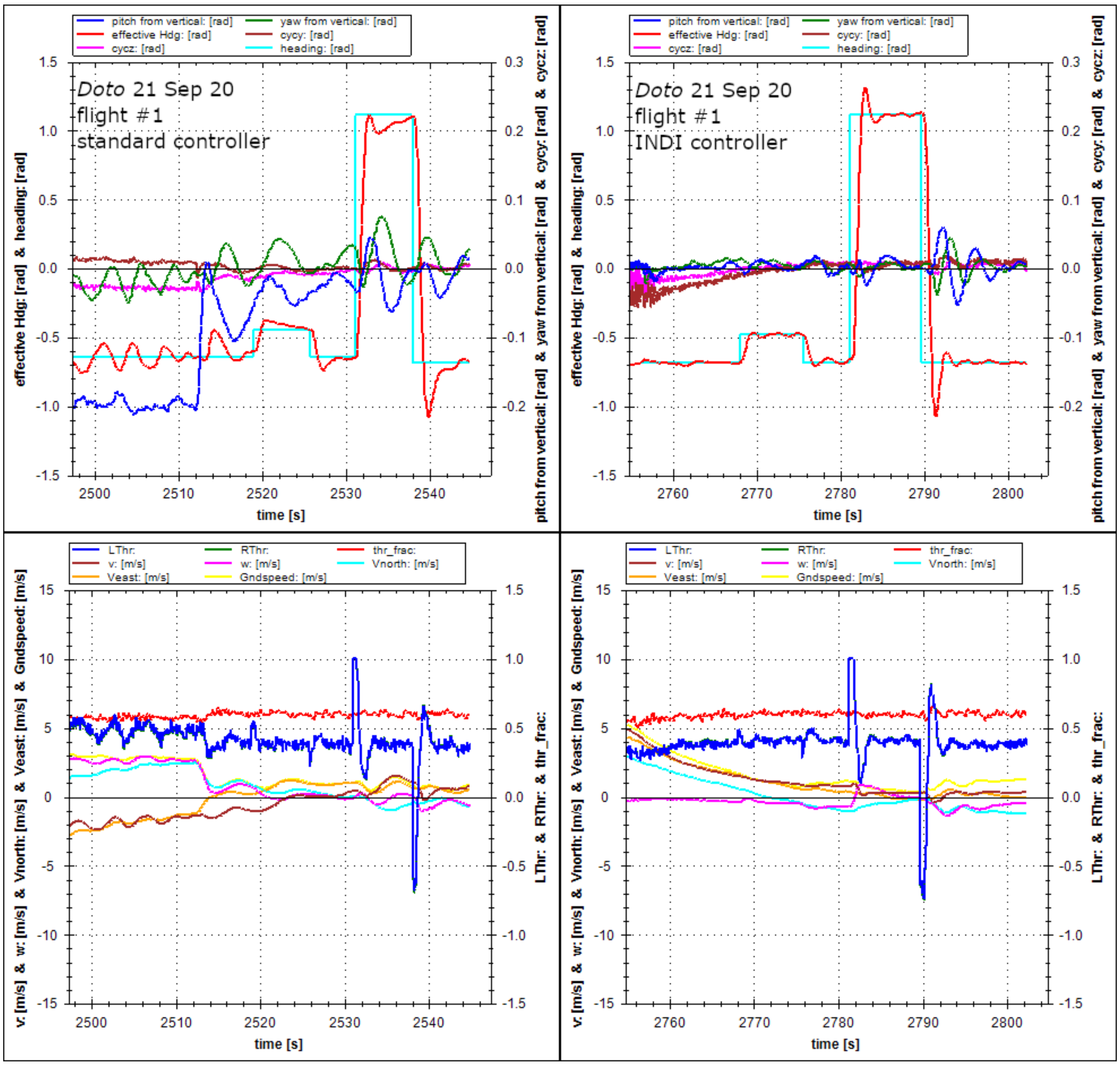


Figure 5.8: Flight heading command steps with fuselage vertical for standard (left) and INDI rotation controllers (right). In the upper plots, the cyan trace is the heading command and red is heading. The lower-plot blue (and green) traces are wingtip thruster commands.

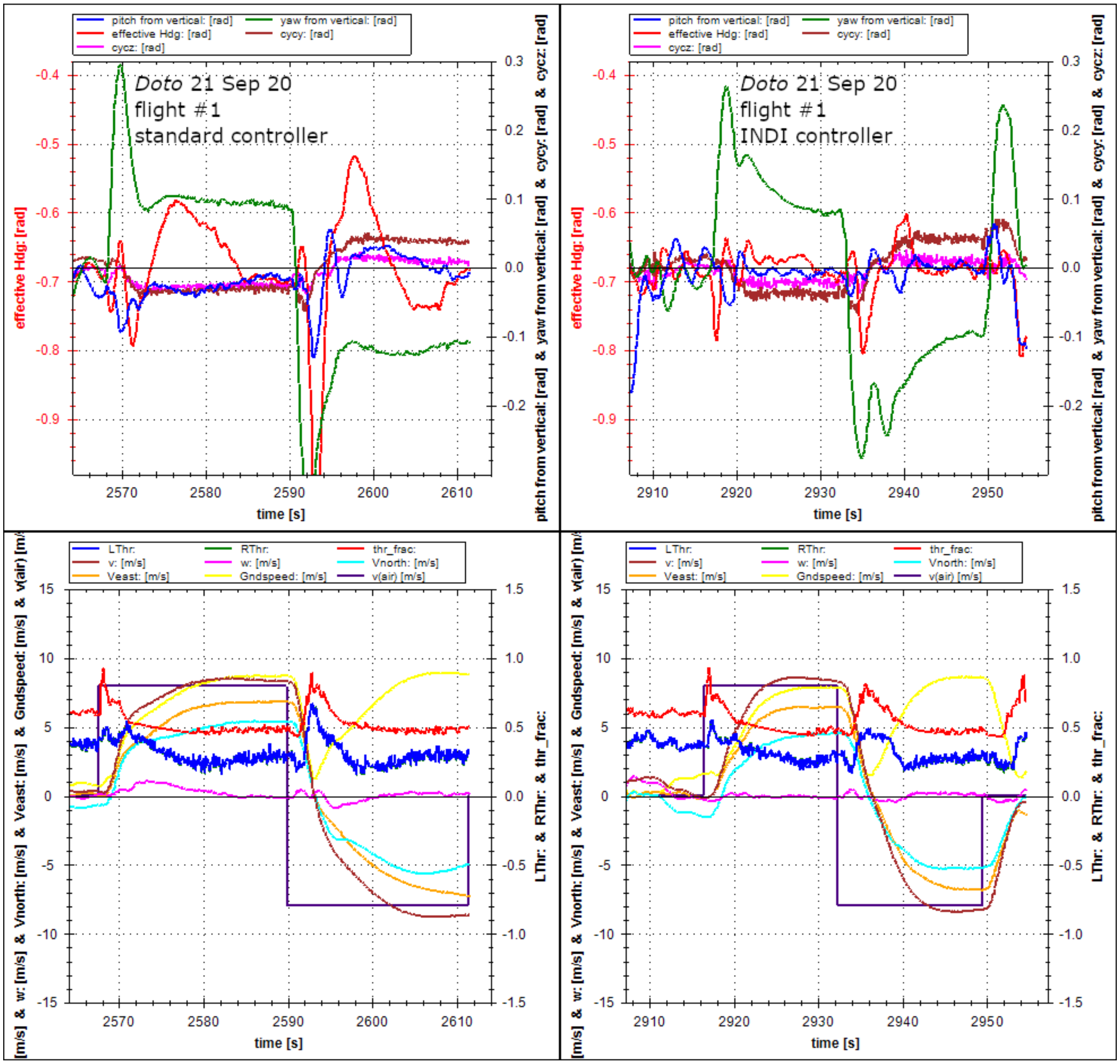


Figure 5.9: Flight spanwise airspeed command steps from position hold for standard (left) and INDI rotation controllers (right). In the lower plots, the purple trace is the spanwise airspeed command and brown is the spanwise airspeed estimate. As in previous figures, in the upper plots green is yaw from vertical, blue is pitch from vertical, and red is heading (commanded here to -0.68 rad).

the standard controller, e.g. 0 to 8 m/s rise times of 10.1 s for the standard controller vs 7.4 s for the INDI controller. The INDI controller also respects the 0.27 rad a priori tilt command limit, e.g. peak yaw tilt for the 8 m/s to -8 m/s command step is -0.38 rad for the standard controller vs -0.28 rad for the INDI controller. Heading regulation is also much better with the INDI controller. Differences between controllers were smaller for steps to maximum tilt of airspeed broadside to the wing (not shown).

Next, INDI controller parameter variations are explored. Consistent with simulation (Figure 5.2), attitude regulation was better with the INDI controller using a long filter time constant than with the standard controller. With filter time constant $\tau_f = 0.48\,\mathrm{s}$, cyclic command activity on a 0.0 to 0.1 rad yaw tilt step was lower than with the nominal $\tau_f = 0.24\,\mathrm{s}$. Cyclic command activity was then at a level similar to that with the standard controller. Yaw tilt regulation was somewhat worse with $\tau_f = 0.48\,\mathrm{s}$ (e.g. 10% settling time of 13.4 s vs 11.2 s with the nominal $\tau_f = 0.24\,\mathrm{s}$) but still much better than with the standard regulator. With $\tau_f = 0.12\,\mathrm{s}$, yaw tilt regulation for a 0.0 to 0.1 rad step was somewhat faster than with the nominal filter time constant (e.g. 10% settling time of 7.1 s vs 11.2 s with the nominal $\tau_f = 0.24\,\mathrm{s}$).

Consistent with simulation (Figure 5.4), decreasing control effectiveness from the nominal $g = 1.0$ improved low-frequency attitude regulation at the cost of increased actuator activity. Increasing control effectiveness degraded attitude regulation, resulting in "under-control". During the first few seconds after a yaw tilt command step from 0.0 to -0.1 rad in Figure 5.10, yaw and pitch tilt are regulated better with pitch-yaw control effectiveness $g_y = g_z = 0.8$ and regulated worse with $g_y = g_z = 1.4$. High-frequency cyclic command activity has a larger amplitude for $g_y = g_z = 0.8$ than for $g_y = g_z = 1.4$.

Figures 5.7–5.10 showed results with cyclic AUTHORITY COMP set to FLOW MODEL (Section 3.1.1). Now, performance is explored with cyclic AUTHORITY COMP set to OFF. Figure 5.10, using the FLOW MODEL, shows cyclic authority estimate varying with estimated airspeed between approximately 0.6 and 1.0. Figure 5.11, with AUTHORITY COMP OFF, has a fixed cyclic authority estimate of 1.0. For *off-0.5* (i.e. AUTHORITY COMP OFF and pitch-

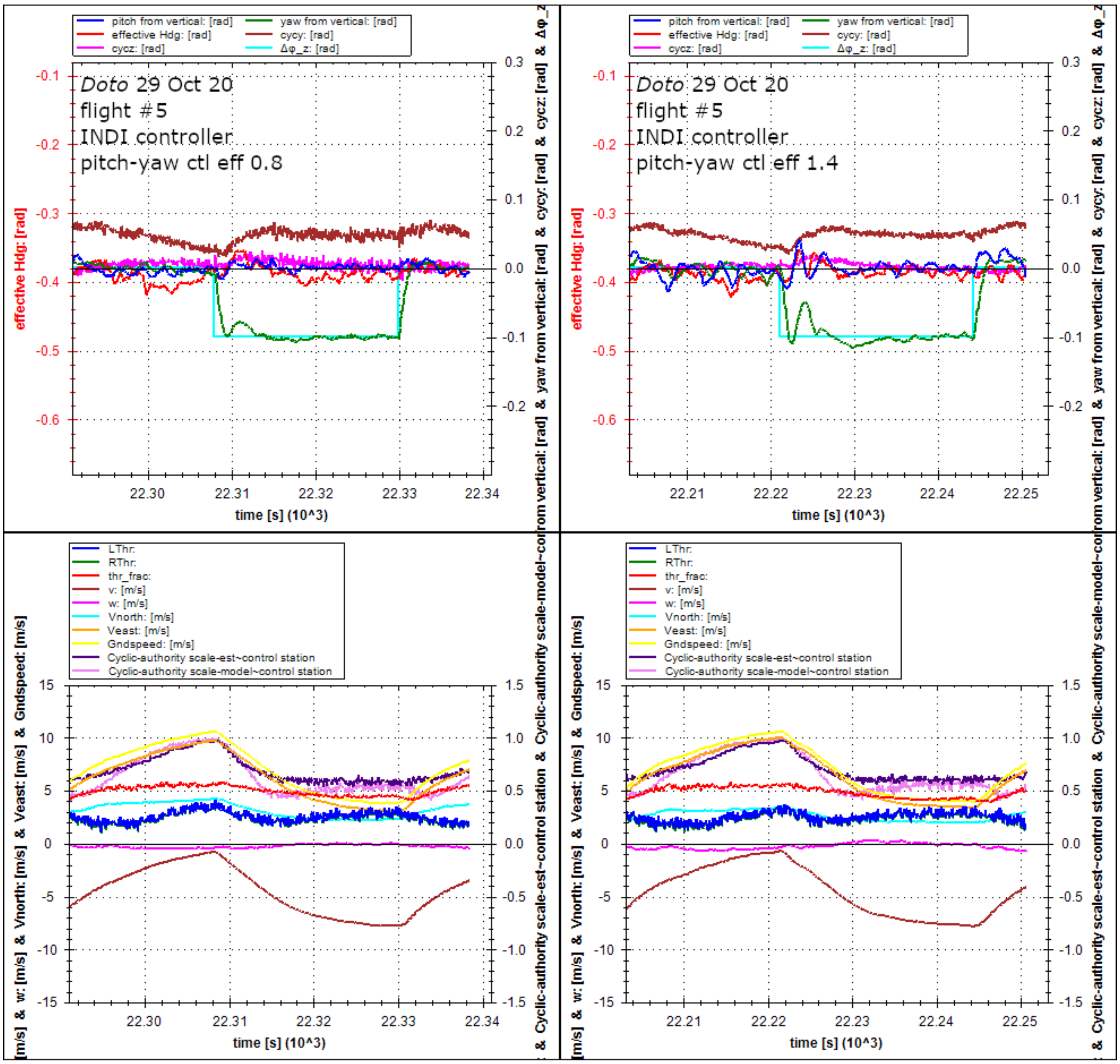


Figure 5.10: Flight yaw tilt command steps from fuselage vertical for INDI rotation controller control effectiveness values $g_y = g_z = 0.8$ (left) and $g_y = g_z = 1.4$ (right). In the lower plots, the purple trace is the cyclic authority estimate and pink is the flow model of cyclic authority, which differ from each other due to the averaging described in Section 3.1.1.

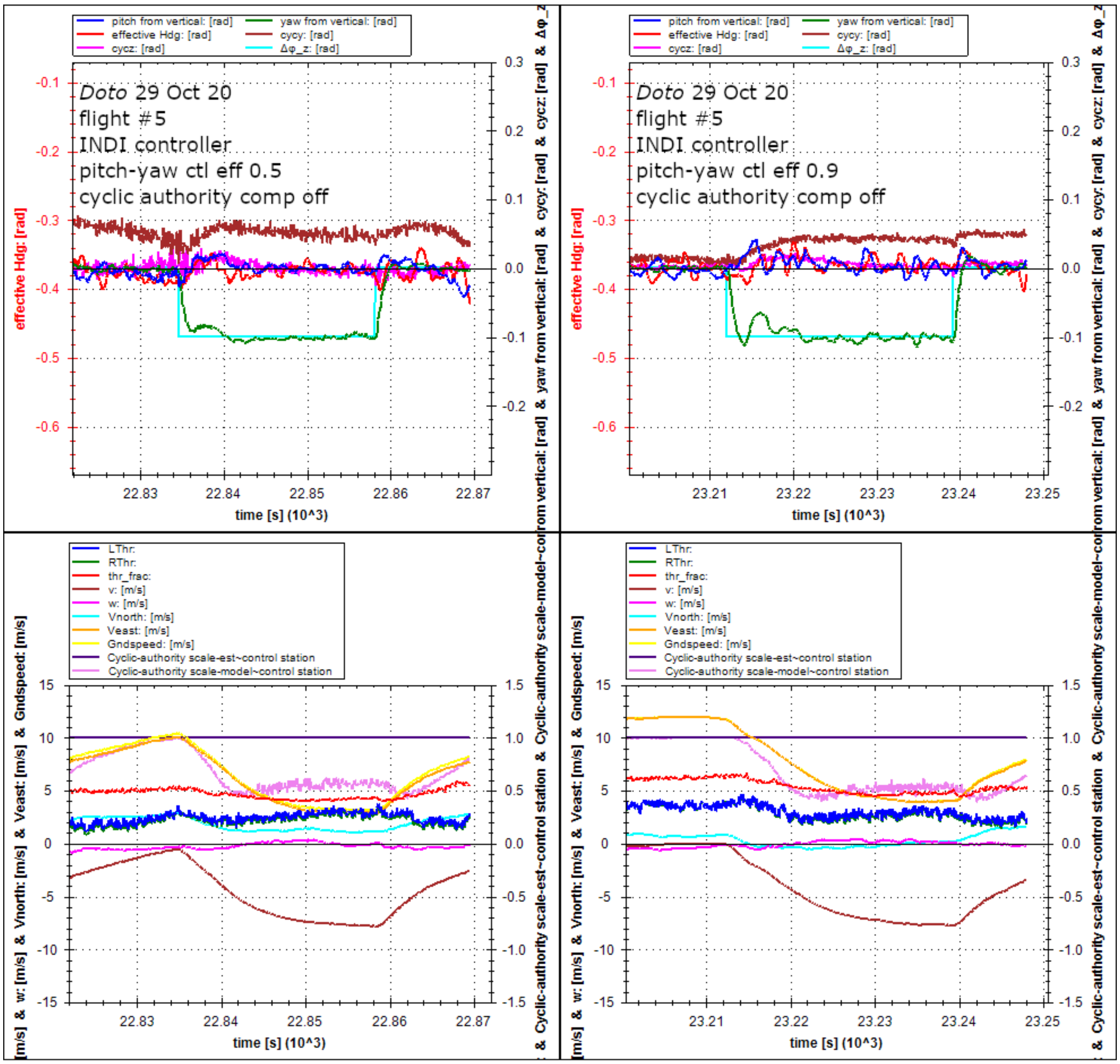


Figure 5.11: Flight yaw tilt command steps from fuselage vertical for INDI rotation controller with cyclic AUTHORITY COMP set to OFF and control effectiveness values $g_y = g_z = 0.5$ (left) and $g_y = g_z = 0.9$ (right).

yaw control effectiveness $g_y = g_z = 0.5$), total *effectiveness scale* was 0.5 (i.e. the product of cyclic authority estimate 1.0 and control effectiveness 0.5). The effectiveness scale for *flow-0.8* in Figure 5.10 varied from approximately 0.5 to 0.8.

Figure 5.11 shows better attitude regulation but larger amplitude high-frequency cyclic command activity with control effectiveness $g_y = g_z = 0.5$ compared to $g_y = g_z = 0.9$ (with cyclic AUTHORITY COMP set to OFF for both).

The remainder of this section examines position regulation with the INDI rotation controller and the tradeoff between regulation performance and actuator activity made by varying control effectiveness parameters. Tests were conducted with the aircraft regulating position relative to a target moving at a series of specified horizontal velocities. Considering wind, this allowed position regulation to be tested across the full tilt and airspeed range. Position regulation with the INDI rotation controller was satisfactory across the range and equal or better than with the standard rotation controller.

Figure 5.12 shows the lowest (left) and highest (right) levels of cyclic command activity during moving-target tracking on 29 October 2020. Position regulation is good as inferred from groundspeed variation relative to target-position rate. At left, target-position rate is 3 m/s, with the aircraft near the yaw tilt limit at a spanwise airspeed estimate -14 to -15 m/s. At right, target-position rate is 11 m/s, with a spanwise airspeed estimate of -1 to -3 m/s.

On 5 January 2021, moving-target tracking was tested with the INDI rotation controller (and standard position controller) for the 97 conditions outlined in Table 5.3. Figures 5.13 and 5.14 show the tradeoff between regulation performance and actuator activity made by varying control effectiveness parameters with cyclic AUTHORITY COMP set to OFF. Use of AUTHORITY COMP OFF avoids the FLOW MODEL vulnerability to airspeed estimation errors.

Figure 5.13 shows the largest cyclic command activity amplitudes observed during moving-target tracking for each of three different pitch-yaw control effectiveness g_y, g_z values with AUTHORITY COMP OFF (and one with FLOW MODEL). The maximum amplitude for *off-0.7* is similar to or slightly smaller (better) than that for *flow-0.9*. As expected, the maximum

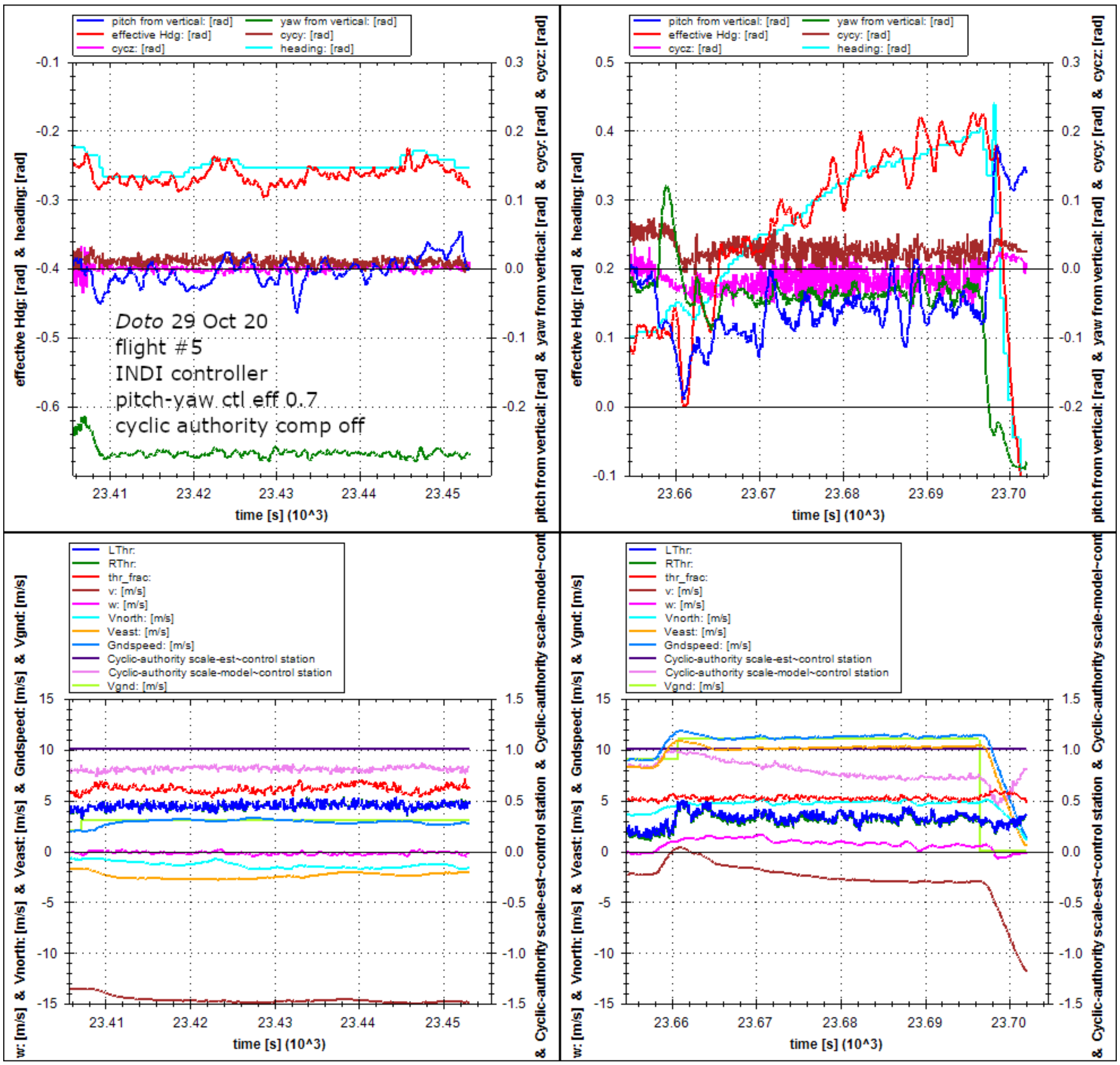


Figure 5.12: Flight moving-target tracking in 12 m/s of wind. In the lower plots, the brown trace is the spanwise airspeed estimate, light green is target-position rate, and azure is aircraft groundspeed.

Table 5.3: Moving-target tracking flight tests with INDI rotation controller and standard position controller on 5 January 2021.

Cyclic AUTHORITY COMP mode	Pitch-yaw control effectiveness g_y, g_z	Number of target-position rates	Spanwise airspeed estimate range, m/s
FLOW MODEL	0.7	1	-8 to -6
	0.8	15	-14 to 14
	0.9	14	-15 to 13
	1.0	14	-15 to 14
OFF	0.5	1	-8 to -7
	0.6	14	-14 to 14
	0.7	14	-15 to 15
	0.8	14	-14 to 15
	0.9	10	-15 to 6

for *off-0.7* is smaller than for *off-0.6* and larger than for *off-0.8*. Note that in some cases the maxima occur at different speeds.

Figure 5.14 shows the worst-damped pitch and yaw tilt oscillations observed during moving-target tracking for each of three different pitch-yaw control effectiveness g_y, g_z values with AUTHORITY COMP OFF (and one with FLOW MODEL). The worst oscillation for *off-0.7* had smaller amplitude than for *flow-0.9*. As expected, the worst for *off-0.7* had worse damping than for *off-0.6* and better damping than for *off-0.8*.

Use of AUTHORITY COMP OFF seems to avoid worst-case performance, e.g. during moving-target tracking *off-0.7* had both similar or better worst-case-speed cyclic activity (Figure 5.13) and better-damped worst-case-speed "undercontrol" oscillations (Figure 5.14) than *flow-0.9*. Note that FLOW MODEL authority (with averaging for airspeed uncertainty)

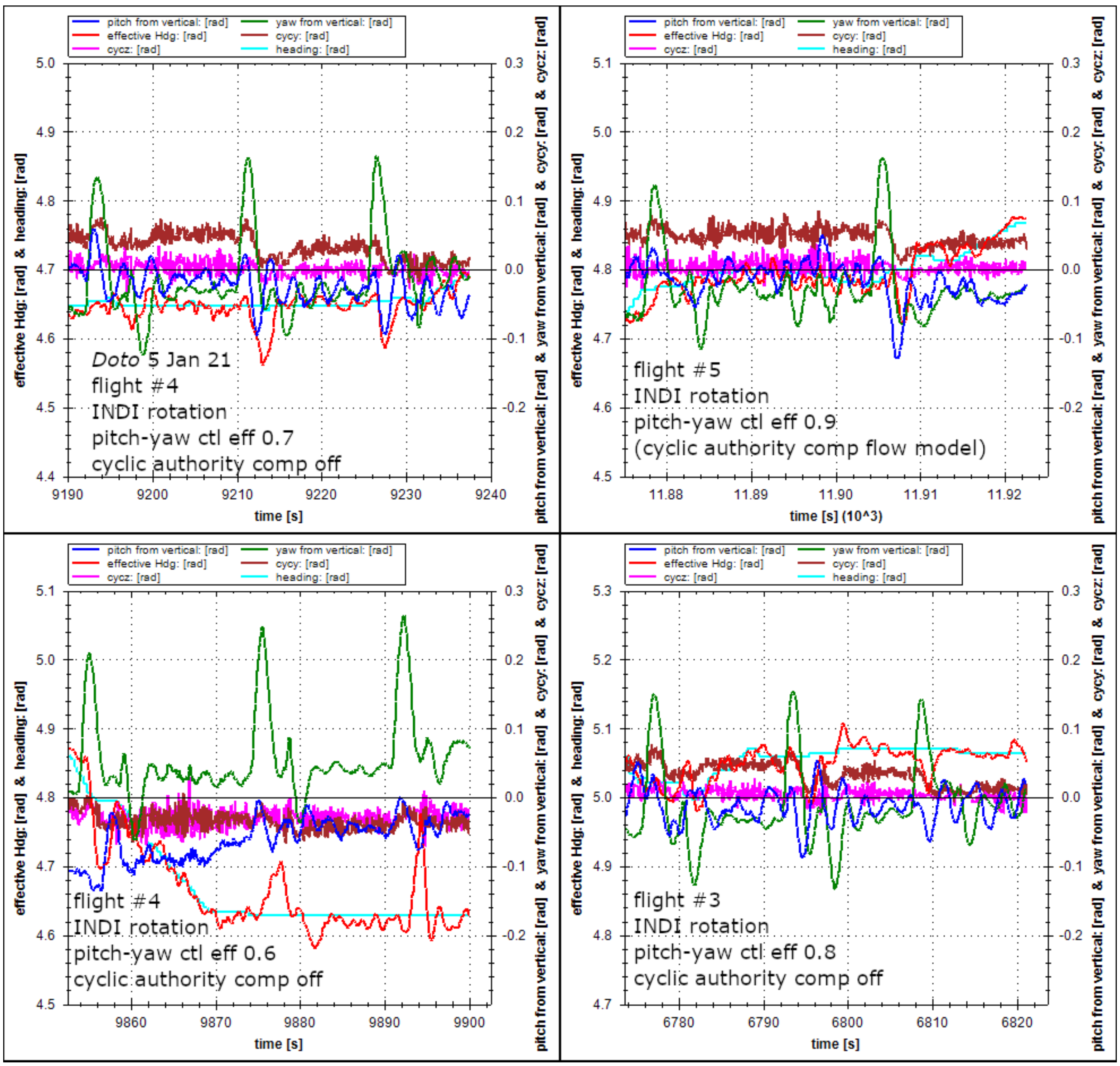


Figure 5.13: Flight moving-target tracking with highest actuator activity for each of four AUTHORITY COMP and pitch-yaw control effectiveness g_y, g_z combinations.

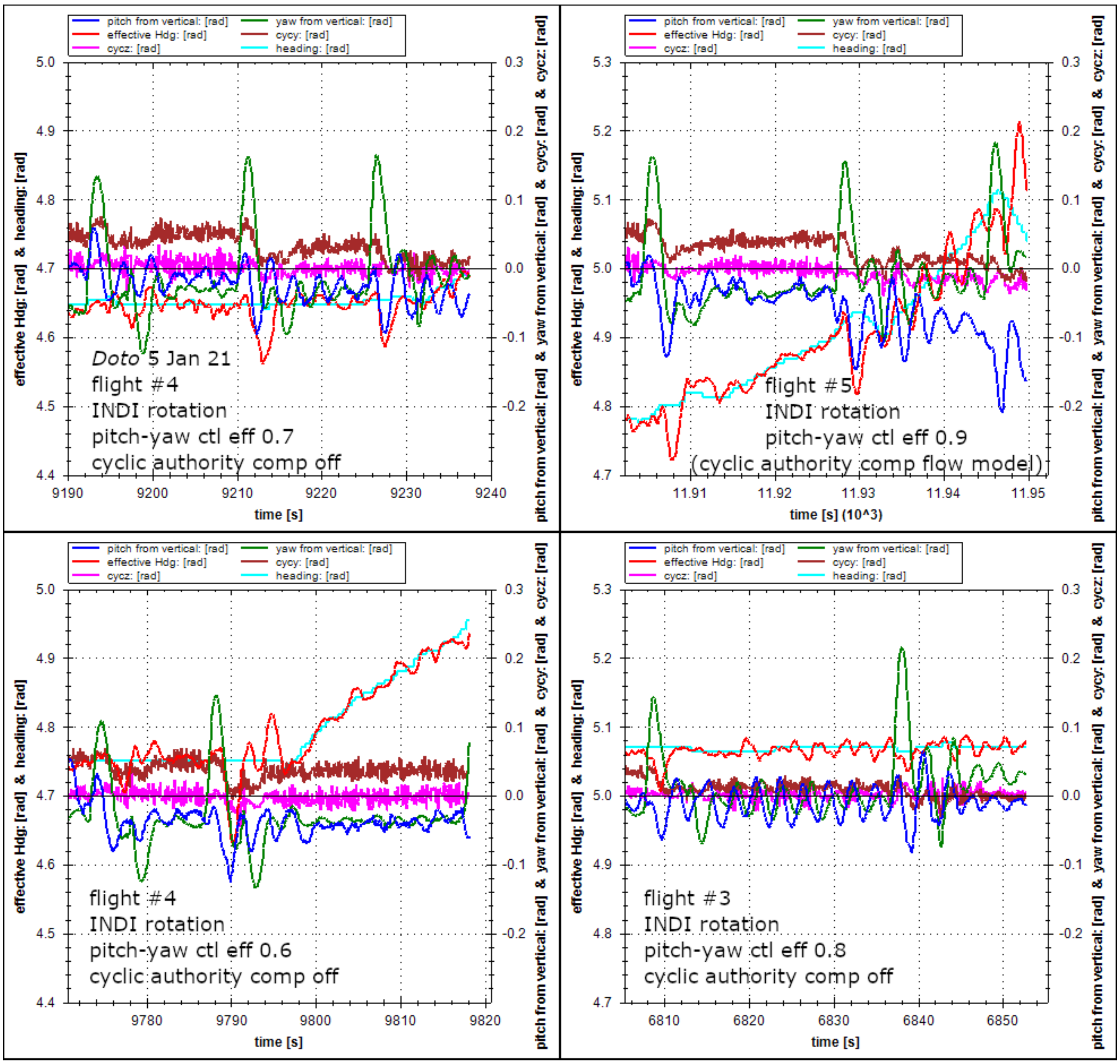


Figure 5.14: Flight moving-target tracking with worst-damped pitch and yaw tilt oscillations for each of four AUTHORITY COMP and pitch-yaw control effectiveness g_y, g_z combinations.

Table 5.4: Translational INDI controller parameters for flight testing.

Category	Parameter	Value
PD gains	natural frequencies $\omega_{n_y}, \omega_{n_z}$	$0.4\,\mathrm{rad/s}$
	damping ratios ζ_y, ζ_z	1.0
Control effectiveness	e_y, e_z	1.0
Acceleration and thrust estimate filters	time constant τ_f	$0.24\,\mathrm{s}$

varies between about 0.6 and 1.0, so *off-0.7* is within the total effectiveness scales of *flow-0.8* and *flow-1.0*.

Figures 5.15 and 5.16 show the tradeoff between regulation performance and actuator activity made by varying control effectiveness parameters with cyclic AUTHORITY COMP set to FLOW MODEL.

Figure 5.15 shows the largest cyclic command activity amplitudes observed during moving-target tracking for each of three different pitch-yaw control effectiveness g_y, g_z values with FLOW MODEL AUTHORITY COMP. As expected, the maximum amplitude for *flow-0.9* is smaller than for *flow-0.8* and larger than for *flow-1.0*.

Figure 5.16 shows the worst-damped pitch and yaw tilt oscillations observed during moving-target tracking for each of three different pitch-yaw control effectiveness g_y, g_z values with FLOW MODEL AUTHORITY COMP. The worst oscillation for *flow-0.9* had similar damping to that for *flow-0.8* and better damping than for *flow-1.0*.

5.4.2 Translational INDI Flight Results

Except where specified otherwise, translational INDI controller parameters for flight testing are as in Table 5.4. The y and z natural frequencies $\omega_{n_y}, \omega_{n_z}$ are conservative (i.e. small) values. The INDI rotation controller was used for the INDI position controller tests.

In initial flight tests, position regulation was worse with the INDI position controller than

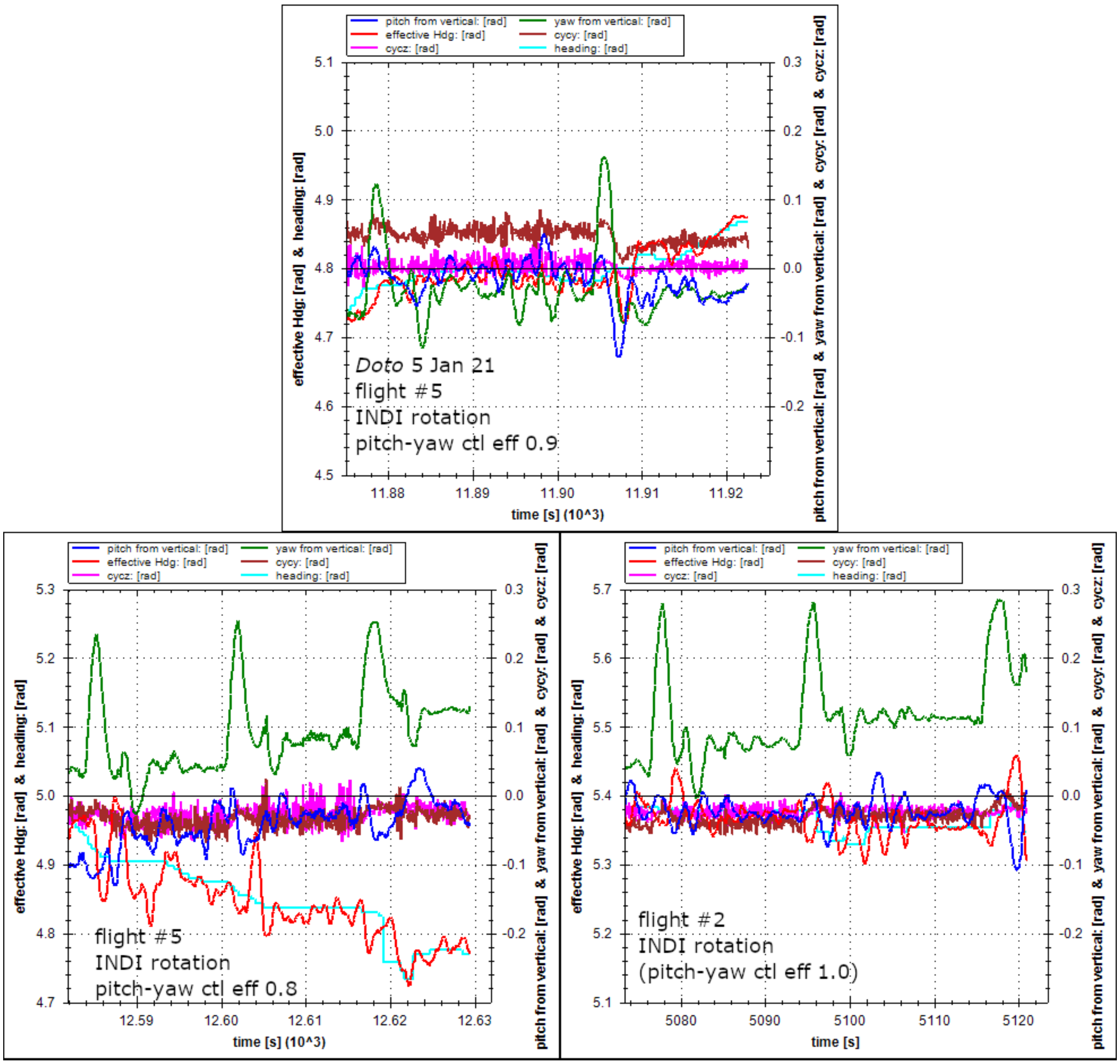


Figure 5.15: Flight moving-target tracking with highest actuator activity for each of three pitch-yaw control effectiveness g_y, g_z values and FLOW MODEL AUTHORITY COMP.

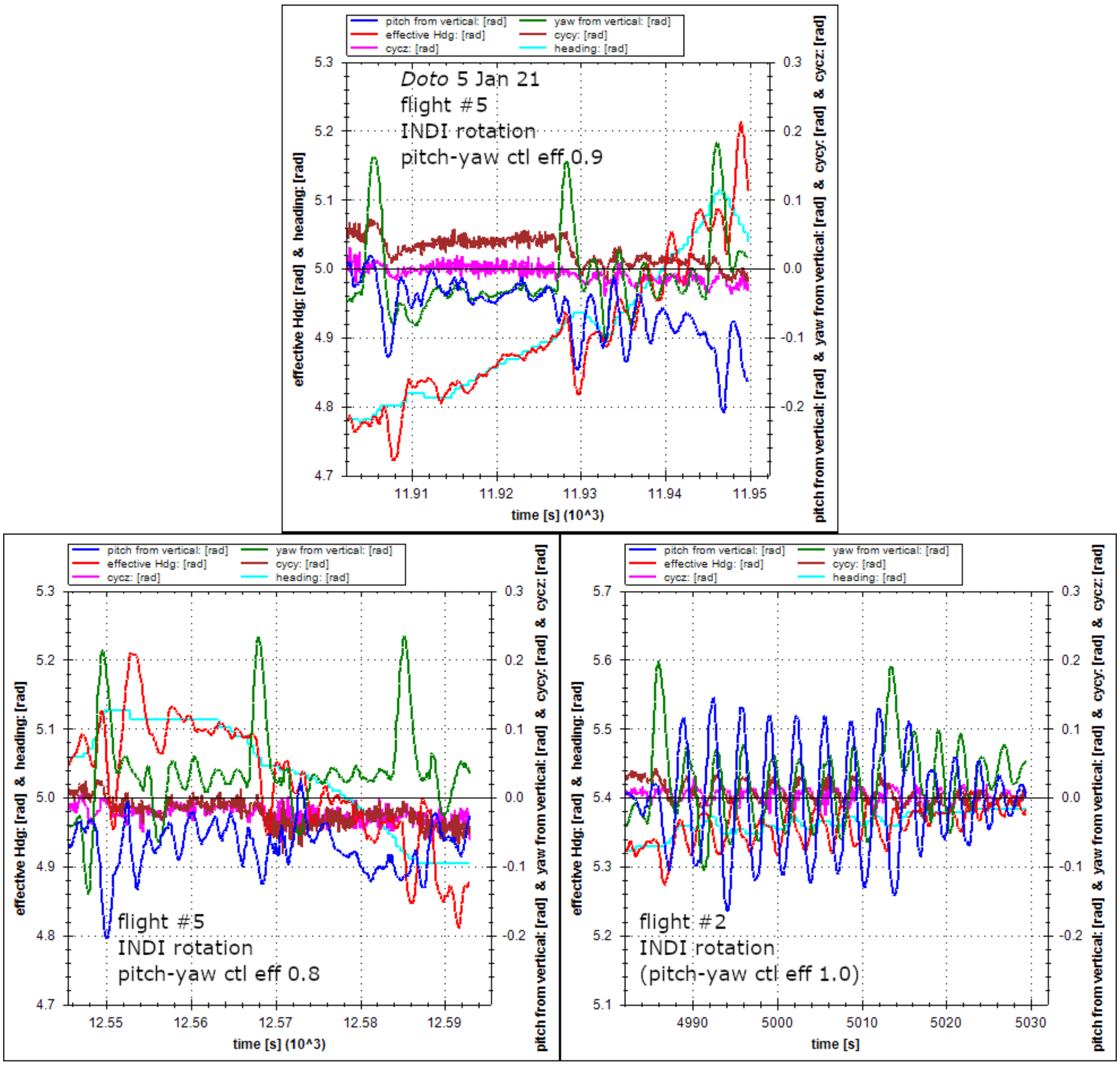


Figure 5.16: Flight moving-target tracking with worst-damped pitch and yaw tilt oscillations for each of three pitch-yaw control effectiveness g_y, g_z values and FLOW MODEL AUTHORITY COMP.

with the standard position controller (Figure 5.22). Position slowly wandered with the INDI position controller, primarily in the body z axis direction (Figures 5.17 and 5.18). There was also an oscillation during a short climb with the INDI position controller (Figure 5.20).

Figure 5.17 shows position slowly wandering in the body y axis direction and especially in the z axis direction after the INDI position controller is enabled. The wandering is confirmed by centimeter-accurate differential GPS. The y axis position jump at 2235 s is a commanded position step.

Wandering with the INDI position controller was reduced after y and z natural frequencies $\omega_{n_y}, \omega_{n_z}$ were doubled to $\omega_{n_y} = \omega_{n_z} = 0.8\,\mathrm{rad/s}$, e.g. from Figure 5.17 to Figure 5.18. In Figure 5.18, the yaw tilt and y axis position oscillations indicate that y natural frequency ω_{n_y} is probably too high. The position jumps are commanded position steps.

The responses to position step commands were similar with an INDI position controller filter time constant of $\tau_f = 0.48\,\mathrm{s}$ and $\tau_f = 0.12\,\mathrm{s}$ (Figure 5.19). Damping was reduced with the shorter time constant.

Figure 5.20 shows an oscillation during a short climb with the INDI position controller.

Figure 5.21 shows moving-target tracking at target-position rates of 4 and 6 m/s resulting in oscillations with both the standard position controller and with the INDI position controller. This was near zero tilt and spanwise airspeed estimates of -2 to 1 m/s. Note the larger heading command change and decrease in high frequency cyclic command activity with the INDI position controller. For the INDI position controller, the oscillation was worse with y and z natural frequencies increased to $\omega_{n_y} = \omega_{n_z} = 0.5\,\mathrm{rad/s}$ (not shown).

Figure 5.22 shows moving-target tracking with AUTHORITY COMP OFF and pitch-yaw control effectiveness $g_y = g_z = 0.7$ for conditions similar to those in Figure 5.21 (e.g. near zero tilt). Oscillations with the standard position controller are substantially reduced compared to Figure 5.21 but are only somewhat reduced with the INDI position controller. Note that the lower plots have slightly different scales. For the standard position controller (left), the position regulation improvement relative to Figure 5.21 came at the cost of increased high frequency cyclic command activity. For the INDI position controller, the oscillation was

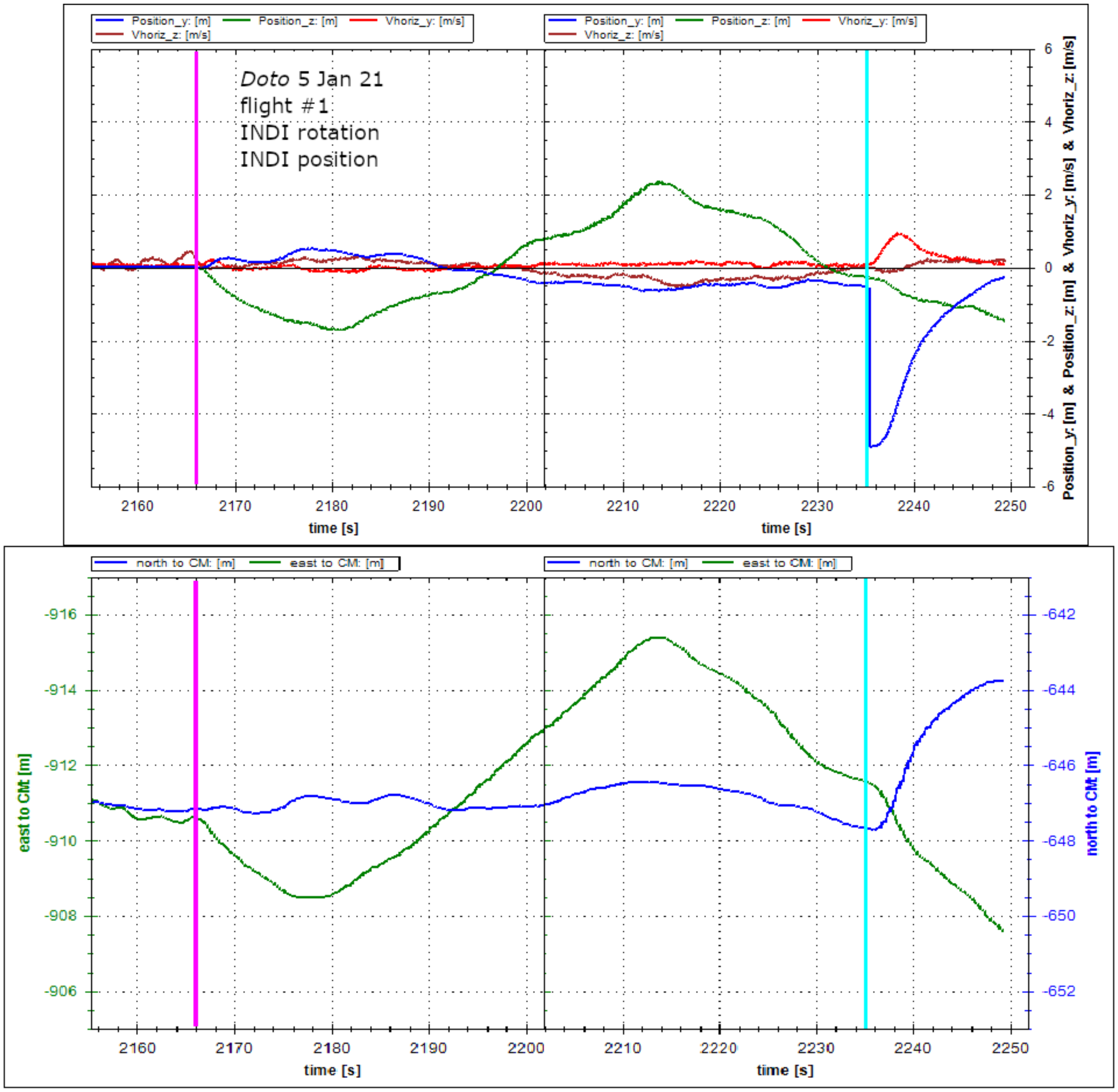


Figure 5.17: Flight INDI position controller enabling and command step. The magenta line indicates INDI position controller enable and the cyan line indicates the position command step. In the upper plots, the blue trace is horizontal position relative to target in the body y axis direction and green is the same in the body z axis direction. In the lower plots, the blue trace is centimeter-accurate differential GPS position in the north direction, and green is differential GPS position in the east direction. Aircraft heading is 5.05 rad.

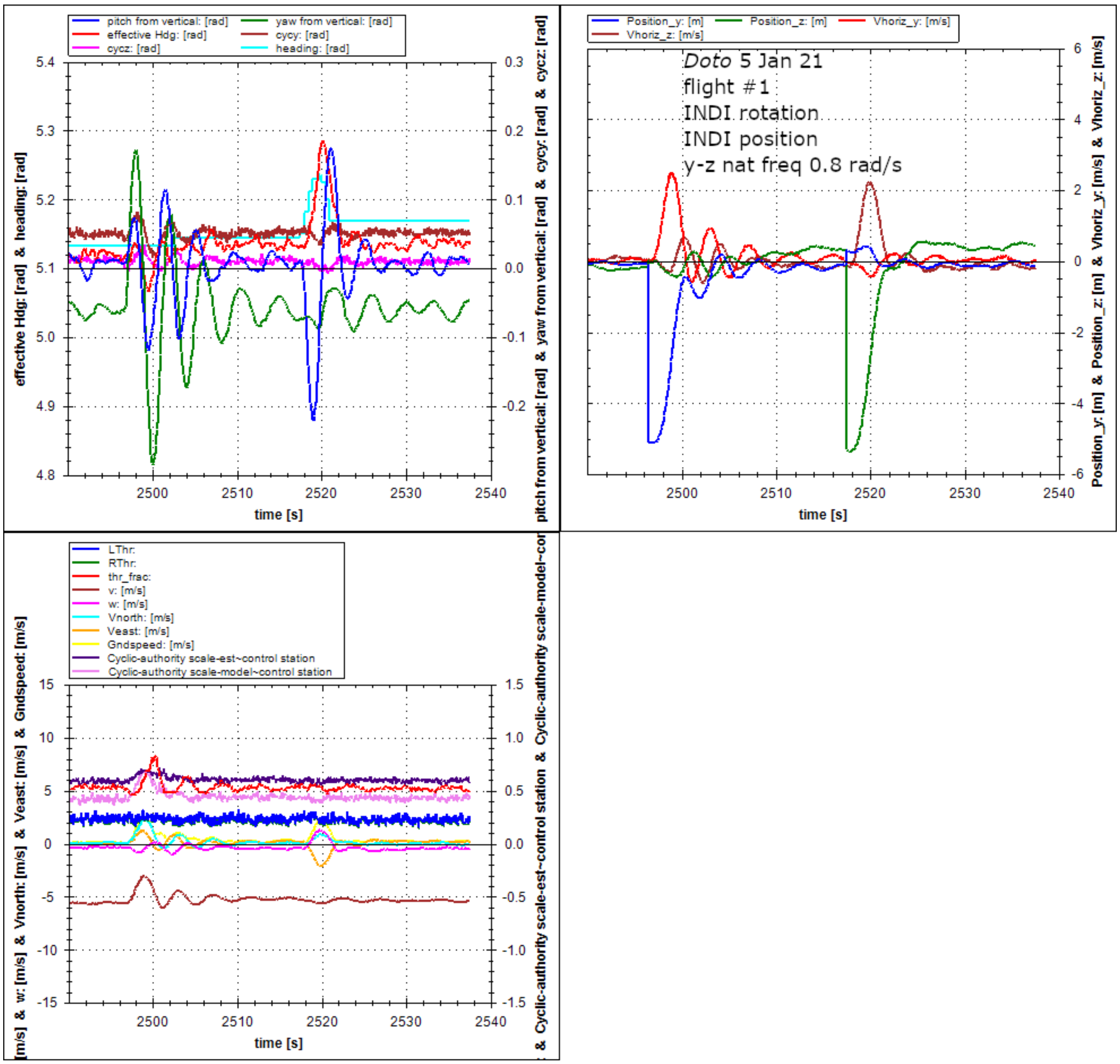


Figure 5.18: Flight INDI position controller command steps with y and z natural frequencies $\omega_{n_y} = \omega_{n_z} = 0.8\,\text{rad/s}$. In the upper-right plot, the blue trace is horizontal position relative to target in the body y axis direction and green is the same in the body z axis direction. In the upper-left plot, the blue trace is pitch angle from vertical and green is yaw from vertical.

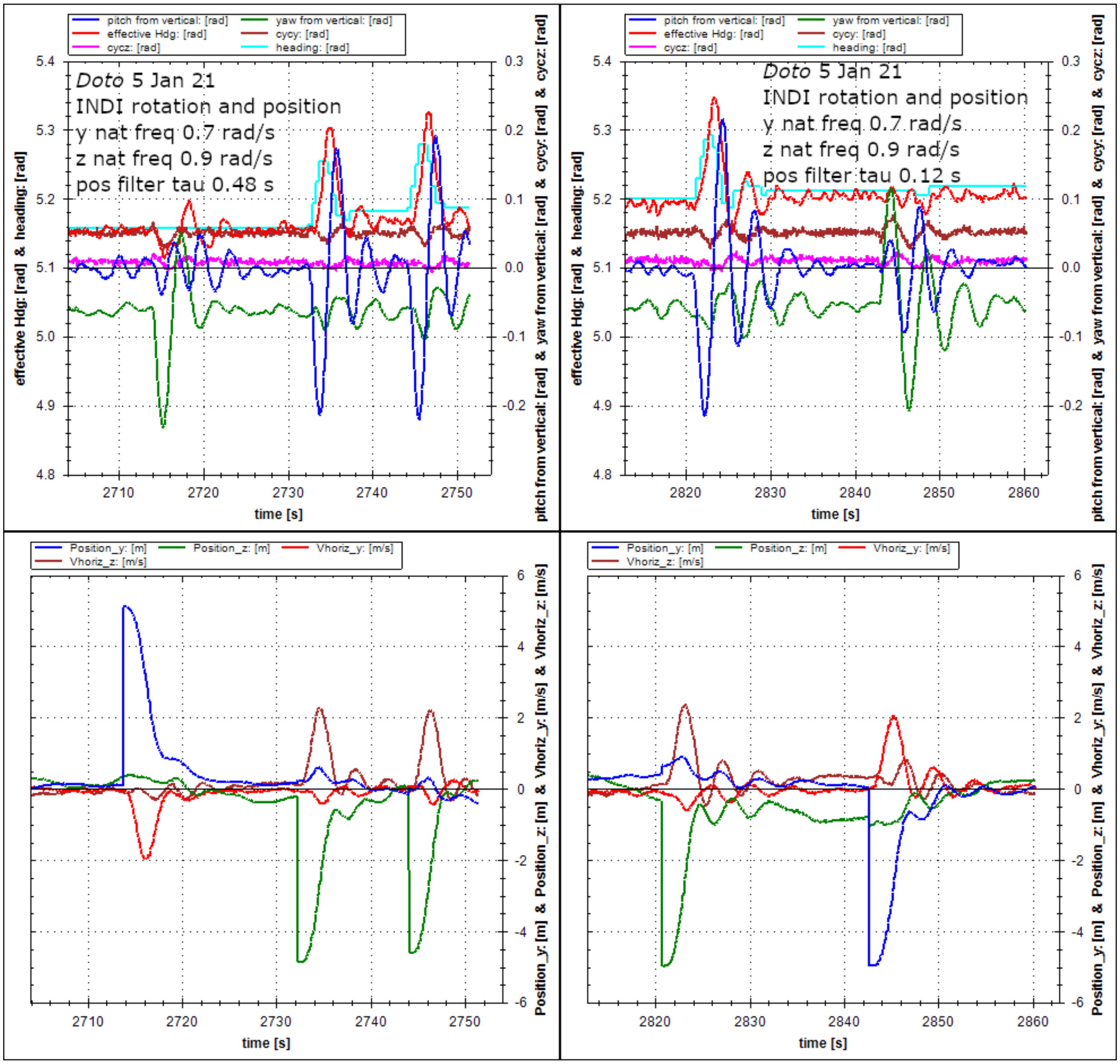


Figure 5.19: Flight INDI position controller command steps for acceleration and thrust estimate filters time constant $\tau_f = 0.48\,\mathrm{s}$ (left) and $\tau_f = 0.12\,\mathrm{s}$ (right). In the upper plots, the blue trace is pitch angle from vertical and green is yaw from vertical. In the lower plots, the blue trace is horizontal position relative to target in the body y axis direction and green is the same in the body z axis direction.

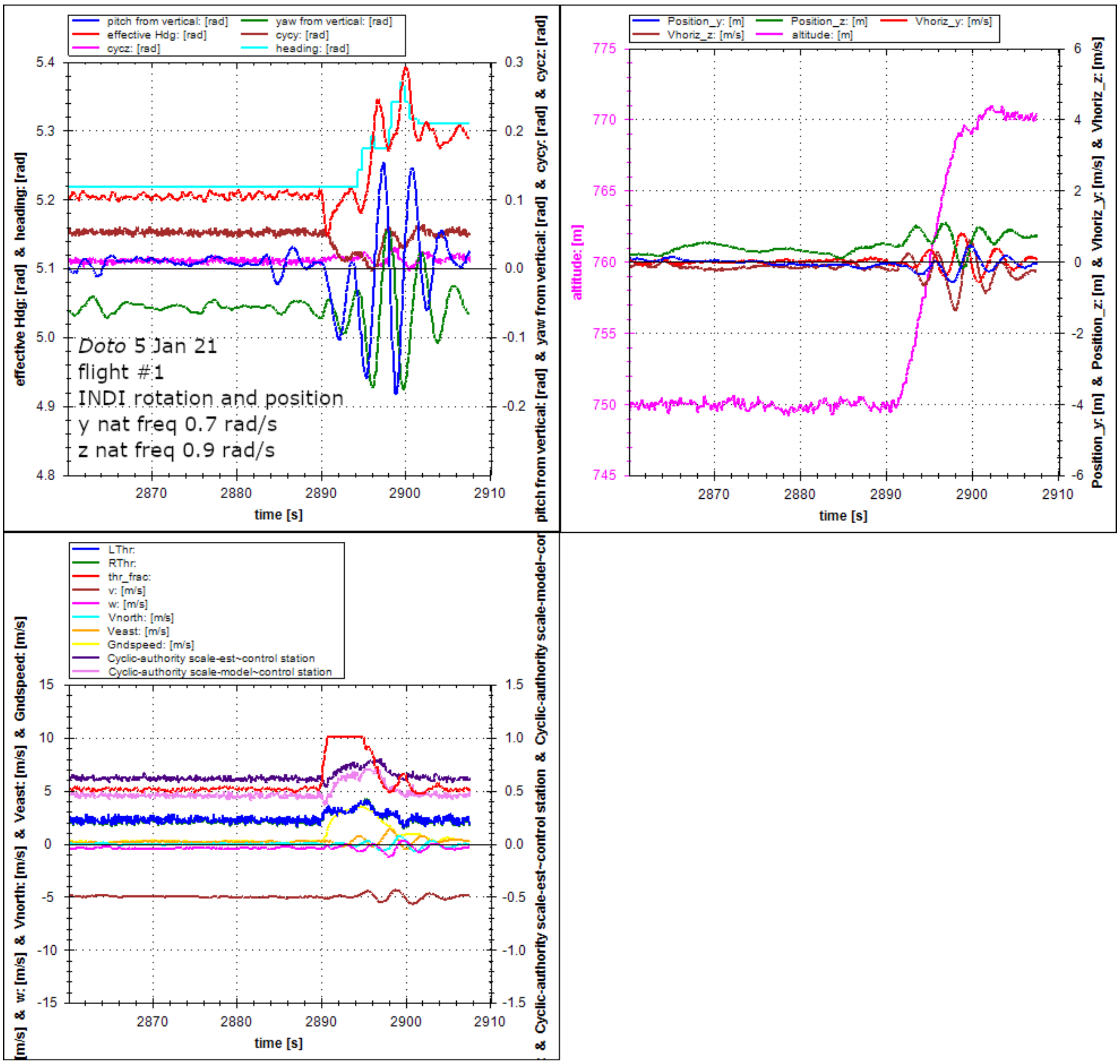


Figure 5.20: Flight INDI position controller regulation during vertical climb. In the upper-right plot, the magenta trace is altitude, blue is horizontal position relative to target in the body y axis direction, and green is the same in the body z axis direction. In the upper-left plot, the blue trace is pitch angle from vertical and green is yaw from vertical.

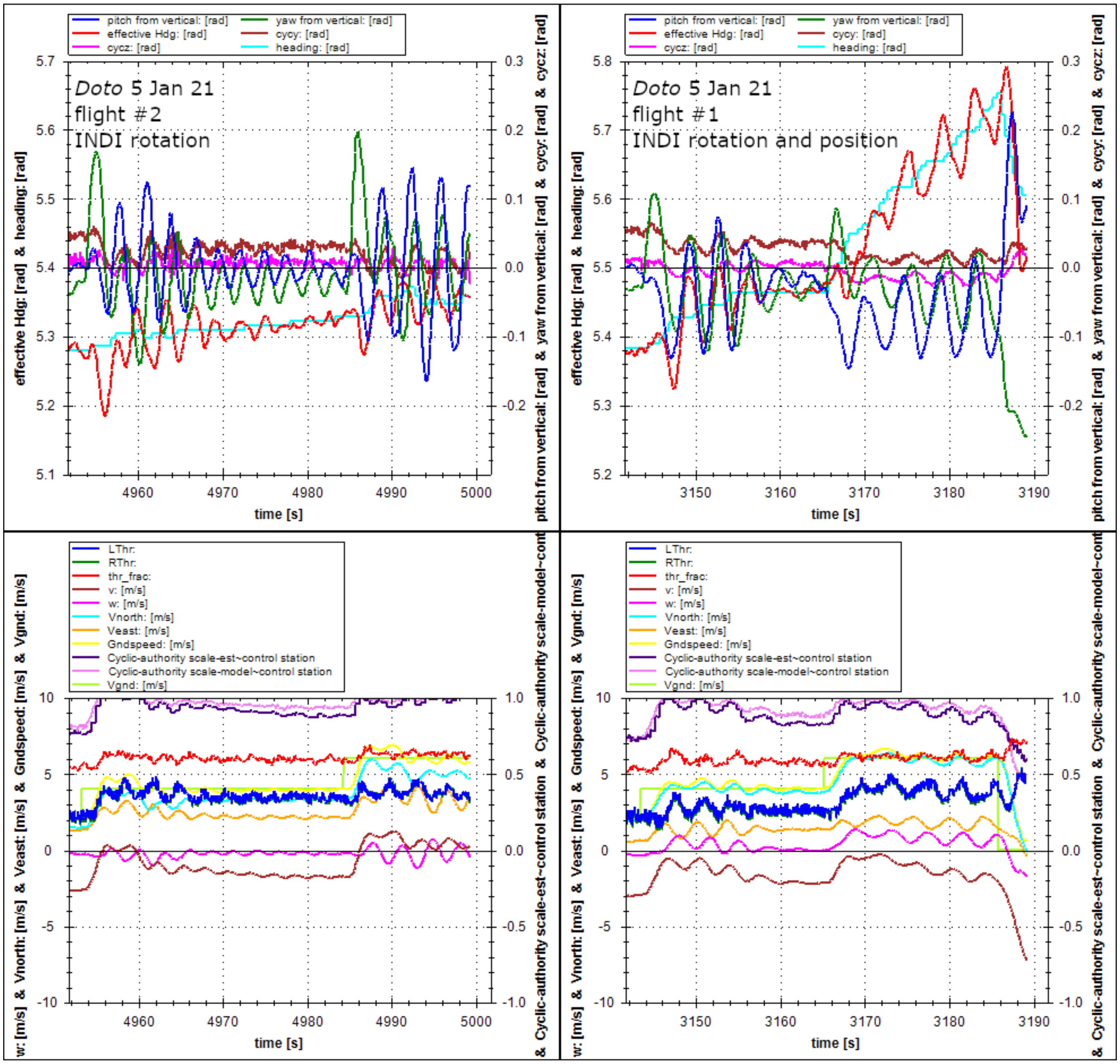


Figure 5.21: Flight moving-target tracking for standard position controller (left) and INDI position controller (right) in 7 m/s of wind. In the upper plots, the blue trace is pitch angle from vertical, green is yaw from vertical, cyan is heading command, and red is heading. The upper-plot brown and magenta traces are cyclic commands. In the lower plots, the light-green trace is target-position rate and brown is the spanwise airspeed estimate.

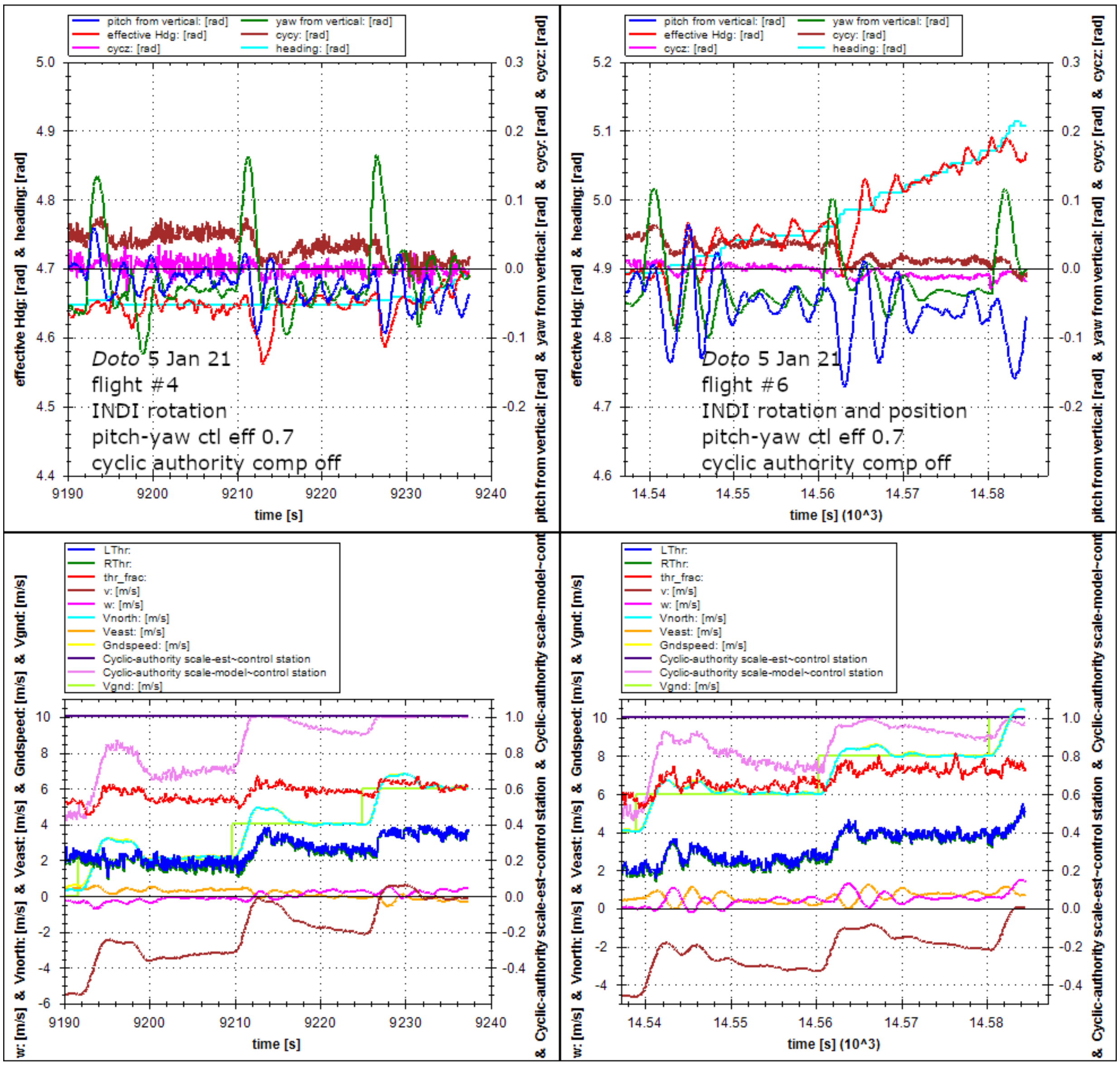


Figure 5.22: Flight moving-target tracking for standard position controller (left) and INDI position controller (right) with AUTHORITY COMP OFF and pitch-yaw control effectiveness $g_y = g_z = 0.7$.

stronger with y and z natural frequencies increased to $\omega_{n_y} = \omega_{n_z} = 0.6\,\mathrm{rad/s}$ (not shown).

5.5 Conclusions

A simple INDI angular acceleration controller combined with attitude and rotation rate feedback regulated attitude well in flight of Flexrotor, i.e. substantially better than the standard controller (e.g. Figure 5.7). Good tilt attitude regulation can be used to increase climb rate during climbout, increasing takeoff weight or safety margin. Good attitude regulation also aids translational control, e.g. speeding regulation (Figure 5.9) or reducing oscillations (Figure 5.22). Furthermore, consider a rotorcraft regulating position when it is disturbed by a wind gust. Good attitude regulation and disturbance rejection aid position regulation performance both by providing an effective thrust tilt actuator and by reducing the attitude disturbance due to the wind gust. For practical rotorcraft operations, improved position regulation and tracking can improve safety margin; relax wind, gust, or deck motion limits; and reduce required approach or departure volumes and deck area.

The INDI position controller tested here needs further work before it is as robust as a PID controller or performs as well as the standard Flexrotor trajectory tracker. Use of the yawed NED frame allowed controller gains to vary with high-drag and low-drag body axes, and could be useful for other applications such as helicopters.

The analysis and experiments in this chapter demonstrate that INDI control laws are sufficiently robust to sensor vibration and controller parameter changes (including actuator models) to allow safe testing and tuning for applications such as Flexrotor. In particular, performance degrades gracefully with long filter time constants and the actuator model does not have to be accurately known for low-bandwidth control. Thus, the concerns with INDI for rotorcraft control raised earlier (i.e. sensor noise, actuator model errors, and time scale separation) have been addressed. Note that a time-accurate actuator model or measurement would be needed for online control effectiveness estimation as used by Smeur et al. (2016) [72].

Chapter 6

CONCLUSIONS

The control problems of interest were investigated culminating in successful flight test of INDI angular acceleration control on the Flexrotor tailsitter. For rotorcraft, induced velocity modeling at least at the level of Pitt-Peters [64] is crucial for capturing behavior of interest at low airspeeds. This model indicates a control effectiveness gradient reversal that is little discussed in the literature. A computationally efficient algorithm has been provided for applying Pitt-Peters effects, enabling applications such as online use on low-power processors. A second efficient calculation of control effectiveness variation is provided in the form of corrected rotor moment lift deficiency functions. The Pitt-Peters model was used in controller simulation testing, providing consistency in controller behavior between simulation and flight.

The variation in trim at low airspeed is a problem for Flexrotor tilt attitude control. Algorithms were developed for online trim estimation and feedforward, but this approach remains dependent on at least one accurate three-dimensional airspeed sensor. Adaptive control based on dynamic inversion was applied and extended to this application, but the resulting controller is dependent on learning rates and airspeed estimation errors.

Incremental nonlinear dynamic inversion (INDI) controllers can by contrast be simple and robust to parameter changes, aiding controller implementation, debugging, and tuning. INDI translational acceleration control was extended to allow the controller to vary for different body tilt axes (e.g. to handle low-drag and high-drag axes on a helicopter). INDI angular acceleration control was applied, analyzed, and successfully tested in flight, addressing all of the problems of interest for this research despite not using a rotor model, trim model, or airspeed estimate. Furthermore, concerns such as sensor noise, actuator model errors, and time scale separation were addressed, indicating the potential for INDI control in numerous applications.

6.1 Future Work

For modeling and system identification, flight tests could be conducted to determine how cyclic control effectiveness varies with airspeed and rotor azimuth, e.g. whether it is consistent with the model. Cyclic control effectiveness variation could also be modeled for different aircraft (e.g. helicopters) and with different models, e.g. advanced dynamic inflow and free wake.

An onboard anemometer (i.e. three-dimensional airspeed sensor) could be flown and tested with control algorithms for gust rejection, e.g. the trim feedforward controller of the Trim Feedforward chapter.

For INDI control, further testing and comparisons to other control methods could be made, e.g. the continuous second-order sliding mode controller of Halbe and Oza [31]. Multi-objective parametric optimization as in Tischler et al. [77] could be used to meet performance and stability specifications. A better roll actuator (i.e. wingtip thruster) model could be developed to handle thruster reversal and improve the performance (e.g. relative to (Figure 5.8). For translational INDI control, further development is needed to address issues such as position wandering and oscillation in climb.

For INDI rotational control, control effectiveness could be modeled using Pitt-Peters lift deficiency functions. While this extension increases dependence on models and airspeed estimates, it could improve the control effectiveness estimate for typical airspeed estimation errors. Moreover, stability should be maintained even for worst case errors.

INDI rotational control will be further tested on Flexrotor, e.g. for approach and landing in various winds. The goal is routine operations across the flight envelope including robust landings on a small boat in strong, gusty wind and stormy seas.

BIBLIOGRAPHY

[1] Aerovel Corporation. About the Flexrotor unmanned aerial system | Aerovel. https://aerovel.com/flexrotor/; accessed 1-October-2018 and 4-March-2021.

[2] Aerovel Corporation. Aerovel Flexrotor flies for 32 consecutive hours | Aerovel. https://aerovel.com/flexrotor-vtol-endurance/; accessed 7-April-2020.

[3] Aerovel Corporation. Aerovel software, September 2019.

[4] PEO Aviation. Future tactical unmanned aircraft system (FTUAS), September 2020. https://www.army.mil/article/239411; accessed 11-February-2021.

[5] Lindsay Barbieri, Stephan Kral, Sean Bailey, Amy Frazier, Jamey Jacob, Joachim Reuder, David Brus, Phillip Chilson, Christopher Crick, Carrick Detweiler, and et al. Intercomparison of small unmanned aircraft system (sUAS) measurements for atmospheric science during the lapse-rate campaign. *Sensors*, 19(9):2179, May 2019.

[6] F Binz and D Moormann. Actuator modelling for attitude control using incremental nonlinear dynamic inversion. *International Journal of Micro Air Vehicles*, 12:1756829320961925, 2020.

[7] Anthony J Calise and Rolf T Rysdyk. Nonlinear adaptive flight control using neural networks. *IEEE Control Systems Magazine*, 18(6):14–25, 1998.

[8] Liyu Cao and Howard Schwartz. A directional forgetting algorithm based on the decomposition of the information matrix. *Automatica*, 36(11):1725–1731, 2000.

[9] G.D. Carico, R. Fang, R.S. Finch, W.P. Geyer Jr., H.W. Krijns, and K. Long. Helicopter/ship qualification testing. Technical report, Research and Technology Organisation, North Atlantic Treaty Organisation, February 2003.

[10] Robert T N Chen. A survey of nonuniform inflow models for rotorcraft flight dynamics and control applications. Technical memorandum, National Aeronautics and Space Administration (NASA), November 1989.

[11] Girish Chowdhary and Eric N Johnson. Concurrent learning for improved convergence in adaptive flight control. In *AIAA Guidance, Navigation, and Control Conference*, 2010.

[12] Wikimedia Commons. File:MBB Bo 105.png — Wikimedia Commons, the free media repository, 2017. https://commons.wikimedia.org/w/index.php?title=File:MBB_Bo_105.png; accessed 4-March-2021.

[13] William Craig, Derrick Yeo, and Derek A Paley. Dynamics of a rotor-pendulum with a small, stiff propeller in wind. In *ASME 2016 Dynamic Systems and Control Conference*, pages V001T05A001–V001T05A001. American Society of Mechanical Engineers, 2016.

[14] Kumeresan A Danapalasingam, Anders la Cour-Harbo, and Morten Bisgaard. Feedforward control of an autonomous helicopter using trim inputs. In *AIAA Infotech@Aerospace and AIAA Unmanned...Unlimited Conference*, 2009.

[15] Christophe De Wagter, Rick Ruijsink, Ewoud J. J. Smeur, Kevin G. van Hecke, Freek van Tienen, Erik van der Horst, and Bart D. W. Remes. Design, control, and visual navigation of the DelftaCopter VTOL tail-sitter UAV. *Journal of Field Robotics*, 35(6):937–960, 2018.

[16] Prism Defence. Lynx helicopter operating limit development, 2010. https://www.youtube.com/watch?v=bC2XIGMI2kM; accessed 30-October-2018.

[17] James Downs, Ron Prentice, Scott Dalzell, Adam Besachio, Christina M Ivler, Mark B Tischler, and Mohammadreza H Mansur. Control system development and flight test experience with the MQ-8B Fire Scout vertical take-off unmanned aerial vehicle (VTUAV). In *Annual Forum Proceedings-American Helicopter Society*, volume 63, page 566. American Helicopter Society International, Inc., 2007.

[18] Mark E Dreier. *Introduction to Helicopter and Tiltrotor Flight Simulation*. American Institute of Aeronautics and Astronautics, 2018.

[19] Enoch Durbin and Tad McGeer. An airspeed vector sensor for V/STOL aircraft. *Journal of Aircraft*, 19(6):449–455, 1982.

[20] George Ellis. *Control System Design Guide: Using Your Computer to Understand and Diagnose Feedback Controllers*. Academic Press, 2000.

[21] Rutger Engbersen, Brian T McGeer, and Michael Baxter. Tail-sitter aircraft with legged undercarriage foldable to form rear fuselage, April 30 2019. US Patent 10,272,999.

[22] AJ Faulkner and F Buchner. Flight investigations of a helicopter low airspeed estimation system based on measurement of control parameters. In *Sixth European Rotorcraft and Powered Lift Aircraft Forum*, 1980.

[23] Federal Aviation Administration. *Helicopter Flying Handbook*, faa-h-8083-21b edition, 2019.

[24] James S Forrest, Ieuan Owen, Gareth D Padfield, and Steven J Hodge. Ship-helicopter operating limits prediction using piloted flight simulation and time-accurate airwakes. *Journal of Aircraft*, 49(4):1020–1031, 2012.

[25] G H Gaonkar and David A Peters. Review of dynamic inflow modeling for rotorcraft flight dynamics. In *27th Structures, Structural Dynamics and Materials Conference*. American Institute of Aeronautics and Astronautics, 1986.

[26] Gopal H. Gaonkar and David A. Peters. Effectiveness of current dynamic-inflow models in hover and forward flight. *Journal of the American Helicopter Society*, 31(2):47–57, April 1986.

[27] Stefano Geluardi, Frank M. Nieuwenhuizen, Joost Venrooij, Lorenzo Pollini, and Heinrich H. Bülthoff. Frequency domain system identification of a Robinson R44 in hover. *Journal of the American Helicopter Society*, 63(1):1–18, 2018.

[28] Stefano Geluardi, Joost Venrooij, Mario Olivari, Heinrich H. Bülthoff, and Lorenzo Pollini. Transforming civil helicopters into personal aerial vehicles: Modeling, control, and validation. *Journal of Guidance, Control, and Dynamics*, 40(10):2481–2495, 2017.

[29] William B. Greer and Cornel Sultan. Shrinking horizon model predictive control method for helicopter-ship touchdown. *Journal of Guidance, Control, and Dynamics*, 43(5):884–900, 2020.

[30] Håvard Fjær Grip, Wayne Johnson, Carlos Malpica, Daniel P. Scharf, Milan Mandić, Larry Young, Brian Allan, Bérénice Mettler, Miguel San Martin, and Johnny Lam. Modeling and identification of hover flight dynamics for NASA's Mars Helicopter. *Journal of Guidance, Control, and Dynamics*, 43(2):179–194, 2020.

[31] Omkar Halbe and Harshal Oza. Robust continuous finite-time control of a helicopter in turbulence. *IEEE Control Systems Letters*, 5(1):37–42, 2021.

[32] Franklin D. Harris. Articulated rotor blade flapping motion at low advance ratio. *Journal of the American Helicopter Society*, 17(1):41–48, 1972.

[33] JunSoo Hong. A fully nonlinear version of the Pitt-Peters dynamic inflow model in axial flow. In *AHS 69th Annual Forum*. American Helicopter Society International, Inc., May 2013.

[34] Joseph F. Horn, Anthony J. Calise, and J. V. R. Prasad. Flight envelope cueing on a tiltrotor aircraft using neural network limit prediction. *Journal of the American Helicopter Society*, 46(1):23–31, 2001.

[35] J. J. Howlett. UH-60A Black Hawk engineering simulation program. Volume 1: Mathematical model. Contractor report, National Aeronautics and Space Administration (NASA), December 1981.

[36] Insitu. Insitu - Insitu ScanEagle - persistent imagery solution for land and sea, 2018. https://www.insitu.com/information-delivery/unmanned-systems/scaneagle; accessed 17-October-2018.

[37] Petros Ioannou and Barış Fidan. *Adaptive control tutorial.* SIAM, 2006.

[38] Eric N Johnson and Suresh K Kannan. Adaptive trajectory control for autonomous helicopters. *Journal of Guidance, Control, and Dynamics*, 28(3):524–538, 2005.

[39] Wayne Johnson. Milestones in rotorcraft aeromechanics Alexander A. Nikolsky honorary lecture. *Journal of the American Helicopter Society*, 56(3):1–24, 2011.

[40] Wayne Johnson. *Rotorcraft Aeromechanics*, volume 36 of *Cambridge Aerospace Series.* Cambridge University Press, 2013.

[41] Hassan K. Khalil. *Nonlinear Systems.* Prentice Hall, third edition, 2002.

[42] J Gordon Leishman. *Principles of Helicopter Aerodynamics.* Cambridge University Press, 2006.

[43] Anemoment LLC. Features - Anemoment, 2020. https://anemoment.com/features/; accessed 11-April-2020.

[44] MathWorks. Splines - MATLAB & Simulink, 2020. https://www.mathworks.com/help/curvefit/splines.html; accessed 13-April-2020.

[45] Tad McGeer. *Minimum-effort and least-squares fitting.* Aerovel Corporation, February 2008.

[46] Tad McGeer. *Adaptive Control Using Rotor Cyclic.* Aerovel Corporation, September 2017.

[47] Tad McGeer. Flexrotor flight notes for 29 June 2017. Technical report, Aerovel Corporation, July 2017.

[48] Tad McGeer. Flexrotor flight notes for 9 June 2017. Technical report, Aerovel Corporation, June 2017.

[49] Tad McGeer. *Measurement and Modeling of Rotor In-Plane Loads*. Aerovel Corporation, March 2017.

[50] Tad McGeer. *Flexrotor Modeling for Dynamics Analysis and Simulation*. Aerovel Corporation, July 2018.

[51] Tad McGeer. Cydippe hard landing...4 May 2019. Technical report, Aerovel Corporation, May 2019.

[52] Tad McGeer. Flexrotor flight notes for 13 March 2019. Technical report, Aerovel Corporation, March 2019.

[53] Tad McGeer. Flexrotor flight notes for 4 April 2019. Technical report, Aerovel Corporation, April 2019.

[54] Tad McGeer. Flexrotor flight notes for 6 February 2019. Technical report, Aerovel Corporation, February 2019.

[55] Tad McGeer. Flexrotor flight notes for 6 November 2018. Technical report, Aerovel Corporation, January 2019.

[56] Tad McGeer. Flexrotor flight notes for 3 February 2021. Technical report, Aerovel Corporation, February 2021.

[57] Mark C. Newkirk. Process improvements for the AH-64 tail rotor vibration analysis. Master's thesis, Air Force Institute of Technology, 2007.

[58] Nhan T Nguyen. *Model-Reference Adaptive Control: A Primer*. Springer, 2018.

[59] Robert Niemiec and Farhan Gandhi. Effects of inflow model on simulated aeromechanics of a quadrotor helicopter. In *72nd Annual Forum of the American Helicopter Society International*, 2016.

[60] Gareth D Padfield. *Helicopter Flight Dynamics: Including a Treatment of Tiltrotor Aircraft*. John Wiley & Sons, 2018.

[61] Marilena D. Pavel, Perumal Shanthakumaran, Qiping Chu, Olaf Stroosma, Mike Wolfe, and Harm Cazemier. Incremental nonlinear dynamic inversion for the Apache AH-64 helicopter control. *Journal of the American Helicopter Society*, 65(2):1–16, 2020.

[62] David A. Peters. How dynamic inflow survives in the competitive world of rotorcraft aerodynamics. *Journal of the American Helicopter Society*, 54(1):11001, January 2009.

[63] David A. Peters and Ninh HaQuang. Technical note: Dynamic inflow for practical applications. *Journal of the American Helicopter Society*, 33(4):64–68, October 1988.

[64] Dale M Pitt and David A Peters. Theoretical prediction of dynamic-inflow derivatives. 1980. Published in Vertica, Vol. 5, (1), March 1981, pp. 21-34.

[65] Nidhish Raj, Ravi N. Banavar, Abhishek, and Mangal Kothari. Robust attitude tracking for aerobatic helicopters: A geometric approach. *IEEE Transactions on Control Systems Technology*, 29(1):150–164, 2021.

[66] Robinson Helicopter Company. Robinson Helicopter Company. https://robinsonheli.com/; accessed 15-January-2020.

[67] Andrew J. Ruddell. Advancing Blade Concept (ABC) development. *Journal of the American Helicopter Society*, 22(1):13–23, 1977.

[68] Robert Shorten, Andreas Schütte, and A D Fagan. Robust parameter tracking through regional forgetting. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1440–1443. IEEE, 1995.

[69] P. Simplício, M.D. Pavel, E. van Kampen, and Q.P. Chu. An acceleration measurements-based approach for helicopter nonlinear flight control using incremental nonlinear dynamic inversion. *Control Engineering Practice*, 21(8):1065–1077, 2013.

[70] B. J. M. M. Slinger. Attitude control of a small helicopter UAV using incremental nonlinear dynamic inversion. Master's thesis, Delft University of Technology, 2016.

[71] E.J.J. Smeur, G.C.H.E. de Croon, and Q. Chu. Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection. *Control Engineering Practice*, 73:79 – 90, 2018.

[72] Ewoud J. J. Smeur, Qiping Chu, and Guido C. H. E. de Croon. Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles. *Journal of Guidance, Control, and Dynamics*, 39(3):450–461, 2016.

[73] Gregory L. Soneson, Joseph F. Horn, and Albert Zheng. Simulation testing of advanced response types for ship-based rotorcraft. *Journal of the American Helicopter Society*, 61(3):1–13, 2016.

[74] Brian L Stevens, Frank L Lewis, and Eric N Johnson. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems.* John Wiley & Sons, 2016.

[75] Textron Systems. Aerosonde | Textron Systems, 2018. https://www.textronsystems.com/what-we-do/unmanned-systems/aerosonde; accessed 17-October-2018.

[76] E. Tal and S. Karaman. Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness. *IEEE Transactions on Control Systems Technology*, 2020.

[77] Mark B Tischler, Tom Berger, Christina M Ivler, Mohammadreza H Mansur, Kenny K Cheung, and Jonathan Y Soong. *Practical Methods for Aircraft and Rotorcraft Flight Control Design: An Optimization-Based Approach.* American Institute of Aeronautics and Astronautics, Incorporated, 2017.