

AOE X224 Homework #2

Due: September 19, 2022 at 8PM

Note: You must submit any computational scripts along with your solutions to receive credit.

Problem 1. Obtain SIDPAC from the following URL:

<https://software.nasa.gov/software/LAR-16100-1>

Once you've downloaded the software and added the subdirectory to your Matlab path, run the command `fly` and begin the simulated flight. Take a screenshot of the scene and include it as your response to this problem.

Problem 2. Create a Matlab simulation of the nonlinear flight dynamic model developed in [1], which is appended to this assignment. (You may omit the actuator model that accounts for time delay and dynamics.) Append your script as your response to this problem.

Problem 3. Using the Matlab simulation developed in Problem 2, simulate a steady flight condition corresponding constant-altitude, wings-level flight at 45 ft/s. Provide plots of all twelve state variables and all four inputs versus time over a 10 second window.

Problem 4. Using the Matlab simulation developed in Problem 2, and starting from the steady flight condition determined in Problem 3, apply a square doublet with a total duration of 1 second. Provide plots of all twelve state variables and all four inputs versus time over a 10 second window.

Problem 5. (Graduate Students) Incorporate the actuator dynamic model developed in [1] into your model and repeat the simulation of Problem 4.

References

- [1] J L Gresham, J-m W Fahmi, b M Simmons, J W Hopwood, W Foster, and C A Woolsey. Flight test approach for modeling and control law validation for unmanned aircraft. In *AIAA SciTech 2022 Forum*, 2022.



Flight Test Approach for Modeling and Control Law Validation for Unmanned Aircraft

James L. Gresham*, Jean-Michel W. Fahmi†, Benjamin M. Simmons‡, Jeremy W. Hopwood§, Joseph W. Foster¶, and Craig A. Woolsey||
Virginia Tech, Blacksburg, Virginia 24061

A nonlinear energy-based control law was flight tested on a small, fixed-wing unmanned aircraft. This paper summarizes the selected aircraft, instrumentation system, data processing techniques, system identification methods, and the control laws that were implemented. The flight test campaign used a build up approach with increasingly complex computer generated system identification excitations and then increasingly complex control laws. This build up approach allowed the team to overcome technical challenges in a progressive manner and then finally test an advanced nonlinear control law in flight. Automated multistep, frequency sweep, and multisine excitation inputs were implemented for system identification. System identification methods were leveraged to develop linear and nonlinear aerodynamic models. A servoactuator model was developed from ground test data and data processing techniques were used to condition the flight test data for analysis. A proportional-derivative attitude commanded system was implemented and tuned using pilot comments, without the use of an *a priori* model. A linear quadratic regulator was tuned using the linear aerodynamic model, and refined during flight tests to improve handling qualities. A port-Hamiltonian energy-based nonlinear control law was tuned in simulation using the nonlinear model, and gains refined during flight tests to improve directional tracking and perturbation response. Implementation techniques for automated system identification maneuvers, as well as feedback control using a Pixhawk and Raspberry Pi co-computer are documented and made available by means of a publicly accessible web repository. Flight test results illustrate the utility of the experimental data collection and analysis methods for testing advanced flight control schemes.

Nomenclature

a_x, a_y, a_z	= body-axis translational acceleration, ft/s ² or g
C_l, C_m, C_n	= body-axis aerodynamic moment coefficients
C_x, C_y, C_z	= body-axis aerodynamic force coefficients
g	= gravitational acceleration, ft/s ²
p, q, r	= body-axis angular velocity components, rad/s or deg/s
$\hat{p}, \hat{q}, \hat{r}$	= non-dimensional body-axis angular velocity components
u, v, w	= body-axis translational velocity components, ft/s
V_t	= true airspeed, ft/s
α	= angle of attack, rad or deg
β	= angle of sideslip, rad or deg
$\delta_a, \delta_e, \delta_r$	= control surface deflections, rad or deg
ϕ, θ, ψ	= inertial orientation Euler angles, rad or deg
σ	= parameter estimate uncertainty bound
$\hat{\theta}$	= parameter estimate

*Graduate Student, Kevin T. Crofton Department of Aerospace and Ocean Engineering, Senior Member AIAA. jamesgresham@vt.edu

†Graduate Student, Kevin T. Crofton Department of Aerospace and Ocean Engineering, Student Member AIAA. fahmi@vt.edu

‡Graduate Student, Kevin T. Crofton Department of Aerospace and Ocean Engineering, Member AIAA. bsimm11@vt.edu

§Graduate Student, Kevin T. Crofton Department of Aerospace and Ocean Engineering, Student Member AIAA. jeremyhopwood@vt.edu

¶Graduate Student, Kevin T. Crofton Department of Aerospace and Ocean Engineering, Student Member AIAA. jwade109@vt.edu

||Professor, Kevin T. Crofton Department of Aerospace and Ocean Engineering, Associate Fellow AIAA. cwoolsey@vt.edu

I. Introduction

AUTONOMOUS aircraft reached a turning point recently for increased national security use in dynamic environments. Unmanned aerial vehicles (UAVs) such as the Kratos XQ-58 Valkyrie and Boeing Airpower Teaming System are capable of aggressive maneuvering and will use complex control laws. A small UAV (sUAV) with a simple commercially available flight computer provides an inexpensive platform to test algorithms destined for these applications. Recent literature on system identification and control law implementation for small aircraft reflects the increased availability and use of instrumented sUAVs to support science and engineering. This paper describes techniques used for data processing and automated co-computer inputs to support low-risk, low-cost flight testing of advanced control laws for fixed-wing aircraft.

The primary contribution of this work is flight test validation of theoretical work on a novel port-Hamiltonian energy-based controller presented in Ref. [1]. This paper also details the data processing and automated input tools used in Ref. [2], which were omitted there to focus on novel pilot input techniques. These useful techniques and algorithms were developed for the combined effort and are being made available to the community through a publicly available web repository [3]. This includes code to process PX4 ULog format (“.ulg”) data for system identification and control law analysis, as well as documentation and code to implement automated inputs and real-time feedback control using a Raspberry Pi co-computer. While a fixed-wing aircraft was used for this effort, the approach and techniques are applicable to other unmanned aircraft categories, such as multirotors.

There is substantial literature concerning the identification and use of flight dynamic models for small unmanned aircraft. Papers describing input excitation techniques and data analysis methods are especially relevant to the work presented here. Reference [4] describes development of a linear state-space model using frequency sweep inputs and CIFER [5] frequency domain analysis tools to then tune a proportional-integral-derivative controller for a unique research aircraft configuration. Reference [6] assesses a variety of computer generated excitation inputs, including frequency sweeps, multisines, and short duration inputs, with the aim of characterizing the handling qualities of a sUAV. Reference [7] describes an approach to online, near real-time system identification to develop local linear models using computer generated excitations including doublets, triplets, and frequency sweeps. References [8, 9] describe an artificial neural network system identification algorithm to rapidly generate linear models using automated doublets as excitations. Reference [10] documents system identification methods tailored for a sUAV using low-cost sensors together with computer generated sum-of-sines inputs to develop a linear aerodynamic model.

Additionally, other engineering efforts on small aircraft control law research were studied. Reference [11] describes development of a Linux based flight control computer for use in sUAS research, prior to the widespread use of the Pixhawk hardware. References [12, 13] introduce a generic sUAV nonlinear guidance control system using backstepping and adaptive modified incremental backstepping, respectively; the control laws were demonstrated using simulated flight data. Reference [14] develops a robust servomechanism linear quadratic regulator for a sUAV and presents simulation results. Reference [15] uses a similar flight test aircraft and instrumentation setup to the present paper; the study conducts aerodynamic system identification and implements an explicit model following control law using the Pixhawk and modified ArduPilot firmware. A downside of modifying the firmware for control law research is the possibility of unintentionally altering critical software components as well as the inability to use future firmware releases without careful review and modification.

This paper addresses a particular need for guidance concerning the effective use of flight computers and co-computers for sUAV system identification and control law implementation. The remainder of the paper is organized as follows. Section II describes the aircraft system identification process, including system identification excitation inputs, data processing techniques, and select control design methods that play a role in this effort. Section III describes the particular sUAV and instrumentation flown in this effort and design of the flight test experiments. Section IV presents flight data processing techniques and control surface servo-actuator modeling techniques. Section V presents results from flight tests. Section VI discusses conclusions, as well as plans for ongoing work.

II. Background

The overall aim of this effort was to experimentally demonstrate, for the first time ever, the passivity-based nonlinear flight control law described in Ref. [1] on a fixed-wing aircraft. This section provides an overview of the steps required to develop and demonstrate the controller, including the system identification process, data processing techniques, the baseline controllers implemented for safety build up, and the advanced nonlinear controller.

A. Aircraft System Identification

1. System Identification Process

System identification is the process of developing a mathematical model of a physical system based on observed data from the system [16]. The objective of system identification is to create a *useful* system model based on measured or calculated *outputs* and *inputs*. It is composed of multiple steps which typically result in a set of differential or algebraic equations describing system behavior. The aircraft system identification methodology, inspired by Ref. [17], and adapted for the small, fixed-wing UAV research in Refs. [2, 18] is shown in Fig. 1.

Aircraft system identification seeks to develop an aerodynamic model that adequately describes the aircraft motion or response (outputs) to controls or perturbations (inputs). System identification is performed using imperfect knowledge of the inputs and outputs due to sensor noise, process noise, sensor scale factors, and bias that must be accounted for during analysis. The overarching aircraft system identification methodology from Ref. [17], and refined for small, low-cost, fixed-wing UAVs in Refs. [2, 18] was used for this effort. Data compatibility, or kinematic consistency, analysis was accomplished using both integration and smoothed differentiation of kinematic states and measurements. Next, model structure determination from experimental data was accomplished using both stepwise regression (SWR) [17, 19] and multivariate orthogonal function (MOF) modeling [17, 20]. Values for the unknown parameter estimates were determined initially using the equation-error method and then final parameter estimates were determined with the output-error method [17, 21]. Finally, validation of each model was accomplished to confirm the model predictions agreed with independent flight data not used to develop the model.

2. Automated System Identification Excitation Inputs

The first known research to estimate stability and control parameters from flight test data occurred in the late 1940s, described in Ref. [22]. Milliken utilized automatically programmed inputs of both step and continuous sinusoidal forcing functions for each control surface independently with graphical analysis of the transient and frequency response data [22]. Digital computers accelerated analysis techniques in the 1960s and 1970s, with new “multistep” inputs including doublets, 1-2-1, and 3-2-1-1 inputs and the addition of other oscillatory inputs including frequency sweeps and Schroeder-phased harmonic signals [23–25]. Multistep inputs are designed to be conducted near the frequencies of the aircraft dynamic modes, to best excite the natural dynamic motion. The amplitude is chosen to obtain a high signal-to-noise ratio without violating the modeling assumptions. Multistep inputs have the advantage of being simple to implement. Assuming the aircraft does not have a feedback control system active, a skilled pilot can provide these excitations without the necessity for automated inputs, and can vary frequency of the input in real-time with observation of the aircraft response. A computer can be programmed for highly repeatable maneuvers with *a priori* knowledge or adjusted in real-time from modeling results [17]. Additionally, step inputs excite broadband frequency content, which is advantageous and disadvantageous for different objectives. For an inaccurate or unknown *a priori* model, exciting a broad range of frequencies provides robustness in the model identification process by exciting frequencies around the intended input. Exciting a broad spectrum of frequencies can be disadvantageous, however, as it results in unintentionally exciting dynamic modes that are of less immediate interest, such as nearby rigid body or structural modes, and which can obscure the signal of interest. For example, a short period doublet excites the phugoid motion because it contains low frequency content [26]. Frequency sweeps and oscillatory inputs such as the Schroeder-phased harmonic signal excite a broad spectrum of frequencies, require minimal *a priori* knowledge of the system, and are well suited for frequency domain analysis techniques [5, 17]. In the early 2000s, orthogonal phase-optimized multisine inputs were developed to simultaneously excite multiple control surfaces and thereby excite the aircraft’s natural longitudinal and lateral-directional motion concurrently [17, 27, 28]. This technique offers the primary advantages of minimizing test time by ensuring the mutual orthogonality of control inputs so that excitation energy is distributed efficiently over a

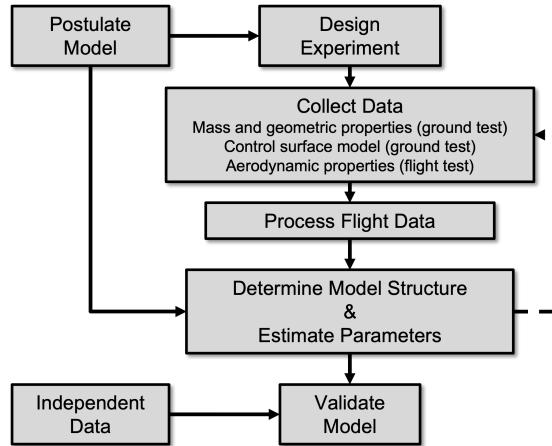


Fig. 1 Aircraft system identification process for a small fixed-wing aircraft (adapted from Ref. [17]).

range of interest. Design of multisine inputs does not require *a priori* information of the aircraft dynamics, but does require relatively precise programmed inputs and thus automated control inputs [29].

The implementation and in flight execution of a subset of the automated inputs discussed here was a key build up step to implementing on-board feedback control.

B. Feedback Control Laws

1. Proportional-Derivative Attitude Control

A typical state-based automatic feedback controller calculates the difference between the current (actual) value of a state, x_{act} , and the desired value of the state, x_{des} , resulting in the error:

$$e(t) = x_{\text{des}}(t) - x_{\text{act}}(t). \quad (1)$$

This error is used to calculate a control signal that will drive the error to zero. Proportional-derivative (PD) feedback control includes a gain multiplied by the error, K_p , as well as a gain multiplied by the time derivative of the error, K_d . The control input for a time invariant PD controller then takes the form:

$$u(t) = K_p e(t) + K_d \dot{e}(t). \quad (2)$$

The proportional control action is a gain which amplifies the error directly. Large K_p values result in faster rise time. In a steady state case, the derivative control action anticipates the error, “initiates early action, and tends to increase the stability of the system” [30]. A well-tuned derivative gain adds damping and allows a larger proportional gain without resulting in undesirable overshoots.

A simple aircraft attitude PD controller was implemented in this effort to demonstrate real-time feedback control. The state and control actuator used for longitudinal control were pitch attitude, θ , and elevator, δ_e , respectively. The state and control actuator used for lateral control were bank angle, ϕ , and aileron, δ_a , respectively. To maintain a simple control law, rudder was controlled directly by the pilot without any feedback. Linear scalings were defined to convert the pilot’s manual pitch and roll inputs into “desired” pitch and roll angles. The range of the control stick’s movement was scaled to correspond to an allowable range of pitch ($\theta : [-30^\circ, 30^\circ]$) and roll ($\phi : [-45^\circ, 45^\circ]$) attitudes. The flight computer extended Kalman filter (EKF) described in Refs. [31, 32] provided the “actual” θ and ϕ , and the co-computer completed the calculations and sent the control input back to the flight computer. The hardware and software architecture is discussed further in Sec. III.

There are numerous methods to experimentally tune a PD controller based on the dynamic response to known inputs, with one example being the Ziegler-Nichols method described in Ref. [33]. However, there is not a rigorous method to tune the control law without prior knowledge of the system characteristics. In this effort, small gains were initially chosen for K_p and K_d for both θ and ϕ and then tuned based on pilot comments for simple path control maneuvers.

2. Linear Quadratic Regulator

The linear quadratic regulator (LQR) is a popular and well-established model-based strategy for closed-loop, full-state feedback control of linear systems. The resultant control law provides the optimal solution for a prescribed cost function.

Consider a linear time-invariant system with a state vector $\mathbf{x} \in \mathbb{R}^n$ and input vector $\mathbf{u} \in \mathbb{R}^m$:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad (3)$$

where \mathbf{A} and \mathbf{B} are the state and control matrices of appropriate dimensions. According to infinite-horizon LQR, for a cost function defined as

$$J(\mathbf{x}(t), \mathbf{u}(t), t) = \int_0^{\infty} \mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t) + \mathbf{u}^T(t) \mathbf{R} \mathbf{u}(t) + 2\mathbf{x}^T(t) \mathbf{N} \mathbf{u}(t) dt, \quad (4)$$

where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is a positive semi-definite state weighting matrix, $\mathbf{R} \in \mathbb{R}^{m \times m}$ is a positive definite control weighting matrix, and $\mathbf{N} \in \mathbb{R}^{n \times m}$ is a cross-term matrix, the optimal control law is

$$\mathbf{u}(t) = -\mathbf{R}^{-1} (\mathbf{B}^T \mathbf{P} + \mathbf{N}^T) \mathbf{x}(t) \quad (5)$$

where $\mathbf{P} \in \mathbb{R}^{n \times n}$ is the unique, positive definite solution of the algebraic Riccati equation:

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - (\mathbf{P} \mathbf{B} + \mathbf{N}) \mathbf{R}^{-1} (\mathbf{P} \mathbf{B} + \mathbf{N})^T + \mathbf{Q} = \mathbf{0} \quad (6)$$

It will be convenient to write the control law using a gain matrix $\mathbf{K} = \mathbf{R}^{-1} (\mathbf{B}^T \mathbf{P} + \mathbf{N}^T)$:

$$\mathbf{u}(t) = -\mathbf{K} \mathbf{x}(t). \quad (7)$$

Since LQR is a model-based, full-state feedback control method, its construction requires knowledge of the system model and its implementation requires knowledge of all states. The resulting control gains not only satisfy control design criteria, but offer the optimal solution for the selected weighting matrices as well as some desirable robustness guarantees. Moreover, the closed-loop system has the same order as the plant system and the gain matrix is invariant, because there are no internal state updates such as integral action. This is contrasted with PD control, which is a model free controller, and requires only a subset of the states. LQR is based on a linear model and may not provide satisfactory performance in all flight regimes.

3. Port-Hamiltonian Energy-Based Nonlinear Control

The nonlinear controller implemented in this paper, which was developed in Ref. [1], is based on modeling the dynamics of an aircraft as a port-Hamiltonian system (PHS). Hamiltonian dynamics is a standard formulation for mechanical systems in classical dynamics. Hamiltonian mechanics is related to Lagrangian mechanics in which the state of a system is defined by a vector of generalized coordinates \mathbf{q} , and the corresponding velocity vector $\dot{\mathbf{q}}$. The canonical Hamiltonian formulation instead uses the vector \mathbf{q} along with the corresponding momentum vector, $\mathbf{p} = \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}$, where $\mathbf{M}(\mathbf{q})$ is a positive definite generalized inertia matrix. Port-Hamiltonian systems are a generalization of canonical Hamiltonian systems which allow for reduced-dimensional state vectors that reflect certain system symmetries, dissipative forces, and inputs and outputs. These systems are characterized by a scalar function H called the Hamiltonian which typically represents the total energy of the system. A port-Hamiltonian system has the form:

$$\dot{\mathbf{x}}(t) = [\mathbf{J}(\mathbf{x}, t) - \mathbf{D}(\mathbf{x}, t)] \frac{\partial H(\mathbf{x}, t)}{\partial \mathbf{x}} + \mathbf{g}(\mathbf{x}, t) \mathbf{u}(t) \quad (8)$$

$$\mathbf{y}(t) = \mathbf{g}^T(\mathbf{x}, t) \frac{\partial H(\mathbf{x}, t)}{\partial \mathbf{x}} \quad (9)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the state vector, $\mathbf{J}(\mathbf{x}, t) = -\mathbf{J}^T(\mathbf{x}, t)$ is an interconnection matrix which represents energy conserving interactions among the state variables, the matrix $\mathbf{D}(\mathbf{x}, t) = \mathbf{D}^T(\mathbf{x}, t) \geq \mathbf{0}$ accounts for the dissipation incurred during system motion, and the matrix $\mathbf{g}(\mathbf{x}) \in \mathbb{R}^{n \times m}$ determines how the m inputs $\mathbf{u}(t) \in \mathbb{R}^m$ affect system motion. The vector $\mathbf{u}(t)$ of inputs and the vector $\mathbf{y}(t) \in \mathbb{R}^m$ of outputs are conjugate in the sense that their inner product expresses the power exchanged with external systems. The port-Hamiltonian system can be shown to be passive [34] by taking the Hamiltonian as a storage function:

$$\frac{dH(\mathbf{x}, t)}{dt} = \mathbf{y}^T \mathbf{u} - \frac{\partial H(\mathbf{x}, t)}{\partial \mathbf{x}} \mathbf{D}(\mathbf{x}, t) \frac{\partial H(\mathbf{x}, t)}{\partial \mathbf{x}}^T \leq \mathbf{y}^T \mathbf{u} \quad (10)$$

Passivity is a useful property for assessing the stability and robustness of an individual system or of interconnections of multiple systems. Multiple energy-shaping techniques which exploit this property have been developed and applied to various electro-mechanical systems such as the control of electric motors [35], robots [36], unmanned underwater vehicles [37–39], and unmanned rotorcraft [40–44].

III. Flight Test Experiments

A. Research Aircraft and Instrumentation

The aircraft selected to serve as the research platform, shown in Fig. 2, was the commercially available, radio-controlled (RC), foam structure, fixed-wing aircraft marketed under the name My Twin Dream (MTD), manufactured by My Fly Dream. The aircraft is electrically powered by a four-cell 14.8V 6750mAh lithium-polymer battery, with a counter-rotating twin propeller configuration using APC 10-in. diameter, 6-in. pitch (10x6) propellers. The MTD aircraft was instrumented with a Pixhawk flight computer for data collection, specifically a Pixhawk CubePilot Cube

Orange flight controller running PX4 firmware v1.11.3 [45]. Additionally, an on-board co-computer was connected via Universal Serial Bus (USB) for automated input generation and control law implementation. The co-computer was a Raspberry Pi 4 running the Ubuntu 20.04 Linux distribution, communicating with the Pixhawk using MAVLink protocols, and using the Robot Operating System (ROS) Noetic libraries. The Pixhawk sensor suite included an attitude and heading reference system (AHRS), a pitot-static probe, and a global positioning system (GPS) receiver, specifically the Here+ Real-Time Kinematic (RTK) GPS capable receiver. The MTD was chosen for its simple construction, benign flight dynamics, endurance of approximately 30 minutes, twin engine redundancy, and sufficient payload to incorporate the desired instrumentation package and on-board co-computer. The MTD mass and geometric properties, tabulated in Table 1, were determined with methods described in Ref. [2]. The principle moments of inertia (MoI) were measured experimentally using the compound pendulum technique presented in Ref. [46] and the product of inertia I_{xz} was determined using a test described in Ref. [17]. The remaining products of inertia, I_{xy} and I_{yz} , were assumed to be zero based on lateral symmetry.



Fig. 2 MTD fixed-wing research aircraft.

Table 1 MTD mass and geometric properties

Property	Symbol	Value	Units
Mass	m	0.211	slug
Mean aerodynamic chord	\bar{c}	0.833	ft
Projected wing span	b	5.91	ft
Projected wing area	S	4.92	ft ²
Roll moment of inertia	I_{xx}	0.2163	slug·ft ²
Pitch moment of inertia	I_{yy}	0.1823	slug·ft ²
Yaw moment of inertia	I_{zz}	0.3396	slug·ft ²
Product of inertia	I_{xz}	0.0364	slug·ft ²
Products of inertia	I_{xy}, I_{yz}	~0	slug·ft ²

Minor modifications to a Pixhawk startup file and PX4 parameters were required prior to flight test to ensure the logging of specific data fields at desired sample rates. These steps are included in documentation available at Ref. [3]. Automated inputs for system identification excitation and feedback control were accomplished using multiple pieces of software on the Raspberry Pi co-computer. Setup of the Raspberry Pi co-computer followed the steps included in documentation available at Ref. [3]. Additionally, minor modifications to a Pixhawk startup file were required to send pilot inputs from the Pixhawk to the co-computer at appropriate data rates, as discussed in Section III.C. This was accomplished concurrently with the modifications for system identification.

B. Pilot Control with Radio Control Transmitter

Flight test data were collected at the Kentland Experimental Aerial Systems (KEAS) Laboratory at Virginia Tech [47]. Flight testing for this effort consisted of nearly 17 hours of flight time, included 56 flights, and was conducted on 24 separate days from April to October, 2021. The initial flight included an airworthiness checkout with piloted elevator and rudder doublet maneuvers flown to characterize short period and dutch roll respectively. This flight collected sufficient data to develop a preliminary model and confirm data collection was successful. Data collected on subsequent days was preceded by a calibration of the instrumentation magnetometer, accelerometer, gyroscope, and pitot-static system using a repeatable, methodical, and precise approach.

During initial testing, the pilot observed the short period and dutch roll mode natural aircraft response to an impulse input and replicated the periodic response for piloted elevator and rudder doublet inputs. Results from initial piloted elevator and rudder doublets were used to design the frequency of inputs for automated elevator and rudder doublet inputs to best capture short period and dutch roll dynamics. Additionally, the phugoid mode natural response was used to determine a starting frequency for designing the frequency sweep and multisine inputs. In industry, automated inputs are routinely referred to as Programmable Test Inputs (PTI). “Automated” and PTI terminology are used interchangeably in this paper.

Each flight test was controlled using a Spektrum DX20, 20 channel, RC transmitter. The switch layout is shown in Fig. 3 and described in Table 2. Piloted system identification excitation inputs were flown with the Pixhawk in “Manual” mode which disables feedback stabilization. Artificial stabilization is undesirable for system identification because it



Fig. 3 Radio-controlled transmitter labeled for use with automated inputs and feedback control.

suppresses the natural aircraft response which is being characterized [17]. For both system identification testing and control law testing, the Pixhawk was placed in “Offboard” mode to allow co-computer inputs. For system identification maneuvers, automated inputs were summed with the pilot inputs to permit attitude control. For control law testing, the pilot inputs provided command signals for the control law as described in Sec. II. The automated system identification inputs and feedback control were initiated when the PTI activation switch was engaged.

Table 2 RC transmitter channels

Channel	Use	Selection	Switch Type
1	Actuator Input	Aileron	Right Control Stick (L/R)
2	Actuator Input	Elevator	Right Control Stick (Fwd/Aft)
3	Actuator Input	Throttle	Left Control Stick (Fwd/Aft)
4	Actuator Input	Rudder	Left Control Stick (L/R)
5	Pixhawk / PX4 Mode	Stabilized - Manual - Offboard	3-place switch
6	PTI Activation	Off - On	2-place switch
7	PTI Mode	Doublet - Frequency Sweep - Multisine	3-place switch
8	PTI Submode	(Long - Lat-Dir - 3-2-1-1) or ($\delta_e - \delta_r - \delta_a$) or (Aero 3-input - Prop 4-input - unused)	3-place switch
9	PTI Amplitude	0% to 100%	Rotary knob
10	PTI Propulsion Amplitude	0% - 10% - 20%	3-place switch

PTI activation, PTI mode, PTI submode, and PTI amplitude were determined by individual channels. The submode functionality depended on which PTI mode was selected, as shown in Fig. 4. For example, if “Doublet” mode was selected in Fig. 4, the second submode (Lat-Dir) performed an automated rudder doublet and then aileron 1-2-1 maneuver. However, if “Frequency Sweep” mode was selected, the 2nd submode (δ_r) performed a rudder frequency sweep and the 3rd submode (δ_a) performed an aileron frequency sweep. The 3-2-1-1 maneuver performed a 3-2-1-1 input serially for each control surface independently, spaced 5 seconds apart. Aerodynamic 3-axis control surface multisine inputs were used for this effort and multisine inputs for control surfaces and the propulsion system were used for a concurrent effort [48]. This hierarchy was determined in the co-computer automation code based on each channel’s pulse width modulation (PWM) signal.

C. Experiment Design and Build Up Approach

The test environment was selected to provide ample maneuvering airspace in all directions, horizontally and vertically, which allowed flexibility to setup and observe aircraft response with minimal concern for airspace boundaries. Additionally, it was desired for flight tests to be conducted with negligible wind, considered at or below one knot.

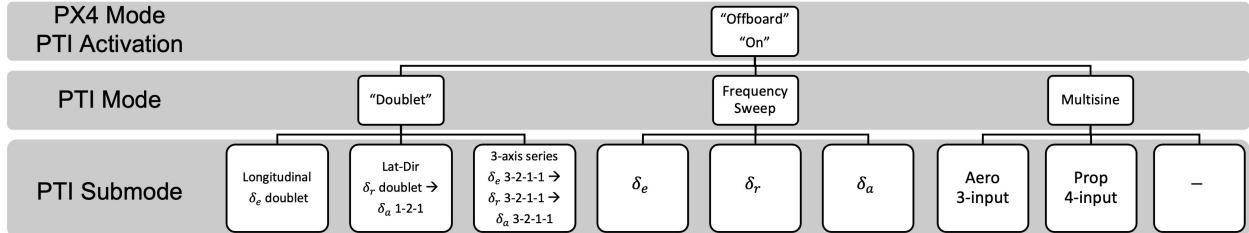


Fig. 4 System identification programmable test input mode hierarchy.

A build up approach was followed for co-computer inputs, with increasingly complex inputs implemented after successful in-flight demonstration of simpler code. For example, PTI doublets were demonstrated first, then vector based single-axis frequency sweeps, and then multi-axis multisine inputs. The same logic was then followed for control law implementation. First, the PD feedback control was implemented, then LQR was demonstrated, and finally, the nonlinear path following control law was tested. This build up approach was critical to developing confidence in the co-computer's ability to process control law algorithms in real-time, in combination with pilot manual inputs. Also, this build up approach facilitated significant code improvements which were initiated during ground testing, but required flight test for detailed verification of functionality. One example of this was adjustment of the rate at which the Pixhawk passed the "manual_control" data field over MAVLink to the co-computer. During initial PD feedback control law implementation, the test pilots noted poor handling qualities due to a time lag in inputs. The default rate for "manual_control" is 5 Hz, but was adjusted to 50 Hz. The test pilots subsequently noted greatly improved handling qualities for the same PD gains.

IV. Data Processing

Flight data processing is a critical step in the system identification process to convert measured flight data to signals conditioned for system identification analyses. The steps include data smoothing, correcting time delays, calculating unmeasured signals, and data compatibility analysis and corrections. Although tedious and often overlooked, proper signal conditioning is necessary to obtain good modeling results. As a part of this effort and Refs. [2, 18, 48–50], a data processing methodology for system identification of aircraft equipped with the Pixhawk flight computer was developed and refined by the third author. The current version of the data processing methodology is documented in this section.

When the Pixhawk flight computer with PX4 firmware was first flight tested, all variants of available signals needed for flight dynamics analysis were compared to determine the best measurements and state estimates to use for post-flight system identification and real-time feedback control. The signals selected for system identification are shown in Table 3, with modifications needed for real-time control applications included in the table footer. Notably, the aircraft instrumentation suite lacks an air-data sensor to measure the aerodynamic angles α and β . The Pixhawk PX4 firmware includes an extended Kalman filter (EKF) used to estimate vehicle state variables, including attitude angles and Earth-relative velocity. The EKF states were leveraged successfully to calculate signals traditionally derived from air-data sensors in previous and concurrent system identification works for flights with negligible wind [2, 18, 48–50]. Therefore α and β were derived from EKF state estimates as discussed in detail later. Further information about the PX4 EKF and output predictor algorithm can be found in Refs. [31, 32].

Table 3 Sources of key measurements from flight test data

Measurement Name	Source/Type	Data Field	Symbol	Default Units	Sample Rate
Body-Axis Accelerations	Accelerometers	sensor_combined	a_x, a_y, a_z	m/s ²	205 Hz
Body-Axis Angular Velocities	Rate Gyros	sensor_combined	p, q, r	rad/s	205 Hz
Earth-Fixed Velocities	EKF*	estimator_status*	V_N, V_E, V_D	m/s	102 Hz
Orientation (Quaternions)	EKF*	estimator_status*	q_0, q_x, q_y, q_z	—	102 Hz
Control Signals	PWM Signals	actuator_outputs	$\delta_e, \delta_a, \delta_r$	μs	41 Hz
Density	Air-Data	vehicle_air_data	ρ	kg/m ³	20 Hz

*For post-flight applications, the "estimator_status" (PX4 EKF states) data fields were observed to have marginally improved data quality over the output predictor states. For real-time applications, the "vehicle_local_position" and "vehicle_attitude" data fields (PX4 output predictor states) sampled at 51 Hz are used in place of "estimator_status" to reduce time delay [32].

One data processing objective is to convert the raw PX4 “.ulg” format to a MATLAB® data format compatible with the employed system identification tools. Several other intermediate signal processing steps must also be taken because the raw PX4 output measurement signals are not suitable for system identification for several reasons, including:

- 1) Message topics have different sample rates.
- 2) Message topic sample intervals are not constant.
- 3) Raw signals are not expressed as the states needed for modeling.
- 4) Raw signals contain bias errors and relative time offsets.
- 5) The data include measurement noise.

Several data processing steps are needed to provide sufficient data quality for system identification and the steps are different depending on the particular signal. The general steps are summarized as a flowchart shown in Fig. 5. The result is processed flight data stored in an *fdata* matrix, which is the standard data format used in the System IDentification Program for AirCraft (SIDPAC) software toolbox [17, 51]. Each step in the flowchart is described in the remainder of this subsection. Note that it is important to include an extra second at the start and end of the data when processing the data. The reasons for this include being able to remove initial filter transients, numerical differentiation startup inaccuracies, and to shift time series signals. After applying the data processing steps, the extra second of time included at the beginning and end of the data is removed before performing further analysis.

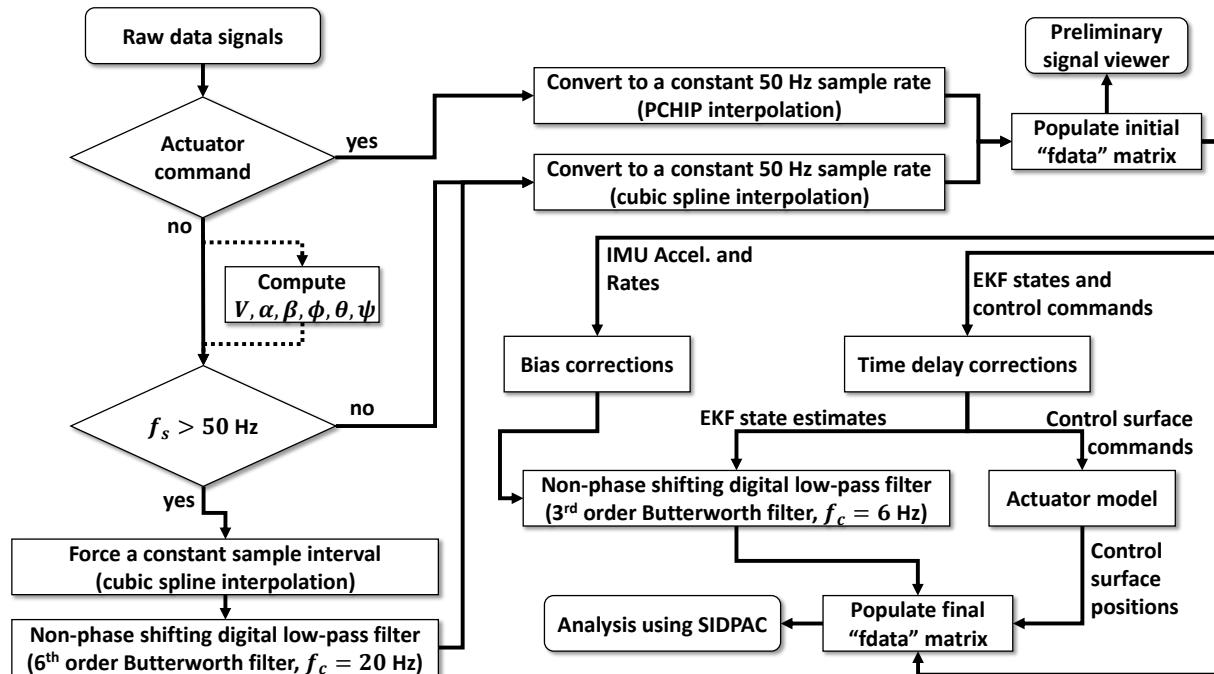


Fig. 5 Flight data processing flowchart.

A. Data Importing, Resampling, and Formatting

The data are first imported into the MATLAB® workspace from the “.ulg” format using the *ulogreader* function available in the MATLAB® UAV Toolbox [52]. The output is a “*ulogreader object*” which is then passed to the *readTopicMsgs* function to extract the data. More information about the “.ulg” file format is available in Ref. [53].

After the data have been imported into the MATLAB® workspace, the raw signals are then converted to have a constant sample rate of 50 Hz. The raw data have different sample rates and nonuniform sample intervals. Control surface actuator command signals are converted to a 50 Hz sample rate using shape-preserving piecewise cubic interpolation with the *pchip* function in MATLAB®. Shape-preserving piecewise cubic interpolation is preferred over cubic spline interpolation for actuator commands (which can be sharp inputs or square waves) because of its tendency to avoid overshoot and oscillatory behavior. Anti-alias protection is not needed for resampling the actuator commands because the PWM signals are recorded digital data which do not have wideband measurement noise. For all

other measured or estimated signals, the data are resampled using cubic spline interpolation. If the original signal's median sample rate is less than 50 Hz, the signal is upsampled using cubic spline interpolation directly. If the original signal's median sample rate is greater than 50 Hz then the data are resampled to have a constant sample interval at the raw signal's median sample interval, smoothed, and then downsampled to 50 Hz. The data are smoothed using a non-phase shifting digital filtering technique with a digital sixth-order low-pass Butterworth filter applied both forward and backward in time using the `filtfilt` function available in the MATLAB® Signal Processing Toolbox [52]. The filter cutoff frequency is selected as $f_c = 20$ Hz to provide at least -20 dB of attenuation at the Nyquist frequency of 25 Hz for the final sampling frequency of 50 Hz. This step removes higher frequency content that would cause aliasing when downsampling the data.

Within these initial processing steps the quaternion states are converted to Euler orientation angles and the air-data signals are computed using:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \cos \psi \sin \theta \sin \phi - \cos \phi \sin \psi & \cos \phi \cos \psi + \sin \theta \sin \phi \sin \psi & \cos \theta \sin \phi \\ \cos \psi \sin \theta \cos \phi + \sin \phi \sin \psi & \sin \theta \cos \phi \sin \psi - \sin \phi \cos \psi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} V_N \\ V_E \\ V_D \end{bmatrix} \quad (11)$$

$$V_t = \sqrt{u^2 + v^2 + w^2} \quad (12)$$

$$\alpha = \tan^{-1}(w/u) \quad (13)$$

$$\beta = \sin^{-1}(v/V) \quad (14)$$

These equations assume that there is no wind, which is a reasonable assumption because the modeling data are collected in negligible wind conditions.

After all signals are resampled and computed, the data are populated into an initial `fdata` matrix. The data at this point were examined using a custom-developed signal viewer to determine the start and end times of each maneuver and assess maneuver quality. Additional steps are needed before using the data for model development.

B. Data Compatibility Analysis and Corrections

Data compatibility, or kinematic consistency, analysis is the process of validating that dependent kinematic signals are compatible using the aircraft equations of motion. Ensuring kinematically consistent data is essential to obtaining quality system identification results. This analysis is performed by using the translational dynamics and rotational kinematics relations,

$$a_x = \dot{u} + qw - rv + g \sin \theta \quad (15)$$

$$a_y = \dot{v} + ru - pw - g \cos \theta \sin \phi \quad (16)$$

$$a_z = \dot{w} + pv - qu - g \cos \theta \cos \phi \quad (17)$$

$$p = \dot{\phi} - \dot{\psi} \sin \theta \quad (18)$$

$$q = \dot{\theta} \cos \phi + \dot{\psi} \sin \phi \cos \theta \quad (19)$$

$$r = \dot{\psi} \cos \phi \cos \theta - \dot{\theta} \sin \phi \quad (20)$$

which are a rearranged form of the rigid body aircraft translational dynamics and rotational kinematics equations [17].

Data compatibility analysis is performed using a numerical integration and/or differentiation procedure. The numerical integration method is performed by integrating Eqs. (15)-(20) using the translational accelerations and angular rates as inputs. Reconstructed air-data are subsequently calculated from the reconstructed body-axis velocity using Eqs. (12)-(14). The reconstructed Euler attitude angles and air-data signals are then compared to the corresponding measured signals to assess kinematic consistency. An alternative procedure is to substitute measured Euler angles, angular rates and body-velocity, as well as computed time derivatives of Euler angles and body-velocity, into Eqs. (15)-(20) to obtain the reconstructed translational accelerations and angular rates. If the traces of the reconstructed and measured signals match, then the data are kinematically consistent. If data are found to be kinematically inconsistent from this analysis, corrective measures must be taken before proceeding in system identification analysis. Potential errors may include bias errors in translational acceleration and angular rate measurements, bias errors and scale factor errors in Euler attitude angles and air-data measurements, position corrections for acceleration and air-data measurements to the aircraft center of gravity, and time skews between measurements [17].

Time delays between data signals used for system identification corrupt analysis [54], because parameter estimation methods assume that all signals are sampled at the same time [55]. Translational acceleration and angular rate signals are generally acquired from the same data stream making their time stamps a good grounding value. Data compatibility analysis can be used to correct air-data time delays by comparing apparent time lags between measured and reconstructed signals. Control variables can be corrected for time delays using a time delay as a parameter in the parameter estimation process. The time delays for air-data and control deflections can be estimated through visual inspection, exhaustive enumeration of a series of possible time delays, or use of frequency domain parameter estimation methods described in Ref. [55]. Servo-actuator models can also bookkeep time delay, as will be discussed in the next subsection.

The data compatibility corrections considered for this work include correction of a constant bias in translational accelerations and angular rates, correction for time delays in Euler angles and air-data measurements, and correction for time delays in control signals as a part of servo-actuator models. Scale factors and biases in Euler angles and air-data measurements are also usually considered in data compatibility corrections [17]; however, these states obtained from the PX4 EKF did not appear to have this deficiency. The acceleration and angular rate biases were corrected by subtracting the median difference between the measured and reconstructed signals. Time lags for aircraft states were estimated visually by comparing reconstructed and measured signals. The instrumentation system IMU was secured at the aircraft CG so position corrections were not necessary.

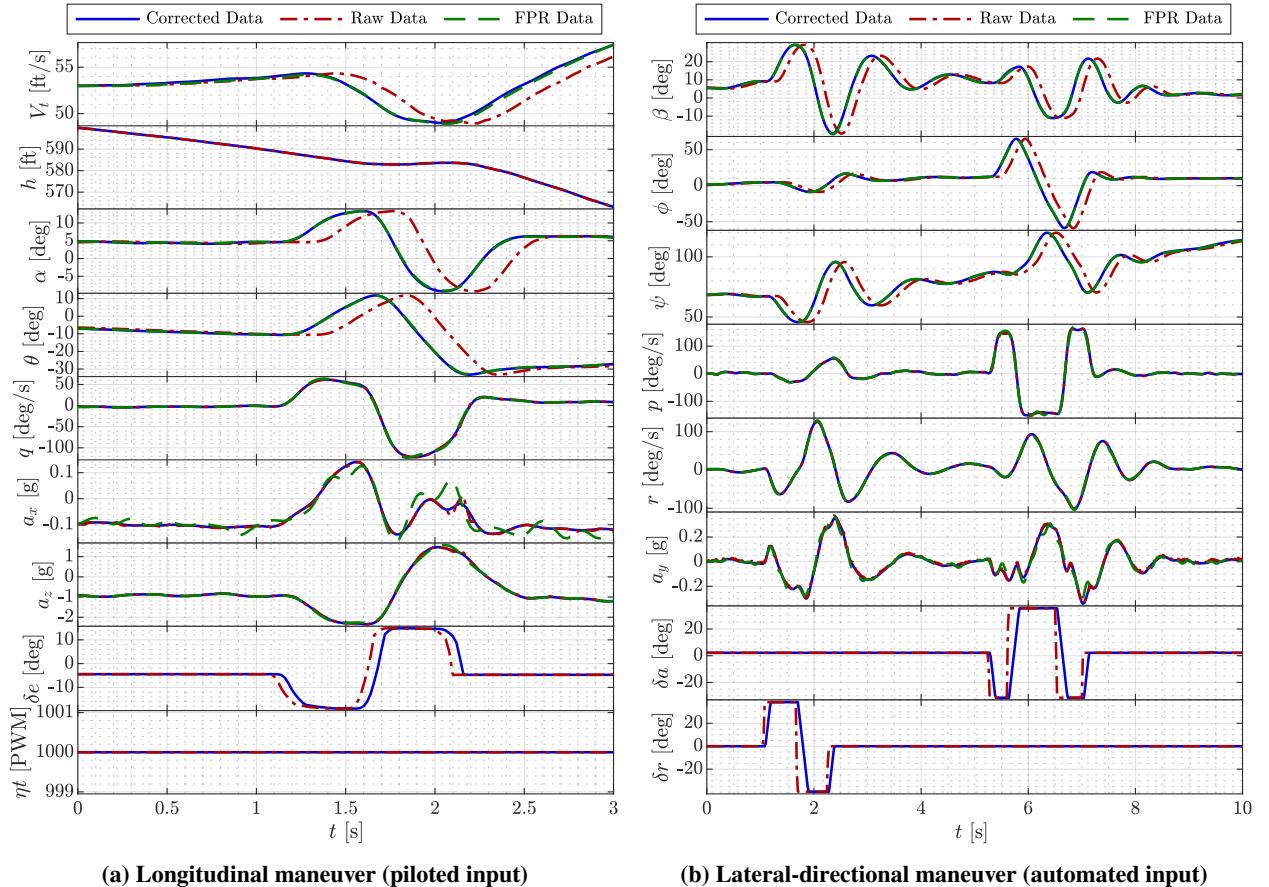


Fig. 6 Measurement kinematic consistency analysis comparing corrected flight data to reconstructed signals.

Figure 6 shows example data compatibility analysis plots for longitudinal and lateral-directional maneuvers. The figure shows corrected data used for system identification analysis, raw data obtained from the instrumentation system, and flight path reconstruction (FPR) signals computed from other measurements to confirm validity of the kinematic consistency corrections. The reconstructed measurements for V_t , α , β , ϕ , θ , and ψ are obtained from integrating Eqs. (15)-(20) using measured IMU signals p , q , r , a_x , a_y , and a_z as integrator inputs and using Eqs. (12)-(14). The reconstructed measurements for p , q , r , a_x , a_y , and a_z are obtained through differentiation of measured u , v , w , ϕ , θ , and

ψ (obtained from the EKF-derived states) and direct substitution into Eqs. (15)-(20). After correction, most measured and reconstructed signals agree well showing that the data have been sufficiently conditioned for the proceeding system identification analyses. Notably the measured axial acceleration, a_x , does not match as well with the reconstructed state because this signal is weakly excited by the pitch doublet maneuver shown for analysis.

C. Control Surface Servo-Actuator Model

Control surface actuator ground testing was performed to develop a servo-actuator model. The model consists of a static model and a dynamic model. The static actuator model relates the PWM command, η , to the control surface deflection angle after all transients have ended, which is referred to as the commanded deflection angle, δ_c . The dynamic actuator model describes the actuator dynamic response to changes in the commanded deflection angle to compute the control surface deflection angle, δ . A block diagram for the actuator model is shown in Fig. 7.

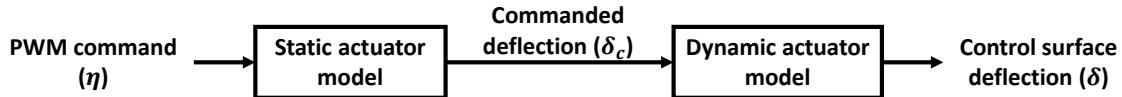


Fig. 7 Control surface servo-actuator model block diagram.

The ground testing used to develop the static and dynamic actuator models consisted of cascaded square wave pulses of 10 different amplitudes applied to each control surface, as shown in Fig. 8. Frequency sweeps and multisines were also applied to establish a baseline understanding of actuator characteristics. Actuator ground testing was performed using the same Pixhawk flight computer used in-flight as the measurement device. Since a control surface deflection sensor, such as a potentiometer, was not readily available, the Pixhawk was affixed to each individual control surface using double sided foam adhesive tape to enable reconstruction of deflection angle, as shown in Fig. 9. This approach proved to be both resourceful and accurate. This strategy requires no additional sensors than are used in-flight, which is useful for characterizing low-cost aircraft systems. Additionally, since the Pixhawk flight computer is used as the measurement device, the method also allows for accurate characterization of time delays relative to other signals measured by the Pixhawk. Furthermore, the presence of the Pixhawk on the control surface appeared to have a negligible effect on the actuator dynamics.

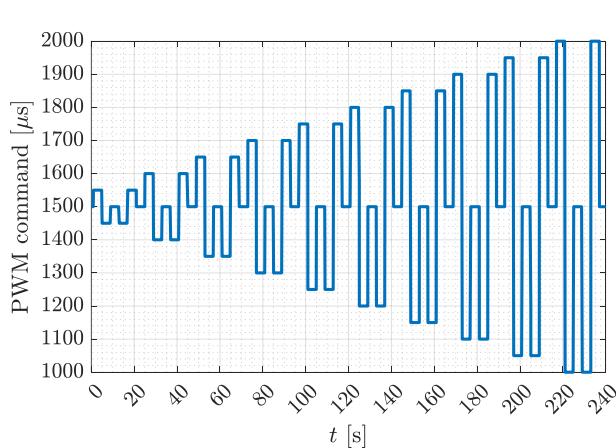


Fig. 8 Cascaded square wave actuator test input.

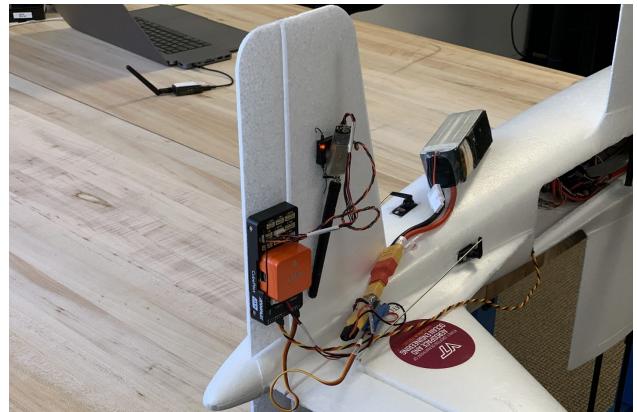


Fig. 9 Pixhawk affixed to the MTD elevator for actuator characterization ground testing.

The angular velocity signal about the axis of rotation was integrated to reconstruct the deflection angle. However, the integrated deflection angle signal drifted over time, corrupting analysis, because the measured angular velocity signal contains a small bias error. Since all control surface inputs started and ended at zero deflection, the trend of the numerically integrated signal was removed, such that the starting and ending deflection angles were both zero. The detrended signal was considered the measured output used for a sequential optimization procedure leveraging the output-error parameter estimation method [17] to correct the angular velocity bias since there was no measured position signal available. This procedure yielded an accurate deflection angle solution. The process is reflected in Figs. 10 and

11 for a rudder deflection ground test. The bias in the rudder deflection rate $\dot{\delta}_r$ measurement (i.e., the Pixhawk angular velocity measurement along the axis of rotation) appears small, but clearly affects the reconstructed deflection angle δ_r . This procedure was performed independently for each doublet square wave input to estimate a deflection angle signal used to identify both the static and dynamic actuator models. The EKF orientation angle solution was also available from the Pixhawk data log, however, these signals appeared to be flawed due to the abnormal use of the measurement hardware out of its intended application and were discarded for analysis.

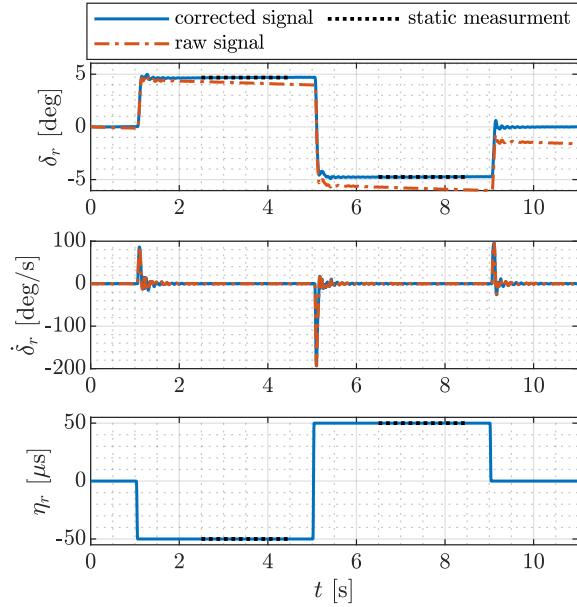


Fig. 10 Rudder response to a square wave input in ground testing.

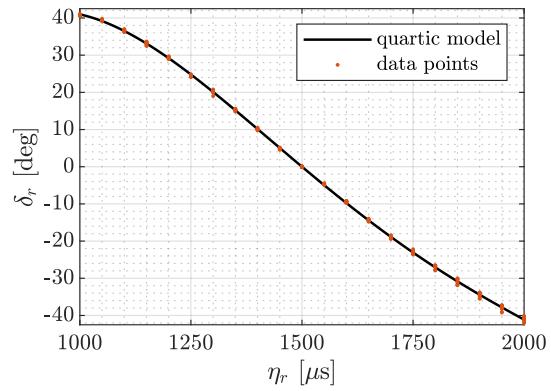


Fig. 11 Rudder PWM command to deflection angle model fit.

1. Static Actuator Model Identification

The static actuator model was developed as a polynomial mapping between the PWM command sent to the actuator, η , and commanded control surface deflection angle, δ_c . Deflection angle data were taken as an averaged value from the static portions of each square wave input, as shown in Fig. 10. A polynomial model was then fit to the data to characterize the relationship between PWM command and deflection angle. The data used to develop the rudder deflection angle mapping and the quartic polynomial model for the rudder control surface are shown in Fig. 11, where a close model fit is observed. Note that the process must be repeated for each individual control surface. This data-driven deflection angle modeling strategy is much more accurate than conventional methods using a protractor to create a control surface deflection angle calibration.

2. Dynamic Actuator Model Identification

Two dynamic model forms were considered to convert commanded deflection angle, δ_c , to actual deflection angle, δ . The first model was a first-order dynamics model with a pure time delay. The transfer function for this model is

$$\delta(s) = \frac{e^{-\tau_0 s}}{\tau_1 s + 1} \delta_c(s) \quad (21)$$

where τ_0 is the time delay and τ_1 is the first-order time constant. The second model consisted of a rate limit and a pure time delay, which is a nonlinear model. The parameters in each model were identified by treating the commanded deflection angle, δ_c , as the input and the reconstructed deflection angle, δ , as the output. The two parameters in each model type were exhaustively swept to determine the combination with the lowest least-squares cost, which were taken to be the model parameter values.

Example modeling results for a doublet square wave input are shown for the first-order dynamics model in Fig. 12a and for the rate limit model in Fig. 12b. The model fit for both model forms are good; however, the rate limit model is

observed to have a superior visual model fit demonstrated by the lower modeling residuals and the closer model fit in the transient regions seen in the zoomed portion of the plot. It was also noted that the estimated first-order model time constant parameter increased with increasing input amplitude, whereas the the estimated rate limit parameter was constant across the range of input amplitudes tested. Furthermore, the estimate of the time constant parameter and the time delay parameter were somewhat confounded, whereas the rate limit parameter and time delay parameter were decoupled. These results suggest that the first-order dynamics model (which is commonly used to characterize actuator behavior) may not be the best representation for an electric servo-actuator, which typically has a trapezoidal response to step commands, as can be seen in Fig. 12. Although the first-order dynamics model may be more convenient for controller design and tuning, the rate limit model appears to be the better representation when the objective is to most accurately estimate control surface deflection angles subject to sharp changes in input command. For this reason, the rate limit model with a pure time delay was used as the actuator model to compute the estimated control surface deflection angle for system identification analysis. The final parameter values for each control surface were taken to be the averaged values computed across the range of amplitudes tested.

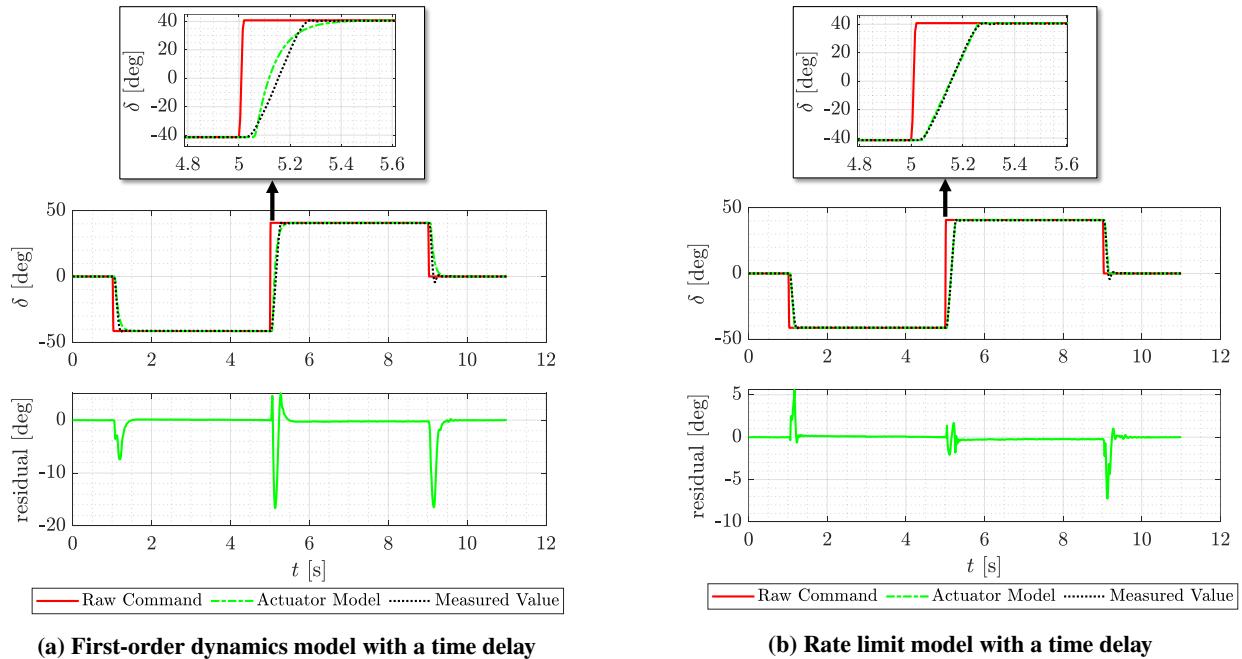


Fig. 12 Actuator command and modeled actuator response compared to the measured deflection angle.

D. Data Smoothing

After all above steps are applied, the flight data, excluding the control surface signals, are smoothed by applying a third-order Butterworth filter both forward and backward in time using the `filtfilt` function in MATLAB®. Measured flight data contains noise which can compromise the accuracy of system identification. Smoothing helps satisfy the assumed noise characteristics of data used for parameter estimation [17, 21]. Additionally, it is best to smooth data before performing numerical differentiation, which is required to compute analysis signals such as the body-axis angular accelerations [17]. A cutoff frequency of $f_c = 6$ Hz was selected to not distort low frequency information needed for modeling, but reject most high frequency noise.

The frequency response of the non-phase shifting digital filter is compared to the frequency response of a global optimal Fourier smoothing technique [17, 56] commonly applied for aircraft system identification in Fig. 13. The magnitude plots show similar roll-off characteristics. The phase plot shows that both the zero-phase shifting digital filter and the global optimal Fourier smoother have no phase lag, unlike a typical digital filter that is only applied forward in time. The non-phase shifting nature of the smoothing techniques allows a lower cutoff frequency, and thus further noise attenuation, without corrupting the deterministic content used for modeling.

After completing all data processing steps the data were stored in an updated `fdata` matrix and then used to perform

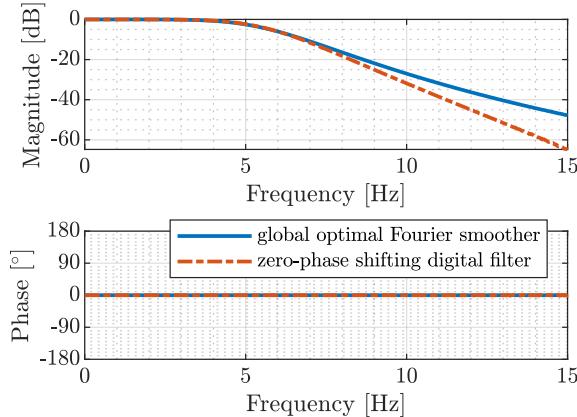


Fig. 13 Comparison of the frequency response for the global optimal Fourier smoother and non-phase shifting third-order Butterworth digital filter, each with a break frequency of 6 Hz.

model identification leveraging methods available in SIDPAC [17, 51].

E. Data Processing Code

A MATLAB[®] code implementing the steps shown in Fig. 5 was used for data processing for multiple system identification efforts performed in parallel to this work, including Refs. [2, 48, 50]. The code is generalized for use with multiple Nonlinear Systems Laboratory aircraft including the MTD aircraft used in this work, the CZ150 aircraft used in Refs. [48, 50], and the eSPAAERO aircraft. The code can be easily applied to future aircraft equipped with the Pixhawk flight computer running the PX4 firmware. An abbreviated version of the code is available in Ref. [3]. The full code cannot be made publicly available because certain functionality leverages SIDPAC codes, which are not in the public domain.

V. Flight Test Results

A. Aerodynamic System Identification

The aircraft system identification methodology presented in Sec. II.A was followed to generate a nonlinear aerodynamic model of the MTD aircraft. The model was developed using automated excitation input maneuvers collected for Ref. [2], excluding data sets with unsuitable data quality due to process noise, data dropouts, or extraneous inputs. The modeling process included automated doublet, 3-2-1-1 multistep, and multisine excitation input maneuvers using the techniques and processing tools leveraged in Ref. [2]. Validation was accomplished with independently collected data for a piloted elevator doublet for longitudinal motion, and a rudder doublet followed by aileron 1-2-1 maneuver for lateral-directional motion. The independently collected validation data were not used for model development. The nonlinear model was linearized about the nominal cruise trim condition for use with LQR tuning. Model structure, parameter estimate, and validation results are shown for the nonlinear models and the linearized model in state space format is shown.

1. Nonlinear Aerodynamic Model

The identified nonlinear model structure is given in Eqs. (22)-(27) and the parameter estimates are given in Tables 4 and 5. Bias parameters C_{y_o} , C_{l_o} , and C_{n_o} were estimated for each maneuver, but discarded as nuisance parameters, as suggested in Ref. [17]. The model structure is valid in the normal operating range, to include just before stall; it does not capture propulsion effects, α - β coupling, stall effects, or spin dynamics which are discussed in related work in

Refs. [48, 50]. A comparison of the nonlinear model predicted outputs to validation flight data is shown in Fig. 14.

$$C_x = C_{x_\alpha} \alpha + C_{x_{\delta_e}} \delta_e + C_{x_{\alpha^2}} \alpha^2 + C_{x_o} \quad (22)$$

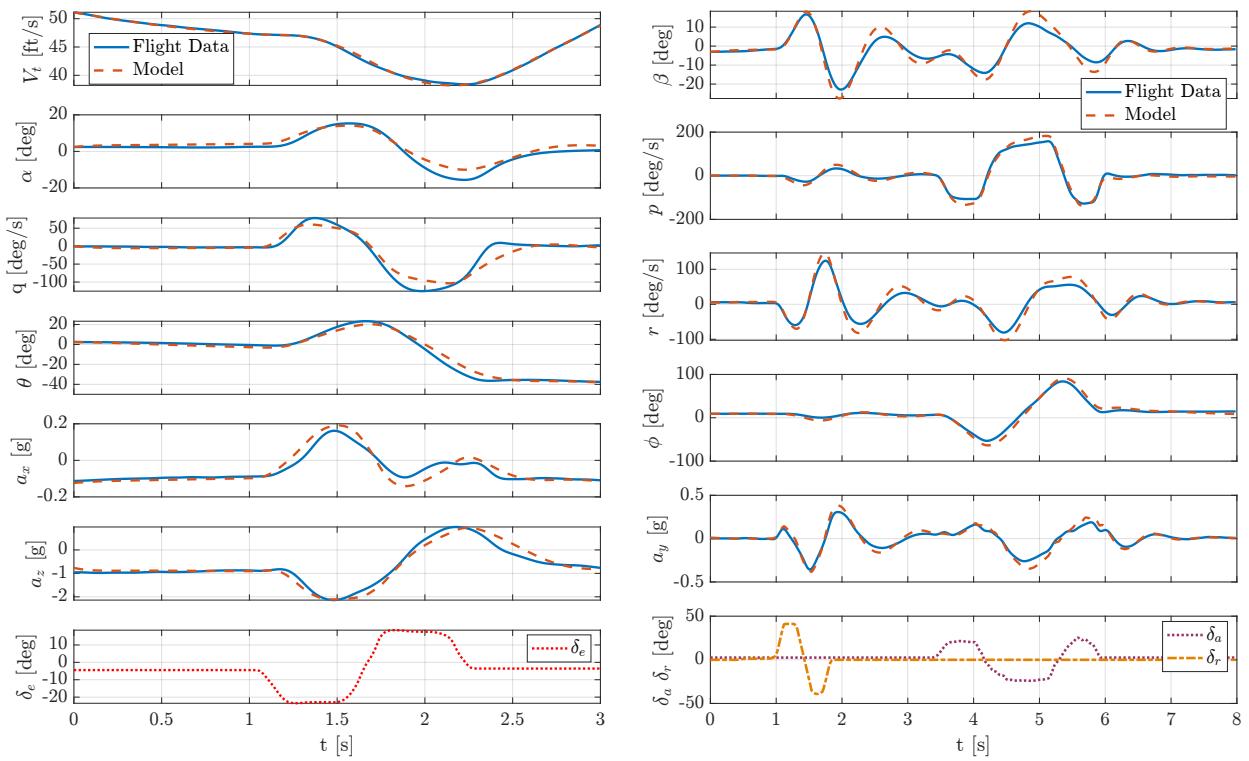
$$C_z = C_{z_\alpha} \alpha + C_{z_{\delta_e}} \delta_e + C_{z_{\alpha^2}} \alpha^2 + C_{z_o} \quad (23)$$

$$C_m = C_{m_\alpha} \alpha + C_{m_q} \hat{q} + C_{m_{\delta_e}} \delta_e + C_{m_{\alpha^2}} \alpha^2 + C_{m_{\alpha^3}} \alpha^3 + C_{m_o} \quad (24)$$

$$C_y = C_{y_\beta} \beta + C_{y_p} \hat{p} + C_{y_r} \hat{r} + C_{y_{\delta_a}} \delta_a + C_{y_{\delta_r}} \delta_r + C_{y_{\beta^3}} \beta^3 \quad (25)$$

$$C_l = C_{l_\beta} \beta + C_{l_p} \hat{p} + C_{l_r} \hat{r} + C_{l_{\delta_a}} \delta_a \quad (26)$$

$$C_n = C_{n_\beta} \beta + C_{n_p} \hat{p} + C_{n_r} \hat{r} + C_{n_{\delta_a}} \delta_a + C_{n_{\delta_r}} \delta_r + C_{n_{\beta^3}} \beta^3 \quad (27)$$



(a) Longitudinal Nonlinear Model Validation

(b) Lateral-Directional Nonlinear Model Validation

Fig. 14 Validation results for nonlinear model developed from automated inputs for piloted doublet maneuver.

Table 4 Nonlinear model: longitudinal parameter estimates

Term	$\hat{\theta} \pm \sigma$	Term	$\hat{\theta} \pm \sigma$	Term	$\hat{\theta} \pm \sigma$
C_{x_α}	$+0.4400 \pm 0.0346$	C_{z_α}	-3.947 ± 0.350	C_{m_α}	-0.8068 ± 0.111
$C_{x_{\delta_e}}$	-0.01398 ± 0.0329	$C_{z_{\delta_e}}$	-21.19 ± 8.80	C_{m_q}	-4.937 ± 3.15
$C_{x_{\alpha^2}}$	$+2.59 \pm 0.401$	$C_{z_{\alpha^2}}$	-0.8207 ± 0.495	$C_{m_{\delta_e}}$	-0.7286 ± 0.160
C_{x_o}	-0.061 ± 0.006	C_{z_o}	-0.206 ± 0.112	$C_{m_{\alpha^2}}$	-1.251 ± 1.916
				$C_{m_{\alpha^3}}$	-29.920 ± 17.988
				C_{m_o}	-0.061 ± 0.022

Table 5 Nonlinear model: lateral-directional parameter estimates

Term	$\hat{\theta} \pm \sigma$	Term	$\hat{\theta} \pm \sigma$	Term	$\hat{\theta} \pm \sigma$
$C_{y\beta}$	-0.3761 ± 0.0273	$C_{l\beta}$	-0.0529 ± 0.0088	$C_{n\beta}$	$+0.1040 \pm 0.0060$
C_{yp}	$+0.6323 \pm 0.1251$	C_{lp}	-0.6586 ± 0.0907	C_{np}	$+0.0427 \pm 0.0168$
C_{yr}	$+0.1951 \pm 0.0888$	C_{lr}	$+0.1365 \pm 0.0500$	C_{nr}	-0.1511 ± 0.0111
$C_{y\delta a}$	$+0.2491 \pm 0.0508$	$C_{l\delta a}$	-0.2729 ± 0.0393	$C_{n\delta a}$	$+0.0522 \pm 0.0061$
$C_{y\delta r}$	$+0.1544 \pm 0.0124$			$C_{n\delta r}$	-0.0726 ± 0.0051
$C_{y\beta^3}$	-0.2294 ± 0.3115			$C_{n\beta^3}$	$+0.1777 \pm 0.1551$

2. Linear Aerodynamic Model

To obtain linear differential equations, the nonlinear equations were linearized about the nominal cruise trim condition of $V_t = 45$ ft/s, $\alpha = 5^\circ$, $\delta_e = -3^\circ$, and with other states at zero. The linear model differential equations adjusted to the form of Eq. (3) are:

$$\dot{\mathbf{x}} = [\dot{p} \quad \dot{q} \quad \dot{r} \quad \dot{u} \quad \dot{v} \quad \dot{w} \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T, \quad \mathbf{u} = [\delta_a \quad \delta_e \quad \delta_r]^T$$

$$\mathbf{A} = \begin{bmatrix} -2.84 & 0 & 0.65 & 0 & -0.09 & 0 & 0 & 0 & 0 \\ 0 & -0.51 & 0 & -0.29 & 0 & -2.86 & 0 & 0 & 0 \\ 0.42 & 0 & -0.47 & 0 & 0.10 & 0 & 0 & 0 & 0 \\ 0 & -4.62 & 0 & -0.12 & 0 & 1.29 & 0 & 32.1 & 0 \\ 7.35 & 0 & -52.0 & 0 & -0.55 & 0 & 32.1 & 0 & 0 \\ 0 & 52.8 & 0 & -1.11 & 0 & -5.92 & 0 & 2.81 & 0 \\ 1.00 & 0 & 0.09 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.00 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.00 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -21.3 & 0 & 0.59 \\ 0 & -65.0 & 0 \\ 4.78 & 0 & -3.54 \\ 0 & -0.23 & 0 \\ 4.05 & 0 & 2.51 \\ 0 & -344 & 0 \\ 0.00 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The developed linear model is valid for small perturbations in the normal operating range near the trim condition.

B. Feedback Control Laws

A build up approach was used to develop confidence in the techniques and establish team flight test discipline. The proportional-derivative (PD) attitude command control law phase of the flight test program was straightforward and confirmed the feedback control application of the co-computer. Next, the effort progressed to advanced control law implementation. The linear quadratic regulator (LQR) phase of the program added complexity, and also confirmed the utility of the model developed during system identification. The effort culminated in tuning and demonstrating the nonlinear control law.

The advanced control laws were designed to stabilize the motion of the aircraft to a desired direction in inertial space corresponding to a climb angle and a course angle. The desired flight profile consists of wings-level flight at zero sideslip and a fixed angle of attack. Rather than solving for an angle of attack which would satisfy equilibrium conditions for a prescribed airspeed, the airspeed which achieves a prescribed angle of attack is solved for instead. The former approach requires a numerical solution for every trim condition whereas the latter has an analytical solution which can be easily evaluated for any trim.

Once the control law is engaged, the aircraft should maintain its current heading and a pitch angle offset based on the desired angle of attack, to achieve zero climb angle. In order to allow the pilot some control authority while experimenting with the control laws or in case of contingencies, the RC control sticks, rather than directly controlling the control surfaces, would effectively control the desired roll, pitch, and yaw of the aircraft and allow for varying altitude in-flight. This allows the pilot to fly the aircraft similarly to “fly-by-wire” modes and would moreover allow the pilot to introduce disturbances to the control law to test its effectiveness in recovering from perturbations. For this effort, airspeed was controlled by pilot throttle adjustment, rather than feedback control using a thrust model. The throttle is manually set to a value which roughly corresponds to the desired airspeed value at steady state. Note that the control law presented in Ref. [1] requires active closed-loop control of airspeed. The flight test results described here effectively incorporate a ‘thrust disturbance’ relative to the control law proposed in Ref. [1]. A follow on effort is planned to control the velocity using a thrust model and real-time sensing of the propellers’ RPM with an inner control loop. Instrumentation and development of a thrust model from flight data are discussed in Ref. [48].

1. Proportional-Derivative Attitude Control

The PD controller was implemented first to establish confidence in the on-board controller architecture. The simplicity of a non-model-based controller allowed for isolation, testing, and evaluation of the co-computer's ability to read data from the Pixhawk and send actuator commands with little delay and at a reliable update rate. The PD feedback controller was implemented as described in Sec. II.B.1 and tuned to have similar handling qualities to that of the built-in "stabilized" mode of the Pixhawk PX4 autopilot described in Ref. [57].

Contrary to the control strategy outlined for the advanced control laws, the PD controller functions as an attitude controller, taking the attitude and rotational velocity as feedback while ignoring the aerodynamic angles. This was done to replicate the performance of the built in Pixhawk controller which does not use the aerodynamic angles. Hence, the ailerons asymptotically stabilize a particular roll angle, and the elevator asymptotically stabilize a particular pitch angle. The rudder was controlled directly by pilot inputs, and thus there are no gains for yaw feedback control. The gains used for the PD controller can be found in Table 6.

2. Linear Quadratic Regulator

After establishing confidence in the communication between the Pixhawk and the co-computer, a control strategy was implemented which leveraged the model of the aircraft to confirm the utility of the model derived from system identification. The LQR gain matrix was tuned initially in simulation, using the identified linear flight dynamic model, so that handling qualities would be comparable to that of the PD control law already developed and of the "stabilized" mode in the PX4 autopilot. The simulation environment used linearized aircraft equations of motion for preliminary tuning. Once agreeable performance was attained, the control law was tested in a simulation environment with the full nonlinear dynamics for further tuning. Afterwards, the LQR was tested on the physical platform in flight and required additional fine-tuning before the performance was satisfactory. A trajectory of a horizontal S-maneuver demonstrating satisfactory cross-track control in flight is shown in Fig. 15. Data from a test maneuver with LQR engaged is shown in Fig. 16. The maneuver demonstrates satisfactory first order response in tracking bank angle, pitch attitude, and heading angle.

Table 6 PD controller gains

Gain	K_p	K_d
roll	1.1	0.08
pitch	1.0	0.08

Table 7 LQR gain matrix

K	<i>p</i>	<i>q</i>	<i>r</i>	<i>u</i>	<i>v</i>	<i>w</i>	ϕ	θ	ψ
δ_a	0.257	0	-0.260	0	0.051	0	1.010	0	0.451
δ_e	0	-0.027	0	-0.011	0	0.055	0	-1.220	0
δ_r	0.090	0	0.506	0	-0.005	0	0.149	0	0.468

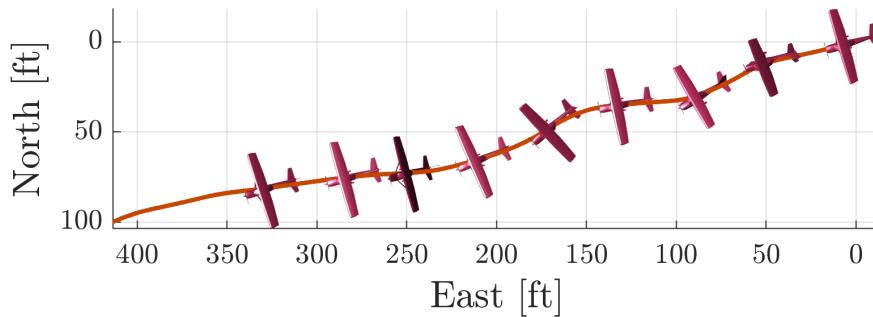


Fig. 15 Trajectory of the aircraft while LQR was engaged.

3. Port-Hamiltonian Energy-Based Nonlinear Control

Having verified the aircraft model and implementation of advanced feedback control using LQR, the nonlinear control scheme presented in Ref. [1] was incorporated into the co-computer. In simulation, the nonlinear control law showed the ability to stabilize the motion of the aircraft with constant thrust. The control law was implemented

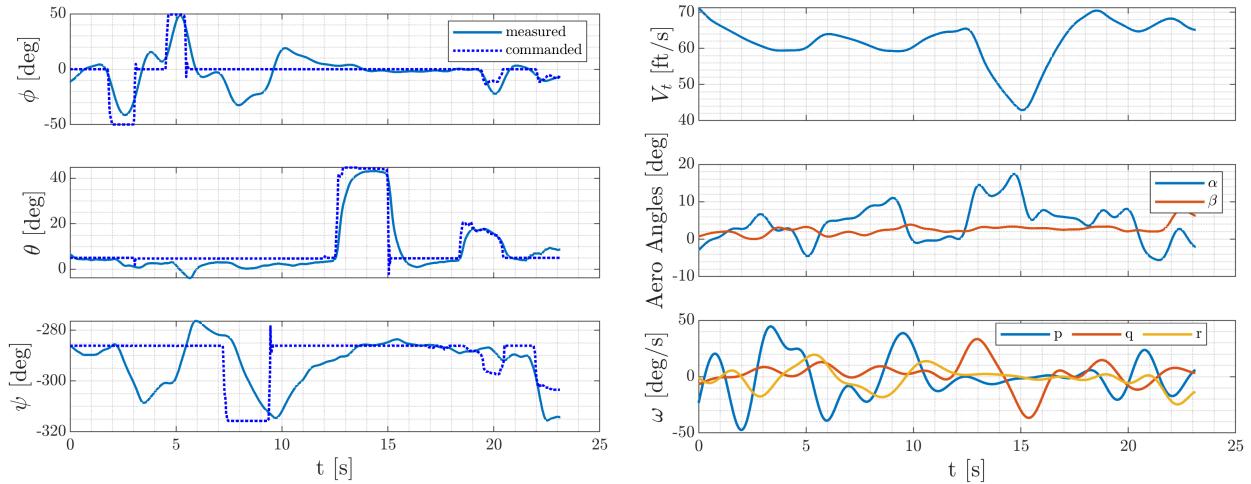


Fig. 16 Experimental results for LQR.

on the research aircraft in flight after achieving good performance in simulation. Oscillations in roll and pitch were initially observed. The discrepancy between the simulated and physical aircraft response was attributed to un-modeled phenomena such as measurement noise and time delays, and mass properties and aerodynamic modeling errors. The relevant gains were tuned to attenuate the adverse effects observed while progressing through a series of step inputs with the aircraft in flight. The process was carried out using the transmitter's variable knob to adjust the magnitude of select gains while the aircraft was airborne. The controller uses the parameters k_v , k_ψ , and k_ζ to determine the controller's stiffness with respect to errors in the aerodynamic angles, the heading, and the aircraft's tilt, determined by pitch and roll, respectively (see [1] for details). The controller also uses the damping matrix $\mathbf{C} = \text{diag}(C_p, C_q, C_r)$. The values of the nonlinear control parameters can be found in Table 8.

Table 8 Nonlinear control gains

Gain	k_v	k_ζ	k_ψ	C_p	C_q	C_r
Value	53.7	125	814	57.2	42.7	102

To demonstrate adequate performance of the final tuned control law, a demonstration maneuver was created with a series of long duration singlet inputs. The control law was engaged with the aircraft on a desired heading, then the commanded heading was shifted to a new desired course, then the commanded climb angle was adjusted, then commanded roll angle was adjusted. (The control law was designed for wings-level flight.) The resulting trajectory and pertinent time trace data are shown in Figs. 17 and 18.

VI. Conclusion

A flight test research aircraft was developed with a Pixhawk flight computer and Raspberry Pi co-computer. Software was developed to implement automated system identification excitations and feedback control. The tools used are offered in a publicly available web repository as a companion product to this paper. This investigation demonstrated increasingly complex feedback control for a sUAV using a the Pixhawk flight computer and Raspberry Pi co-computer. Results show a high quality model was developed using automated excitations inputs. Additionally, results show the tuned linear and nonlinear control laws met the design objective of performance similar to the baseline PX4 “stabilized” mode autopilot. The flight test results include the first experimental demonstration of a new energy-based, nonlinear flight control law described in an earlier publication.

Ongoing work includes comprehensive flight testing of the nonlinear, energy-based path following control law described here with both the MTD fixed-wing UAV and a multi-copter air vehicle. Additionally, this process will be implemented with more complex aircraft, additional air-data sensors, and other control laws.

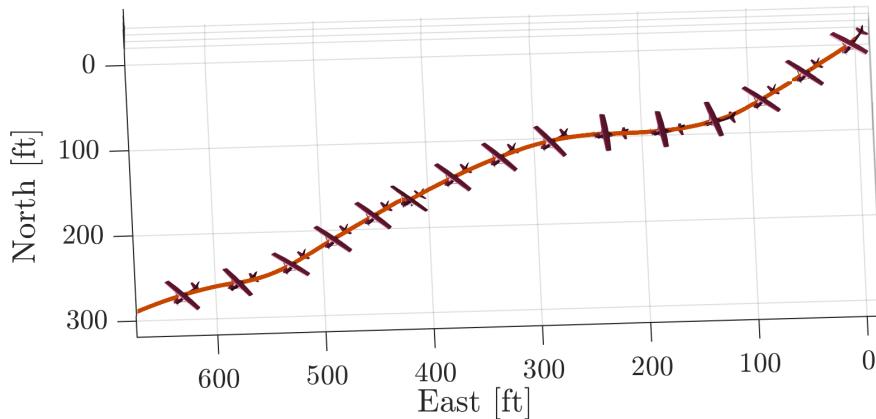


Fig. 17 Trajectory of the aircraft while the nonlinear controller was engaged.

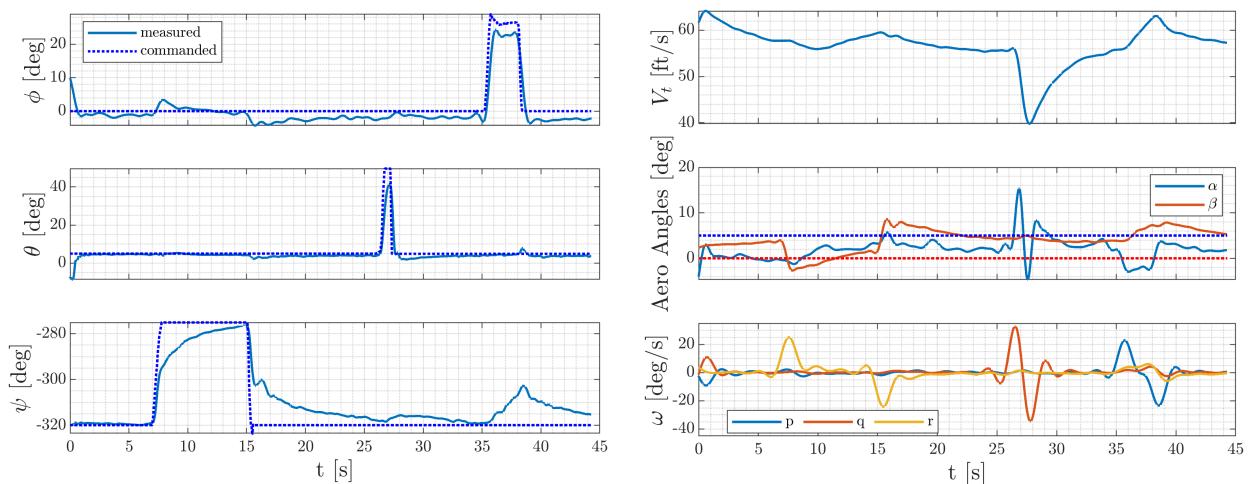


Fig. 18 Experimental results for the nonlinear controller.

Acknowledgments

The authors wish to thank Mekonen Halefom and Sourav Sinha for experiment support. The authors also gratefully acknowledge financial support from the United States Air Force Test Pilot School under USAF Contract No. FA9302-18-P-5227 and from NASA under Grant No. 80NSSC20M0162. The first author is grateful to the United States Air Force and the Dominion Energy Research Fellowship at Virginia Tech for financial support.

References

- [1] Fahmi, J.-M., and Woolsey, C. A., “Port-Hamiltonian Flight Control of a Fixed-Wing Aircraft,” *IEEE Transactions on Control Systems Technology*, 2021, pp. 1–8.
- [2] Gresham, J. L., Simmons, B. M., Fahmi, J.-M. W., and Woolsey, C. A., “Remote Uncorrelated Pilot Inputs for Nonlinear Aerodynamic Model Identification from Flight Data,” *AIAA AVIATION 2021 FORUM*, AIAA Paper 2021-2792, 2021, pp. 1–26. <https://doi.org/10.2514/6.2021-2792>.
- [3] Virginia Tech Nonlinear System Laboratory, “NSL@VT GitHub,” , 2021. URL <https://github.com/NSL-VT>.
- [4] Smith, G., Bixler, B., Babcock, J. T., Osterroos, R., McLaughlin, T. E., and Tischler, M. B., “System Identification of the ICE/SACCON UAS Aircraft,” *AIAA Scitech 2020 Forum*, AIAA Paper 2020-0289, 2020, pp. 1–17. <https://doi.org/10.2514/6.2020-0289>.
- [5] Tischler, M. B., and Remple, R. K., *Aircraft and Rotorcraft System Identification: Engineering Methods With Flight-Test Examples*, AIAA, Reston ,VA, 2006. <https://doi.org/10.2514/4.861352>.

- [6] Schulze, P. C., Miller, J., Klyde, D. H., Regan, C. D., and Alexandrov, N., "System Identification of a Small UAS in Support of Handling Qualities Evaluations," *AIAA Scitech 2019 Forum*, AIAA Paper 2019-0826, 2019, pp. 1–16. <https://doi.org/10.2514/6.2019-0826>.
- [7] Lu, H. H., Rogers, C., Goecks, V. G., and Valasek, J., "Online Near Real Time System Identification on a Fixed-Wing Small Unmanned Air Vehicle," *2018 AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper 2018-0295, 2018, pp. 1–18. <https://doi.org/10.2514/6.2018-0295>.
- [8] Kirkpatrick, K., May, J., and Valasek, J., "Aircraft System Identification Using Artificial Neural Networks," *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2013-878, 2013, pp. 1–10. <https://doi.org/10.2514/6.2013-878>.
- [9] Harris, J., Arthurs, F., Henrickson, J. V., and Valasek, J., "Aircraft System Identification Using Artificial Neural Networks with Flight Test Data," *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE Paper ICUAS.2016.7502624, 2016, pp. 679–688. <https://doi.org/10.1109/ICUAS.2016.7502624>.
- [10] Scheper, K. Y., Chowdhary, G., and Johnson, E. N., "Aerodynamic System Identification of Fixed-wing UAV," *AIAA Atmospheric Flight Mechanics (AFM) Conference*, AIAA Paper 2013-4920, 2013, pp. 1–10. <https://doi.org/10.2514/6.2013-4920>.
- [11] Klenke, R., McBride, J., and Nguyen, H., "A Reconfigurable, Linux-Based, Flight Control System for Small UAVs," *AIAA Infotech@Aerospace 2007 Conference and Exhibit*, AIAA Paper 2007-2957, 2007, pp. 1–10. <https://doi.org/10.2514/6.2007-2957>.
- [12] Safwat, E., Weiguo, Z., Kassem, M., and Mohsen, A., "Robust Nonlinear Flight Controller For Small Unmanned Aircraft Vehicle based on Incremental BackStepping," *AIAA Scitech 2020 Forum*, AIAA Paper 2020-0854, 2020, pp. 1–12. <https://doi.org/10.2514/6.2020-0854>.
- [13] Safwat, E., Abozied, M. A., and Kamel, A., "Design and Analysis of A Nonlinear Guidance Law for Small UAV," *AIAA Scitech 2021 Forum*, AIAA Paper 2021-0372, 2021, pp. 1–14. <https://doi.org/10.2514/6.2021-0372>.
- [14] D'Antuono, V., DeMatteis, G., Trotta, D., and Zavoli, A., "Optimal Tuning for Robust Control of a Small Fixed-Wing UAV," *AIAA Scitech 2021 Forum*, AIAA Paper 2021-1057, 2021, pp. 1–17. <https://doi.org/10.2514/6.2021-1057>.
- [15] Sanders, F. C., Tischler, M., Berger, T., Berrios, M. G., and Gong, A., "System Identification and Multi-Objective Longitudinal Control Law Design for a Small Fixed-Wing UAV," *2018 AIAA Atmospheric Flight Mechanics Conference*, AIAA Paper 2018-0296, 2018, pp. 1–23. <https://doi.org/10.2514/6.2018-0296>.
- [16] Ljung, L., *System Identification: Theory for the User*, 2nd ed., Prentice Hall PTR, Upper Saddle River, NJ, 1999.
- [17] Morelli, E. A., and Klein, V., *Aircraft System Identification: Theory and Practice*, 2nd ed., Sunflyte Enterprises, Williamsburg, VA, 2016.
- [18] Simmons, B. M., McClelland, H. G., and Woolsey, C. A., "Nonlinear Model Identification Methodology for Small, Fixed-Wing, Unmanned Aircraft," *Journal of Aircraft*, Vol. 56, No. 3, 2019, pp. 1056–1067. <https://doi.org/10.2514/1.C035160>.
- [19] Klein, V., Batterson, J. G., and Murphy, P. C., "Determination of Airplane Model Structure from Flight Data by Using Modified Stepwise Regression," Technical Report NASA-TP-1916, NASA, 1981.
- [20] Morelli, E. A., "Global Nonlinear Aerodynamic Modeling Using Multivariate Orthogonal Functions," *Journal of Aircraft*, Vol. 32, No. 2, 1995, pp. 270–277. <https://doi.org/10.2514/3.46712>.
- [21] Jategaonkar, R., *Flight Vehicle System Identification: A Time Domain Methodology*, 2nd ed., AIAA/Aerospace Press, Reston, VA, 2015. <https://doi.org/10.2514/4.102790>.
- [22] Milliken, W. F. J., "Progress in Dynamic Stability and Control Research," *Journal of Aeronautical Sciences*, Vol. 14, No. 9, 1947, pp. 493–519.
- [23] Schroeder, M., "Synthesis of Low-Peak-Factor Signals and Binary Sequences with Low Autocorrelation (Corresp.)," *IEEE Transactions on Information Theory*, Vol. 16, No. 1, 1970, pp. 85–89. <https://doi.org/10.1109/TIT.1970.1054411>.
- [24] Klein, V., "Estimation of Aircraft Aerodynamic Parameters from Flight Data," *Progress in Aerospace Sciences*, Vol. 26, No. 1-A, 1989, pp. 1–77.
- [25] Young, P., and Patton, R. J., "Comparison of Test Signals for Aircraft Frequency Domain Identification," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 3, 1990, pp. 430–438. <https://doi.org/10.2514/3.25355>.

- [26] Muzzey, C., and Kidd, E., "Measurement and Interpretation of Flight Test Data for Dynamic Stability and Control," *AGARD Flight Test Manual*, Vol. 2, Pergamon, New York, NY, 1959, Chap. 11, pp. 1–95.
- [27] Morelli, E. A., "Multiple Input Design for Real-Time Parameter Estimation in the Frequency Domain," *13th IFAC Conference on System Identification*, 2003, pp. 1–6. [https://doi.org/10.1016/S1474-6670\(17\)34833-4](https://doi.org/10.1016/S1474-6670(17)34833-4).
- [28] Morelli, E. A., "Flight-Test Experiment Design for Characterizing Stability and Control of Hypersonic Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, 2009, pp. 949–959. <https://doi.org/10.2514/1.37092>.
- [29] Morelli, E., "Efficient Global Aerodynamic Modeling from Flight Data," *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2012-1050, 2012, pp. 1–21. <https://doi.org/10.2514/6.2012-1050>.
- [30] Ogata, K., *Modern Control Engineering*, 5th ed., Prentice Hall, Boston, MA, 2010.
- [31] "Using the Estimation and Control Library Extended Kalman Filter," https://docs.px4.io/master/en/advanced_config/tuning_the_ecl_ekf.html, Accessed Feb 2021.
- [32] Riseborough, P., "PX4 State Estimation Update," Jun. Jun. 2019. URL <https://www.youtube.com/watch?v=HkYRJJoyBwQ>, Accessed 21 November 2021.
- [33] Ziegler, J. G., and Nichols, N. B., "Optimum Settings for Automatic Controllers," *Transactions of the A.S.M.E.*, Vol. 64, 1942, pp. 759–765.
- [34] van der Schaft, A., *L2-Gain and Passivity Techniques in Nonlinear Control*, Vol. 2, Springer, 2000.
- [35] Gonzalez, H., Duarte-Mermoud, M. A., Pelissier, I., Travieso-Torres, J. C., and Ortega, R., "A Novel Induction Motor Control Scheme Using IDA-PBC," *Journal of Control Theory and Applications*, Vol. 6, No. 1, 2008, pp. 59–68.
- [36] Neves, L. C., Paim, G. V., Queinnec, I., Moreno, U. F., and Pieri, E. R. D., "Passivity and Power Based Control of a Robot with Parallel Architecture," *IFAC Proceedings Volumes*, Vol. 44, No. 1, 2011, pp. 14608–14613.
- [37] Valentinis, F., Donaire, A., and Perez, T., "Energy-Based Motion Control of a Slender Hull Unmanned Underwater Vehicle," *Ocean Engineering*, Vol. 104, 2015, pp. 604–616.
- [38] Valentinis, F., Donaire, A., and Perez, T., "Energy-Based Guidance of an Underactuated Unmanned Underwater Vehicle on a Helical Trajectory," *Control Engineering Practice*, Vol. 44, 2015, pp. 138–156.
- [39] Valentinis, F., and Woolsey, C., "Nonlinear Control of a Subscale Submarine in Emergency Ascent," *Ocean Engineering*, Vol. 171, 2019, pp. 646–662.
- [40] Acosta, J. A., Sanchez, M. I., and Ollero, A., "Robust Control of Underactuated Aerial Manipulators via IDA-PBC," *53rd IEEE Conference on Decision and Control*, IEEE, 2014, pp. 673–678.
- [41] Guerrero, M. E., Mercado, D. A., Lozano, R., and García, C. D., "IDA-PBC Methodology for a Quadrotor UAV Transporting a Cable-Suspended Payload," *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2015, pp. 470–476.
- [42] Mersha, A. Y., Carloni, R., and Stramigioli, S., "Port-Based Modeling and Control of Underactuated Aerial Vehicles," *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 14–19.
- [43] Yüksel, B., Secchi, C., Bülthoff, H. H., and Franchi, A., "Reshaping the Physical Properties of a Quadrotor Through IDA-PBC and its Application to Aerial Physical Interaction," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 6258–6265.
- [44] Muñoz, L. E., Santos, O., Castillo, P., and Fantoni, I., "Energy-Based Nonlinear Control for a Quadrotor Rotorcraft," *2013 American Control Conference*, IEEE, 2013, pp. 1177–1182.
- [45] "Cube Orange Flight Controller," https://docs.px4.io/master/en/flight_controller/cubepilot_cube_orange.html, Accessed May 2021.
- [46] Miller, M. P., "An Accurate Method of Measuring the Moments of Inertia of Airplanes," *NACA Technical Note*, 1930.
- [47] "Kentland Experimental Aerial Systems (KEAS) Laboratory," <https://www.aoe.vt.edu/research/facilities/keas.html>, Accessed Apr 2021.
- [48] Simmons, B. M., Gresham, J. L., and Woolsey, C. A., "Aero-Propulsive Modeling for Propeller Aircraft Using Flight Data," *AIAA SciTech 2022 Forum*, 2022, pp. 1–24. To be published.

- [49] Simmons, B. M., “System Identification of a Nonlinear Flight Dynamics Model for a Small, Fixed-Wing UAV,” Master’s thesis, Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, VA, 2018.
- [50] Gresham, J. L., Simmons, B. M., Hopwood, J. W., and Woolsey, C. A., “Spin Aerodynamic Modeling for a Fixed-Wing Aircraft Using Flight Data,” *AIAA SciTech 2022 Forum*, 2022, pp. 1–15. To be published.
- [51] “System IDentification Programs for AirCraft (SIDPAC),” <https://software.nasa.gov/software/LAR-16100-1>, Accessed 21 November 2021.
- [52] MATLAB, *MATLAB Documentation, version 9.9.0.1538559 (R2020b Update 3)*, The MathWorks Inc., Natick, Massachusetts, 2020.
- [53] “ULog File Format,” https://docs.px4.io/master/en/dev_log/ulog_file_format.html, Accessed 22 November 2021.
- [54] Steers, S. T., and Iliff, K. W., “Effects of Time-shifted Data on Flight Determined Stability and Control Derivatives,” Tech. Rep. NASA TN D-7830, NASA, Mar. 1975.
- [55] Morelli, E. A., “Dynamic Modeling from Flight Data with Unknown Time Skews,” *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 8, 2017, pp. 2084–2092. <https://doi.org/10.2514/1.G002008>.
- [56] Morelli, E. A., “Estimating Noise Characteristics from Flight Test Data Using Optimal Fourier Smoothing,” *Journal of Aircraft*, Vol. 32, No. 4, 1995, pp. 689–695. <https://doi.org/10.2514/3.46778>.
- [57] “PX4 Stabilized Mode (Fixed Wing),” https://docs.px4.io/v1.12/en/flight_modes/stabilized_fw.html, Accessed Nov 2021.