Synthesis and Flight Test of Automatic Landing Controller Using Quantitative Feedback Theory

Timothy Woodbury* and John Valasek†
Texas A&M University, College Station, Texas 77843-3141

DOI: 10.2514/1.G001758

Landing is a challenging flight phase for automatic control of fixed-wing aircraft. For unmanned air vehicles in particular, it is imperative that model uncertainty be considered in the control synthesis. These vehicles tend to have limited sensors and instrumentation yet must achieve sufficient performance in the presence of modeling uncertainties and exogenous inputs such as turbulence. Quantitative feedback theory has been reported in the literature for design of automatic landing control laws, but none of these controllers has been flight-tested. In this paper, quantitative feedback theory is employed to synthesize robust discrete-time controllers for automatic landing of an unmanned air vehicle. A low-cost flight vehicle with standard aileron, rudder, elevator, and throttle controls is used. Dynamic simulation is conducted using uncertain aircraft models and sensor noise profiles derived from flight hardware. Controllers are initially synthesized in deterministic simulations. Control validation is performed using a Monte Carlo analysis of stochastic simulations. Sources of uncertainty considered are sensor noise, model uncertainty, and static winds. Landing-phase simulations presented in this paper indicate a routinely high probability of a successful landing in relatively calm wind conditions. The flight-testing process is discussed, and time histories from two automatic landings are presented. Dynamic responses in flight test are found to be similar to the simulation, but a significant amount of control redesign is still required to achieve adequate experimental performance. The methodology is judged to be a promising candidate for an automatic landing controller for unmanned air vehicles.

Nomenclature

P	=	aircraft roll rate
Q	=	aircraft pitch rate
R	=	aircraft yaw rate
X	=	inertial position vector of aircraft along the inertial x axis
$\boldsymbol{x}$	=	vector x
Y	=	inertial position vector of aircraft along the inertial y axis
Z	=	inertial position vector of aircraft along the inertial z axis
α	=	angle of attack
β	=	sideslip angle
ϕ	=	aircraft roll angle
Γ	=	aircraft glide slope in inertial space
δ_a	=	aileron deflection
δ_e	=	elevator deflection
δ_r	=	rudder deflection
δ_t	=	throttle setting
θ	=	aircraft pitch angle
λ	=	aircraft azimuth in inertial space
ψ	=	aircraft heading angle

I. Introduction

APPROACH and landing are among the most critical flight regimes for automatic control of fixed-wing aircraft. As in any flight phase, the effects of external disturbances, which appear primarily as air turbulence and static wind, must be mitigated. Landing should not damage the aircraft. Finally, performance of the controller should be repeatable for many flights in potentially different environmental conditions. Additional challenges are introduced when landing lightweight unmanned air vehicles (UAVs). Accurate models are not always available because traditional design methods do not generally scale down reliably. Parameter estimation may not be feasible without wind-tunnel testing or computational-fluid-dynamics modeling. Further, this scale of aircraft is limited in payload capacity and therefore is limited in the quality of sensor data available for system identification from flight data. These sensor limitations also constrain any automatic controllers. Each of these challenges must be accounted for in the control design and data management.

Design of automatic landing controllers for manned fixed-wing aircraft is a problem that dates back several decades [1]. Recently, automatic landing for smaller unmanned research aircraft has become a topic of research attention. A variety of different control and sensor approaches have been presented. The literature generally focuses on innovative applications of sensors with relatively traditional control approaches.

Barrows et al. [2] describe flight tests in which optical flow is used for terrain following and attitude estimation at low altitudes. In [3], an inexpensive attitude filter for a UAV is designed using a GPS receiver, accelerometer, and gyroscope with an extended Kalman filter. Kim et al. [4] explore the use of inexpensive inertial measurement unit (IMU) and GPS hardware for navigation and control. Flight-test results are presented for nominal maneuvering, but landing is performed by a remote operator. Barber et al. [5] experiment with landings using a fusion of barometric sensors and optical flow to estimate altitude and demonstrated repeated landings within meters of a target. Proportional–integral–derivative (PID) loops are used for longitudinal-axis control with a lateral/directional control method based on course vector fields. This work is extended in [6], which presents a vision-based system for longitudinal and lateral control of an unmanned aircraft during landing. A bias correction scheme is implemented to account for errors in relative orientation, parameter estimation, wind estimation, and target velocity estimation. In flight test, the vehicle successfully demonstrated landing onto both fixed and moving platforms and consistently landed within 5 m of the target position. Roos and Peddle [7] perform autonomous takeoff and landing of a lightweight (less than 5 kg) air vehicle using onboard processing only. Sequential feedback loop closures are used for guidance and control. An ultrasonic rangefinder is used for landing to obtain precise altitude values in the absence of optical flow or visual

Presented as Paper 2015-1073 at the AIAA Guidance, Navigation and Control Conference, Kissimmee, FL, 5–9 January 2015; received 5 October 2015; revision received 5 March 2016; accepted for publication 12 April 2016; published online 20 July 2016.

*Graduate Research Assistant, Land, Air, and Space Robotics Laboratory, Department of Aerospace Engineering; twoodbury@tamu.edu. Student Member AIAA.

†Professor and Director, Vehicle Systems & Control Laboratory, Department of Aerospace Engineering; valasek@tamu.edu. Associate Fellow AIAA.

imagery. Huh and Shim [8] use a vision-based system for visual servoing control of pitch and heading angles to land a micro air vehicle (MAV) on a small inflatable dome. Laiacker et al. [9] present an automatic landing scheme for UAVs in which a vision algorithm for runway detection is fused with GPS. Flight-test results are included. Thurrowgood et al. [10] use a combination of optic flow and stereo vision to achieve automatic landing of a UAV in unstructured environments. Senpheng and Ruchanurucks present an application of edge detection for identifying runways using vision [11].

Other recent papers have emphasized applications of automatic control. Fuzzy-logic controllers are demonstrated in simulation in the presence of disturbances [12] and for control of a nonlinear aircraft model [13]. Similarly, Lungu et al. [14] present fuzzy-logic PID controllers with dynamic inversion for automatic longitudinal control during landing. Simulations are conducted in the presence of wind shear and sensor uncertainty. Joos et al. [15] present a constrained nonlinear model predictive control architecture for automatic landing. This solution is implemented on field-programmable gate arrays. This is a guidance solution, and the air vehicle is modeled as already having inner-loop control. Various constraints, arising from both flight mechanics and from collision avoidance, are considered. Hardware-in-the-loop testing is presented. You et al. present a line-of-sight-based guidance law with inner-loop control augmented by L1 adaptive control for precision landing with an arresting hook [16]. Both hardware-in-loop and flight-test results of the full system are presented. Jianfeng and Caijuan [17] present the control design for an automatic landing controller. The dynamic model is identified from wind-tunnel test data. Linear multi-input/multi-output feedback controllers are implemented, and simulation results are presented. In [18], an adaptive neural network using online learning is employed as the primary longitudinal-axis control for automatic landing of a simulated full-scale aircraft. Good disturbance rejection capabilities are demonstrated, and a real-time implementation of the algorithm on embedded hardware is shown.

Quantitative feedback theory (QFT) is a robust control methodology for synthesizing control laws for linear plants. QFT guarantees satisfaction of frequency domain tracking, stability, and disturbance rejection requirements for a set of uncertain plants specified by the control designer [19]. Much of the QFT work in the literature focuses on fault-tolerant control that is to be applied over a wide range of flight regimes. In [20], use of QFT was motivated by a desire for a single feedback system to be used across the aircraft's flight envelope. A complete design process for a single-input/single-output (SISO) synthesis, including plant uncertainty and flight-test results, is presented for a pitch axis controller for the U.S. Air Force "Lambda" UAV. Keating et al. [21] expand upon the previous work to develop a longitudinal-axis regulator with robustness to elevator damage. A pitch-axis regulator is developed, with no additional outside control loops. A complete SISO design is presented in [22], in which an existing longitudinal-axis controller for a remotely piloted vehicle is improved using QFT. Again, both variations in operating conditions and fault-induced loss of control effectiveness are considered. This paper gives the stability and tracking requirements used for synthesis; it also indicates that disturbance rejection performance can frequently be satisfied as a byproduct of tracking specifications. Houpis and Rasmussen [23] designed robust inner-loop controllers for the Lambda UAV using QFT to achieve robustness to varying flight regimes. The design work included both longitudinal and lateral/directional control designs with flight tests and multiple iterations; however, only a fraction of the design process is presented. Santander and Aranda [24] present multi-input/multi-output (MIMO) longitudinal and lateral/directional QFT controllers validated in nonlinear simulation. More recently, Schuck et al. [25] use QFT to achieve level 1 handling requirements for a manual control of a manned aircraft. Although the utility of QFT for MIMO control design has been questioned [26], recent results in the literature suggest that the QFT approach provides a sound basis for low-dimensional MIMO systems [27]. Furthermore, the history of successful QFT designs in flight-test research supports the viability of this approach for control synthesis. However, it should be noted that the QFT process tends to be conservative, which can make synthesis for high-order systems difficult or impractical [27].

The good, simulation-based results of [28] motivate and form a basis for the present implementation and flight-testing work. In [28], sequential loop closures are used to develop QFT-based control laws for automatic landing of a Rockwell Commander 700 aircraft in simulation. The autopilot is intended for a general aviation class aircraft and therefore uses a very different sensor package than would be available on a UAV. The fundamental control structure, however, can be adapted to any vehicle. The thesis of [29] applies the QFT automatic landing control law to two UAVs and is the basis for the current paper. Simulation results are shown for both UAVs, and flight-test results are shown for one.

The objective of the current paper is to evaluate the suitability of QFT for practical control design in small-scale UAVs. The recent increase in lightweight autopilots makes small-scale UAVs attractive platforms for automatic controls testing, due to their low costs and ease of operation. Model-based control for these vehicles remains challenging due to the difficulty of identifying accurate dynamic models. These vehicles also have greater susceptibility to aerodynamic disturbances than more traditional, larger flight vehicles [30]. Computationally simple control is also considered advantageous because makes a control law relatively simple to integrate with existing autopilot sensor loops. In light of these considerations, QFT has three attractive properties. 1) It can directly synthesize linear, discrete-time control laws suitable for flight-test implementation. 2) It explicitly addresses plant uncertainty specified by the designer. 3) It is reported to offer good disturbance rejection as a side effect of addressing robust tracking.

Additionally, it is noted that no flight-test results using QFT for automatic landing are reported in the literature, outside of the authors' other work. The current paper synthesizes, develops, and flight tests an automatic landing controller. Motivated by a desire for computationally simple robust control, QFT is used for the majority of the control synthesis. For risk reduction and ease of testing, the control law is designed for and flown on a low-cost Easy Star platform that has a full set of standard control surfaces and can support a lightweight autopilot (Fig. 1).

This paper is organized as follows. Section II introduces the QFT theory and synthesis methodology. Section III details the specifics of the approach and landing problem. Section IV describes the vehicle and sensor modeling. Section V presents the digital controller synthesis and Monte Carlo results, and Sec. VI presents the flight-test results. Conclusions are presented in Sec. VII.

II. Quantitative Feedback Theory

Quantitative feedback theory (QFT) is a control synthesis methodology for satisfying closed-loop performance specifications

Fig. 1 Hobby King Bixler flight vehicle modified for autonomous flight testing. The Bixler has the same configuration and nearly identical dimensions to the Multiplex Easy Star.

in the presence of quantified plant uncertainty and/or disturbances. QFT is a graphical control synthesis method, and therefore a CAD process, in which plots of the system response are manipulated by the designer to achieve a desired shape that remains outside boundaries defined by performance requirements. QFT, at its most fundamental level, designs linear control laws for single-input/single-output (SISO) linear plants; this formulation is readily extended to multi-input/single-output and multi-input/multi-output (MIMO) linear plants as well as nonlinear plants. Both continuous-time and discrete-time controllers can be synthesized.

A generic SISO QFT feedback loop is shown in Fig. 2. This loop consists of a prefilter F, a feedback controller G, a plant P, and sensor dynamics H. This is the assumed form of the control loops used in the control synthesis. The QFT design process is to add, remove, and change the poles and zeros of the control elements to shape the open- and closed-loop responses of the feedback system.

The Nichols plot is a tool used extensively in QFT; a representative example is shown in Fig. 3. The Nichols plot shows the open-loop response magnitude of a transfer function $Q(s)$ versus the open-loop phase on the x axis. Contours defining the closed-loop $Q(s)/[1 + Q(s)]$ response magnitude can also be computed. The Nichols plot is used both in defining the plant uncertainty and performance requirements and during the actual control synthesis.

To account for plant uncertainty and performance requirements, the control designer specifies representative discrete sets of frequencies and plant transfer functions. Selection of these sets is largely a matter of the designer's discretion; ideally, one would choose the smallest discrete sets for which satisfaction of the performance requirements also ensures satisfaction for the physical system at all operational frequencies. From a practical standpoint, selection of the frequency set generally involves defining an initial set based on intuition or the problem background, designing a preliminary controller, and then iterating on the frequency set. For example, an initial controller may be synthesized that satisfies performance requirements for the discrete frequency set but does not satisfy requirements at intermediate frequencies. Frequencies near these points should be added, and some frequencies for which the requirements are trivially satisfied may also be removed from the set to streamline computation. Plant uncertainty is often known in terms of a range of parametric uncertainty for the plant transfer function(s). In this case, a set of factorial combinations of the chosen discrete values of the variable parameters can be defined. The response of a candidate set of plants can be plotted on a Nichols plot for the designer to evaluate if the set is representative of the expected continuous set of plant responses. The Nichols plots of the set of plants evaluated at a given frequency ω is termed the plant template at ω.

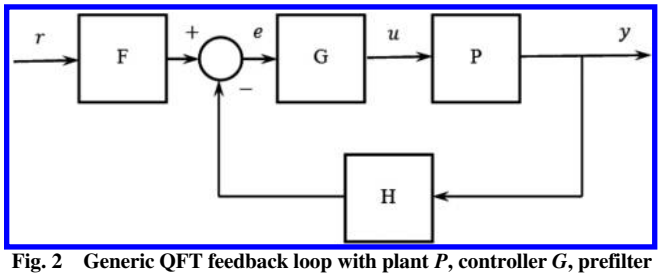

Fig. 2 Generic QFT feedback loop with plant P, controller G, prefilter F, and sensor dynamics H [19].

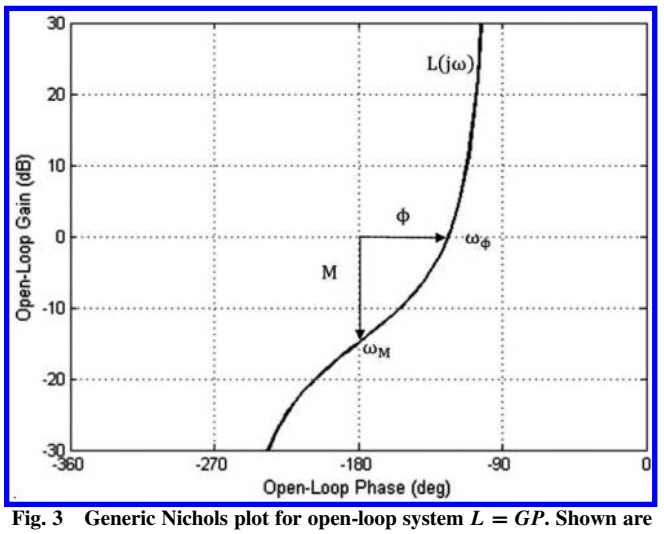

Fig. 3 Generic Nichols plot for open-loop system $L = GP$. Shown are crossover frequency ω_ϕ, phase margin ϕ, gain margin frequency ω_M, and gain margin M [19].

Once the plant and frequency sets are defined, the open- and closed-loop magnitude and phase responses of the plant sets are evaluated, and boundaries for the desired response are generated over the set of frequencies based on the performance specifications. The CAD process is realized by updating the plot of the system response as the designer adds, deletes, or changes the magnitude of the controller poles and zeros. As long as the discrete sets are judiciously defined, controllers can be designed to satisfy the performance requirements for the physical system at any frequency.

The Nichols plot of the open-loop system $L = GP$ is the primary design tool in synthesizing the controller G. The prefilter is designed by observing the Bode magnitude response of the closed-loop system as the prefilter parameters change. The Terasoft MATLAB QFT Frequency Domain Control Design Toolbox supports the CAD process by generating interactive plots and automating the computation of the bounds for performance requirements. Direct digital design is achieved by designing a controller and prefilter with the usual continuous-time poles and zeros, then converting the resulting controller into an equivalent discrete-time controller using a zero-order hold [31].

The primary motivation in selecting QFT for control synthesis is the robustness to specified plant uncertainties. Small-scale UAVs tend to experience relatively large nonlinear dynamic effects, due to both their susceptibility to aerodynamic disturbances as well as modeling inaccuracy [30]. This challenge increases the difficulty of both model identification and control design. By using QFT, controller performance can be ensured for a wide range of linear model uncertainties. QFT also allows for direct design of discrete-time linear control laws. These controllers are relatively computationally simple and should be straightforward to integrate with an existing autopilot system without interfering with the sensor loops.

III. Automatic Landing Control Structure and Coordinates

For the automatic landing system, sequential loop closures of SISO feedback loops are used to stabilize the aircraft dynamics and guide the vehicle to a reference flight path. Motivated by the work of [28], in which a QFT automatic landing controller is designed in simulation, the sensor dynamics H are assumed to be unity for all control loops. Control loops are to consist of prefilter F with feedback G. The control system, which is further detailed later in this section, consists of three primary loop sequences: lateral/directional control loop, in which the aileron is modulated to direct the aircraft's heading to match the runway direction (three sequential loops); longitudinal control loop, in which the elevator is modulated to control the vehicle's descent rate (two sequential loops); and airspeed command and hold (single feedback loop).

The feedback loops used for control are typified by Fig. 2. Two coordinate frames are used in modeling the aircraft dynamics: an inertial frame, designated n^+, and a body-fixed reference frame designated b^+. The inertial reference frame is defined such that the 1 axis is parallel to the target runway, and the 3 axis points down. For convenience, the origin is placed at zero altitude at the X-Y coordinates used to define the aircraft glide slope. The body reference frame is a standard body-fixed reference frame traditionally used in aircraft dynamics; the 1 axis points from the center of mass to the vehicle nose, and the 2 axis is parallel to the vehicle right wing [32].

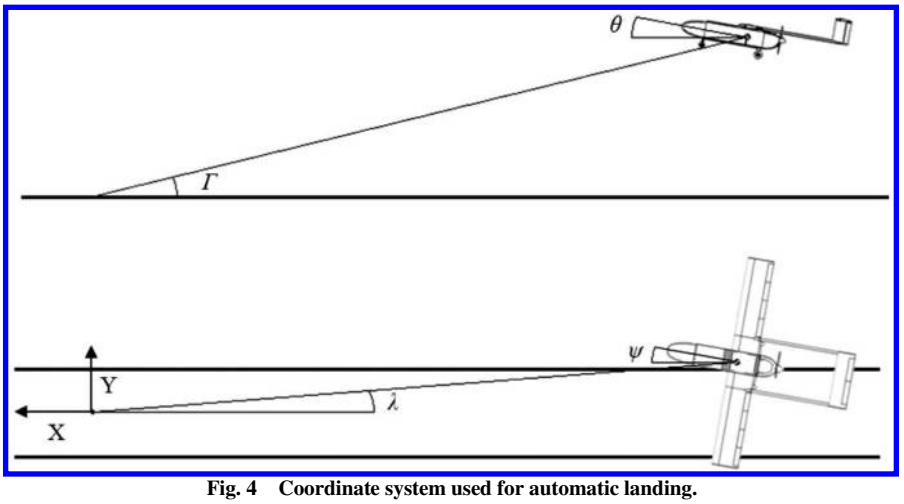

Fig. 4 Coordinate system used for automatic landing.

The aircraft attitude is parameterized by a sequential 3/2/1 rotation from the inertial frame to the body frame through angles $\psi/\theta/\phi$ (heading, pitch, and roll angles, respectively). The remaining aircraft states are parameterized using traditional aircraft dynamics nomenclature: inertial velocity of the aircraft parameterized in the body frame: $[\boldsymbol{V}_T]_b = [U \quad V \quad W]^T$; angular velocity of the body fixed frame relative to inertial frame parameterized in the body frame: $[\boldsymbol{\omega}]_b = [P \quad Q \quad R]^T$; and vector from the origin of the inertial frame to the aircraft center of mass, parameterized in the inertial frame: $[\boldsymbol{r}]_b = [X \quad Y \quad Z]^T$. The glide-slope angle Γ and relative azimuth to the runway λ are defined as in Eqs. (1) and (2) and are shown graphically in Fig. 4:

$$\tan \Gamma = -\frac{Z}{\|X\|} \tag{1}$$

$$\tan \lambda = \frac{Y}{\|X\|} \tag{2}$$

The automatic landing solution can be broken into two stages [1].

1) Approach to the runway: reference azimuth and glide-slope angles are tracked until the aircraft altitude decreases to a prespecified flare height. Constant airspeed is maintained.

2) Flare maneuver: once the flare height is reached, the aircraft performs a flare by tracking a reference descent rate until touchdown. Target airspeed is reduced in preparation for landing.

Three variables are controlled directly by control modulation: aircraft bank angle ϕ using aileron control; aircraft pitch angle θ using elevator control; and aircraft airspeed using throttle control. Guidance to the runway and landing are achieved by inputting reference angles to the bank angle and pitch angle command and hold loops, while airspeed is maintained at a constant value.

Guidance of the vehicle in the X-Y plane is achieved by designing a control loop to track a reference azimuth of 0. This loop is maintained throughout the entire automatic landing. Guidance in the X-Z plane is effected by a glide-slope tracking SISO loop that tracks a constant reference Γ. For the flare maneuver, a separate descent rate tracking loop is designed. The flare altitude is $h = 4$ m, and the reference descent rate is $\dot{h}_{\text{ref}} = -0.4$ h. Airspeed is maintained in both phases of the flight by a SISO loop.

IV. Vehicle Description and Modeling

Control design is performed for the Easy Star UAV and implemented on the ArduPilot Mega (APM) autopilot, an open-source commercial product capable of enabling waypoint navigation on standard remote control (RC) hardware. Once the control laws are designed and validated on the Easy Star, the same experimental software with updated control gains can be used to achieve automatic landing on larger, more expensive platforms. This section begins with a brief description of the Easy Star vehicle. Next, some further variables are defined to supplement those presented in the previous section. From there, the modeling process used to identify dynamic models is described, followed by a description of the sensors used by the autopilot. Finally, the simulation environment and associated assumptions are presented.

A. Easy Star System Description

The Easy Star is an inexpensive commercial RC product. With its high wing position and large aspect ratio, it has relatively benign flying qualities and is designed for manual flying by inexperienced RC pilots.[‡] The vehicle's payload bay is large enough to carry the required RC avionics and battery as well as an APM autopilot. This characteristic has made this vehicle a popular choice among hobbyists for autonomous flying with the APM. The vehicle has dual ailerons, rudder, and elevator control, making it an appropriate controls analog for a standard-configuration fixed-wing aircraft. The small size and all-electric power system enable multiple test flights of short duration with software changes as required in between.

B. Equations of Motion and Variable Conventions

Aircraft reference frames and states are defined in the previous section. The aircraft dynamic states U, V, W, P, Q, and R are heavily influenced by the aerodynamic angles between the aircraft and the relative wind vector, and the governing equations of motion cannot be derived for a general case. The equations of motion for the kinematic states X, Y, Z, ψ, θ, and ϕ are exactly known for an aircraft modeled as a rigid body and are written as follows:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\phi\sin\theta - \cos\phi\sin\psi & \sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\phi\sin\psi\sin\theta & -\sin\phi\cos\psi + \cos\phi\sin\psi\sin\theta \\ -\sin\theta & \cos\theta\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{3}$$

[‡]RTF EasyStar II, http://www.multiplex-rc.de/en/products/categories/products/details/productgroup/rc-accessories/productcategory/rtf-modelle/product/rtf-easystar-ii-mode-2-und-4.html?tx_lwshopitems_pi15Baction5D=show&tx_lwshopitems_pi15Bcontroller5D=Product&cHash=e0dde4b0f1417cf1f5760685031f312b [retrieved 19 July 2013].

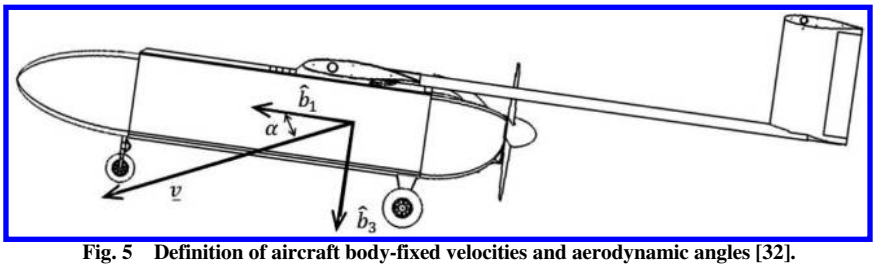

Fig. 5 Definition of aircraft body-fixed velocities and aerodynamic angles [32].

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \tag{4}$$

Aircraft linear dynamic models are commonly defined about the steady-state $P = 0, Q = 0, R = 0, \phi = 0$ at constant values of U, V, W, θ, ψ. With zero steady-state bank angle, the aircraft dynamics can be decoupled into longitudinal states U, W, Q, θ and lateral/directional states V, P, R, ϕ. Under the assumption of constant steady states, the dynamics can be written in terms of perturbations about the steady state, and the perturbation variables u, v, w, p, q, r are defined for the body-axis translational and angular velocities.

The primary aircraft controls are elevator, throttle, aileron, and rudder. The decoupling outlined previously has a similar effect on the controls, such that only elevator and throttle deflections δ_e and δ_t are assumed to influence the longitudinal states, and only aileron and rudder deflections δ_a and δ_r are assumed to influence the lateral/directional states. Under this model, the longitudinal and lateral/directional dynamics can be decomposed into two fourth-order linear systems having two controls each.

The aircraft body y and z axis velocities are parameterized in terms of the aerodynamic angle of attack and sideslip angles, α and β. In the absence of external wind, these angles are defined as in Fig. 5. In symbolic form,

$$\alpha = \arctan\frac{W}{U} \tag{5}$$

$$\beta = \arctan\frac{V}{\sqrt{U^2 + W^2}} \tag{6}$$

C. Easy Star Modeling

The QFT process requires selection of a discrete set of linear plants. This is achieved by assuming significant model uncertainties about a nominal linear model. Identification of a model from flight data is theoretically possible but, for reasons already discussed, is considered impractical. The X-Plane flight simulation program is used to generate flight histories of longitudinal and lateral/directional axis maneuvers. X-Plane uses blade-element theory to estimate aerodynamic characteristics based on the geometry of the aircraft [33], and basic aerodynamic properties can be obtained from this technique. The thrust characteristics are determined experimentally from the vehicle hardware, as detailed in Sec. IV.C.1. A three-dimensional model of the Bixler, which has nearly identical dimensions and configuration to the Easy Star, is flown in simulation to generate the data.[§] The simulation data are then used to fit and evaluate linear aircraft models. A discrete-time least-squares fit is used in model identification of longitudinal and lateral/directional models.

In addition to the identified model, the control surfaces are assumed to have first-order dynamics with a 0.1 s time constant, such that the actual deflection δ is related to the commanded deflection δ_c by the transfer function of Eq. (7). The Hitec HS-5055MG servos used on the Easy Star will be powered at 4.8 V; the 0.1 s time constant provides a 90% rise time of just under 0.2 s, which matches the manufacturer's specification for the time to reach 60 deg at 4.8 V[¶]:

$$\frac{\delta}{\delta_c} = \frac{10}{s + 10} \tag{7}$$

1. Thrust Modeling

To increase the vehicle's available thrust, the stock electric motor is replaced with a Model Motors AXI 2217/12.[**] The manufacturer does not provide a thrust rating with the 7 × 6 in. Easy Star propeller, and so static testing is conducted to determine the maximum thrust. The measured thrust at full throttle is between 1.5 lbf (6.7 N) and 2.0 lbf (8.9 N) in each of several trials. This is consistent with the manufacturer data, which provide thrust values between 9 and 12 N using the same battery and slightly larger propellers. To account for any potential loss of thrust as the batteries are depleted, the maximum thrust at landing is conservatively assumed to be only 1.0 lbf (4.4 N). This value replaces the X-Plane estimate for the throttle influence on the $\dot{u}$ channel. The throttle's influence on the other states cannot readily be determined experimentally without a substantially more complex experiment but should also be less significant. The values derived from X-Plane are assumed to be sufficiently accurate.

The dynamic response of the motor to throttle inputs is also characterized for modeling and simulation purposes. A separate experiment is conducted to determine the dynamic response. An MPX7002DP air data sensor with pitot-static probe is placed in the thrust wash at five different radii outward from the thrust centerline. The thrust dynamics are assumed to be first-order, such that the actual throttle output δ_T is related to the commanded throttle δ_{T_C} by

$$\dot{\delta}_T = \gamma(\delta_{T_C} - \delta_T) \tag{8}$$

Using the preceding assumption and a first-order forward difference approximation for $\dot{\delta}_T$, a least-squares solution for γ is computed using the data from all five tests. It is assumed that thrust is proportional to the velocity magnitude squared V^2, and so the value of γ can be computed by a best fit for the measured value $\tilde{V}^2$ as a function of the recorded throttle inputs. The least-squares solution for γ is 1.90. Figure 6 plots the measured time histories against the computed response using Eq. (8) with the same inputs and initial conditions. The motor dynamics are also included in the dynamic simulation and QFT transfer functions.

2. Parametric Uncertainty

Aircraft stability properties suffer from large uncertainties in modeling when flight-test data are not available. Typical errors may be as large as 20% (see Table 1). Additive uncertainty is assumed in modeling uncertain longitudinal and lateral-directional plants for the QFT control synthesis. Because typical model accuracy differs between the longitudinal and lateral/directional axes, a different approach is used to populate the set of uncertain plants for each axis.

1) For the longitudinal axis, the uncertainties in the nondimensional coefficients listed in Table 1 are assumed to encompass the most significant model uncertainty. Continuous-time models of the form of $\dot{\boldsymbol{x}} = [A]\boldsymbol{x} + [B]\boldsymbol{u}$ are assumed. Nondimensional stability derivatives are computed from the identified coefficients of the $[A]$ and $[B]$ matrices

[§]HiLStar17 Public Release, http://www.diydrones.com/profiles/blogs/hilstar17-public-release [retrieved 11 June 2013].

[¶]Hobby, H., HS-5055MG Digital Metal Gear Feather Light Servo, http://www.horizonhobby.com/products/hs-5055mg-digital-metal-gear-feather-light-servo-HRC35055S#t2 [retrieved 31 March 2014].

[**]Motors, M., AXI 2217/12 GOLD LINE, https://www.modelmotors.cz/product/detail/195/ [retrieved 15 June 2016].

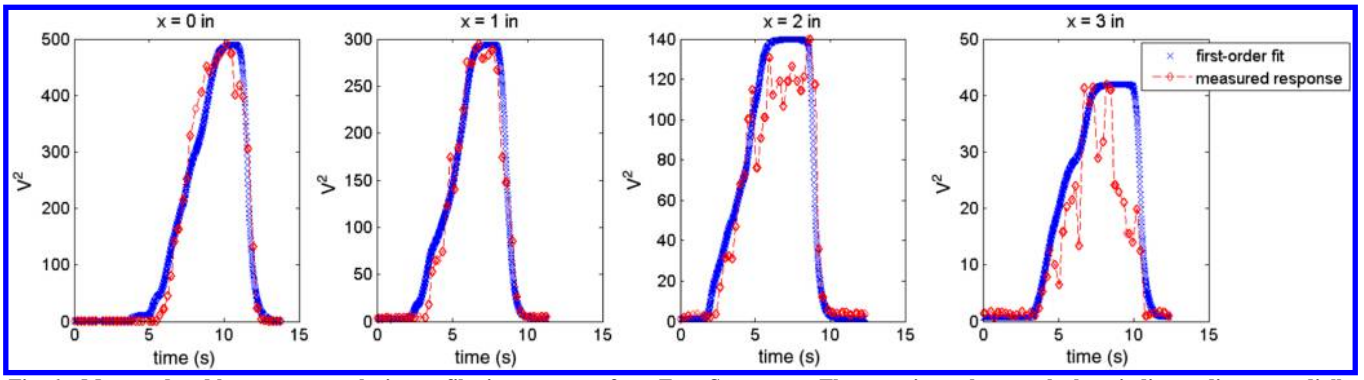

Fig. 6 Measured and least-squares velocity profile time response from Easy Star motor. The quantity x above each chart indicates distance radially outward from the thrust centerline.

Table 1 Typical longitudinal-axis modeling errors [28]

Derivative	Accuracy, %
C_{L_α}	±5
C_{m_α}	±10
C_{D_α}	±10
C_{L_u}	±20
C_{m_u}	±20
C_{D_u}	±20
C_{L_q}	±20
C_{m_q}	±20
C_{D_q}	±20

[32]. The extrema of the appropriate dimensional stability derivatives are then computed and used in QFT control synthesis.

2) For the lateral-directional axis, stability derivative errors may be as high as 90% [28], which is infeasibly large for use with QFT. Instead, each of the dimensional $[A]$ and $[B]$ matrix entries for nonkinematic equations is assumed to have additive errors of up to ±20% of the nominal value. Rudder was not used in the autolanding control laws, and so rudder derivative errors were not considered. The choice to neglect rudder control follows the example of [28] and simplifies the control synthesis by allowing all control loops to be considered as decoupled SISO loops. However, this choice does limit the performance of the lateral-directional control overall and represents a good starting point for future work building on the current paper.

For both the longitudinal and lateral/directional axes, the set of uncertain plants is populated by computing the extrema of each uncertain matrix element, then computing a set of models using factorial combinations of the extrema. The nominal model is also retained for the control design process. There are a total of 32,768 lateral-directional models and 2048 longitudinal models.

Table 2 shows the minimum and maximum values of the modal characteristics associated with the set of uncertain plants. Results are given in terms of natural frequency ω_n and damping ratio ζ for second-order poles, and in terms of the associated eigenvalues for first-order poles. The longitudinal axis is assumed to have two second-order modes, whereas the lateral/directional axis is assumed to have one second-order mode and two first-order modes.

D. Sensor Modeling

To implement the automatic landing feedback control laws, the following aircraft states must be measured or estimated: body 1-axis airspeed U; inertial position X, Y, Z; and body attitude ψ, θ, ϕ. The APM[††] includes a full sensor suite capable of providing the required measurements for normal flight. An ultrasonic rangefinder is added to the standard sensor package for accurate height-above-ground measurements in landing. The sensors used and errors assumed are shown in Table 3. The IMU, GPS, and airspeed sensors are all stock APM hardware, and sensor noise values are obtained directly from manufacturer data. The remainder of this section details the assumptions made in modeling the attitude estimator and ultrasonic rangefinder.

1. *Attitude and Heading Reference System*

The APM's attitude and heading reference system (AHRS) estimates the aircraft attitude from gyroscope measurements. Attitude is parameterized with a 3/2/1 Euler angle rotation sequence through $\psi/\theta/\phi$. The digital motion processor used for attitude determination on the MPU-6000 IMU is not documented.[‡‡] Rather than replicating the complete APM calculations in simulation, the AHRS error performance is analyzed experimentally. In the experiment, a rotating platform is driven by a Hitec HS-311 servo. The APM is then attached to the platform with either the roll, pitch, or yaw axis normal to the platform. The servo rotates the APM through 90 deg at an average rate of 66.67 deg · s^{-1} holds for 1 s, then returns to 0 deg at the same rate. This rotation is performed three times for each axis to improve accuracy.

The AHRS output is modeled as a low-pass-filtered output of the commanded servo angle with a constant bias. A first-order discrete-time model is fit to each axis in terms of the commanded angle at time t_i, $(\theta_c)_i$, the previous angle measurement θ_{i-1}, a constant offset b_θ, and a standard deviation σ_θ:

$$\theta_i = a\theta_{i-1} + b(\theta_c)_i + b_\theta + \mathcal{N}(0, \sigma_\theta^2) \tag{9}$$

The measured standard deviation between the AHRS outputs and the predicted outputs based on the model is taken as σ_θ. This metric admits some more error than may be actually present in the AHRS because the servo actuator dynamics are not accounted for. Because the actuation rate is much lower than the rated limit of the servo,[§§] this error is assumed to be relatively small, and the additional error is retained as a conservative estimate. The 3σ error limits for the AHRS are shown in Table 3. The pitch-axis fit is shown in Fig. 7.

2. *Ultrasonic Rangefinder*

Preliminary landing simulations using the standard sensor package make it clear that the worst-case barometer error is too large for reliable autoflare maneuvers. Because of the minimal available onboard processing power, typical filtering schemes are considered impractical for implementation on the Easy Star. Lightweight ultrasonic rangefinders can provide accurate range data but typically are limited in maximum range to distances right above the planned autoflare altitude. An ultrasonic rangefinder cannot be used to

[††]Official Arduplane Repository, http://plane.ardupilot.com/ardupilot/index.html [retrieved 16 June 2016].

[‡‡]MPU-6000/6050 Six-Axis (Gyro + Accelerometer) MEMS Motion-Tracking Devices, https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/ [retrieved 16 June 2016].

[§§]City, S., HS-311 Standard, http://www.servocity.com/html/hs-311_standard.html [retrieved 14 July 2013].

Table 2 Assumed modal extrema for control design

Mode	Longitudinal axis		Lateral/directional axis	
1	$\omega_n = [3.531, 4.756$ rad/s]	$\zeta = [0.2307, 0.4627]$	$p_1 = [-0.7441, -0.03832]$	$p_2 = [-16.11, -9.862]$
2	$\omega_n = [0.4272, 0.5403$ rad/s]	$\zeta = [0.5084, 0.8559]$	$\omega_n = [4.665, 7.809$ rad/s]	$\zeta = [0.1078, 0.3912]$

Table 3 Sensors and assumed errors for simulation

Sensor	Measurement	Assumed 3σ error bound	Units
MediaTek MT3329 GPS	Global north-east position	2.5	Meters
MPX7002DP differential pressure sensor	Body one-axis airspeed	100 (= 6.5)	Pascals (meters per second)
Measurement Specialties MS5611-01BA03 barometric pressure sensor	Altitude ($h > 6$ m)	250 (= 20.8)	Pascals (meters)
AHRS	Heading, pitch, roll angles	11.1, 11.1, 10.3	Degrees
XL-MaxSonar-EZ4 ultrasonic rangefinder	Altitude ($h < 6$ m)	2.5	Centimeters

determine altitude throughout the flight envelope but can be incorporated for use exclusively during the flare maneuver.

The XL-MaxSonar-EZ4 provides range data for wings level flight with a nominal 1 cm accuracy at ranges between 20 cm and 7.6 m.[¶] Furthermore, it has a footprint of approximately 1 by 1 in. and weighs only 5.9 g. At distances less than the minimum or greater than the maximum sensitive range, it is assumed the sensor returns the minimum or maximum range, respectively. As a worst-case estimate, a maximum effective 3σ error bound of 2.5 cm is assumed to be present in the range readings for simulation. This value is considered to be very conservative.

E. Dynamic Simulation

Controller synthesis and initial validation are performed in MATLAB. The inner QFT loops (heading, roll, pitch, and airspeed) are evaluated by simulating the step response of the set of uncertain linear plants selected for control design. Sensor uncertainty is not considered. For the outer loops, the potential nonlinear effects of large Euler angles are considered important. A full 12-degree-of-freedom (DOF) simulation is conducted using the identified linear models to propagate the vehicle's velocity-level variables, whereas the nonlinear kinematic equations of motion of Eqs. (3) and (4) are used to propagate the translational and rotational states. To address model uncertainty, simulations are conducted using each of the uncertain lateral/directional or longitudinal models. For preliminary controller evaluation, no process or measurement noise is incorporated. Later, batch Monte Carlo simulations are performed including these effects to analyze overall controller performance.

V. Control Design and Validation

QFT is used to develop robust feedback controllers with guaranteed tracking performance for the set of uncertain plants identified in Sec. IV. The automatic landing controller is designed as a set of sequential closed single-input/single-output loops. Each digital controller has an update rate of 10 Hz. This update rate is compatible with real-time operation of the APM, whose inner loop runs at 50 Hz with outer loops of 10, 5, and 1 Hz (see footnote ††). The basic structure of each loop consists of a prefilter $F(z)$, controller $G(z)$, and plant or inner-loop model $P(s)$ or $P(z)$. The QFT performance specifications in each loop are

$$\left\| \frac{P(z)G(z)}{1 + P(z)G(z)} \right\| \leq \text{SM} \tag{10}$$

$$T_{rl}(s) \leq \left\| \frac{F(z)P(z)G(z)}{1 + P(z)G(z)} \right\| \leq T_{ul}(s) \tag{11}$$

The first specification ensures robust stability margins to compensate for the uncertain plant model. The second ensures robust tracking performance in the frequency domain between the lower bound T_{rl} and the upper bound T_{ul}. Both performance criteria place constraints on the controller; the prefilter is restricted only by the tracking bound.

This section summarizes the automatic landing controllers designed for the Easy Star and presents a Monte Carlo analysis of stochastic simulations including sensor noise, dynamic model uncertainty, and static winds.

A. Easy Star Pitch Axis Inner Loop

A comprehensive description of the design of every compensator is beyond the scope of this work. This subsection discusses the design of the pitch angle command loop, for which the control input is the elevator deflection δ_e and the controlled variable is θ. The design process is representative of that used elsewhere (e.g., the bank angle command loop). For the pitch axis control loop, a stability margin of SM = 1.2 is required. The upper and lower bound transfer functions for robust tracking are as follows:

$$T_{rl}(s) = \frac{1.25}{s^3 + 6s^2 + 5.25s + 1.25} \tag{12}$$

$$T_{ul}(s) = \frac{0.1389s + 0.6944}{s^2 + 0.7599s + 0.6944} \tag{13}$$

These limits are selected to correspond approximately to a time-domain rise time of between 2.0 and 6.75 s, with a maximum of 20% overshoot. Note also that the stability margin used in the pitch angle command loop is smaller (i.e., more restrictive) than in other QFT loops because the need for robust stability is felt to be more significant in designing the pitch angle compensator. This constrains the design space. To make the QFT compensator design more feasible, the maximum rise time in $T_{ul}(s)$ was increased from its initial value of 5.0 s.

Figures 8 and 9 show plots produced by the Terasoft QFT control design toolbox for the pitch angle command loop design. Figure 8 shows Nichols plot contours defining the inadmissable regions for the system response. The left figure shows the raw requirements at the chosen frequency array from both the robust stability and tracking bounds. The frequency array used is as follows:

$$\omega \in [0.1, 0.25, 0.5, 1.0, 4.2, 6.5] \text{ Hz} \tag{14}$$

The upper and lower limits for the array are selected by noting that the templates at frequencies of 0.01 and 30.0 Hz are not qualitatively different than at 0.1 and 6.5 Hz, respectively. The robust stability bounds form closed contours that the response must not fall within at a particular frequency. The robust tracking bounds form open curves that the response at a given frequency must be above in the Nichols plot. Note that robust performance curves at frequencies above 1.0 Hz are not computed to simplify the display; however, as can be seen in

[¶]XL-MaxSonar-Ez Series, http://maxbotix.com/documents/XL-MaxSonar-EZ_Datasheet.pdf [retrieved 17 July 2013].

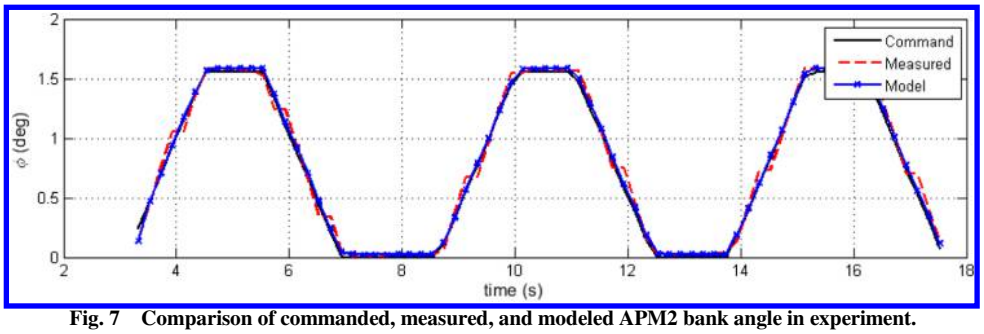

Fig. 7 Comparison of commanded, measured, and modeled APM2 bank angle in experiment.

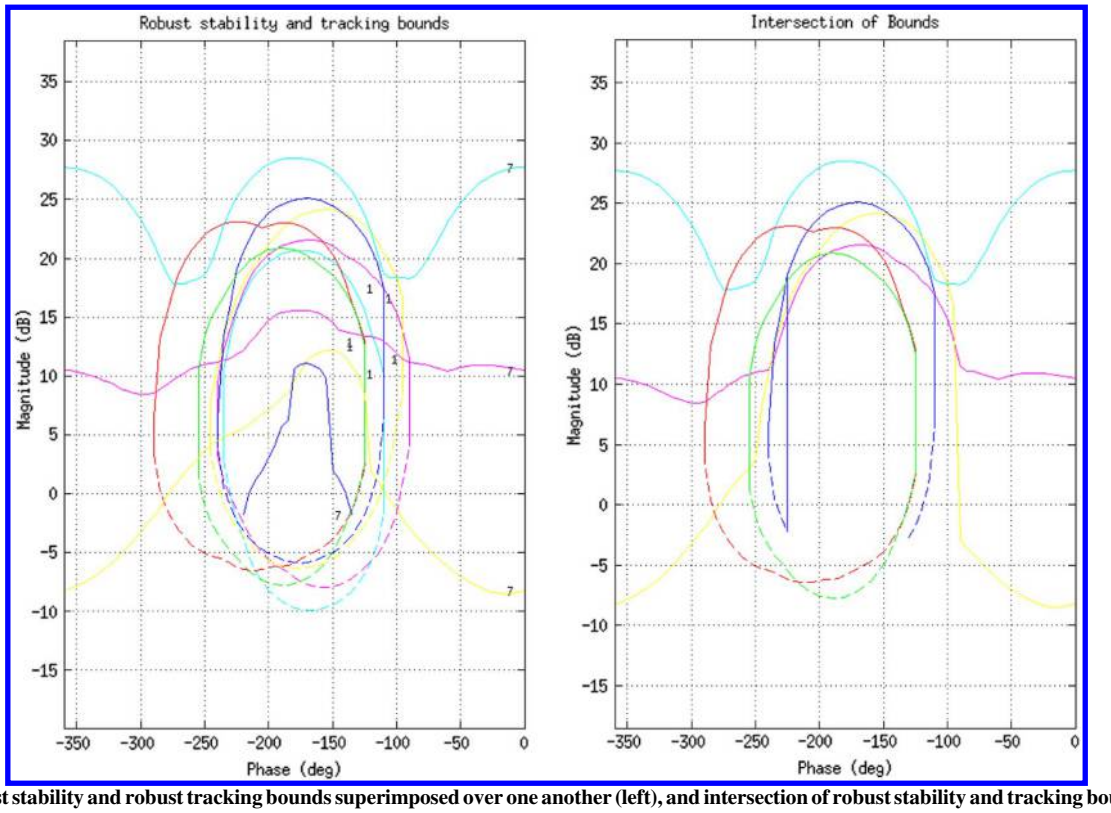

Fig. 8 Robust stability and robust tracking bounds superimposed over one another (left), and intersection of robust stability and tracking bounds (right).

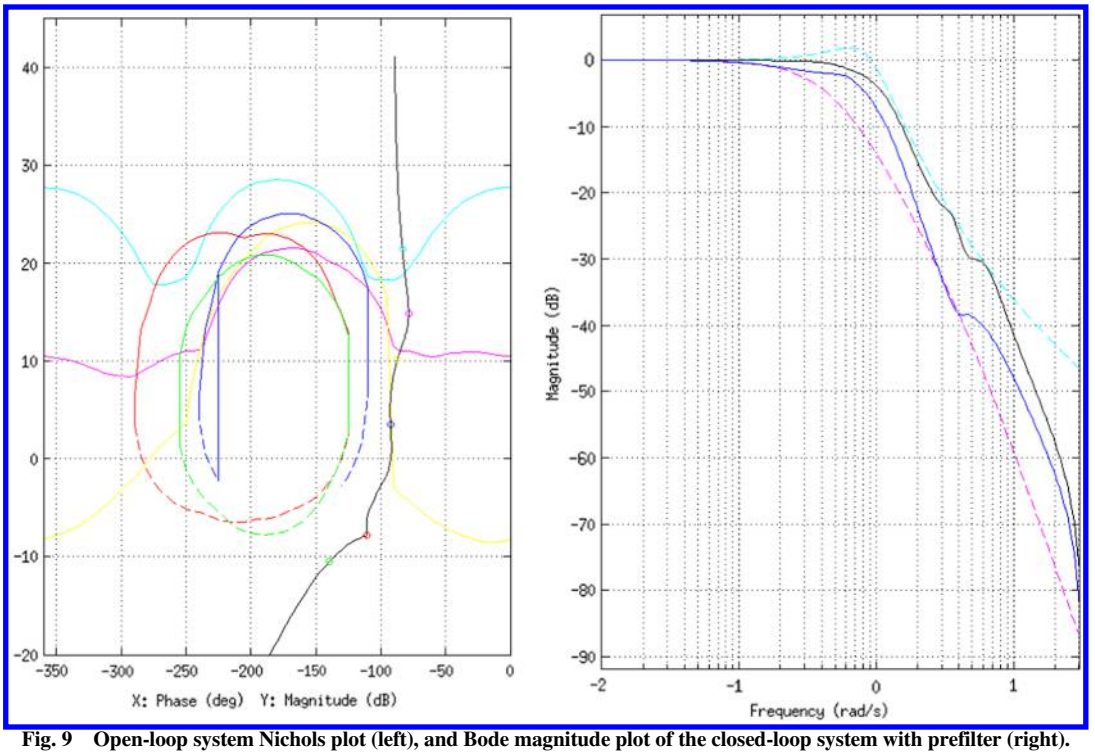

Fig. 9 Open-loop system Nichols plot (left), and Bode magnitude plot of the closed-loop system with prefilter (right).

Table 4 Table of automatic landing controller and prefilter values

Control loop	$G(z)$	$F(z)$
Bank angle command	$\frac{z^4-2.5817z^3+2.1787z^2-0.5969z}{-1.6136z^4+2.6016z^3+0.03421z^2-1.4361z+0.4139}$	$\frac{z^2}{46.29z^2-68.812z+23.517}$
Heading angle command	$\frac{z-0.9668}{0.03072z-0.01192}$	$\frac{z^2-1.6567z+0.6661}{15.268z^2-29.926z+14.67}$
Azimuth regulation	$\frac{2.5001z^2-4z+1.5}{z^2-z}$	$\varnothing$
Pitch angle command	$\frac{z^3-2.5588z^2+2.2991z-0.7231}{-2.5636z^3+6.0048z^2-4.6755z+1.2343}$	$\frac{z^3-2.2095z^2+1.6386z-0.3984}{23.0885z^3-63.6612z^2+58.7204z-18.1170}$
Glide-slope tracker	$\frac{5.51z^2-6.5z+1}{z(z-1)}$	1
Automatic flare	$\frac{2.525z^2-5.025z+2.500}{z^2-z}$	$\frac{.00726}{z^2-1.876z+0.8831}$
Airspeed command	$\frac{z^2-1.7143z+0.7367}{1.3797z^2-1.6695z+0.2901}$	$\frac{z^2-1.8362z+0.8468}{9.9495z^2-19.3958z+9.4587}$

Fig. 9, the tracking bounds are still satisfied for the closed-loop system. The right-hand plot of Fig. 8 shows the intersection of the robust stability and tracking design contours.

Figure 9 shows the compensated system responses that satisfy the design requirements. The left-hand plot shows the Nichols plot of the nominal response of the open loop system as a solid curve extending from the top right to the bottom left. The circles on the response contour correspond to the response at particular frequencies. In the plot, the uppermost circle corresponds to the response at 0.1 Hz, the second highest circle corresponds to 0.25 Hz, and so on. Note that the compensator is excessively conservative; that is, the plant responses at particular frequencies are not as close as possible to the associated contour. This reflects a design effort to try to inject damping into the longitudinal axis system by increasing the phase margin in the pitch-axis compensator. The right-hand plot of Fig. 9 shows the minimum and maximum closed-loop Bode magnitude response of the set of plants with the compensator and prefilter. The dashed lines indicate the tracking bounds defined by $T_{rl}(s)$ and $T_{ul}(s)$. This part of the design is highly constrained, with the extreme responses falling more or less on top of the tracking bounds.

B. Easy Star Controllers

Table 4 summarizes the controller and prefilter values for each of the linear control loops. Some controllers are redesigned after evaluation in flight test; the final values used in flight are shown in this table. To minimize computational and memory overhead, each discrete-time transfer function is fourth-order or less. In general, the control complexity is concentrated in the innermost loops wherever possible. Fulfilling the QFT performance requirements in the outer loops is substantially more difficult than for the inner loops, and frequency-domain tracking in the outer loops does not in general translate to acceptable time-domain performance. Outer loops are synthesized to satisfy QFT requirements where this can be feasibly done with a low-order controller. However, in the azimuth, glide slope, and flare loops, discrete-time PID controllers are implemented. These are evaluated in Monte Carlo simulations with uncertain plants, and the inner-loop robustness in general is sufficient to provide adequate robustness in the outer loops.

C. Monte Carlo Simulations with Uncertainty

To evaluate the overall controller performance, simulations including sensor noise as well as environmental effects are conducted. For 12-DOF simulation, the sets of uncertain longitudinal and lateral/directional models are too large to exhaustively evaluate all combinations of models. A random pairing of one uncertain longitudinal and one uncertain lateral/directional model is used in each simulation to propagate the velocity-level states. In evaluating the performance for this work, a vertical speed at landing of 6 ft/s is considered a "soft" or nominal landing, a speed of 10 ft/s in considered a "hard" landing (minor damage to vehicle, repairable with no significant effect on performance), and a greater speed at landing is considered a crash (effective loss of vehicle or damage requiring major repair) [28]. The target runway for simulation is 10 m wide; this figure is conservative compared to the actual airfield used in the experiment, which has runways closer to 30 m wide. Results are presented in the presence of static winds of varying strength and direction.

The controllers given in Table 4 are evaluated in the simulation using the sensor noise characteristics assumed in Sec. IV.D. To improve performance of the flare maneuver, the target body 1-axis speed during the flare is set at -2 m/s relative to the steady-state value. The primary performance metrics of interest are descent rate at touchdown, horizontal distance traveled during flare, pitch angle at touchdown, and Y position at touchdown.

Figure 10 shows the simulation trajectory and attitude history for a typical Monte Carlo simulation. The vertical speed at touchdown is within the soft landing threshold of 1.8 m/s. The lateral performance shows a slight overshoot in the Y position, which is common for the large initial Y value. The attitude histories appear slightly underdamped but produce reasonably damped outer-loop performance.

To evaluate control law performance, four sets of 1000 Monte Carlo simulations are run at different levels of static winds with uncertain plants and sensor noise. The initial vehicle states for all simulations are $\psi(0) = -20$ deg, $X(0) = -500$ m, $Y(0) = 100$ m, and $\Gamma(0) = 5$ deg. All other states are initially their steady-state values. When crosswinds are present, the vehicle is initialized on the side of the runway toward which the crosswind is blowing. Otherwise, at the initial X coordinate specified, the control law does not compensate for the crosswind fast enough when approaching from the opposite direction. It should be noted that crosswind landings are avoided whenever possible but are considered here to address how performance may be affected if conditions change within the course of a flight. Four sets of simulations are conducted. Case 1 is no wind; case 2 is a 1.35 m/s crosswind; case 3 is a 2.25 m/s wind along the direction of landing (a tailwind), to determine if glide-slope singularities are encountered; and case 4 is a 2.81 m/s wind at 36.5 deg to the runway. This represents 125% of the maximum steady wind and crosswinds allowable for an Easy Star test flight and is considered the worst-case scenario. The orientation is such that there is a tailwind again.

Monte Carlo results are summarized in Table 5. With no wind, landing performance is limited primarily by state uncertainty at the beginning of the flare maneuver. Despite significant sensing errors, 85% of landings are within the target tolerance for vertical speed. The remainder of landings are split evenly between hard landings and crashes. Less than 5% of landings are outside the target 10-m-wide strip. The mean flare length is well below the 150 m target.

With the maximum rated crosswind, approximately 15% of landings are outside the target runway. This represents a performance drop from the no-wind simulation. The longitudinal-axis performance in terms of the number of soft and hard landings is essentially unchanged from the zero-wind case. This indicates that the crosswind does not in general destabilize the controller and merely shifts the mean lateral landing location. Because the physical runways available are much wider than the 10 m simulation target, this performance is considered acceptable.

The third case evaluates the performance with the nominal maximum steady wind blowing along the direction of landing. Performance degrades significantly from the zero-wind case with fewer than 70% soft landings. The mean flare length is slightly longer than in the other cases. The Y position at landing is off the runway in a large number of simulations, despite the absence of a crosswind. Because the tailwind causes the vehicle to approach the landing site faster than before, the lateral/directional controller is unable to

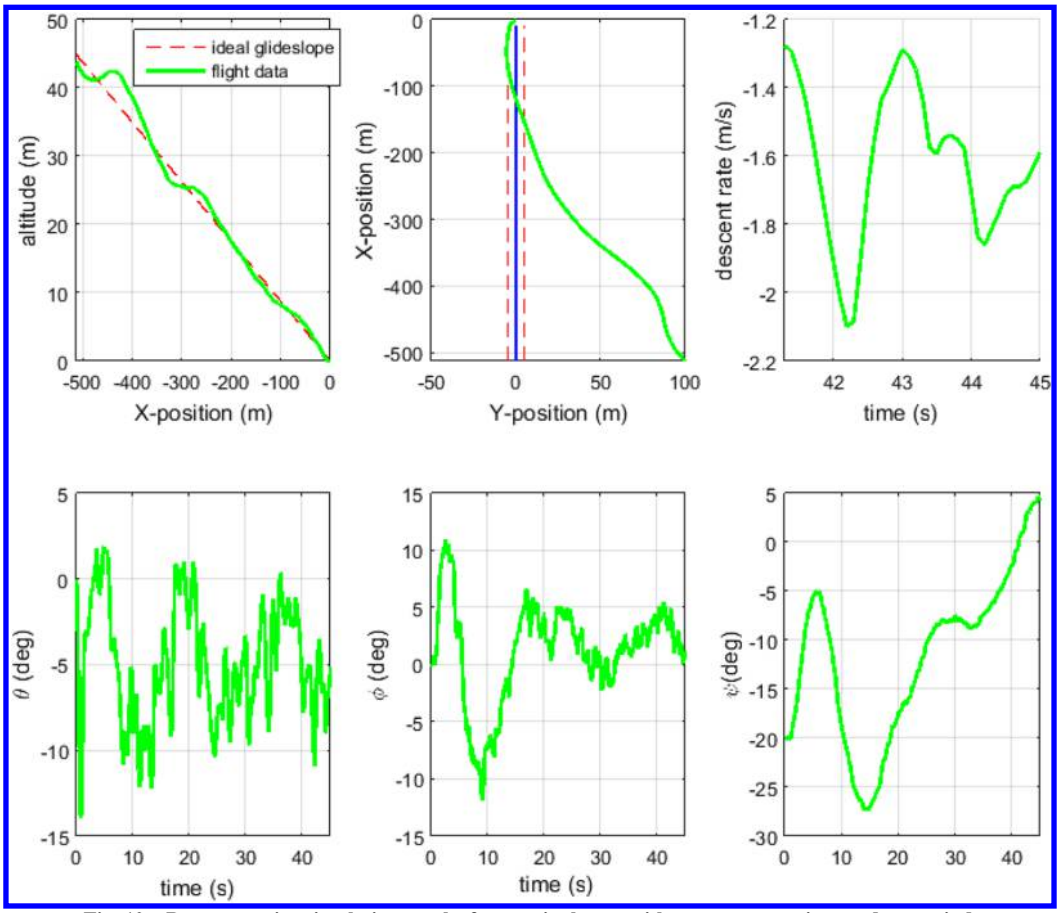

Fig. 10 Representative simulation results from a single run with sensor uncertainty and zero wind.

compensate for the initial condition as well. The short range to the origin induces singularities in the glide-slope and azimuth calculations. Normally, the flare altitude is reached well before the localizer, and so the singularities are not a problem. Because the direction of a steady wind can usually be determined readily in advance of flight, the runway direction should be selected to be into the wind field to maximize the chances of a safe landing. This simulation considers the effect of substantial changes in prevailing wind patterns during flight and indicates that there is approximately a 75% chance of landing without severe damage to the vehicle.

Table 5 Summary of Monte Carlo results for the Easy Star flight vehicle

Case	Soft landings	Hard landings	Landings off runway	Mean flare length, m
No wind	854	73	36	38.3
1.35 m/s crosswind	839	87	144	35.9
2.25 m/s tailwind	617	160	430	41.6
Worst case	708	150	239	31.1

The worst-case scenario assumed has a wind magnitude of 125% the rated maximum of 5 mph, with a crosswind component of 125% of the maximum rated crosswind of 3 mph. The number of soft landings is actually much higher in the worst-case analysis than in the pure tailwind case, and the number of landings on the runway is also larger. This most likely occurs because the combination of a larger tailwind and crosswind tend to keep the vehicle farther from the localizer when $X = 0$, reducing the effect of the trigonometric singularity on the landing performance. The control law retains an approximately 80% chance of landing without loss of the vehicle when landing with a tailwind and significant relative crosswind. Overall, performance is tolerable, given the relatively large disturbances considered and large dynamic uncertainty assumed.

VI. Flight-Test Results

This section presents the flight-test results using the automatic landing controller on the Easy Star platform. Modifications in the heading command and hold loop and automatic flare loop based on flight-test performance are highlighted, before the two successful automatic landings are summarized. The full automatic landing is

Table 6 Test matrix for Easy Star automatic control flight tests

Date	Wind, mph	Objectives	Number of flights
24 Jan. 2014	5–10	Manual flying, operator familiarization	1
25 Jan. 2014	5	Bank and heading hold	2
29 Jan. 2014	0–5	Bank, pitch, and heading hold	5
1 Feb. 2014	0–5	APM2 waypoint navigation	1
3 Feb. 2014	5–10	APM2 waypoint navigation	1
8 Feb. 2014	5–10	APM2 waypoint navigation, full state regulation	2
9 Feb. 2014	5–10	Azimuth and glide slope	2
11 March 2014	10	Azimuth and glide slope	1
12 March 2014	15	Azimuth and glide slope	1
13 March 2014	5–10	Flare, automatic landing	10

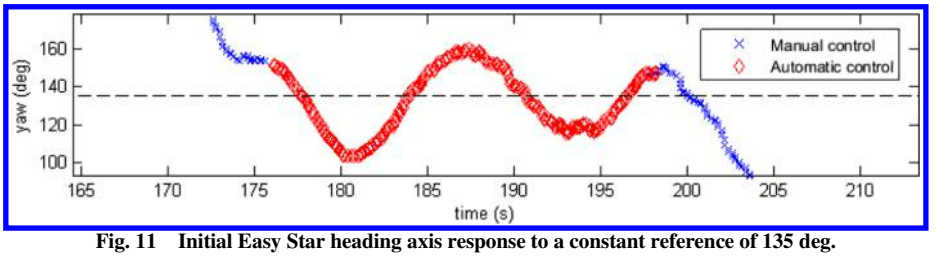

Fig. 11 Initial Easy Star heading axis response to a constant reference of 135 deg.

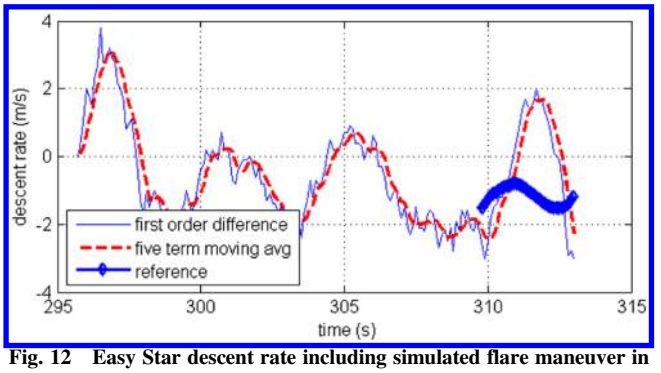

Fig. 12 Easy Star descent rate including simulated flare maneuver in flight testing.

realized across a sequence of tests. Starting with only the inner loop, control loops are evaluated by sequential tests. Additional loops are closed as testing progresses. The sequence in which each control loop is added and evaluated follows: 1) bank angle command and hold to $\phi = 0$; 2) heading command and hold to $\psi = 0$; 3) pitch angle command and hold to $\theta = 0$; 4) airspeed command and hold to $u = 0$; 5) azimuth and glide-slope tracking loop down to the flare altitude; 6) full landing sequence with automatic flare.

A total of 26 flights were conducted between January and March of 2014 for the purpose of evaluating the automatic landing controller on the Easy Star. A summary of the flight dates and test conditions and objectives is given in Table 6. This table includes all tests for which the vehicle was instrumented with the autopilot and does not include prior flights for operator familiarization; the target center of gravity for all flights is 1/4 of the wing root chord. This list includes several flights evaluating the effectiveness of waypoint navigation with the autopilot's built-in waypoint navigation, which was considered for use in initiating the automatic landings. Ultimately, the RC pilot preferred to manually set up and engage the automatic landing, rather than using waypoint navigation for this task.

A. Lateral/Directional Control Redesign

Initial flight tests using the heading command and hold loop to track a constant heading indicate that the initial heading control loop was significantly underdamped, as in Fig. 11. In this test, the vehicle was flown under manual control, then the autopilot was engaged to regulate heading. When the bank angle controller was evaluated with a reference bank of 0 deg, the response was relatively well-damped, so the heading loop was modified. The controller G_ψ was added to the heading control loop to improve performance. The azimuth tracking loop was updated to account for the redesigned inner loop.

B. Longitudinal Control Redesign

Initial flight tests of the Easy Star automatic flare loop indicated that the controller as designed was significantly underdamped. Representative results are shown in Fig. 12. In this flight, the landing waypoint was placed 15 m above the actual runway to allow the human pilot time to recover in the event of a problem. To some extent, the underdamped performance was a result of a design choice to reduce the chance of a hard landing in favor of more lightly damped, longer flares. The actual flight performance exhibited less damping than was present in the simulation, and the autoflare controller and prefilter were redesigned to add damping and improve the reference tracking in the flare mode.

C. Landing Performance

On 13 March 2014, two automatic landings were completed. The vehicle was flown under manual control to a position between 300 and 400 m uprange of the GPS waypoint used to define the glide slope, then switched into the automatic landing mode.

Figure 13 shows images taken from an onboard camera during the approach and landing. The top left image was taken at $t = -3$ s,

Fig. 13 Still images taken from onboard video during automatic landing 1.

under manual control. The target waypoint is indicated by a circle, added in postprocessing. The top center image is taken at $t = 2$ s, showing the initial dive to reach the target glide slope. The top right image was taken at $t = 5$ s, showing the vehicle levelling out after the initial dive. The bottom left image at $t = 19$ s shows the continued approach to the runway. At the bottom center the start of the flare maneuver at $t = 31$ s is shown. Ground station and pilot operators are visible at the left. The bottom right image was taken at $t = 44$ s, just after the vehicle reached a full stop.

The inertial position histories and flare performance are shown in Figs. 14 and 15. The approach performance is evaluated in terms of the inertial frame position histories. The reference glide slope is plotted in the X-Z plane, and the approximate lateral position of the waypoint with 5 m error lines are plotted for the X-Y plane history. In both cases, the vehicle begins with significant initial errors in altitude. This error is corrected by the time approximately 100 m have been traveled along the ground path. The azimuth tracking loops also experience initial errors of a few degrees, which are corrected within roughly 150 m. The azimuth tracking loop still appears to be slightly underdamped, despite the modifications to the heading control loop. This effect is exacerbated by the lightweight aircraft flying in somewhat unsteady winds.

Flare performance is examined in terms of the position histories below the flare altitude and the descent rate during that flight segment. The flare altitude was 4 m. In Figs. 14 and 15, descent rates are computed by a first-order finite difference of the altitude histories; to reduce noise, the presented results include a five-term moving average, which was explicitly used in flight. Flare performance is nearly ideal for the first test. The vehicle travels less than 50 m downrange of the target waypoint and remains within about 1 m/s of the reference descent rate. The final descent rate is nearly zero and is well within the margin for a soft landing. During the second automatic landing, the vehicle overcorrected to reduce its descent rate to match the reference, and the touchdown descent rate is on the threshold of a hard landing (defined earlier as a descent rate between 1.83 and 3.05 m/s). This manifested as damage to the horizontal tail and some minor scratching to the pitot-static probe, which was installed in the nose. The aircraft could not be fixed and flown again in situ but was repaired with approximately two person hours of labor.

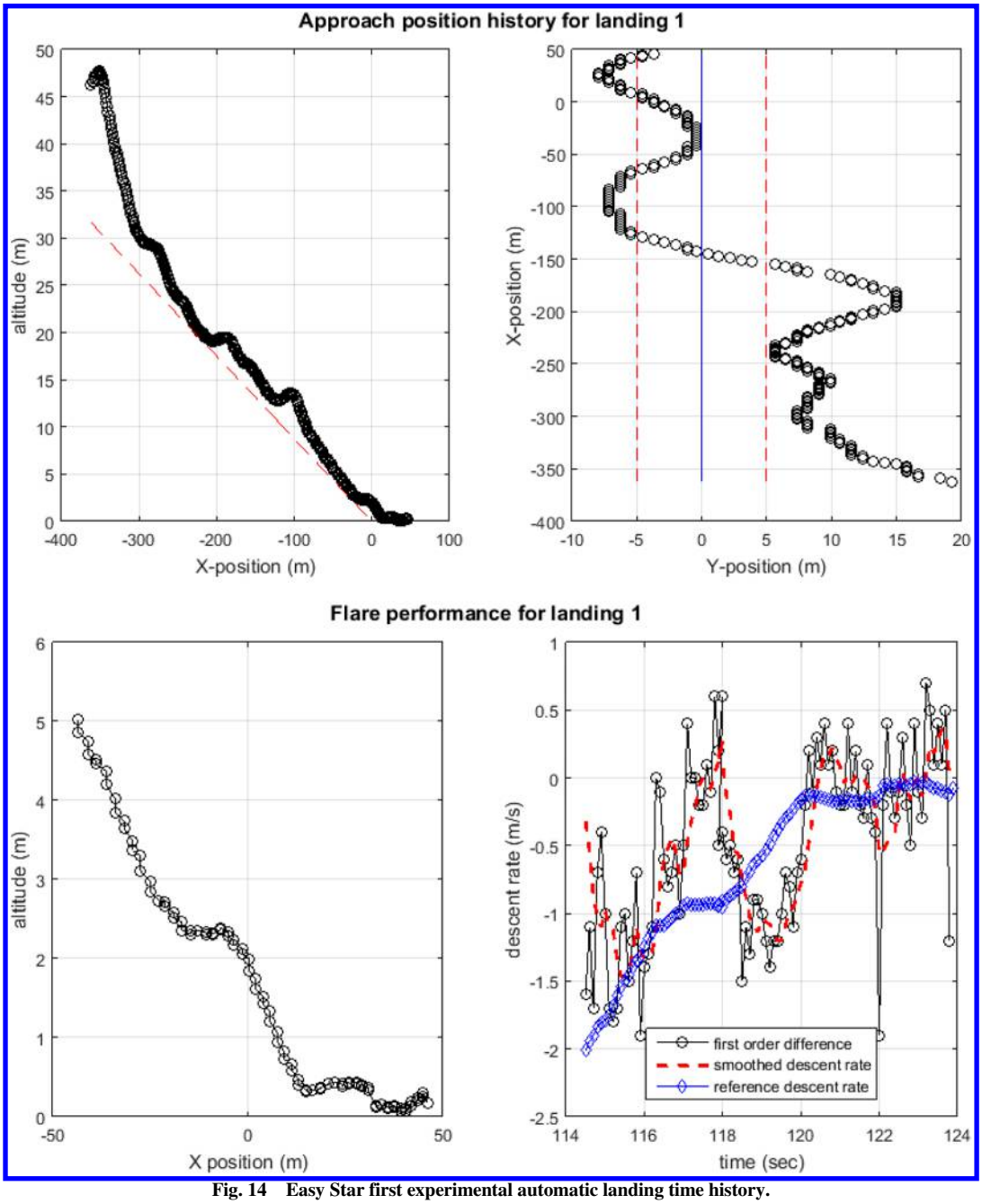

Fig. 14 Easy Star first experimental automatic landing time history.

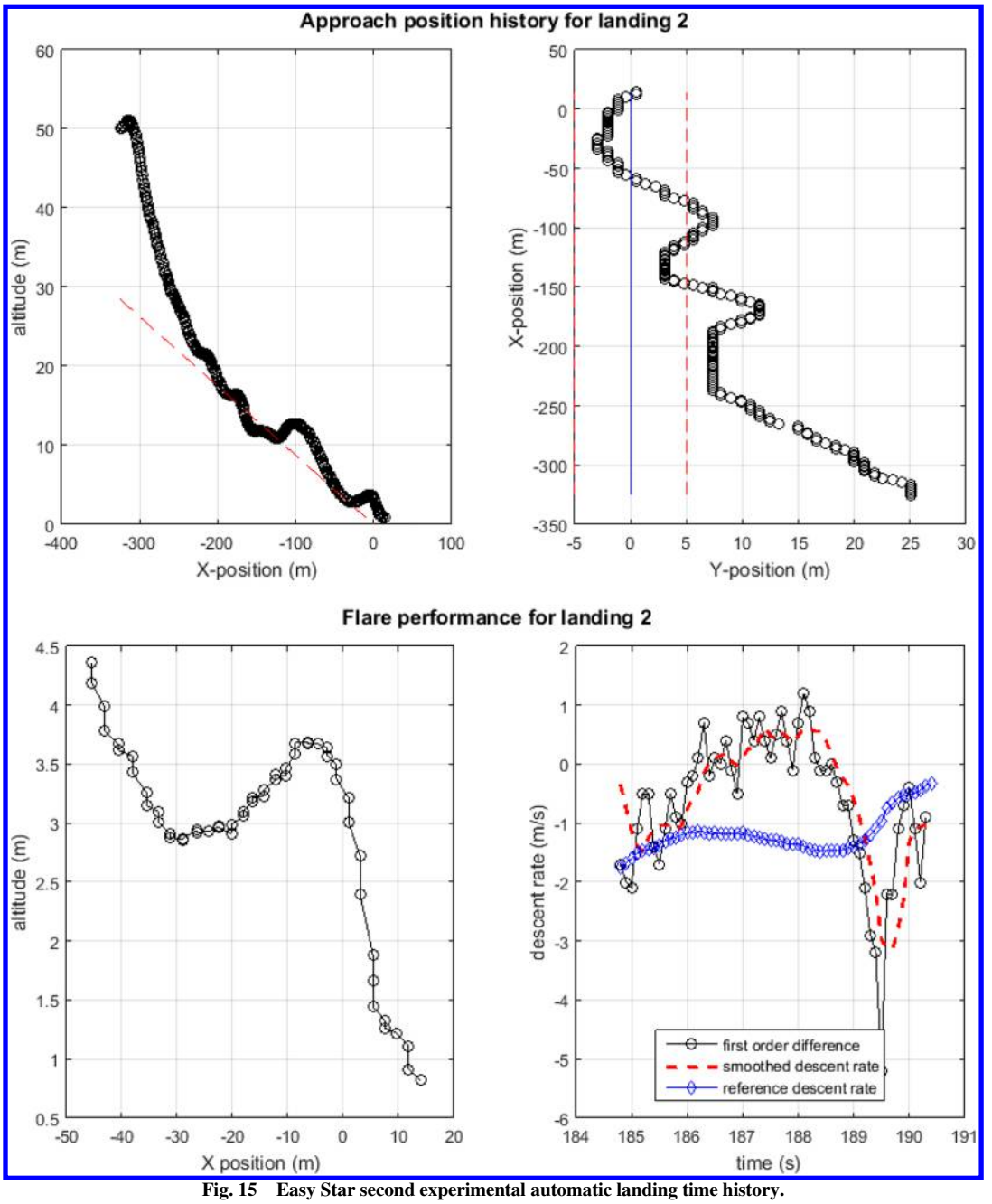

Fig. 15 Easy Star second experimental automatic landing time history.

In examining the flare performance, the response still appears to be underdamped. It is not immediately apparent whether this is due more to the design of the inner or outer loop, and unsteady winds most likely are a factor as well. There is room to improve performance with further gain tuning. However, the two flights shown here demonstrate that the basic implementation of the control law on the hardware works as intended and is capable of safely guiding the vehicle to a landing.

D. Comparison Against Simulation Model

Figure 16 shows the approach and flare performance for the second flight compared against a representative simulation using the Easy Star model from controls development. Figure 17 shows the velocity and attitude history for the same case. The simulation contains sensor uncertainty but zero wind, and the nominal model is used for the vehicle dynamics. Additionally, the simulation used the final controllers implemented in flight test. The initial position, velocity, and attitude for the simulation are set to the measured values of the states at the first time step after the automatic landing mode is triggered. (Note that the position histories are differenced with a first-order Euler approximation and smoothed with a simple low-pass filter to determine the velocity $\hat{v}_k = \alpha\hat{v}_{k-1} + (1-\alpha)\tilde{v}_k, \alpha = 0.8$, where $\tilde{v}_k$ is the differenced value.)

In Fig. 16, the longitudinal axis in terms of the altitude and X-position histories match reasonably well between the data and simulation. The simulated trajectory overshoots less than the real trajectory, and the simulation is noticeably more damped. The lateral-directional model is clearly a poor match. Recall that the damping was increased substantially in the lateral-directional control between the simulation and the final landing. The simulated response is extremely sluggish, and the final landing is actually off the runway. This contradicts the consistent performance in flight test of the heading command and hold loop. It is clear from Fig. 17 that the largest discrepancy is between the dynamic response for the body-axis velocity state v_2, which is simply the perturbed state associated with velocity component V. The simulated state exhibits very small oscillation about a zero mean state, but the data suggest that the model should be much more oscillatory. The other states in Fig. 17 are difficult to compare, although there appear to be some qualitative

Table 7 Monte Carlo comparison of performance at the initial conditions for landing 2

Simulation value	μ_{100}	S_{100}	μ_{200}	S_{200}	Value from flight
Time, s	30.5	3.13	29.6	3.57	34.5
Downrange touchdown, m	−7.08	29.4	−15.3	33.9	14.0
Crossrange touchdown, m	22.4	12.7	22.4	13.2	0.519
Vertical speed, m/s	2.96	1.47	3.28	1.89	1.28
Hard landings	75	— —	160	— —	— —

similarities in some of the longitudinal axis states. The pitch angle in the simulation appears to match the flight data reasonably well in some mean sense but is much better damped in the simulation. This is consistent with the other comparisons between flight and simulation.

For the flare, the simulation exhibits small-amplitude oscillations not present in the flight data and achieves a soft landing. The flare performance does tend to vary noticeably between simulations; sometimes it exhibits small oscillations, as in this example, but in other simulations, it exhibited larger oscillations that were more like those in the flight data.

Table 7 shows Monte Carlo results for a small batch of simulations. Simulations begin at the same initial conditions as the single simulation used to generate Figs. 16 and 17. The values μ_x and S_x indicate the mean and standard deviation of Monte Carlo outputs with x simulations and are used to give some idea of the convergence. Most noticeably, the flare performance in the simulation appears to be worse on average than in the flight test. The simulations typically land several meters short of the target waypoint, although there is a lot of variation in that value based on the standard deviation. The vertical speed at touchdown is consistently around 3 m/s, although the flight value of 1.28 m/s is within one standard deviation of that value. Overall, most simulations terminate in hard landings (or crashes). Also, the simulation landings are consistently off the runway laterally. This last fact is consistent with the observed difference between the simulation and flight-test performance of the lateral-

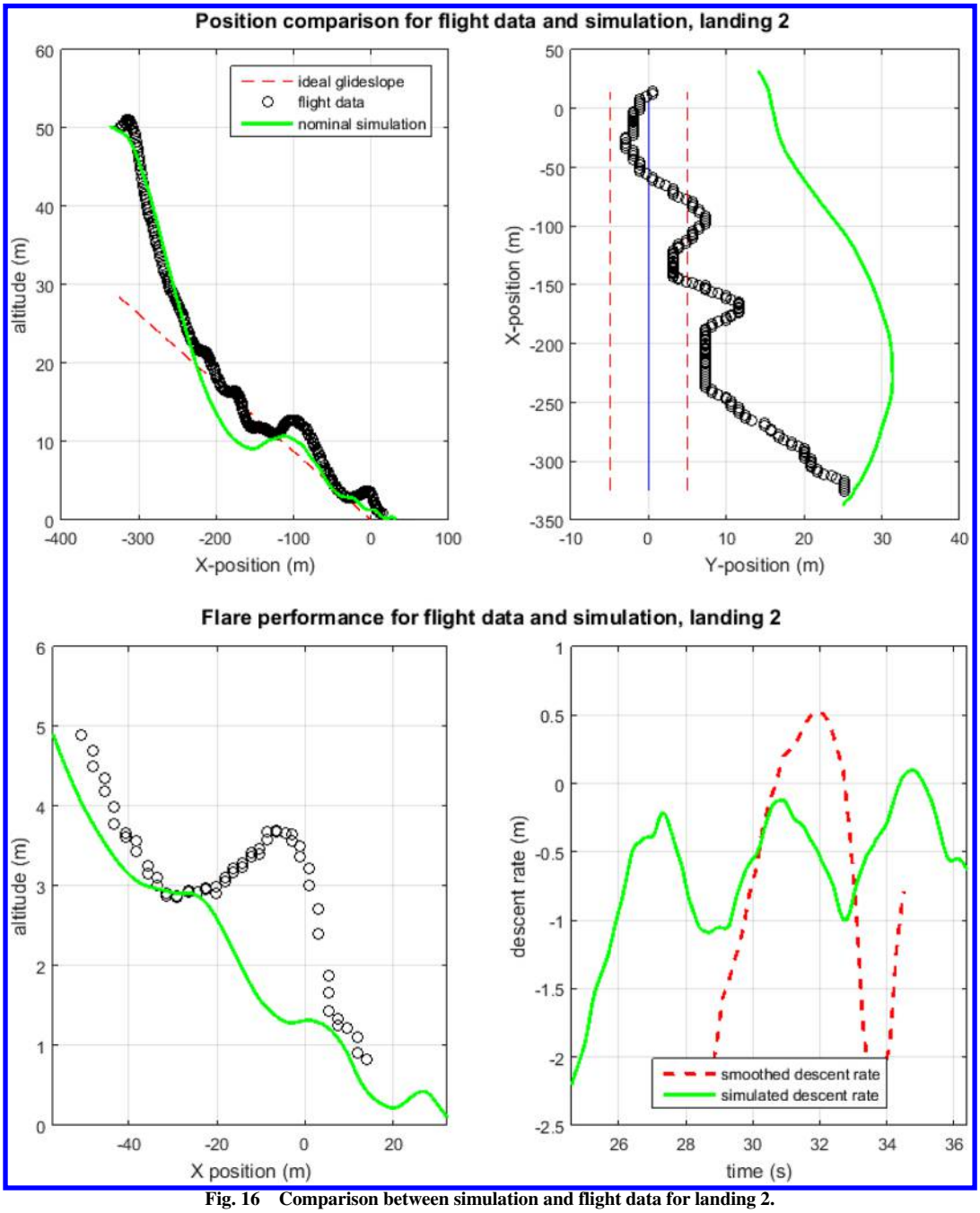

Fig. 16 Comparison between simulation and flight data for landing 2.

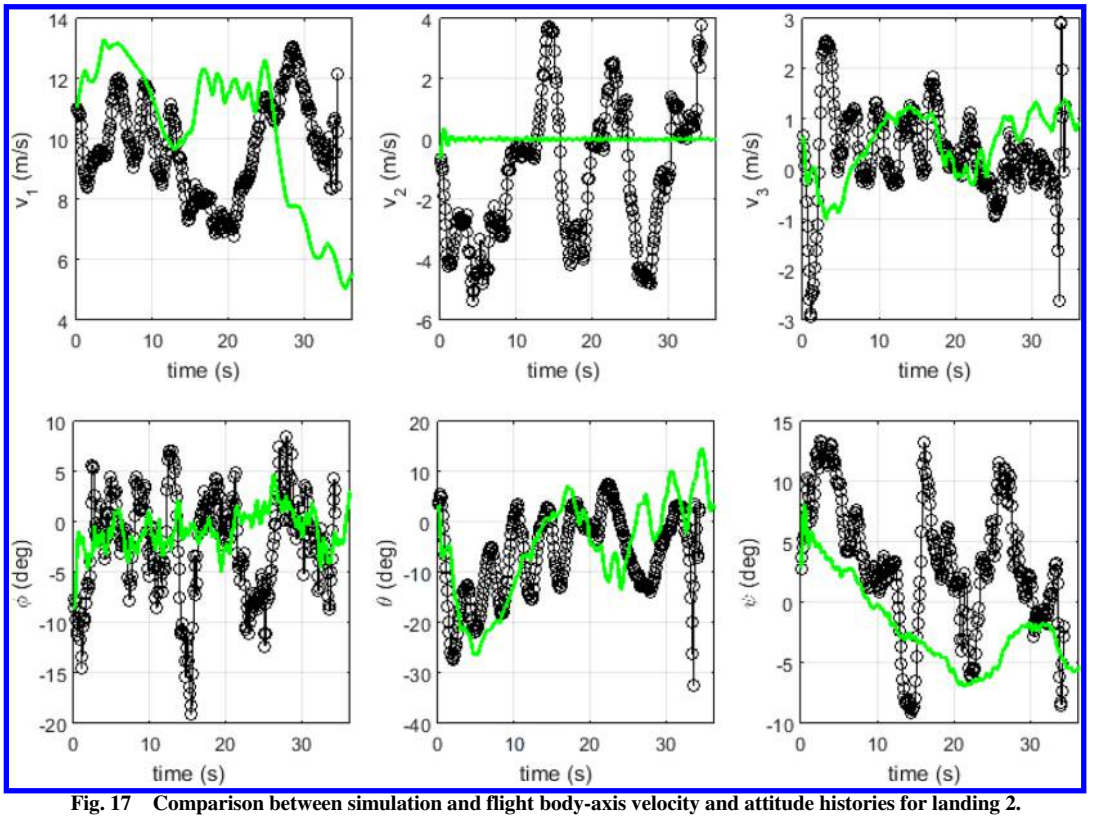

Fig. 17 Comparison between simulation and flight body-axis velocity and attitude histories for landing 2.

directional control and may be one source of the large number of hard landings in the simulation. If the simulated vehicle is far from the runway and tends to be at a relatively aggressive bank angle, as part of correcting the position, then the nonlinear attitude dynamics may affect the longitudinal-axis performance during touchdown adversely.

VII. Conclusions

A QFT-based automatic landing controller has been demonstrated to work in simulation and in flight test. Monte Carlo landing-phase simulations indicate a soft landing rate of approximately 85% in nominal conditions. In the worst-case simulation scenario, this rate drops to about 60%, which is tolerated due to the vehicle's low mass and significant model uncertainty in simulation. In flight test, most of the linear controllers synthesized in simulation are found to give adequate performance; a small number require redesign. The complete landing controller performed well but exhibited lightly damped pitch-axis responses, which can be addressed by modifications to the controllers.

As a control synthesis technique, QFT was found to work well for the inner-loop design and not for the outer-loop design. This is consistent with existing QFT literature, which indicates that QFT in general is excessively conservative for high-order plants. Satisfying robust performance requirements while providing adequate time-domain performance was straightforward on the uncertain inner-loop plants. For the outer control loops, it was not generally possible to achieve adequate time-domain performance with a low-order controller while satisfying the robust performance constraints. Where QFT controllers could not be designed successfully, PID controllers were implemented and gave adequate performance. The inner loops are relied on to ensure robustness and disturbance rejection. Performance in stochastic simulations and the flight test indicate the validity of this control synthesis approach.

Qualitatively, the X-Plane-derived dynamics compare favorably to the true dynamics, but the set of uncertain plants considered for control synthesis failed to fully predict the true system dynamics. Many of the designed control loops offered adequate performance in flight with no redesign. However, both the lateral/directional and longitudinal dynamics appear to be noticeably more lightly damped in experiment than in simulation. The set of dynamic models considered in controller synthesis was intended to be conservatively large. This is symptomatic of the modeling difficulties with lightweight UAVs in general. The conclusion is that the X-Plane dynamics are not sufficiently accurate at this scale for complete controller design and, in general, must be supplemented by flight-test experiments to achieve adequate controller performance.

Appendix: Easy Star Model Used in Control Design

The identified Easy Star longitudinal continuous-time model has the following dynamic modes and eigenvalues λ:

$$\lambda_1 = -1.4 \pm 3.9i \qquad \lambda_2 = -0.33 \pm .36i$$
$$\zeta_1 = .33 \qquad \zeta_2 = .67$$
$$\omega_1 = 4.2 \text{ rad/s} \qquad \omega_2 = .48 \text{ rad/s}$$

The lateral/directional dynamic characteristics are

$$\lambda_1 = -.28 \qquad \lambda_2 = -1.6 \pm 6.2 \qquad \lambda_3 = -13$$
$$\tau_1 = 3.6 \text{ s} \qquad \zeta_2 = .25 \qquad \tau_3 = 0.076 \text{ s}$$
$$\omega_2 = 6.4 \text{ rad/s}$$

The continuous-time linear longitudinal model is

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.548 & 0.0493 & -0.149 & -8.09 \\ -0.0117 & -0.268 & 0.464 & 0.0152 \\ 0.158 & -33.8 & -2.59 & -1.26 \\ 0 & 0 & 1.00 & 1 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} -1.41 & 4.80 \\ 0.201 & -0.0100 \\ -34.9 & -1.57 \\ 0.000 & 0.000 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} \tag{A1}$$

$$U_1 = 12.6 \text{ m/s} \tag{A2}$$

$$\alpha_1 = -0.0293 \text{ rad} \tag{A3}$$

$$\theta_1 = -0.0197 \text{ rad} \tag{A4}$$

The continuous-time lateral-directional model is

$$\begin{bmatrix} \dot{\beta} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -2.23 & 0.248 & 0.770 & -0.326 \\ 35.2 & -12.0 & 4.73 & -4.62 \\ -54.1 & 2.17 & -2.27 & 0.282 \\ 0 & 1.00 & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} 1.83 & 0.606 \\ -101 & -6.02 \\ 16.5 & 21.1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \tag{A5}$$

Acknowledgments

This work has been supported by the National Science Foundation Graduate Research Fellowship Program. This support is gratefully acknowledged by the authors.

References

[1] Blakelock, J. H., *Automatic Control of Aircraft and Missiles*, Vol. 6, Wiley, New York, 1991, pp. 81–98, 176–188.

[2] Barrows, G. L., Chahl, J. S., and Srinivasan, M. V., "Biomimetic Visual Sensing and Flight Control," *Proceedings of the Bristol UAV Conference*, Dept. of Aerospace Engineering, Univ. of Bristol, Bristol, 2002, pp. 159–168.

[3] Kingston, D. B., and Beard, R. W., "Real-Time Attitude and Position Estimation for Small UAVs Using Low-Cost Sensors," *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, AIAA Paper 2004-6488, Sept. 2004.

[4] Kim, J.-H., Sukkarieh, S., and Wishart, S., "Real-Time Navigation, Guidance, and Control of a UAV Using Low-Cost Sensors," *Field and Service Robotics*, Springer, Berlin, 2006, pp. 299–309.

[5] Barber, D. B., Griffiths, S. R., McLain, T. W., and Beard, R. W., "Autonomous Landing of Miniature Aerial Vehicles," *Journal of Aerospace Computing, Information, and Communication*, Vol. 4, No. 5, 2007, pp. 770–784. doi:10.2514/1.26502

[6] Barber, B., McLain, T., and Edwards, B., "Vision-Based Landing of Fixed-Wing Miniature Air Vehicles," *Journal of Aerospace Computing, Information, and Communication*, Vol. 6, No. 3, 2009, pp. 207–226. doi:10.2514/1.36201

[7] Roos, J.-C., and Peddle, I., "Autonomous Take-Off and Landing of a Low Cost Unmanned Aerial Vehicle," *R & D Journal of the South African Institution of Mechanical Engineering*, Vol. 25, South African Inst. of Mechanical Engineering, Stellenbosch, South Africa, 2009.

[8] Huh, S., and Shim, D. H., "A Vision-Based Automatic Landing Method for Fixed-Wing UAVs," *Journal of Intelligent and Robotic Systems*, Vol. 57, No. 1, Jan. 2010, pp. 217–231.

[9] Laiacker, M., Kondak, K., Schwarzbach, M., and Muskardin, T., "Vision Aided Automatic Landing System for Fixed Wing UAV," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE Publ., Piscataway, NJ, 2013, pp. 2971–2976.

[10] Thurrowgood, S., Moore, R. J., Soccol, D., Knight, M., and Srinivasan, M. V., "A Biologically Inspired, Vision-Based Guidance System for Automatic Landing of a Fixed-Wing Aircraft," *Journal of Field Robotics*, Vol. 31, No. 4, 2014, pp. 699–727. doi:10.1002/rob.2014.31.issue-4

[11] Senpheng, M., and Ruchanurucks, M., "Automatic Landing Assistant System Based on Stripe Lines on Runway Using Computer Vision," *Proceedings of the IEEE International Conference on Science and Technology*, IEEE Publ., Piscataway, NJ, 2015, pp. 35–39.

[12] Fan, Z., and Yong, W., "Automatic Landing of Unmanned Aerial Vehicle Using Fuzzy Control," *Proceedings of the IEEE International Conference on Information and Automation*, IEEE Publ., Piscataway, NJ, 2013, pp. 472–477.

[13] Nho, K., and Agarwal, R. K., "Automatic Landing System Design Using Fuzzy Logic," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 2, 2000, pp. 298–304. doi:10.2514/2.4522

[14] Lungu, R., Lungu, M., and Grigorie, L. T., "Automatic Control of Aircraft in Longitudinal Plane During Landing," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 49, No. 2, 2013, pp. 1338–1350. doi:10.1109/TAES.2013.6494418

[15] Joos, A., Müller, M., Baumgärtner, D., and Allgöwer, F., "Nonlinear Predictive Control Based on Time-Domain Simulation for Automatic Landing," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2011-6298, Aug. 2011.

[16] You, D. I., Jung, Y. D., Cho, S. W., Shin, H. M., Lee, S. H., and Shim, D. H., "A Guidance and Control Law Design for Precision Automatic Take-Off and Landing of Fixed-Wing UAVs," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2012-4674, Aug. 2012.

[17] Jianfeng, Z., and Caijuan, J., "Automatic Landing Controller Design and Simulation of Flying-Wing Unmanned Aerial Vehicle," *Proceedings of the IEEE 2nd International Conference on Measurement, Information, and Control*, IEEE Publ., Piscataway, NJ, 2013, pp. 893–896.

[18] Juang, J.-G., Chien, L.-H., and Lin, F., "Automatic Landing Control System Design Using Adaptive Neural Network and its Hardware Realization," *IEEE Systems Journal*, Vol. 5, No. 2, 2011, pp. 266–277. doi:10.1109/JSYST.2011.2134490

[19] Yaniv, O., *Quantitative Feedback Design of Linear and Nonlinear Control Systems*, Vol. 509, Springer, Boston, MA, 1999, pp. 8–9, 15–56.

[20] Sheldon, S. N., and Rasmussen, S. J., "Development and First Successful Flight Test of a QFT Flight Control System," *Proceedings of the IEEE National Aerospace and Electronics Conference, (NAECON'94)*, IEEE Publ., Piscataway, NJ, 1994, pp. 629–636.

[21] Keating, M., Pachter, M., and Houpis, C., "QFT Applied to Fault Tolerant Flight Control System Design," *Proceedings of the American Control Conference*, Vol. 1, IEEE Publ., Piscataway, NJ, 1995, pp. 184–188.

[22] Wu, S.-F., Grimble, M. J., and Wei, W., "QFT Based Robust/Fault Tolerant Flight Control Design for a Remote Pilotless Vehicle," *Proceedings of the IEEE International Conference on Control Applications*, Vol. 1, IEEE Publ., Piscataway, NJ, 1999, pp. 57–62.

[23] Houpis, C., and Rasmussen, S., "Unmanned Research Vehicle: Development, Implementation, and Flight Test of a MIMO Digital Flight Control System Designed Using Quantitative Feedback Theory," *Proceedings of the International Symposium on Quantitative Feedback Theory and Robust Frequency Domain Methods*, Defense Technical Information Center, 1999, pp. 1–13.

[24] Santander, A., and Aranda, J., "QFT for the Design of an Aircraft Flight Control," *IFAC Proceedings Volumes*, Vol. 38, No. 1, 2005, pp. 138–143. doi:10.3182/20050703-6-CZ-1902.01984

[25] Schuck, F., Heller, M., Baier, T., and Holzapfel, F., "Longitudinal Robust Controller for Excellent Handling Qualities Design of a General Aviation Aircraft Using QFT," *AIAA Guidance, Navigation, and Control (GNC) Conference*, AIAA Paper 2013-5180, Aug. 2013.

[26] Doyle, J. C., "Quantitative Feedback Theory (QFT) and Robust Control," *Proceedings of the American Control Conference*, IEEE Publ., Piscataway, NJ, 1986, pp. 1691–1698.

[27] Kerr, M. L., Lan, C.-Y., and Jayasuriya, S., "Non-Sequential MIMO QFT Control of the X-29 Aircraft Using a Generalized Formulation," *International Journal of Robust and Nonlinear Control*, Vol. 17, Nos. 2–3, 2007, pp. 107–134. doi:10.1002/rnc.1106

[28] Wagner, T., and Valasek, J., "Digital Autoland Control Laws Using Direct Digital Design and Quantitative Feedback Theory," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1399–1413. doi:10.2514/1.27761

[29] Woodbury, T. D., "Synthesis and Hardware Implementation of an Unmanned Aerial Vehicle Automatic Landing System Utilizing Quantitative Feedback Theory," M.S. Thesis, Dept. of Aerospace Engineering, Texas A&M Univ., College Station, TX, 2014.

[30] Chao, H., Cao, Y., and Chen, Y., "Autopilots for Small Fixed-Wing Unmanned Air Vehicles: A Survey," *Proceedings of the International*

Conference on Mechatronics and Automation (ICMA 2007), IEEE Publ., Piscataway, NJ, 2007, pp. 3144–3149.

[31] Borghesani, C., Chait, Y., and Yaniv, O., *The QFT Frequency Domain Control Design Toolbox for Use with MATLAB: User's Guide*, Terasoft, July 2003, pp. 3-1–3-30, 5-3, 5-10–5-15, http://www.terasoft.com/qft/QFTManual.pdf.

[32] Roskam, J., *Airplane Flight Dynamics and Automatic Flight Controls*, DARCorporation, Lawrence, KA, 2009, pp. 3–6, 65–67, 78.

[33] "X-Plane Operation Manual," Laminar Research, Columbia, SC, Nov. 2014, http://www.x-plane.com/files/manuals/X-Plane_Desktop_manual.pdf.