Old Dominion University

ODU Digital Commons

Spring 2018

Learn-To-Fly Control System Design

Alexander Brent Streit
Old Dominion University, brent.streit@gmail.com

Follow this and additional works at: https://digitalcommons.odu.edu/ece_etds

Part of the Aerospace Engineering Commons, and the Electrical and Computer Engineering Commons

LEARN-TO-FLY CONTROL SYSTEM DESIGN

by

Alexander Brent Streit
B.S. Electrical Engineering 2014, Old Dominion University

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
May 2018

Approved by:

Oscar R. González (Director)

W. Steven Gray (Member)

Dimitrie Popescu (Member)

ABSTRACT

LEARN-TO-FLY CONTROL SYSTEM DESIGN

Alexander Brent Streit
Old Dominion University, 2018
Director: Dr. Oscar R. González

In order to improve aircraft flight control system development life cycle, new flight control techniques are being explored to allow the system to "learn-to-fly" with limited a priori information of the aircraft's aerodynamic characteristics. One approach is to have a system identification process operating on-the-fly to generate mathematical models which can be used to update control laws. In this thesis, a wind tunnel experiment was conducted with a model aircraft set up to be free-to-roll, so system identification and control methods could be explored for a one-degree-of-freedom case. In particular this thesis covers the design of a novel control system that uses mathematical models of a predetermined form with certain unknown parameters to achieve satisfactory roll control for the free-to-roll experiment. The mathematical model was updated on-the-fly by a separate process that is outside of the scope of this thesis, which performs system identification in real-time.

I dedicate this work to my loving parents Doug
and Alix.

ACKNOWLEDGEMENTS

NOMENCLATURE

$\%OS$ Percent overshoot of reference model, (No Units)

α Angle of attack, degrees

α_e Equilibrium angle-of-attack, degrees

α_p Perturbation angle-of-attack, degrees

$\bar{q}$ Dynamic pressure, pound-force per square foot

β Sideslip angle, degrees

β_e Equilibrium sideslip angle, degrees

β_p Perturbation angle-of-attack, degrees

Δ_ϕ Roll angle perturbation, degrees

δ_{ail-l} Deflection of left aileron, degrees

δ_{ail-r} Deflection of right aileron, degrees

δ_{elev-l} Deflection of left elevator, degrees

δ_{elev-r} Deflection of right elevator, degrees

δ_{flap-l} Deflection of left flap, degrees

δ_{flap-r} Deflection of right flap, degrees

δ_{rudder} Deflection of rudder, degrees

$\hat{a}_1$ First coefficient of reference model transfer function, per second

$\hat{a}_2$ Second coefficient of reference model transfer function, per square second

$\hat{p}$ Nondimensional roll rate, (No Units)

ω_n Natural frequency of reference model, radians per second

ϕ Roll angle, degrees

ϕ_c Commanded roll angle, degrees

ϕ_p Perturbation roll angle, degrees

ψ Yaw angle of free-to-roll rig, degrees

$\tau_{cg,\phi}$ Derivative of τ_{cg} with respect to ϕ, foot pound-force per degree

τ_{cg} Torque about longitudinal body-axis due to center-of-gravity, foot pound-force

θ Pitch angle of free-to-roll rig, degrees

ξ Angle between center of gravity and force of gravity, degrees

ζ Damping ratio of reference model, (No Units)

a_1 First coefficient of perturbation linear system, per second

a_2 Second coefficient of perturbation linear system, per square second

b Wing span, feet

b_2 Input coefficient of perturbation linear system, degrees per square second

C_l Nondimensional aerodynamic rolling moment , (No Units)

$C_{l,0}$ Base value of nondimensional rolling moment, (No Units)

$C_{l,\phi}$ Derivative of C_l with respect to ϕ, per degree

$C_{l,p}$ Roll damping coefficient, seconds per degree

$C_{l,u}$ Vector of control effectiveness coefficients, per degree

F Right-hand-side of state equations, degrees per square second

F_p Derivative of F with respect to p, per second

F_v Derivative of F with respect to v, degrees per square second

F_ϕ Derivative of F with respect to ϕ, per square second

g Gravitational acceleration, feet per square second

I_{xx} Moment of inertia about longitudinal body-axis, slug square foot

K Virtual control input scaling factor, (No Units)

k_1-k_{30} Coefficients of rolling moment model polynomial, (No Units)

K_d Derivative gain, seconds per degree

K_p Proportional gain, per degree

l_{cg} Length of lever arm of center of gravity, feet

m Mass of aircraft, slugs

p Roll rate, degrees per second

p_e Equilibrium roll rate, degrees

p_p Perturbation roll rate, degrees

S Wing reference area, square feet

T_s Settling time of reference model, seconds

u Vector of control surface deflections, degrees

u_{lb} Vector of lower limits of control surface deflections, degrees

u_{max} Control surface deflection vector for maximum virtual control input, degrees

u_{min} Control surface deflection vector for minimum virtual control input, degrees

u_{ub} Vector of upper limits of control surface deflections, degrees

V Air speed of wind tunnel, feet per second

v Virtual control input, (No Units)

v_e Equilibrium virtual control input, (No Units)

v_p Perturbation virtual control input, (No Units)

v_x Longitudinal body-axis component of wind tunnel velocity, feet per second

v_y Lateral body-axis component of wind tunnel velocity, feet per second

v_z Vertical body-axis component of wind tunnel velocity, feet per second

v_{max} Maximum attainable virtual control input, (No Units)

v_{min} Minimum attainable virtual control input, (No Units)

x Longitudinal tunnel-axis position, feet

x_b Longitudinal body-axis position, feet

y Lateral tunnel-axis position, feet

y_b Lateral body-axis position, feet

y_{cg} Lateral position of center of gravity, feet

z Vertical tunnel-axis position, feet

z_b Vertical body-axis position, feet

z_{cg} Vertical position of center of gravity, feet

AMS   Attainable moment set

DC   Direct current

LQR   Linear quadratic regulator

NASA   National Aeronautics and Space Administration

PC   Personal computer

PD   Proportional and derivative

PTI   Programmed test inputs

RMS   Root mean square

UDP   User Datagram Protocol

TABLE OF CONTENTS

Chapter

APPENDICES

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

The conventional paradigm for development of aircraft automatic flight control systems involves iterative wind tunnel testing and evaluation, numerous flight tests with precisely-flown maneuvers, and extensive efforts in the post-flight analysis. This process is expensive and time consuming and has some inherent fidelity limitations due to wind tunnel model scale and geometry differences with respect to the full-scale aircraft, aerodynamic interference from the wind tunnel walls and sting, and Reynolds number differences. The NASA Learn-To-Fly initiative represents a paradigm shift that could reduce the time and expense of development as well as avoiding some of the scaling and wind tunnel fidelity issues through real-time aerodynamic system identification and flight control. This means that algorithms are being developed that identify the aerodynamic model of an aircraft during a flight test based on flight data alone and use that model to tune control laws in real-time. This would be an enabling technology for rapid prototype aircraft development with additional applications in fault detection, self-learning vehicles, flight envelope protection, and autonomous unmanned aerial vehicles [1].

While the vision of Learn-To-Fly invlolves self-learning aircraft in free flight, the algorithms were first demonstrated and tested in a wind tunnel environment. In particular, this thesis focuses on a free-to-roll wind tunnel experiment in which the learning algorithms were tested in 1-degree-of-freedom. While the overall purpose of Learn-To-Fly is the learning algorithms, this thesis focuses on the design of a control system to take a model that has been learned and apply it to control the aircraft.

In the free-to-roll experiment, the aircraft can be oriented at different pitch and yaw angles and is free to roll about its longitudinal axis. The equation of motion for this type of test is known to be a second order nonlinear ordinary differential equation

of the roll angle. Some of the parameters are known or can be estimated ahead of time (e.g., mass, moment of inertia, wing dimensions, and center of gravity), but the aerodynamic rolling moment coefficient is a to-be-identified function of aircraft attitude, velocity, acceleration, and control inputs (control surface deflections). In the Learn-To-Fly free-to-roll experiment, there are seven independent control surfaces available for roll control: left and right ailerons, left and right flaps, left and right elevators, and rudder. Thus, it is an over-actuated system and some form of control allocation will be required. The aerodynamic rolling moment equation is assumed to be in a given form with coefficients that are unknown a priori. The goal is to design a control law in terms of the parameters of this equation in such a way that the roll performance of the model aircraft meets specifications.

1.2 PROBLEM STATEMENT

The goal of the project is to design and test a control system to control an aircraft's roll dynamics with limited a priori knowledge of the aerodynamic rolling moment function, and based on in-flight estimates of the aerodynamic properties provided by a real-time system identification process. The main mode of operation for the experiment has the aircraft at a fixed attitude, so a linear time invariant model may be derived. A second mode of operation was also tested where the free-to-roll rig is moved through a range of attitudes over time. Since the system is over actuated, the model will be expressed in terms of a virtual control input and a controller designed for that signal, then a control allocation will be designed to select a control actuator input to achieve the desired virtual control input [2].

So, in order to reach the goal there are a series of objectives to meet. First, based on the given form of the mathematical model, derive a linearized model for the case where the pitch and yaw angles are fixed. Then, based on that linearized model, design a controller in terms of a virtual control input. Thirdly, design a control allocator to allocate the virtual control input to the seven control surfaces. Finally, implement the design for wind tunnel runs and analyze the results.

1.3 OVERVIEW

The design and testing process are presented in two main chapters. Chapter 2 discusses the theoretical design from specifications to modeling to linear control and control allocation. In Chapter 3, the setup and results of the wind tunnel

experiments are presented and analyzed. Finally, Chapter 4 discusses the conclusions and suggested future work.

CHAPTER 2

THEORETICAL DESIGN

In order to control the roll dynamics of an aircraft, a control system must be designed to meet performance specifications. Since the aerodynamic model is to remain uncertain until the system is on-line, the control system should be designed to be general for any model of the given form. As a note of terminology, the experiment setup uses an airframe from a radio-controlled model airplane, and the control system theory uses the term 'model' referring to the mathematical description of the airplane's flight response characteristics. To avoid confusion over the meaning of 'model', the term 'aircraft' will be used for the physical model aircraft in the wind tunnel: it is not a full-scale flying aircraft. In this chapter, control specifications are laid out, the mathematical model of the aircraft is derived leaving some parameters to be identified, and then a control system is designed to control the model to achieve the specification. The control parameters will be expressed in terms of the aerodynamic coefficients so that when those parameters are identified, they can be used to update the control law in order ideally to meet the desired specifications.

2.1 SPECIFICATIONS

The roll angle response to a step command should have a settling time of less than 1.5 seconds, an overshoot less than 15%, and less than 10% steady state error.

2.1.1 REFERENCE MODEL SELECTION

In order to meet the specification, the control system is designed to track a reference model. This model is a roll response transfer function in the following form:

$$T_{ideal}(s) = \frac{\phi(s)}{\phi_c(s)} = \frac{\hat{a}_2}{s^2 + \hat{a}_1 s + \hat{a}_2}, \tag{1}$$

where the coefficients $\hat{a}_1$ and $\hat{a}_2$ can be chosen to match desired settling time and overshoot characteristics [3]. This ideal transfer function has unit DC gain. In general, a second order response can be expressed as

$$T(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2},$$

with the damping ratio ζ and the natural frequency ω_n. The damping ratio and natural frequency are approximate functions of the settling time (T_s) and percent overshoot ($\%OS$) of a linear system response [3], as specified by:

$$\zeta = -\frac{\ln(\%OS/100)}{\sqrt{\pi^2 + \ln^2(\%OS/100)}} \text{ and}$$

$$\omega_n \approx \frac{4}{\zeta T_s}.$$

So, given a desired percent overshoot and settling time specification, a damping ratio and natural frequency can be calculated, and thence the coefficients of the reference model:

$$\hat{a}_1 = 2\zeta\omega_n, \text{ and}$$

$$\hat{a}_2 = \omega_n^2,$$

which can be used in the linear control design to match the coefficients characteristic equation.

For example, using the specified 1.5 s settling time and 10% overshoot would result in a damping ratio of $\zeta = 0.59$, and a natural frequency of $w_n = 4.51$ rad/s, which would result in ideal characteristic equation coefficients of $\hat{a}_1 = 5.33$, and $\hat{a}_2 = 20.35$, which when simulated results in a settling time of 1.31 s and a percent overshoot of 9.98%.

2.2 MATHEMATICAL MODEL

In order to design a control system to meet the specifications, first a model of the system must be known. In the case of Learn-to-Fly, a general structure of the model is known and some of the mechanical parameters are measured ahead of time, while a parameterized aerodynamic rolling moment function is to be identified during the experiment. There is a coupling effect between some attitude angles with the roll

angle that is unique to the Free-to-roll type of experiment. Based on the structure of the aerodynamic model, some simplifications can be made to put the system in a form which is useful for designing a control system whose parameters depend on the aerodynamic model.

2.2.1 COUPLING OF BODY-AXIS AND TUNNEL-AXIS ATTITUDES

Fig. 1: Photograph of the aircraft on the free-to-roll rig in the wind tunnel.

In the free-to-roll experiment, the aircraft is mounted on a rig, as seen in Fig. 1, which can hold it at a fixed or slowly varying pitch (θ) and yaw (ψ) angles and allow the aircraft to roll freely. The rig consists of a sting to which the aircraft is mounted, which allows for rolling movement. The sting is mounted on a C-strut, and can be positioned at different pitch angles by moving up and down on the C-strut. Different yaw angles are achieved by rotating the C-strut from its base. The relationship between the pitch and yaw of the sting and the angle of attack (α) and sideslip (β)
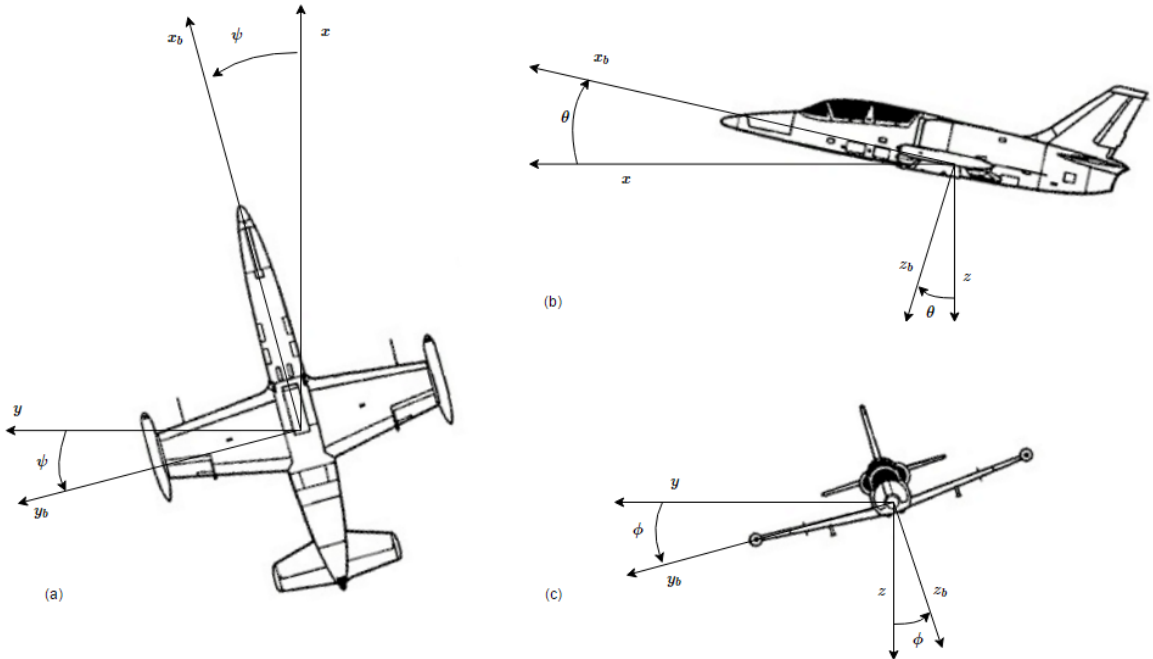
Fig. 2: (a) Yaw angle is positive-nose-right (perspective is looking up from beneath aircraft). (b) Pitch angle is positive-nose-up. (c) Roll angle is positive-right-wing-down.

are dependent on the roll angle (ϕ) of the aircraft on the sting in a nonlinear way. The sting pitch and yaw and the roll angle can be measured, but for estimating the aerodynamics, the independent variables should be the body-axis angles (α, β), thus they need to be calculated in terms of the sting attitude and roll angle. To show the mathematical relationship between these angles, consider a coordinate system oriented with the wind tunnel and then rotations are performed mathematically along the yaw, pitch, and roll directions to find the relationship between the body-axis coordinate system, which is defined by the aircraft, as shown in Fig. 2. Both coordinate systems would normally be centered at the center of gravity of the aircraft, but because the free-to-roll setup constrains the possible movement, the coordinates are here taken to be centered on the axis of ration of the free-to-roll rig.

Start by defining the wind tunnel coordinate system (x, y, z) with the air flowing from the positive toward the negative x direction, and with z vertically down (in the direction of earth's gravity). The aircraft body axis (x_b, y_b, z_b) is defined with x_b oriented toward the nose of the aircraft along the axis of rotation of the sting, and with y_b in the direction of the starboard wing. It is a right-handed coordinate
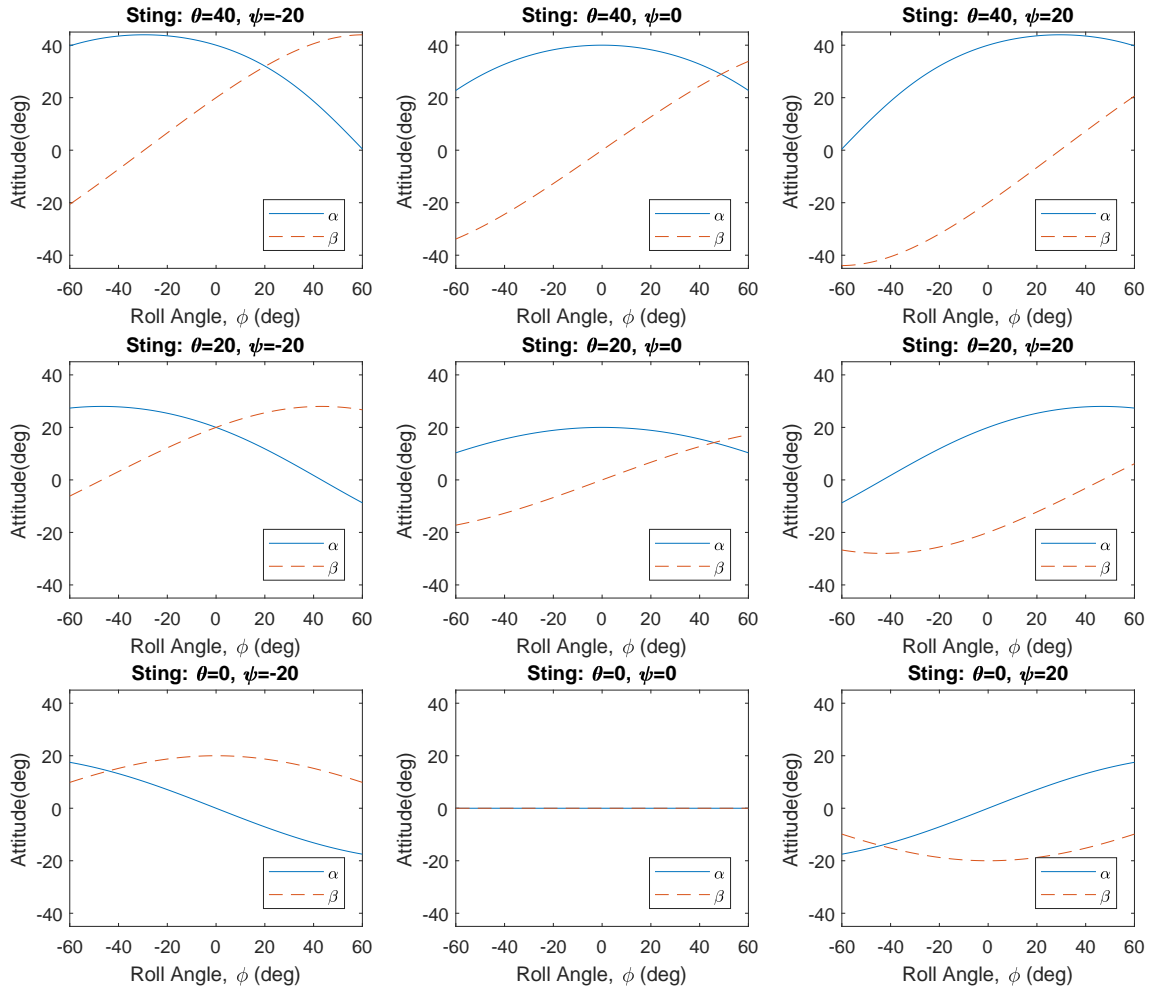
Fig. 3: Body-axis attitude as a function of roll angle for several fixed sting attitudes.

system, so z_b is oriented downward beneath the belly of the aircraft. The air flow in the wind tunnel is assumed to have a constant velocity flowing from positive toward the negative x direction. Aerodynamically this is equivalent to the case where the aircraft is traveling in the positive x direction with a constant velocity V. To express this velocity with respect to the body-axis, only a rotation is needed [4] which gives

$$v_x = V \cos\theta \cos\psi, \tag{2}$$

$$v_y = V \left(\sin\phi \sin\theta \cos\psi - \sin\psi \cos\phi\right), \text{and} \tag{3}$$

$$v_z = V \left(\cos\psi \cos\phi \sin\theta + \sin\psi \sin\phi\right), \tag{4}$$

where v_x, v_y, v_z are the velocity components in the x_b, y_b, z_b axes, respectively. The angle of attack and angle of sideslip define the orientation of the velocity vector with

respect to the body axes. The projection of the velocity vector onto the vertical symmetry plane ($x_b - z_b$ plane) defines the direction of the stability axis, and the angle between this axis and the x_b axis is called the angle of attack. The angle between the stability axis and the velocity vector is called the sideslip. These angles can be expressed in terms of the body-axis velocity components as:

$$\alpha = \tan^{-1}\left(\frac{v_z}{v_x}\right), \tag{5}$$

$$\beta = \sin^{-1}\left(\frac{v_y}{V}\right). \tag{6}$$

In order to express α and β in terms of the Euler angles (θ, ψ, ϕ), substitute (2),(3), and (4) into (5) and (6) resulting in the following expression:

$$\alpha = \tan^{-1}\left(\frac{\cos\psi\cos\phi\sin\theta + \sin\psi\sin\phi}{\cos\theta\cos\psi}\right), \tag{7}$$

$$\beta = \sin^{-1}\left(\sin\phi\sin\theta\cos\psi - \sin\psi\cos\phi\right). \tag{8}$$

Equations (7) and (8) can be used to determine, based on the sting position and roll angle, what is the angle of attack and sideslip. This is important because the given rolling moment model is defined in terms of angle of attack and sideslip, whereas the feedback measurements in the experiment measure the pitch, yaw, and roll but neither angle of attack nor sideslip directly. In Fig. 3 the relationship of α and β to roll angle is illustrated at several sting attitudes. These plots are calculated directly from equations (7) and (8). At the straight and level condition ($\theta = 0°$, $\psi = 0°$) both α and β are zero and independent of roll angle. As the pitch increases with yaw fixed at zero ($\theta = 20°$ and $\theta = 40°$, $\psi = 0°$) Fig. 3 shows that in the neighborhood of $\phi = 0°$, β can be approximated as a linear function of the roll angle, and θ is approximately constant. However, when the yaw angle is nonzero, this is no longer the case. Similarly, when $\theta = 0°$ and the roll angle is in the neighborhood of zero, $\beta \approx -\psi$.

2.2.2 FREE-TO-ROLL DYNAMICS

The governing equation for an aircraft in a free-to-roll experiment configuration is [5]:

$$I_{xx}\frac{\pi}{180}\dot{p} = \tau_{cg}(\phi) + \bar{q}SbC_l, \tag{9}$$

where p is the roll rate, I_{xx} is the moment of inertia of the aircraft, $\bar{q}$ is the dynamic pressure, S is the reference wing area, b is the wing span, C_l is the nondimensional aerodynamic rolling moment, and $\tau_{cg}(\phi)$ is the torque due to the center of gravity being off center from the axis of rotation. If the center of gravity is offset by z_{cg} above the axis of rotation and y_{cg} to the left, in the $-z_b$ and $-y_b$ directions respectively, then the torque due to this offset is:

$$\tau_{cg}(\phi) = mgl_{cg} \sin\left(\frac{\pi}{180}\xi\right), \tag{10}$$

where

$$l_{cg} = \sqrt{y_{cg}^2 + z_{cg}^2}, \tag{11}$$

and

$$\xi = \phi - \sin^{-1}\left(\frac{y_{cg}}{\sqrt{y_{cg}^2 + z_{cg}^2}}\right). \tag{12}$$

The aerodynamic rolling moment coefficient C_l is a function of α, β, p, and the vector of control surface deflections (u) [1]. There are seven control surfaces used in this experiment and they may each have different levels of effectiveness in different flight conditions, which will be taken into account by the system identification process. A positive deflection of each control surface is defined as trailing edge down (except the rudder is trailing edge left): left aileron δ_{ail-l}, right aileron δ_{ail-r}, left flap δ_{flap-l}, right flap δ_{flap-r}, rudder δ_{rudder}, left elevator δ_{elev-l}, and right elevator δ_{elev-r}. The control input u is a vector of the control surface deflections:

$$u = \begin{bmatrix} \delta_{ail-l} & \delta_{ail-r} & \delta_{flap-l} & \delta_{flap-r} & \delta_{rudder} & \delta_{elev-l} & \delta_{elev-r} \end{bmatrix}'. \tag{13}$$

The nonlinear state equation can be expressed as follows:

$$\begin{bmatrix} \dot{\phi} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} p \\ \dfrac{180}{\pi I_{xx}}\left(\tau_{cg}(\phi) + \bar{q}SbC_l\left(\alpha, \beta, p, u\right)\right) \end{bmatrix}. \tag{14}$$

The rolling moment coefficient function is assumed to be in the form of a polynomial

whose coefficients are to be identified during the experiment:

$$C_l\left(\alpha, \beta, \hat{p}, u\right) = k_1 + k_2\alpha + k_3\beta + k_4\hat{p}$$
$$+ k_5\delta_{ail-l} + k_6\delta_{ail-r} + k_7\delta_{flap-l} + k_8\delta_{flap-r} + k_9\delta_{rudder} + k_{10}\delta_{elev-l}$$
$$+ k_{11}\delta_{elev-r} + k_{12}\alpha^2 + k_{13}\beta\alpha + k_{14}\hat{p}\alpha + k_{15}\delta_{ail-l}\alpha + k_{16}\delta_{ail-r}\alpha$$
$$+ k_{17}\delta_{flap-l}\alpha + k_{18}\delta_{flap-r}\alpha + k_{19}\delta_{rudder}\alpha + k_{20}\delta_{elev-l}\alpha + k_{21}\delta_{elev-r}\alpha$$
$$+ k_{22}\beta^2 + k_{23}\hat{p}\beta + k_{24}\delta_{ail-l}\beta + k_{25}\delta_{ail-r}\beta + k_{26}\delta_{flap-l}\beta$$
$$+ k_{27}\delta_{flap-r}\beta + k_{28}\delta_{rudder}\beta + k_{29}\delta_{elev-l}\beta + k_{30}\delta_{elev-r}\beta, \tag{15}$$

where $\hat{p} = \dfrac{\left(\frac{\pi}{180}p\right)b}{2V}$ is the nondimensional roll rate [1], [6]. The C_l polynomial is defined in terms of the nondimensional roll rate because it is conventional in aerodynamic experiments to use a nondimensional roll rate. Collecting like terms of the C_l function and switching to the pure roll rate as one of the inputs gives the following expression:

$$C_l\left(\alpha, \beta, p, u\right) = C_{l,0}\left(\alpha, \beta\right) + C_{l,p}\left(\alpha, \beta\right)p + C_{l,u}\left(\alpha, \beta\right)u, \tag{16}$$

where

$$C_{l,0}\left(\alpha, \beta\right) = k_1 + k_2\alpha + k_3\beta + k_{12}\alpha^2 + k_{13}\beta\alpha + k_{22}\beta^2, \tag{17}$$

$$C_{l,p}\left(\alpha, \beta\right) = \left(k_4 + k_{14}\alpha + k_{23}\beta\right)\frac{b}{2V}\frac{\pi}{180}, \tag{18}$$

$$C_{l,u}\left(\alpha, \beta\right) = \begin{bmatrix} k_5 + k_{15}\alpha + k_{24}\beta \\ k_6 + k_{16}\alpha + k_{25}\beta \\ k_7 + k_{17}\alpha + k_{26}\beta \\ k_8 + k_{18}\alpha + k_{27}\beta \\ k_9 + k_{19}\alpha + k_{28}\beta \\ k_{10} + k_{20}\alpha + k_{29}\beta \\ k_{11} + k_{21}\alpha + k_{30}\beta \end{bmatrix}' = \begin{bmatrix} C_{l,ail-l}(\alpha, \beta) \\ C_{l,ail-r}(\alpha, \beta) \\ C_{l,flap-l}(\alpha, \beta) \\ C_{l,flap-r}(\alpha, \beta) \\ C_{l,rudder}(\alpha, \beta) \\ C_{l,elev-l}(\alpha, \beta) \\ C_{l,elev-r}(\alpha, \beta) \end{bmatrix}', \tag{19}$$

and $'$ denotes transposition. For control purposes, define a virtual control input v that represents the nondimensional moment coefficient due to the linear combination of the control surface deflections,

$$v = C_{l,u}\left(\alpha, \beta\right)u. \tag{20}$$

Then the controller can be designed with one control input and then the control input can be allocated to the different surfaces in a separate step. Substituting the

resulting expansion of C_l allows the nonlinear state equation to be expressed as

$$\begin{bmatrix} \dot{\phi} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} p \\ \dfrac{180}{\pi I_{xx}} \left(\tau_{cg}(\phi) + \bar{q} S b \left(C_{l,0}\left(\alpha,\beta\right) + C_{l,p}\left(\alpha,\beta\right) p + v \right) \right) \end{bmatrix}. \tag{21}$$

Note that at a given sting position (θ, ψ) the terms $C_{l,0}\left(\alpha,\beta\right)$ and $C_{l,p}\left(\alpha,\beta\right)$ are nonlinear functions of ϕ because of the dependence of α and β on $\theta, \psi,$ and ϕ as per equations (7) and (8). In order to apply linear control methods, the model is linearized with respect to ϕ and p.

2.3 LINEARIZATION

When the free-to-roll rig is at a fixed position and the roll angle is commanded to a set value ϕ_c, the model is linearized at a constant θ, ψ and the roll angle command is taken as the equilibrium point, so that the linear system is centered around the desired position. Let $\phi(t) = \phi_c$ and $p = 0$ be the desired equilibrium values, and the perturbation values are defined as $\phi(t) = \phi_c + \phi_p(t)$, and $p(t) = p_e + p_p(t)$, but with $p_e = 0$, so $p_p(t) = p(t)$. Define the α, β at the equilibrium roll angle for a given θ, ψ, ϕ_c as α_e, β_e. The virtual control input is separated into an equilibrium virtual control input (i.e.,virtual trim) v_e and a perturbation virtual control input v_p such that $v = v_e + v_p$. This way, v_e can be used to balance the equilibrium equation and v_p can be used to control the linear perturbation equation. Since the first entry in equation (21) is already linear with respect to the state, it is only the second entry, or

$$F(\phi, p, v) = \frac{180}{\pi I_{xx}} \left(\tau_{cg}(\phi) + \bar{q} S b \left(C_{l,0}\left(\alpha,\beta\right) + C_{l,p}\left(\alpha,\beta\right) p + v \right) \right)$$

that is to be linearized by a first order Taylor series expansion at the equilibrium point, that is,

$$F(\phi, p, v) \approx F(\phi_c, 0, v_e) + F_\phi(\phi_c, 0, v_e)\phi_p + F_p(\phi_c, 0, v_e)p + F_v(\phi_c, 0, v_e)v_p,$$

where it follows by substitutions and differentiations that

$$F(\phi_c, 0, v_e) = \frac{180}{\pi I_{xx}} \left(\tau_{cg}(\phi_c) + \bar{q} S b \left[C_{l,0}\left(\alpha_e,\beta_e\right) + v_e \right] \right),$$

$$F_\phi(\phi_c, 0, v_e) = \frac{180}{\pi I_{xx}} \left(\tau_{cg,\phi}(\phi_c) + \bar{q} S b C_{l,\phi}\left(\alpha_e,\beta_e\right) \right),$$

$$F_p(\phi_c, 0, v_e) = \frac{180}{\pi I_{xx}} \bar{q} S b C_{l,p}\left(\alpha_e,\beta_e\right), \text{ and}$$

$$F_v(\phi_c, 0, v_e) = \frac{180}{\pi I_{xx}},$$

with $C_{l,\phi}(\alpha_e, \beta_e)$ the derivative of $C_{l,0}(\alpha, \beta)$ with respect to ϕ evaluated at ϕ_c, and $\tau_{cg,\phi}(\phi_c)$ the derivative of $\tau_{cg}(\phi)$ with respect to ϕ evaluated at ϕ_c. To calculate $C_{l,\phi}(\alpha_e, \beta_e) = C_{l,\phi}(\theta, \psi, \phi_c) = \left.\dfrac{\partial}{\partial\phi}C_{l,0}(\theta, \psi, \phi)\right|_{\phi_c}$, substitute α_e and β_e for θ, ψ, ϕ_c in (17) using (7) and (8) and then differentiate resulting in

$$C_{l,\phi}(\alpha_e, \beta_e) = k_2\alpha_\phi + k_3\beta_\phi + 2k_{12}\alpha\alpha_\phi + k_{13}(\alpha\beta_\phi + \beta\alpha_\phi) + 2k_{22}\beta\beta_\phi \tag{22}$$

$$\alpha_\phi(\theta, \psi, \phi_c) = \frac{c_{\phi_c} s_\psi - c_\psi s_{\phi_c} s_\theta}{c_\psi c_\theta \left(\dfrac{(s_{\phi_c} s_\psi + c_{\phi_c} c_\psi s_\theta)^2}{c_\psi^2 c_\theta^2} + 1\right)} \tag{23}$$

$$\beta_\phi(\theta, \psi, \phi_c) = \frac{s_{\phi_c} s_\psi + c_{\phi_c} c_\psi s_\theta}{\sqrt{1 - (c_{\phi_c} s_\psi - c_\psi s_{\phi_c} s_\theta)^2}} \tag{24}$$

with the shorthand notation $c_\theta = \cos\theta$, and $s_\theta = \sin\theta$ used to save space, and where $\alpha_\phi = \dfrac{\partial}{\partial\phi}\alpha$, and $\beta_\phi = \dfrac{\partial}{\partial\phi}\beta$. For the purposes of implementation, however, this derivative is estimated numerically using a roll angle perturbation Δ_ϕ at a given sting position. The perturbed position (α_p, β_p) is calculated from (7) and (8) evaluated at the given sting position θ, ψ and the perturbed roll angle $\phi_c + \Delta_\phi$. Assuming the coefficients of (15) are known, evaluate $C_{l,0}$ at the equilibrium (α_e, β_e) and at the perturbed position (α_p, β_p) and evaluate the difference

$$C_{l,\phi}(\alpha_e, \beta_e) = \frac{C_{l,0}(\alpha_e, \beta_e) - C_{l,0}(\alpha_p, \beta_p)}{\Delta_\phi}. \tag{25}$$

In the experiment, a perturbation of $1°$ was used. The fidelity of this numerical approximation can be checked by using the aerodynamic model that was identified during the experiment and perform both the analytical calculation from (22)-(24), and the numerical approximation from (25). Running this check with the model generated during Run 1 on 10/21/2015 resulted in a maximum percent difference of 2.6%, which indicates that the numerical approximation was reasonably accurate.

The derivative of the center-of-gravity torque with respect to ϕ is:

$$\tau_{cg,\phi}(\phi_c) = \frac{\pi}{180}mgl_{cg}\cos\left(\frac{\pi}{180}\left(\phi_c - \sin^{-1}\left(\frac{y_{cg}}{\sqrt{y_{cg}^2 + z_{cg}^2}}\right)\right)\right).$$

Thus, the linearized system takes the form

$$\begin{bmatrix} \dot{\phi}_c \\ \dot{p}_e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ F(\phi_c, 0, v_e) \end{bmatrix},$$

$$\begin{bmatrix} \dot{\phi}_p \\ \dot{p}_p \end{bmatrix} = \begin{bmatrix} p_p \\ F_\phi(\phi_c, 0, v_e)\phi_p + F_p(\phi_c, 0, v_e)p_p + F_v(\phi_c, 0, v_e)v_p \end{bmatrix},$$

which simplifies to

$$0 = F(\phi_c, 0, v_e),$$

$$\begin{bmatrix} \dot{\phi}_p \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ F_\phi(\phi_c, 0, v_e) & F_p(\phi_c, 0, v_e) \end{bmatrix} \begin{bmatrix} \phi_p \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ F_v(\phi_c, 0, v_e) \end{bmatrix} v_p.$$

Substituting the derivatives calculated previously gives the equilibrium trim equation

$$0 = \frac{180}{\pi I_{xx}} \left(\tau_{cg}(\phi_c) + \bar{q}Sb \left[C_{l,0}\left(\alpha_e, \beta_e \right) + v_e \right] \right), \tag{26}$$

and the linear state space equation:

$$\begin{bmatrix} \dot{\phi}_p \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \dfrac{180}{\pi I_{xx}} \left(\tau_{cg,\phi}(\phi_c) + \bar{q}SbC_{l,\phi}\left(\alpha_e, \beta_e \right) \right) & \dfrac{180}{\pi I_{xx}} \bar{q}SbC_{l,p}\left(\alpha_e, \beta_e \right) \end{bmatrix} \begin{bmatrix} \phi_p \\ p \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ \dfrac{180}{\pi I_{xx}} \bar{q}Sb \end{bmatrix} v_p,$$

alternatively notated as

$$\begin{bmatrix} \dot{\phi}_p \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} \phi_p \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ b_2 \end{bmatrix} v_p, \tag{27}$$

where

$$a_1 = -\frac{180}{\pi I_{xx}} \bar{q}SbC_{l,p}\left(\alpha_e, \beta_e \right),$$

$$a_2 = -\frac{180}{\pi I_{xx}} \left(\tau_{cg,\phi}(\phi_c) + \bar{q}SbC_{l,\phi}\left(\alpha_e, \beta_e \right) \right), \text{ and}$$

$$b_2 = \frac{180}{\pi I_{xx}} \bar{q}Sb.$$

The system having been thus linearized is in a form conducive to linear control system design.

2.4 CONTROL

A modular approach to control design allows for a degree of flexibility. The control architecture has a trim offset, a linear controller and a control allocator. The trim offset will set the equilibrium at the commanded roll angle, the linear controller will effect the desired transient response, and the control allocator will distribute the virtual control input to the various control surfaces.
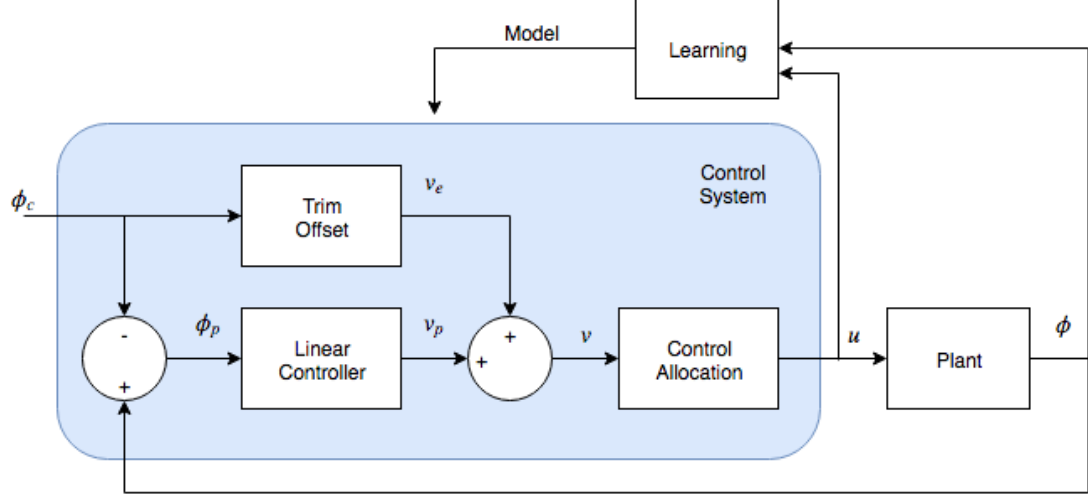
Fig. 4: High level control architecture.

2.4.1 TRIM OFFSET

The trim virtual control input is calculated as a solution, if one exists, to (26).

$$v_e = -\frac{\tau_{cg}(\phi_c)}{\bar{q}Sb} - C_{l,0}\left(\alpha_e, \beta_e\right).\tag{28}$$

This trim virtual control input should cancel the mechanical and aerodynamic forces acting on the aircraft at steady state at the commanded roll angle, so that without perturbations, it would maintain that position. The next step is to derive a linear controller to compensate for perturbations that will happen and keep the aircraft stable at the equilibrium.

2.4.2 LINEAR CONTROL

Pole placement is a linear control approach which allows gains to be chosen to achieve a desired characteristic equation. Let $\hat{a}_1, \hat{a}_2$ be the coefficients of the desired characteristic equation $(0 = s^2 + \hat{a}_1 s + \hat{a}_2)$. Given the linear plant model from (27), a control law with a proportional and a derivative gain can be chosen in order to move the poles of the system to the desired location. The time response of the plant is described by:

$$\ddot{\phi}_p = -a_1 \dot{\phi}_p - a_2 \phi_p + b_2 v_p.\tag{29}$$

The proposed control law is:

$$v_p = -K_d\dot{\phi}_p - K_p\phi_p. \tag{30}$$

Substituting (29) in (30) gives the closed loop system

$$\ddot{\phi}_p = -(a_1 + b_2 K_d)\dot{\phi}_p - (a_2 + b_2 K_p)\phi_p.$$

Based on the specifications, the ideal response should be described by

$$\ddot{\phi}_p = -\hat{a}_1\dot{\phi}_p - \hat{a}_2\phi_p,$$

so by matching the coefficients, it follows that the controller gains are

$$K_d = \frac{\hat{a}_1 - a_1}{b_2} \text{ and}$$
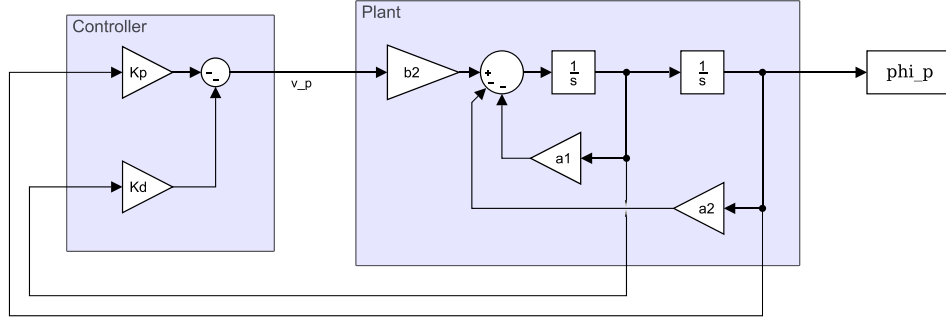
$$K_p = \frac{\hat{a}_2 - a_2}{b_2}.$$

Fig. 5: Control diagram of linear regulator.

This linear portion of the control system is a regulator that is designed to keep the perturbation roll angle ϕ_p close to zero, since the equilibrium roll angle is selected at the commanded roll angle ϕ_c, thus the perturbation roll angle $\phi_p = \phi - \phi_c$ is essentially a tracking error. A Simulink®model diagram of this part of the system is shown in Fig. 5.

To illustrate this part of the design, consider a setpoint of $\theta = 5°$, $\psi = 0°$, $\phi_c = -20°$ and using plant parameters from experiment run 1 on 10/21/2015 which result in $a_1 = 6.4690$, $a_2 = 5.1279$, and $b_2 = 17598$. If the example reference model in Section 2.1.1 is used, then the ideal characteristic polynomial coefficients
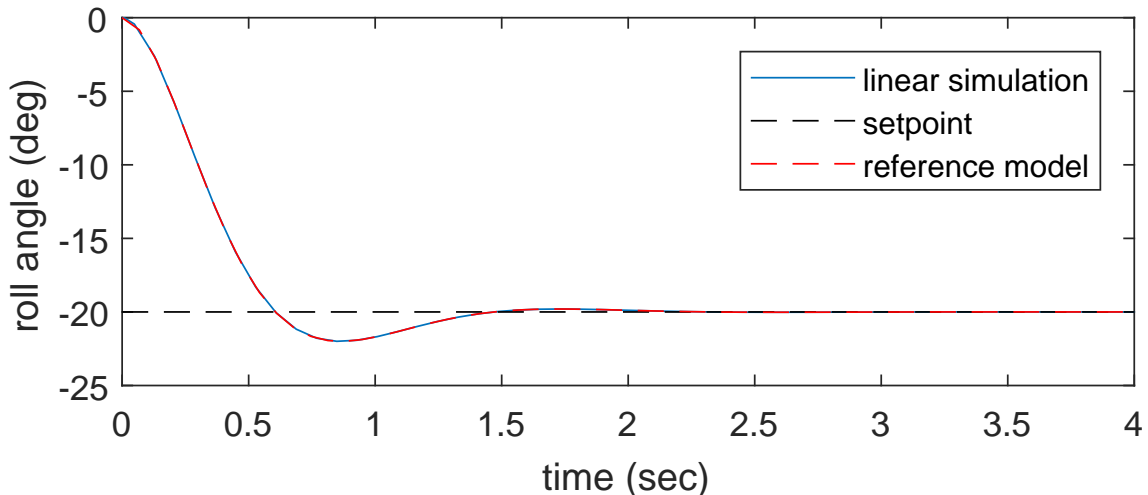
Fig. 6: Simulated response of linear regulator with initial conditions.

are $\hat{a}_1 = 5.33$ and $\hat{a}_2 = 20.35$ so the resulting gains at this setpoint with this reference model would be $K_d = -6.4537 \times 10^{-5}$ and $K_p = 8.6493 \times 10^{-4}$. These values are provided to the Simulink®model, and the model is simulated with an initial condition of $\phi_{p,0} = 20°$, which would be the initial condition if ϕ started from zero with $\phi_c = -20°$. The response is shown in Fig. 6, with a settling time of $T_s = 1.3149s$, and overshoot of $\%OS = 9.9765$ which are within the specifications.

2.4.3 CONTROL ALLOCATION

Once the perturbation virtual control input v_p has been determined by the linear control law, it is combined with the trim value v_e to get the total virtual control input $v = v_e + v_p$. This virtual control input represents the desired nondimensional rolling moment coefficient due to control surfaces ($C_{l,u}u$), and must then be distributed to the available control surfaces by choosing among the infinity of solutions of (20) for u within the deflection constraints. Since all seven control surfaces are available for roll control, the aircraft is over-actuated. In a standard aircraft configuration, the ailerons primarily are used for roll control, but in this experiment the flaps, elevators and rudder are also used for additional roll input. The control allocation problem can become complex because of allocating three moments (pitch, roll, and yaw) among multiple control effectors, and there is much research in the optimal and/or efficient ways to do this (see, for example, [2] [7]). In the free-to-roll case, only the roll moment is being considered, making it possible to utilize a simpler allocation

method. In this paper, a direct allocation is selected, which uses all the surfaces by the same proportion of their deflection limit in the direction of the desired moment, which is a simplified case of the method presented in [8]. The main advantage of this method is that it is guaranteed to make full use of the available control surfaces in cases where maximum control input is needed. Each control surface has an upper and lower deflection limit given by

$$u_{ub} = \begin{bmatrix} 25° & 25° & 40° & 40° & 30° & 30° & 30° \end{bmatrix}' \text{ and}$$

$$u_{lb} = \begin{bmatrix} -25° & -25° & 0° & 0° & -30° & -30° & -30° \end{bmatrix}',$$

where each element corresponds to each control surface in the same order as in u, defined in (13). In other words, the ailerons can deflect plus or minus 25°, the flaps can deflect only down up to 40°, and the elevators and rudder can deflect plus or minus 30°. The set of moments that can be achieved with some combination of the control effectors within their constrains is known as the attainable moment set (AMS), and since the control effectors are linear and have a continuous attainable region in this case and only the roll moment is considered, then the AMS can be assumed to be a continuous segment with upper bound v_{max} and lower bound v_{min}. The control deflections that correspond to the upper and lower boundaries of the AMS (call them u_{max} and u_{min} respectively) can be found by saturating each control surface in the direction that results in either a positive or negative moment respectively. The direction in which each control surface is effective is determined by the sign of the corresponding element of $C_{l,u}$, so the selection of u_{max} and u_{min} can be defined as:

If $C_{l,u_i} > 0$, then $u_{max,i} = u_{ub,i}$, and $u_{min,i} = u_{lb,i}$,

else if $C_{l,u_i} < 0$, then $u_{max,i} = u_{lb,i}$, and $u_{min,i} = u_{ub,i}$,

else if $C_{l,u_i} = 0$, then $u_{max,i} = u_{min,i} = 0$, for all $i \in \{1, 2, ..., 7\}$.

For example, if $C_{l,u} = [0.0015, -0.0014, 0.00025, -0.00028, 0.00014, 0.000032, -0.00014]$, the signs are $[+, -, +, -, +, +, -]$, and therefore $u_{max} = [u_{ub,1}$ $u_{lb,2}$ $u_{ub,3}$ $u_{lb,4}$ $u_{ub,5}$ $u_{ub,6}$ $u_{lb,7}]' = [25, -25, 40, 0, 30, 30, -30]'$, with a resulting $v_{max} = C_{l,u}u_{max} = [0.0015, -0.0014, 0.00025, -0.00028, 0.00014, 0.000032, -0.00014][25, -25, 40, 0, 30, 30, -30]' = 0.0919$ and the lower bound similarly would be $u_{min} = [-25, 25, 0, 40, -30, -30, 30]$ resulting in $v_{min} = -0.0931$.

The direct control allocation finds the point on the boundary of the AMS that is in the direction of the desired moment (in this case v_{max} or v_{min}), and takes
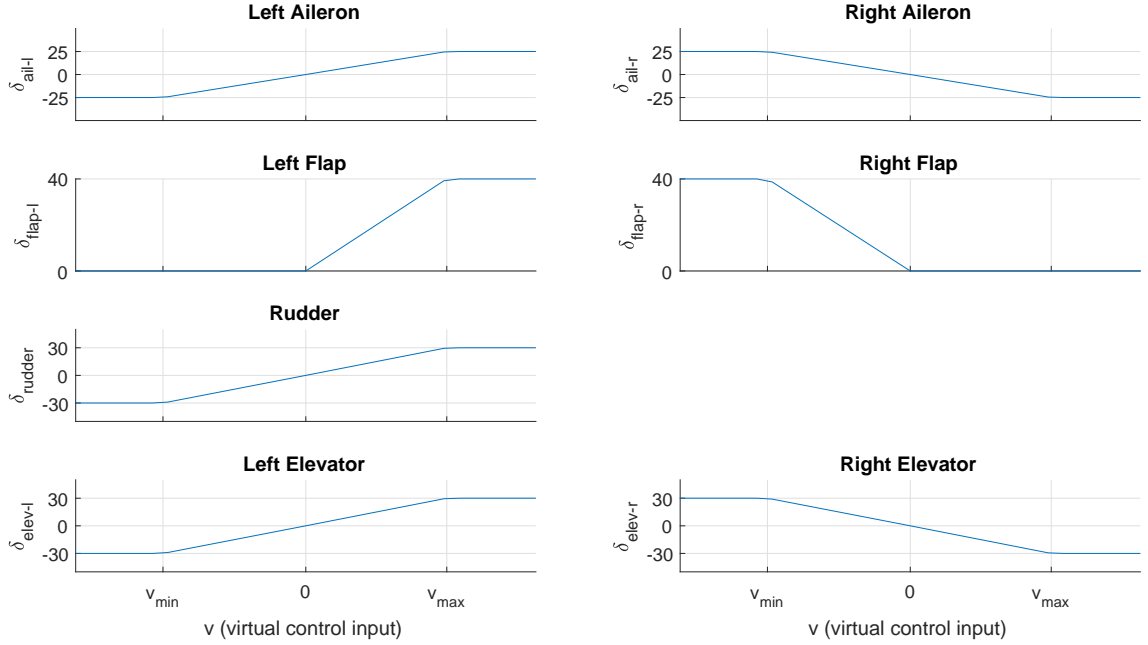
Fig. 7: Control Allocation.

the corresponding control vector (either u_{max} or u_{min}) and scales it to match the magnitude of the desired moment. In other words, if $0 < v < v_{max}$ then $u = Ku_{max}$, where $K = \dfrac{|v|}{|v_{max}|}$. If $v > v_{max}$, then u_{max} represents the best attempt to achieve a moment in the desired direction with all the surfaces saturated. The same is done in the negative direction, so the control allocation algorithm can be summarized as:

$$\text{If } v \leq v_{min}, \text{ then } u = u_{min},$$

$$\text{else if } v_{min} < v < 0, \text{ then } u = \frac{|v|}{|v_{min}|}u_{min},$$

$$\text{else if } v = 0, \text{ then } u = 0,$$

$$\text{else if } 0 < v < v_{max}, \text{ then } u = \frac{|v|}{|v_{max}|}u_{max},$$

$$\text{else if } v_{max} \leq v, \text{ then } u = u_{max}.$$

Using the above mentioned example value of $C_{l,u}$, the plots in Fig. 7 show the control surface deflections versus the desired virtual control input to illustrate the outcome of the direct control allocation. Within the attainable moment set, each control surface is deflected by the same proportion of its limit in the direction of the desired control input, and all saturate at the maximum or minimum virtual control input in the

positive and negative directions respectively.

2.4.4 SIMULATION

The control design is illustrated with a simulation based on sample values taken from an experiment run. The same model and setpoint are used from the example in Section 2.4.2, namely $\theta = 5°$, $\psi = 0°$, $\phi_c = -20°$ and roll angle initial condition of $\phi = 0°$. In this simulation the full nonlinear plant model and control allocation are evaluated. The Simulink®model can be found in Appendix A, and the step response is shown in Fig. 8, with a settling time of 1.3138 s, and a percent overshoot of 10.2258, which are as specified.
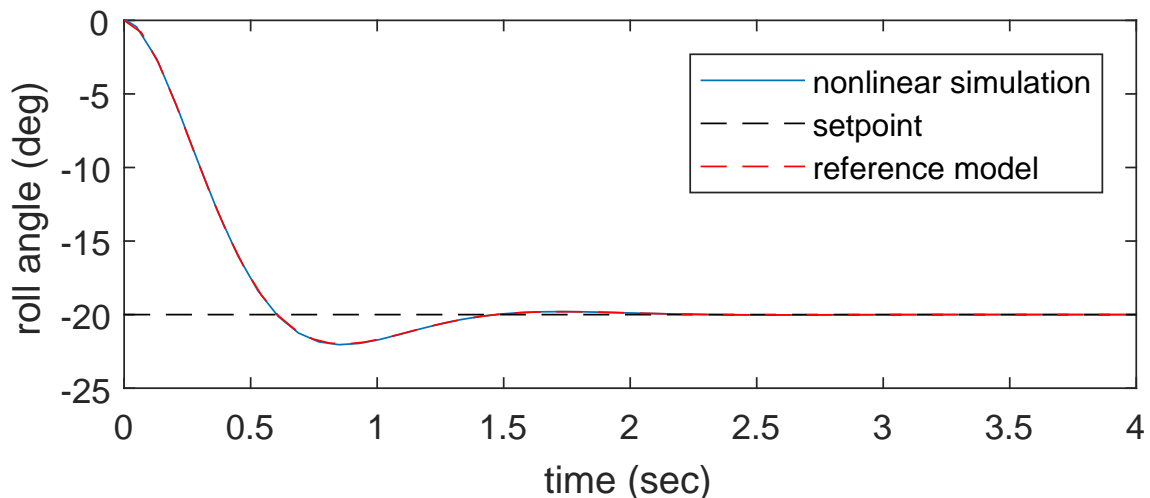

Fig. 8: Simulated response of nonlinear system.

2.4.5 GAIN SCHEDULING

The control design heretofore assumes a fixed θ and ψ. The gains and trim offset can be calculated for any sting attitude for a given aerodynamic model. During the experiment, the sting can be moved and the aerodynamic model can be updated. A controller can be calculated at each time sample for the current sting attitude. Alternatively, for a given model, the gains can be calculated for a sampling of sting positions and then interpolated during the run, assuming that the parameters vary slowly with sting position. Either way, whenever the aerodynamic model is updated, the new coefficients are immediately used in the calculations of the control parameters.

CHAPTER 3

EXPERIMENT

3.1 INTRODUCTION

The control system proposed in the previous chapter is evaluated with wind tunnel experiments conducted in NASA's 12-ft low-speed wind tunnel at Langley Air Force Base in Hampton, Virginia. A model airplane designed for recreational radio-controlled flying was adapted to be mounted to the tunnel's free-to-roll rig, as shown in Fig. 1. The aircraft was modeled after the Aero L-59 Super Albatross military jet, but is not necessarily a precise replica. It was chosen for low cost since the purpose of the experiment was not to design control systems for a specific airframe, but to demonstrate the algorithm's ability to learn the flight characteristics of an aircraft. This chapter will cover the experimental setup and results.

3.2 STATIC TESTING

Before the free-to-roll experiment, static tests were conducted to measure the constant parameters characterizing the free-to-roll dynamics in (9) - (12). The measured values of these static parameters are found in Table 1. The weight and moment of inertia of the aircraft, as well as the wing area and span are all constant parameters that are found in the governing equation (9) and are measured before the dynamic experiment. The location of the center of gravity relative to the axis of rotation is of importance because of the pendulum effect that results if the center of gravity is not aligned with the axis of rotation, modeled by the torque-due-to-center-of-gravity (τ_{cg}) term in (9). This effect does not exist in normal free flight of an aircraft, but is an artifact of the free-to-roll experiment setup. In order to quantify the effect, the vertical (z_{cg}) and horizontal (y_{cg}) offsets from the axis of rotation are measured as well as the weight of the aircraft. At the outset of the experiment, the existence of the horizontal offset was not known, so the experiment was designed with only the vertical offset taken into account. Thus, whatever torque was due to the horizontal offset during the experiment was treated as an aerodynamic torque and its effect is

Parameter	Value	Units	Description
z_{cg}	0.0122	ft	Vertical position of c.g. above roll axis
y_{cg}	0.0023	ft	Horizontal position of c.g. left of roll axis
I_{xx}	0.0805	slug-ft^2	Moment of Inertia
mg	21.72	lb	Weight
S	3.14	ft^2	Wing reference area
b	3.937	ft	Wing reference span

Table 1: Static parameters.

reflected in the aerodynamic model, thus a later attempt to account for the horizontal offset using the same model only induced an additional error because the model was derived without accounting for it. Therefore, the equations to account for the horizontal offset are included, but the experimental results and code in the thesis are implemented assuming no horizontal offset.

3.3 INSTRUMENTATION AND ACTUATION

The wind tunnel experiment setup includes instrumentation to measure the roll angle and roll rate independently using an analog angle resolver (Data Device Corp. Part #AP1-30604-301) and roll rate gyroscope (Systron Donner QRS116-0200-200 Quart Rate Sensor) respectively. The control surfaces are actuated by seven servo motors (ailerons: Futaba S9602 servos, the rest: Futaba S3150 servos) which are controlled through an Arduino Due that had a Adafruit PWM/Servo shield (https://www.adafruit.com/product/1411) mounted on it with commands sent via USB connection from the experiment operating PC. The positions of the control surfaces are measured by potentiometers, whose analog outputs are captured by National Instruments 6052E data acquisition board along with the roll angle and rate as well as the free-to-roll rig angles θ and ψ and the dynamic pressure $\bar{q}$ which is held at approximately 2 lbf/ft^2 during this experiment. Roll angle and rate (ϕ and p) are filtered with a Krohn-THE Model 3362 filter using 50 Hz butterworth low-pass DC filters before entering the NI board.
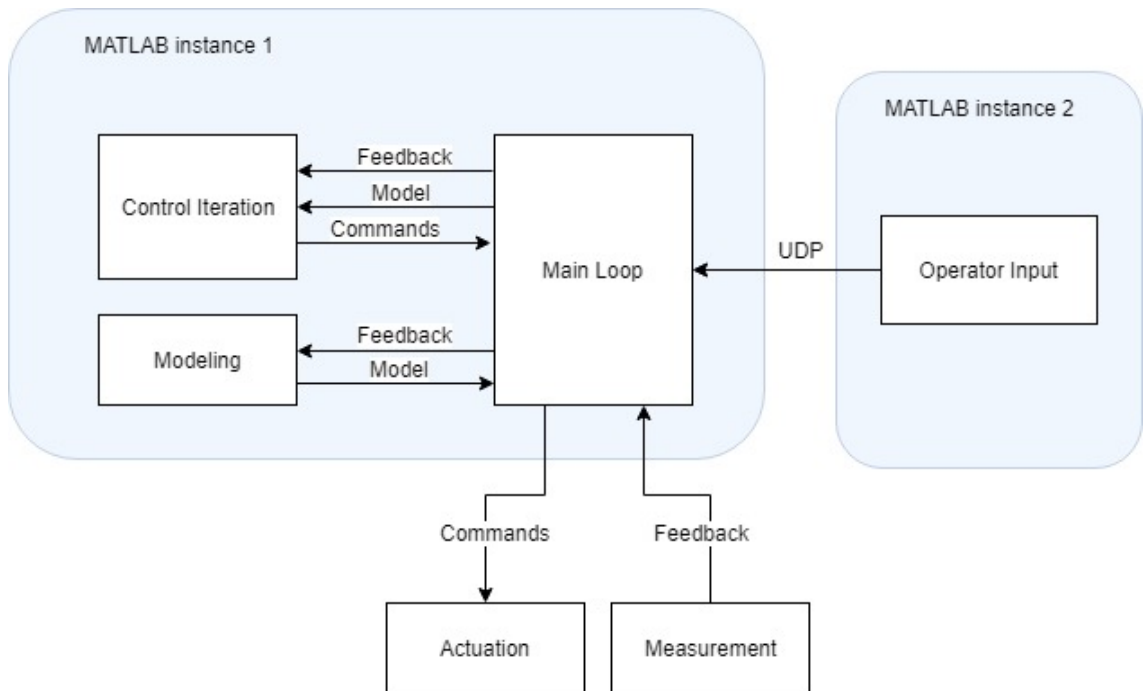
Fig. 9: High level software architecture.

3.4 SOFTWARE ARCHITECTURE

The high-level software architecture is illustrated in Fig. 9, showing the main elements of the software that is used to execute the experiment. The experiment is executed from a Windows, 64-bit PC running MATLAB r2015b connected to National Instruments data acquisition hardware for feedback of measured variables, and an Arduino Due for control surface actuation. In order to allow the experiment operator to control the mode of the experiment, a second instance of MATLAB runs in parallel with the first, allowing the operator to type in commands to send messages via User Datagram Protocol (UDP) to the main MATLAB process to change certain operating modes during the experiment. The mode commands are: 'PTI_on' to begin the programmed test inputs (PTI), 'PTI_off' to turn off the PTI, 'Model_on' to begin the modeling process, and 'STOP_aero' to terminate the experiment run. The PTI are pseudo-random perturbations that are added to the control surface deflections during system identification in order to decouple the effects of each control surface on the aerodynamic moment. The feedback data received from the measurement system (roll angle, roll rate, sting position, dynamic pressure, control surface deflections) is

provided to the Control Iteration function and the Modeling function at each sample time (approximately 50 Hz sample rate), and the Modeling code returns an updated aerodynamic model based on the system identification algorithm. The Control Iteration function takes the current feedback data and the most recent aerodynamic model as inputs, and returns a control surface deflection command for each of the seven control surfaces. This command is sent to the actuation system at each sample time.

3.5 EXPERIMENTAL PROCEDURE

For the experiment, several types of runs were executed and several other control methods that are not presented in this thesis were tested [9], [10]. In the first type of run, only a few sting positions were used (ψ: 0°, θ: 5° to 10° to 15°), and the sting was held at a fixed position for enough time to demonstrate the aircraft's response at that attitude with a roll angle doublet command, which alternates the commanded roll angle ϕ_c from 0° to −20° to 20°, then back to 0°, holding it constant for about 5 seconds at each set point. The system started off with a baseline controller, which was a simple PD controller using only the ailerons with gains selected by trial and error to give a basically stable response from which the system can begin the learning process. Then a period of learning was initiated by the experiment operator. During the learning part of an experiment run, a pseudo-random perturbation was added to the control surfaces which was intended to decouple the effects of the various control surfaces. Next, the learning process would be switched off and the newly calculated control system's performance observed. This process was then repeated at a few other attitudes. The run shown in Fig. 11 follows this procedure. In the first doublet, the baseline control is active, then the response gets visibly noisier, which indicates a period where learning is active and the system identification perturbations are being applied to the control surfaces. In the period right after the learning noise stops the controller has been updated with new parameters from the learning. At around 70 and 100 seconds, the sting pitch is changed, which is followed by a deterioration in response before the learning is switched back on.

The second type of run also uses a few fixed sting positions, but rather than alternating between learning and observation within the same run, a complete run is done with learning switched on and with the controller being updated continuously as the model is learned, as in Run 1 of 10/21/2015, and then the model generated
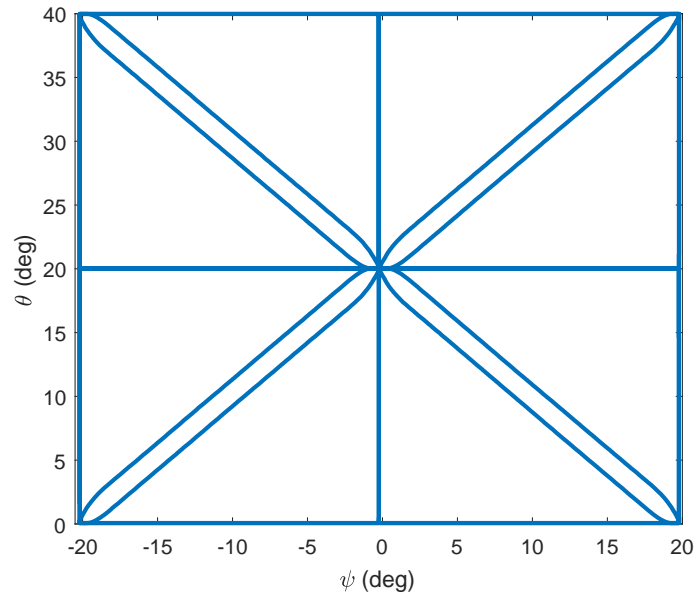
Fig. 10: The sting is moved during the experiment covering a range of pitch and yaw angles in a pattern referred to as the cross-box.

at the end of that run is used in several independent runs to observe the resulting performance of several control methods based on the same identified model. For example, Runs 2 and 4 of 10/21/2015 use the model generated from Run 1 on that day. Run 2 of 10/21/2015 tests a control setup that calculates the control parameters from the model at each time step for the present conditions, resulting in an instantaneous gain calculation, whereas Run 4 calculates the gains for a sampling of sting positions and roll commands and stores them in tables to be looked-up during the run.

The third type of run involves covering a wide range of sting attitudes to test the control system over a wider flight envelope. In this type, the sting moves in a pre-programmed pattern of boxes crossed by diagonals, nicknamed a 'cross-box' pattern, as shown in Fig. 10.

In addition to the control method presented herein, a Linear Quadratic Regulator (LQR) method was tested as well as a nonlinear backstepping method [9], [10].

3.6 EXPERIMENTAL RESULTS

3.6.1 INTRODUCTION

The experimental results presented in this section are set up to show that the objectives were indeed met. The experiment Run 1 on 9/23/2015 shows the control system applying newly learned model information to improve control in real-time, while Runs 2 and 4 from 10/21/2015 show the performance of the controller with a given identified model, with and without a precomputed table of control gains. The results of Run 23 of 10/21/2015 show the performance of the system over a wide range of the flight envelope. While the baseline control used in some parts of the runs before or during initial system identification uses only the ailerons for control, the identified models include information about the other control surfaces because of the decoupling inputs, and the control system uses all of the surfaces according to the control allocation method described in Section 2.4.3. The controller in each of the experimental runs presented is as presented in Chapter 2, with the reference model chosen to target a settling time of 0.75 seconds, and damping ratio of 0.8. In Run 1 on 9/23/2015, Run 2 on 10/21/2015, and Run 23 on 10/21/2015 the gains are calculated at each time step from whatever model information is available, while in Run 4 on 10/21/2015 the gains are precomputed in a lookup table from which the gains are interpolated at each time step.

3.6.2 RUN 1 ON 9/23/2015

The results of Run 1 of 9/23/2015 are shown in Fig. 11. During this run, the sting is held at zero yaw ($\psi = 0°$), and the pitch is stepped up from $\theta = 5°$ to $\theta = 10°$ to $\theta = 15°$. In the first doublet command, the baseline controller is running, then the system identification process is started by the experiment operator and the response gets more noisy because of the perturbations being added to the surfaces to decouple the effects of each surface. Once the perturbations are switched off and a model is received, the updated model is used to update the control gains, and the response smooths out and shows an improvement from the baseline (around 40-70 seconds). At a fixed sting position, there is a different linearized controller for each roll angle setpoint, and the gains are updated whenever the roll angle command changes. Around 70-80 seconds, the experiment operator initiated raising the sting pitch to
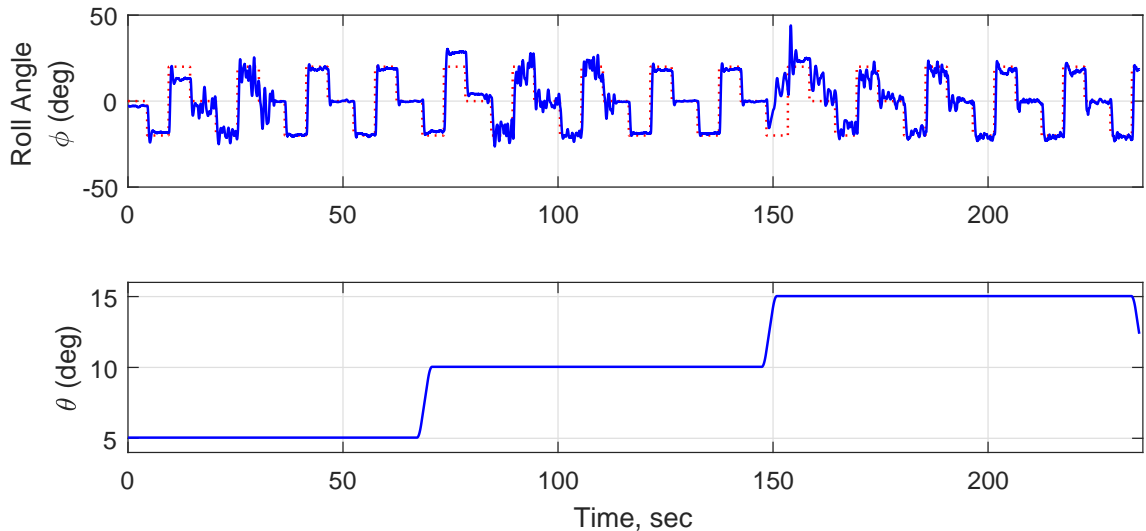
Fig. 11: Run with learning followed by new controller at each sting pitch setting. Run 1 on 9/23/2015.

$\theta = 10°$, and the performance noticeably deteriorated because the model being used for control did not include information at this part of the envelope. Then the operator switched on the system identification mode from about 80-110 seconds, and the the resulting performance, seen during 110-150 seconds, shows notable improvement compared to the steady state error at 70-80 seconds. The operator repeated this procedure for $\theta = 15°$ starting around 150 seconds, and the same pattern was observed whereby upon moving to a new sting attitude the performance is worse, then after learning it improves.

3.6.3 RUN 2 ON 10/21/2015

In Run 1 on 10/21/2015, a model was estimated for $\theta = 5°, 10°, 15°$ with the controller being updated continuously and using all of the control surfaces. The resulting model is used in Runs 2 and 4 of 10/21/2015. In run 2, the control method developed in this work was implemented with reference model settings of $T_s = 0.75$, and $\zeta = 0.8$. At every time step, the control parameters were calculated from the aerodynamic model at the present conditions of θ, ψ, and ϕ_c. Since the run covered three positions of θ with $\psi = 0°$, and with ϕ_c alternating between three values $(-20°, 0°, 20°)$, there were nine distinct linear controllers active at various times during the run, not counting the periods while the sting was transitioning between
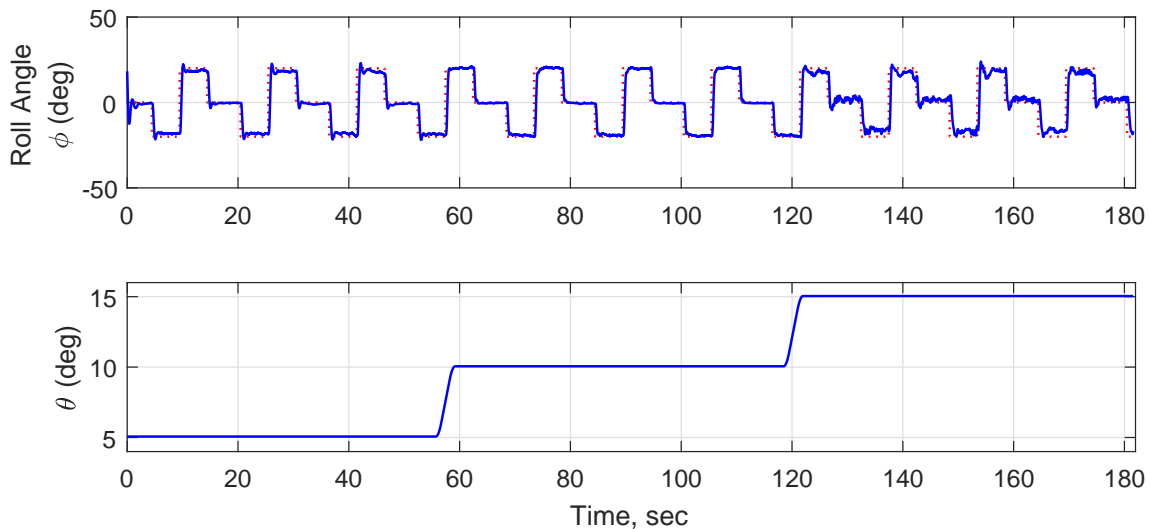
Fig. 12: Run 2 on 10/21/2015, with gains derived at each sample time from model identified in Run 1 on 10/21/2015.

attitudes. The linear controller and the trim offset changed whenever ϕ_c or θ changed, and the control allocation was calculated at the present value of ϕ throughout the experiment. In the results shown in Fig. 12, the response at $\theta = 5°$ shows a greater overshoot than designed, as well as a noticeable steady-state error, but the settling time seems to be good. At the next sting position at $\theta = 10°$, there is a good settling time and less steady-state error. However, when the sting is held at $\theta = 15°$, the performance is noticeably worse with some additional oscillations and steady-state errors.

To investigate closer what is happening at this attitude, Fig. 13 shows the commanded and measured control surface deflections for one of the steps at $\theta = 15°$. In this run sample, the roll angle command steps down from $0°$ to $-20°$. There appears to be about 100 ms of latency between the commanded and measured positions of each of the control surfaces and there seems, especially around 133 s to 134 s, to be an oscillation that could be caused, or exacerbated, by this apparent control surface delay.

To investigate the effect of this delay, a simulation can be set up using a fixed time-delay of the control surfaces of 0.1 s, which is the apparent approximate delay observed in Fig. 13. The aerodynamic model used in this experiment run is used for the simulation. The results of this simulation are shown in Fig. 14. The simulation at $\theta = 5°$ shows an extra overshoot, and at $\theta = 15°$ there is an extra oscillation, both
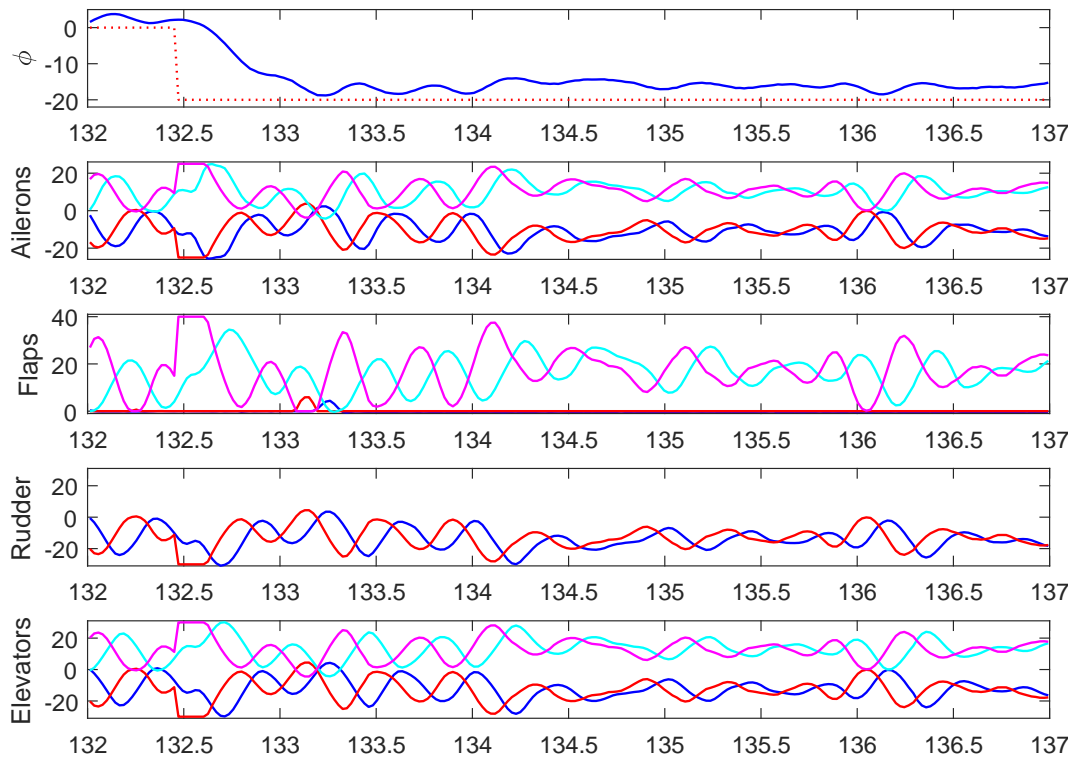
Fig. 13: Control surface deflections for approximately 5 s of Run 2 on 10/21/2015. The red and magenta lines for each pair of surfaces are the left and right commanded deflections respectively, while the blue and cyan are the left and right measured deflections respectively.

of which are similar to the irregularities observed in the experiment, which could indicate that the control surface latency is a contributing factor.

One reason for the increased oscillation at $\theta = 15°$ in the simulation and experiment could be that greater control deflection is required due to reduced aerodynamic roll rate damping, as seen in Fig. 15, which breaks out the torque components based on the C_l aerodynamic model identified in Run 1 on 10/21/2015. The pair of dashed lines in this plot are the maximum and minimum torque due to aerodynamic roll rate damping, calculated by multiplying the $C_{l,p}$ coefficient by the peak roll rate observed during the experiment (which is about $180°/s$). This torque component is clearly smaller at $\theta = 15°$ compared to $\theta = 5°$, which means that there is less aerodynamic resistance to roll rate, so in order to achieve the ideal response, more control input would be required to dampen any disturbances. This increased control input would amplify the effect of any control surface latency because there is a greater error between the commanded surface deflection and actual surface deflection in the
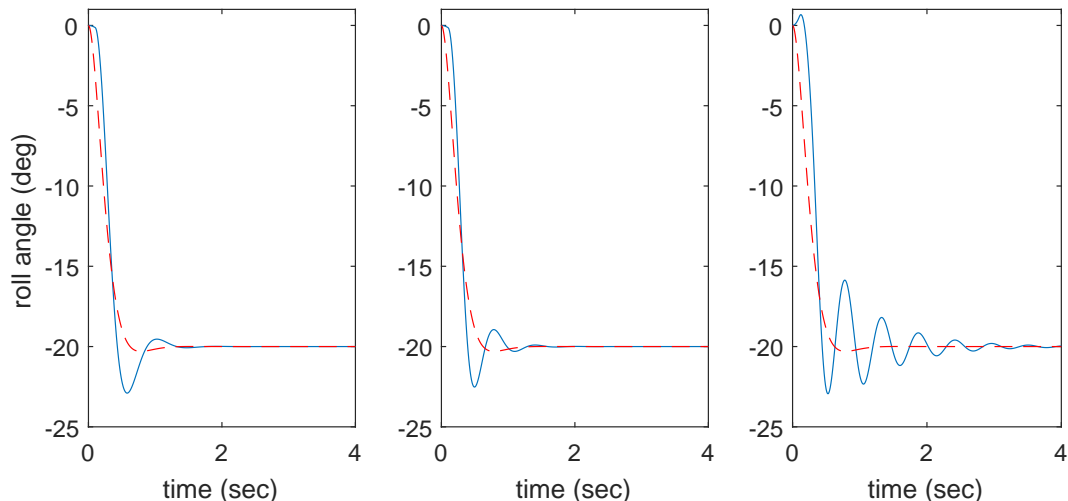
Fig. 14: Simulation results including a 0.1 second control surface input delay. Solid line is the simulation result, while dashed line is the reference model response. (a) Sting pitch is 5 degrees. (b) Sting pitch is 10 degrees. (c) Sting pitch is 15 degrees.

dead time. Also seen in Fig. 15 are the torques due to center of gravity, baseline aerodynamic effect, and control surface deflections. The blue and red lines plot the torque due to center of gravity (τ_{cg}), with and without accounting for y_{cg}, respectively. Compared to the other torque components, the difference between these two torques is small (at most, 2.1% of the total possible torque). The solid curve in each subplot in Fig. 15 is the baseline aerodynamic torque, which is the torque due to the $C_{l,0}$ term, or the aerodynamic torque with zero roll rate and zero control deflections. The dotted lines are the maximum and minimum torque due to control surface deflections, i.e., with all control surfaces saturated in the direction of maximum and minimum torque respectively. The control surfaces appear to have a similar range of torque capability at each of the three positions, but since the aerodynamic roll-rate damping is smaller as θ increases, it would stand to reason that a greater control effort would be required to dampen out any disturbances at higher θ, which could amplify the effect of any delays.

3.6.4 RUN 4 ON 10/21/2015

In order to reduce online computation time in gain-scheduled systems it is sometimes advantageous to use a precomputed lookup table of the control gains. In Run 4 on 10/21/2015, a look-up table approach was used with the same parameters as Run
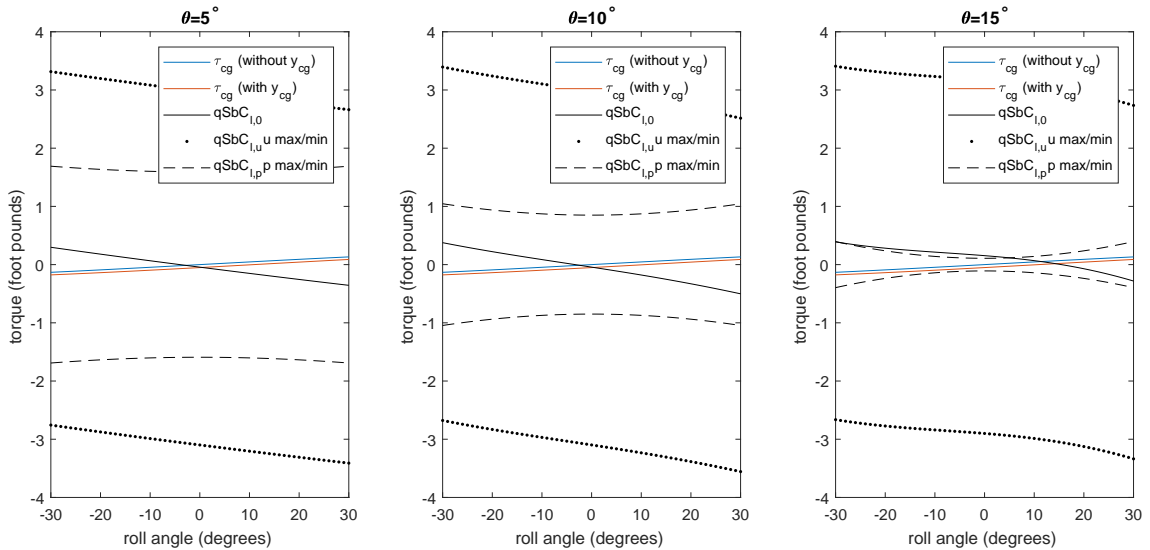
Fig. 15: Based on the aerodynamic model identified in Run 1 on 10/21/2015, the components of torque involved in the experiment are calculated and shown in this plot versus roll angle for each pitch attitude.

2. The sample grid used is $\theta = 5, 10, 15; \phi_c = -20, 0, 20$, which are the set-points of the experiment run in both Runs 2 and 4 on 10/21/2015, so the control gains should be the same in both runs. There is a slight difference in the gains that results from interpolation. The experimental response, seen in Fig. 16, is very similar to the response from Run 2, with only minor variations.
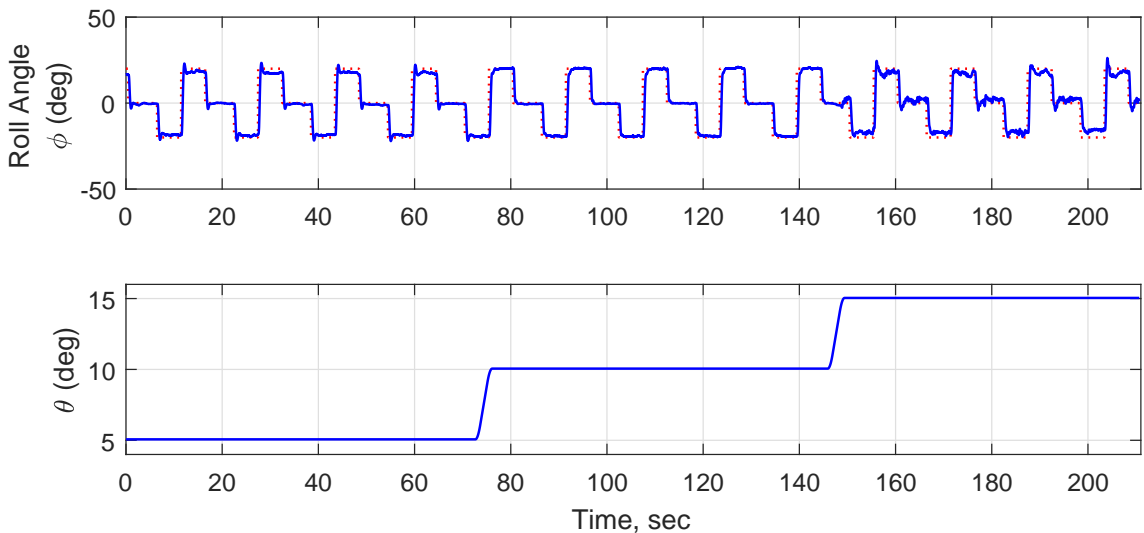

Fig. 16: Run with gains derived from a table based on previously identified model. Run 4 on 10/21/2015.
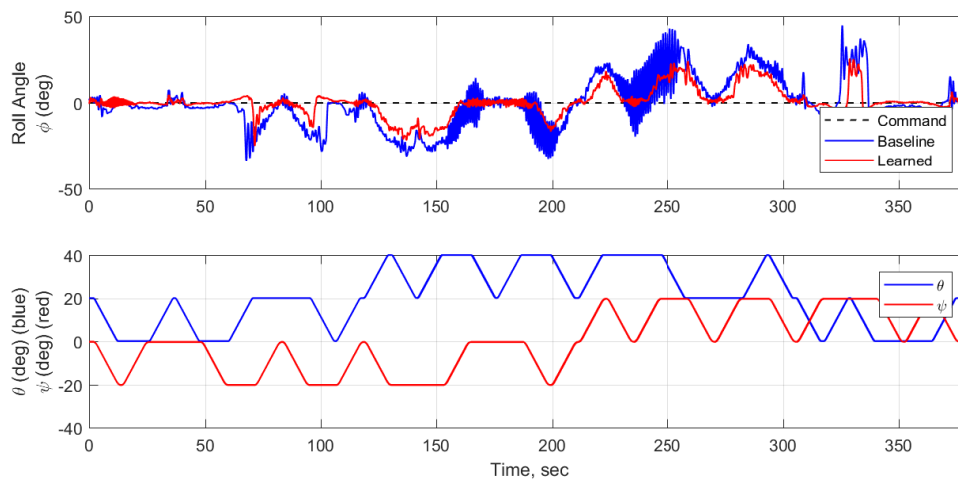
3.6.5 CROSS-BOX RESULTS

Fig. 17: Baseline and learned runs with full cross-box pitch and yaw coverage. Run 7 on 9/23/2015 (Baseline), and Run 23 on 10/21/2015 (Learned).

To observe the system's performance over a wide flight envelope, several runs were carried out with the sting moving in a pre-programmed 'cross-box' pattern, as shown in Fig. 10. The roll angle command was set at zero, and there are some parts of the flight envelope where the plane cannot hold that position because of stall conditions. A baseline run was executed with a PD controller with hand-tuned gains controlling only the ailerons. An identification run was then executed which covered the entire cross-box and generated an aerodynamic rolling moment model. The model was used to calculate the control parameters for another run to observe the effect of the learning. The baseline run and the run after learning are juxtaposed in Fig. 17, where it is apparent that the post-learning controller can hold closer to zero more often then the baseline. A simple comparison of the Root Mean Square (RMS) error shows that the baseline controller had an RMS error of 13.9221, while the learned controller had an RMS error of only 7.9882, which shows an improvement. A more rigorous analysis of these runs requires considering the time-varying nature of the plant and can be studied in a future work.

CHAPTER 4

CONCLUSIONS AND FUTURE RESEARCH

In Chapter 2, a mathematical model for the free-to-roll wind tunnel aircraft model was derived from first principles with the aerodynamic rolling moment function coefficients remaining unknown. This model was then linearized for fixed pitch and yaw points and a linear controller was designed for a virtual control input, which was distributed to the various control surfaces with control allocation. In chapter 3, the experiment was set-up and executed, and the results showed that applying the model learned by the system identification algorithm resulted in improved performance which included satisfactory settling times at some conditions. The system struggled at some conditions, but still showed better control after applying the learned model than before. Thus, the experiment was successful in showing that the control system can be designed in such a way that the parameters are learned on-the-fly and still maintain a sufficient level of control.

In the future, the time-varying case could be analyzed to better account for when the attitude is not fixed at a certain set point. Some additional analysis of the actuator dynamics could also improve the control in more difficult attitudes. Additional nonlinear and adaptive control methods could be explored as well. It will also be useful to expand to more degrees-of-freedom so that the principles can be applied to full-scale aircraft in free flight. The control system designed in this thesis served to validate the Learn-To-Fly learning algorithm in a wind tunnel experiment which is a step toward free-flight testing of these techniques. The sucessful development of this type of algorithm has the potential to change the way aircraft flight control systems are developed by enabling rapid prototyping as well as fault detection and flight envelope protection while simultaneously bringing down the time and expense of the development process.

REFERENCES

[1] E. A. Morelli, "Real-time global nonlinear aerodynamic modeling for learn-to-fly," in *AIAA Atmospheric Flight Mechanics Conference*, no. AIAA 2016-2010, (San Diego), pp. 1–22, January 2016.

[2] T. A. Johansen and T. I. Fossen, "Control allocation - a survey," *Automatica*, vol. 49, pp. 1087–1103, 2013.

[3] N. S. Nise, *Control Systems Engineering*. Wiley, 6 ed., 2011.

[4] R. F. Antoniewicz, E. L. Duke, and K. D. Krambeer, "Derivation and definition of a linear aircraft model," *NASA Reference Publication*, vol. 1207, 1988.

[5] D. B. Owens, J. M. Brandon, M. A. Croom, C. M. Fremaux, E. H. Heim, and D. D. Vicroy, "Overview of dynamic test techniques for flight dynamics research at NASA LaRC," in *AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, (San Francisco), 2006.

[6] E. A. Morelli and V. Klein, *Aircraft system identification theory and practice*. American Institute of Aeronautics and Astronautics, 2006.

[7] O. Harkegard and S. T. Glad, "Resolving actuator redundancy-optimal control vs. control allocation," *Automatica*, vol. 41, pp. 137–144, 2005.

[8] W. C. Durham, "Constrained control allocation," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 4, pp. 717–725, 1993.

[9] A. Mekky and O. R. González, "LQ control for the NASA learn-to-fly free-to-roll project," 2016.

[10] T. Tran, *Nonlinear Flight Control Design Using Backstepping Methodology*. Doctoral dissertation, Old Dominion University, May 2016.

APPENDIX A

SIMULATIONS

In this appendix are the simulation diagrams and code from the Simulink model of the control system, the simulation results of which are discussed in section 2.4.4. The high level system diagram is shown in Fig. 18, with the subsystems for the controller and plant in Fig. 19 and Fig. 20. There are two MATLAB function blocks and the code for those blocks is also listed in this appendix.
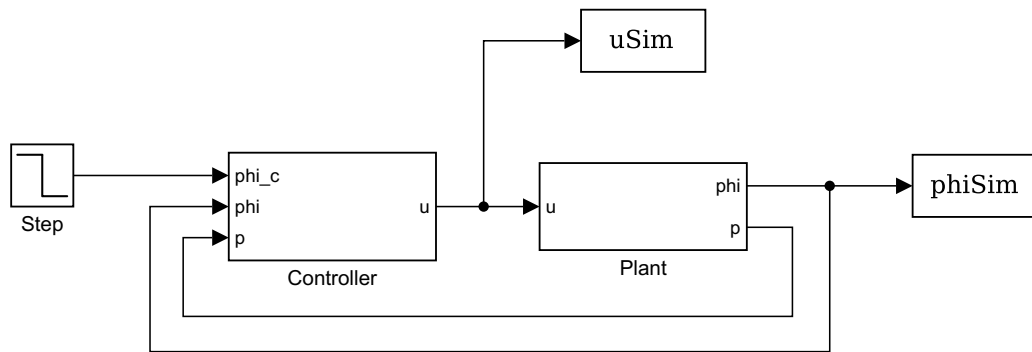
Fig. 18: Top level Simulink diagram

Fig. 19: Controller subsystem

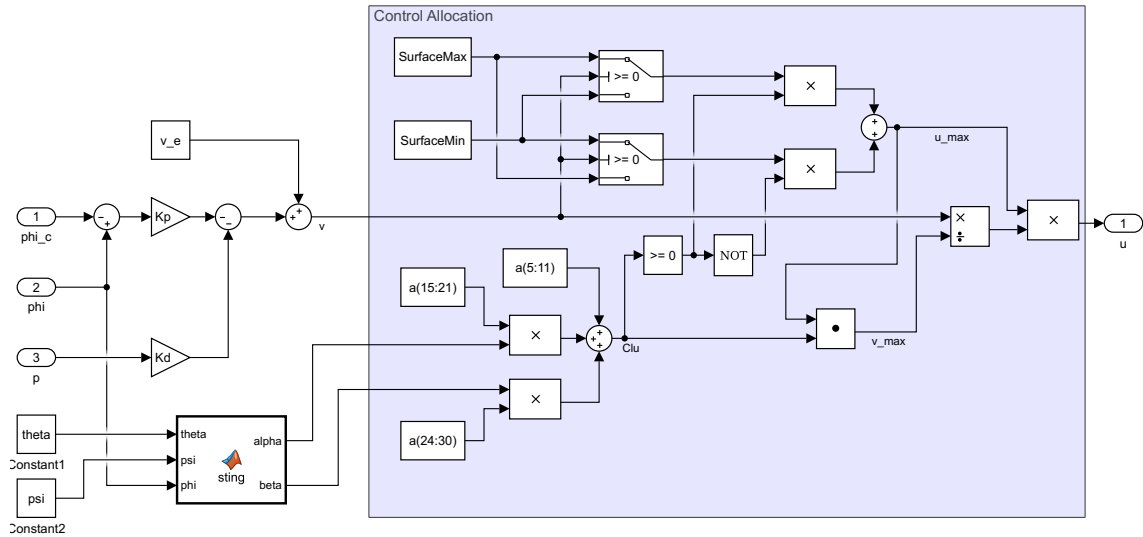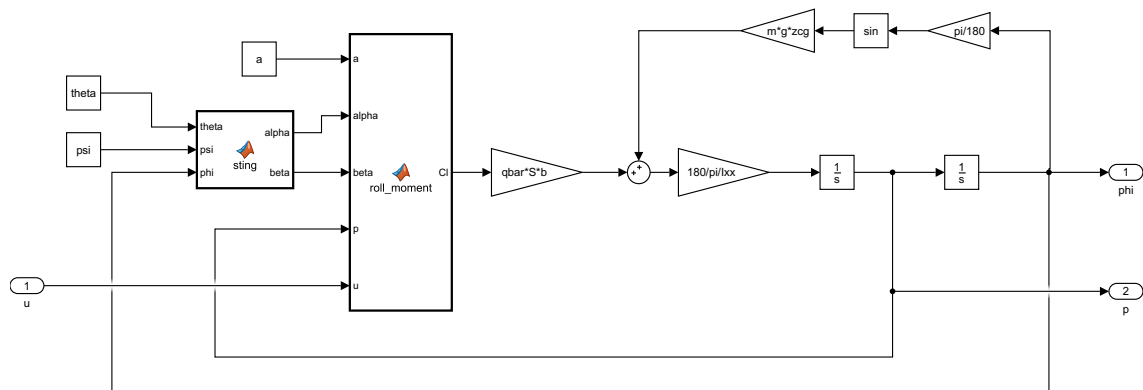
Fig. 20: Plant subsystem

The 'sting' function block in both the plant and controller subsystems contain the following code to model the FTR rig as in (5) and (6).

```
function [alpha,beta] = sting(theta,psi,phi)
%#codegen
alpha = atand((cosd(psi)*cosd(phi)*sind(theta)+sind(psi)*sind(phi))...
    /(cosd(theta)*cosd(psi)));
beta = asind(sind(phi)*sind(theta)*cosd(psi)−sind(psi)*cosd(phi));
```

The 'roll_moment' function block in the plant contains the equation for the aerodynamic rolling moment coefficient (16).

```
function Cl = roll_moment(a,alpha,beta,p,u)
%#codegen
Cl0 = 0;
Clp = 0;
Cldelta = zeros(7,1);
uu = zeros(1,7);
uu(1:7) = u(1:7);


Cl0 = a(1) + a(2)*alpha + a(3)*beta + a(12)*alpha.^2 + ...
    a(13)*beta.*alpha + a(22)*beta.^2;


b = 3.94;
V = 29*sqrt(2);
Clp = (a(4) + a(14)*alpha + a(23)*beta) *pi/180 * (b/(2*V));


Cldelta = a(5:11) + a(15:21).*alpha + a(24:30).*beta;


Cl = Cl0 + Clp * p + uu*Cldelta;
```

APPENDIX B

EXPERIMENT CODE

The code used in the experiment for controller implementation is contained in this appendix. There are two types of controller implemented: an instantaneous (i.e., online calculated gains), and a gain-scheduled (i.e., pre-computed table of gains). Both call the same function for control allocation, and both have an initialization function and a 'control_iteration' function which is called at each time step during the experiment.

Control allocation function:

```matlab
function [Surface] = ...
    uniformControlAllocation2(v,Cldelta,SurfaceMax,SurfaceMin)

% Streit's analytic control allocation

if      v > 0
    u_max = (Cldelta > 0).*SurfaceMax + (Cldelta < 0).*SurfaceMin;
    v_max = Cldelta*u_max';
    v_hat = v/v_max;
    Surface = v_hat * u_max;

elseif v < 0
    u_min = (Cldelta > 0).*SurfaceMin + (Cldelta < 0).*SurfaceMax;
    v_min = Cldelta*u_min';
    v_hat = v/v_min;
    Surface = v_hat * u_min;

else % v = 0
    Surface = zeros(1,7);
end

Surface = max(min(Surface,SurfaceMax),SurfaceMin);

end
```

Control iteration for instantaneous:

```matlab
function [CtrlSurfCmd,CtrlParms,ctrllog]=...
    control_iteration(feedback,model,model_flag,CtrlParms)

%Identified coefficients
a = model.C1.p .* model.C1.xi;


%Equilibrium point
theta = feedback.theta;
psi = feedback.psi;
phi_c = feedback.phi_cmd;
alpha_e = ...
    atand((cosd(psi)*cosd(phi_c)*sind(theta)+sind(psi)*sind(phi_c))...
    /(cosd(theta)*cosd(psi)));
beta_e = ...
    asind(sind(phi_c)*sind(theta)*cosd(psi)-sind(psi)*cosd(phi_c));
Cl0 = a(1) + a(2)*alpha_e + a(3)*beta_e + a(12)*alpha_e^2 + ...
    a(13)*beta_e*alpha_e + a(22)*beta_e^2;


%Roll angle linearization
delta_phi = 1;
alpha_p = atand((cosd(psi)*cosd(phi_c+delta_phi)*sind(theta)...
    +sind(psi)*sind(phi_c+delta_phi))/(cosd(theta)*cosd(psi)));
beta_p = asind(sind(phi_c+delta_phi)*sind(theta)*cosd(psi)...
    -sind(psi)*cosd(phi_c+delta_phi));
Cl0p = a(1) + a(2)*alpha_p + a(3)*beta_p + a(12)*alpha_p^2 + ...
    a(13)*beta_p*alpha_p + a(22)*beta_p^2;
Clphi = (Cl0p-Cl0)/delta_phi;


%Roll rate coefficient
b = 3.94;
V = 29*sqrt(2);
Clp = (a(4) + a(14)*alpha_e + a(23)*beta_e) *pi/180 * (b/(2*V));


% Plant parameters
m    = CtrlParms.m;
g    = CtrlParms.g;
zcg  = CtrlParms.zcg;
qbar = CtrlParms.qbar;
```

```matlab
S    = CtrlParms.Sref;
b    = CtrlParms.Bref;
Ixx  = CtrlParms.Ixx;


% Trim virtual control input
v_e = -m*g*zcg/qbar/S/b*sind(phi_c) - Cl0;


% Linear control
dr=CtrlParms.dr;        %Damping ratio
wn=CtrlParms.wn;        %Natural frequency


a0 = -(m*g*zcg*cosd(phi_c) + 180/pi*qbar*S*b*Clphi)/Ixx;
a1 = -(180/pi/Ixx*qbar*S*b*Clp);
b0 = 180/pi/Ixx*qbar*S*b;
Kp = (wn^2 - a0)/b0;
Kd = (2*dr*wn - a1)/b0;


% Perturbation virtual control input
v_p = -Kp * (feedback.phi - phi_c) - Kd * feedback.p;


% Total virtual control input
v = v_e + v_p;


% Control Allocation
Cldelta = a(5:11) + a(15:21)*feedback.aoa + a(24:30)*feedback.beta;
Cldelta = Cldelta';


Surface = uniformControlAllocation2(v,Cldelta...
    ,CtrlParms.SurfaceMax,CtrlParms.SurfaceMin);
CtrlSurfCmd.Surface = Surface;


ctrllog.Kp=Kp;
ctrllog.Kd=Kd;
ctrllog.v_e=v_e;


end
```

Control initialization for instantaneous:

```matlab
function ctrlparms = control_setup(feedback,model,ctrlparms)
```

```matlab
Tset = 0.75;
ctrlparms.Tset = Tset;
ctrlparms.dr = 0.8;
ctrlparms.wn = 4/Tset/ctrlparms.dr;

% ctrlparms.SurfaceMax = [25,25,40,40,30,30,30];
ctrlparms.SurfaceMax = [ ctrlparms.AileronUpperLimit,...
                         ctrlparms.AileronUpperLimit,...
                         ctrlparms.FlapUpperLimit,...
                         ctrlparms.FlapUpperLimit,...
                         ctrlparms.RudderUpperLimit,...
                         ctrlparms.ElevatorUpperLimit,...
                         ctrlparms.ElevatorUpperLimit];
% ctrlparms.SurfaceMin = [-25,-25,0,0,-30,-30,-30];
ctrlparms.SurfaceMin = [ ctrlparms.AileronLowerLimit,...
                         ctrlparms.AileronLowerLimit,...
                         ctrlparms.FlapLowerLimit,...
                         ctrlparms.FlapLowerLimit,...
                         ctrlparms.RudderLowerLimit,...
                         ctrlparms.ElevatorLowerLimit,...
                         ctrlparms.ElevatorLowerLimit];

end
```

Control iteration for gain-scheduled:

```matlab
function [CtrlSurfCmd,CtrlParms,ctrllog]=...
    control_iteration(feedback,model,model_flag,CtrlParms)
v_e = CtrlParms.v_e(feedback.theta,feedback.psi,feedback.phi_cmd);
Kp = CtrlParms.Kp(feedback.theta,feedback.psi,feedback.phi_cmd);
Kd = CtrlParms.Kd(feedback.theta,feedback.psi,feedback.phi_cmd);

% Perturbation virtual control input
v_p = -Kp * (feedback.phi - feedback.phi_cmd) - Kd * feedback.p;

% Total virtual control input
v = v_e + v_p;

% Control Allocation

a = model.C1.p .* model.C1.xi;
```

```matlab
Cldelta = a(5:11) + a(15:21)*feedback.aoa + a(24:30)*feedback.beta;
Cldelta = Cldelta';
Surface = uniformControlAllocation2(v,Cldelta...
    ,CtrlParms.SurfaceMax,CtrlParms.SurfaceMin);
CtrlSurfCmd.Surface = Surface;


ctrllog.Kp=Kp;
ctrllog.Kd=Kd;
ctrllog.v_e=v_e;



end
```

Control initialization for gain-scheduled:

```matlab
function ctrlparms = control_setup(feedback,model,ctrlparms)


Tset = 0.75;
ctrlparms.Tset = Tset;
ctrlparms.dr = 0.8;
ctrlparms.wn = 4/Tset/ctrlparms.dr;
ctrlparms.thetaVec = -5:5:40;
ctrlparms.psiVec   = -30:10:30;
ctrlparms.phiVec   = -20:20:20;


% ctrlparms.SurfaceMax = [25,25,40,40,30,30,30];
ctrlparms.SurfaceMax = [ ctrlparms.AileronUpperLimit,...
                         ctrlparms.AileronUpperLimit,...
                         ctrlparms.FlapUpperLimit,...
                         ctrlparms.FlapUpperLimit,...
                         ctrlparms.RudderUpperLimit,...
                         ctrlparms.ElevatorUpperLimit,...
                         ctrlparms.ElevatorUpperLimit];
% ctrlparms.SurfaceMin = [-25,-25,0,0,-30,-30,-30];
ctrlparms.SurfaceMin = [ ctrlparms.AileronLowerLimit,...
                         ctrlparms.AileronLowerLimit,...
                         ctrlparms.FlapLowerLimit,...
                         ctrlparms.FlapLowerLimit,...
                         ctrlparms.RudderLowerLimit,...
                         ctrlparms.ElevatorLowerLimit,...
                         ctrlparms.ElevatorLowerLimit];
```

```
end
```

Gain table generation for gain-scheduled:

```matlab
function [CtrlParms] = make_table(model,CtrlParms)

thetaN = length(CtrlParms.thetaVec);
psiN = length(CtrlParms.psiVec);
phiN = length(CtrlParms.phiVec);

v_e = zeros(thetaN,psiN,phiN);
Kp  = zeros(thetaN,psiN,phiN);
Kd  = zeros(thetaN,psiN,phiN);

for iTheta = 1:thetaN
    for iPsi = 1:psiN
        for iPhi = 1:phiN
            theta = CtrlParms.thetaVec(iTheta);
            psi   = CtrlParms.psiVec(iPsi);
            phi_c  = CtrlParms.phiVec(iPhi);


            %Identified coefficients
            a = model.C1.p .* model.C1.xi;

            %Equilibrium point
%             theta = feedback.theta;
%             psi = feedback.psi;
%             phi_c = feedback.phi_cmd;
            alpha_e = atand((cosd(psi)*cosd(phi_c)*sind(theta)...
                +sind(psi)*sind(phi_c))/(cosd(theta)*cosd(psi)));
            beta_e = asind(sind(phi_c)*sind(theta)*cosd(psi)...
                -sind(psi)*cosd(phi_c));
            Cl0 = a(1) + a(2)*alpha_e + a(3)*beta_e + ...
                a(12)*alpha_e^2 + a(13)*beta_e*alpha_e + a(22)*beta_e^2;

            %Roll anlge linearization
            delta_phi = 1;
            alpha_p = atand((cosd(psi)*cosd(phi_c+delta_phi)...
                *sind(theta)+sind(psi)*sind(phi_c+delta_phi))...
```

```matlab
                /(cosd(theta)*cosd(psi)));
            beta_p = ...
                asind(sind(phi_c+delta_phi)*sind(theta)*cosd(psi)...
                 -sind(psi)*cosd(phi_c+delta_phi));
            Cl0p = a(1) + a(2)*alpha_p + a(3)*beta_p + ...
                a(12)*alpha_p^2 + a(13)*beta_p*alpha_p + a(22)*beta_p^2;
            Clphi = (Cl0p-Cl0)/delta_phi;


            %Roll rate coefficient
            b = 3.94;
            V = 29*sqrt(2);
            Clp = (a(4) + a(14)*alpha_e + a(23)*beta_e) *pi/180 * ...
                (b/(2*V));



            % Plant parameters
            m    = CtrlParms.m;
            g    = CtrlParms.g;
            zcg  = CtrlParms.zcg;
            qbar = CtrlParms.qbar;
            S    = CtrlParms.Sref;
            b    = CtrlParms.Bref;
            Ixx  = CtrlParms.Ixx;

            % Trim virtual control input
            v_e(iTheta,iPsi,iPhi) = -m*g*zcg/qbar/S/b*sind(phi_c) - ...
                Cl0;

            % Linear control
            dr=CtrlParms.dr;        %Damping ratio
            wn=CtrlParms.wn;        %Natural frequency

            a0 = -(m*g*zcg*cosd(phi_c) + 180/pi*qbar*S*b*Clphi)/Ixx;
            a1 = -(180/pi/Ixx*qbar*S*b*Clp);
            b0 = 180/pi/Ixx*qbar*S*b;
            Kp(iTheta,iPsi,iPhi) = (wn^2 - a0)/b0;
            Kd(iTheta,iPsi,iPhi) = (2*dr*wn - a1)/b0;
        end
    end
end
```

```matlab
[theta,psi,phi] = ...
    ndgrid(CtrlParms.thetaVec,CtrlParms.psiVec,CtrlParms.phiVec);
CtrlParms.v_e = ...
    griddedInterpolant(theta,psi,phi,v_e,'linear','nearest');
CtrlParms.Kp = griddedInterpolant(theta,psi,phi,Kp,'linear','nearest');
CtrlParms. Kd = griddedInterpolant(theta,psi,phi, ...
    Kd,'linear','nearest');
end
```

VITA

Alexander Brent Streit

Department of Electrical and Computer Engineering

Old Dominion University

Norfolk, VA 23529

EDUCATION

B.S. in Electrical Engineering, Old Dominion University, 2014, Norfolk, VA