REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
10-12-2003	Final Technical Report	October 2000 - December 2002

4. TITLE AND SUBTITLE

GPS Auto-Navigation Design for Unmanned Air Vehicles

5a. CONTRACT NUMBER
N/A

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
N/A

6. AUTHOR(S)

Nilsson, Caroline C. A; Heinzen, Stearns N.; Hall Jr. Charles E.; Chokani, Ndaona

5d. PROJECT NUMBER
NCC1-01-008

5e. TASK NUMBER
N/A

5f. WORK UNIT NUMBER
N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

North Carolina State University
Office of Contracts and Grants
Campus Box 7214
Raleigh NC 27695-7214

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Flow Physics and Control Branch

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA

11. SPONSOR/MONITOR'S REPORT NUMBER
N/A

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution unlimited.

13. SUPPLEMENTARY NOTES
N/A

14. ABSTRACT

A GPS auto-navigation system is designed for Unmanned Air Vehicles. The objective is to enable the air vehicle to be used as a test-bed for novel flow control concepts. The navigation system uses pre-programmed GPS waypoints. The actual GPS position, heading, and velocity are collected by the flight computer, a PC104 system running in Real-Time Linux, and compared with the desired waypoint. The navigator then determines the necessity of a heading correction and outputs the correction in the form of a commanded bank angle, for a level coordinated turn, to the controller system. This controller system consists of 5 controllers (pitch rate PID, yaw damper, bank angle PID, velocity hold, and altitude hold) designed for a closed loop non-linear aircraft model with linear aerodynamic coefficients. The ability and accuracy of using GPS data, is validated by a GPS flight. The autopilots are also validated in flight. The autopilot unit flight validations show that the designed autopilots function as designed. The aircraft model, generated on Matlab SIMULINK is also enhanced by the flight data to accurately represent the actual aircraft.

15. SUBJECT TERMS
GPS. Auto-navigation. Unmanned Air Vehicles.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Ndaona Chokani
Unclassified	Unclassified	Unclassified	None	125	19b. TELEPHONE NUMBER *(include area code)* 757-864-1103

1.0 Introduction

1.1 UAV Background

From the dawn of aviation, the Unmanned Air Vehicles (UAVs) have proved to hold a significant role in the world of aviation. As early as World War I, the British military used remotely piloted vehicles (RPV) to counter German airships[1]. Throughout the twentieth century, UAVs have been used for reconnaissance, monitoring, and surveillance in many wars including World War II, Vietnam War and as recently as the Gulf War, and in the former Yugoslavia[1]. The interest and motivation in UAVs for military use are very real in today's world as the loss of human life in combat is now a primary issue concern in the leading nations. Thus UAVs can be used for specialized tasks where the risks to pilots are high, where beyond normal human endurance is required, or where human presence is not necessary[2]. For example, in the Gulf War UAVs were used to perform reconnaissance missions to provide real-time intelligence without endangering pilots. Furthermore, the cost of development, production, and maintenance of UAVs are significantly lower than for manned military aircrafts.

In addition to military use, UAVs have many civilian applications. These include environmental monitoring, agricultural support, search and rescue operations, and weather reconnaissance. The Insitu group and the University of Washington developed an autonomous UAV, named Aerosonde, for weather reconnaissance[3]. This UAV has flown autonomously across the Atlantic[3].

The worldwide growing interest in UAVs is also due to technological advances that include the availability of compact, lightweight, low-cost sensors and computers, the

maturation of control systems, and the development of autonomous flight control[2]. Indeed, autonomous flight opens up many potential applications for UAVs and has become a key issue of interest. Autonomous navigation enables the aircraft to be flown out-of-pilot sight, and enhances flight test capability. Under autonomous flight, the UAV can be pre-programmed to perform a certain task repeatedly under the same flight condition and flight pattern[4].

Autonomous UAV flight is an emerging research field in the military, civilian, and university settings. There are very few completely autonomous UAVs where the aircraft takes-off, flies, and lands autonomously[2]. The Global Hawk, a military surveillance UAV that is currently in use, is capable of partial autonomous flight[5]. NASA is currently working on an auto-return function[6] for the X48a.

In the university setting, autonomous UAVs are becoming the choice for test bed vehicles. From a cost standpoint, a UAV research testbed is more affordable in the University setting than the use of a full-scale aircraft. Consequently, the growing interest in UAVs has also encouraged universities to participate in UAV research. The University of Sydney research group is working towards fully autonomous capabilities[2] on a UAV. Stanford University is using a versatile UAV for the advancement of aircraft navigation and control[7]. The Georgia Institute of Technology is studying guidance and flight control systems for UAVs[8].

1.2 Navigation System

Autonomous flight is achieved using either inertial or Global Positioning System (GPS) navigation. An inertial navigation system (INS) requires the use of very accurate

accelerometers and angular rate gyros, and a computing unit. The computing unit calculates the position and attitude of the aircraft by integrating the information from the accelerometers and angular rate gyros. The limiting factor of the INS is the error in the instrument reading. Even if the error is small, the integration of the angular rates and the double integration of the accelerations over long periods of time cause the error to drift very quickly. The INS is advantageous for calculations over a short time period as the position and attitude information is frequently required. Active rocket stabilization is one such application. Thus for applications, where attitude and position are not as frequently required, the GPS navigation is becoming the navigation system of choice.

GPS consists of 24 satellites orbiting at an altitude of 12,000 miles above earth and their ground stations. The satellites are arranged such that there are at least four satellites in view from any given point on earth. These satellites send radio signals to a GPS receiver, which then uses the signals to calculate position information in latitude and longitude, heading, and velocity. The position updates are generated at a frequency of 1Hz. The availability of compact, low-weight, low-cost off the shelf GPS receivers has enhanced the popularity of GPS.

1.3 Objectives of Present Work

The objective of the present work is to develop and test fly a GPS auto-navigation system for UAV. The auto-navigation system is composed of an avionics system, an autopilot unit that consists of five controllers for attitude control, and a navigator that is programmed to fly through a pre-determined waypoint pattern. In the following chapter, the

design approach of the auto-navigation system is described. This chapter starts with a description of the UAV test vehicle and its avionics system and instruments. Next, the aircraft dynamics system, autopilot unit, and navigator design are explained in detail. In chapter three, the extensive flight testing of the GPS auto-navigation system is described. Finally, this work concludes with an overview of suggested work and recommendations for future study.

2.0 Approach

This chapter is divided into four main sections. In the first section, the UAV selected from the NC State University Flight Research fleet and its payload are described. Then, in Section 2.2 the modeling of the non-linear aircraft dynamics is detailed. Next, the development of the autopilot unit that consists of five controllers is described in detail in Section 2.3. Finally, the design of the navigator is explained in Section 2.4.

2.1 Test Vehicle Description

The Stingray UAV, designed and built at NC State University, is used as the technology demonstrator in this project. This vehicle was selected due to its good handling qualities and performance, payload capacity, and its requirement of little ground support. The aircraft is equipped with an avionic system and instruments required for auto-navigation flight.

2.1.1 Stingray UAV design

The Stingray UAV was designed and built by the NC State Aerospace Engineering senior design class of 1997-1998. The design specifications[9] for Stingray UAV are summarized in Table 2.1. During its design the UAV underwent a thorough analysis of the various aspects of an airplane design including aerodynamics, stability and control, structures, and performance using computer aided tools such as Unigraphics®,[10] PMARC,[11] Matlab®,[12] and ANSYS.[13]

The primary characteristics of Stingray are the blended body, low wing configuration and fixed tricycle landing gear. The fixed tricycle landing gear requirement was unique for this design class. Traditionally, the NC State Senior design RPVs are catapult launched, and landed using skids. This is because the landing gear adds weight and drag to the aircraft, and complexity to the aircraft's design, but on the other hand facilitates the flight operation at the airfield. Indeed, when the aircraft is catapult launched, a minimum of three people (excluding the pilot) are needed, whereas only one person is needed to prepare an aircraft with landing gear for take-off. However, since the aircraft with landing gear takes off under its own power, its weight is an important factor during its operation. The fully instrumented Stingray UAV is flown 1 lb above its designed weight of 24 lbs. Additionally, the prevailing winds, and atmospheric temperature are also taken into account before clearing the aircraft for flight.

Table 2.1 – Stingray UAV Design requirements

Tractor configuration
Conventional blended body
Low wing configuration with a rectangular wing platform
Outboard 25% of wing panels must be removable for ease of transportation
The maximum span of the wing is 80 inches
The maximum aspect ratio is 9
Fixed tricycle landing gear (main gear will have brakes and the nose wheel will be steerable)
$750 Budget
The design load factors are +4 and –2.67 with a 1.5 safety factor
The aircraft is designed for minimum weight
The test flights will be conducted at the NCSU flight facility
One servo will be used per control surface and placed as close to the surface as possible
A linear control system will be designed for use with the aircraft
The control surfaces will be analyzed to determine if mass balancing is required
An "iron bird" of the landing gear will be built and tested in the Fall semester
The landing distance must be within 150 feet

The final configuration of the aircraft, shown in Figure 2.1, is a conventional blended body, low wing configuration with a rectangular wing platform. The wing span and chord are 77in. and 12in. respectively, and the airfoil is NACA 2415. The wings have a dihedral of 6° and an incidence of 2°. The fuselage shape is a symmetrical airfoil NACA 0011 with a length of 63in. and maximum height of 7.5in. The wing-fuselage blend is formed by systematically increasing the thickness of the wing airfoil until the fuselage airfoil size is reached. The horizontal and vertical stabilizers are of a conventional configuration and have a NACA 0012 airfoil shape. The aileron and plain flaps are located on the outboard panels of the wing.[14]

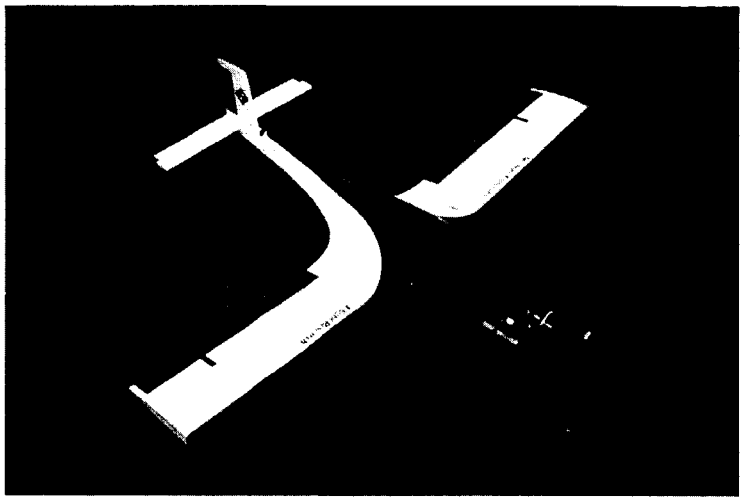


Figure 2.1 – Stingray UAV

The control surfaces and wing are constructed from a blue foam core that is sheeted with balsa and glassed with a thin layer of fiberglass. The wing spar consists of a balsa core wrapped with carbon fiber. The fuselage was formed by laying-up fiberglass skins in a mold that was CNC milled from files generated in Unigraphics®.

There are two available payload compartments in the aircraft to store the required systems. The main compartment is located over the wing measures 11in. long, 8in. wide, and has a maximum height of 7.5in. The forward compartment is located between the engine's firewall and the main compartment and measures 11in. long, 6 in. high and its width ranges between 5 and 8 in. The main cargo compartment with the installed avionics system is shown in Figure 2.2.

Figure 2.2 – Stingray UAV Main Compartment

The aircraft is powered by an O.S. Max 1.08 cubic inch displacement, 2.25 hp, two-cycle reciprocating engine. A 14x8 APC propeller was selected for the engine and aircraft using an iterative process. During the flight-testing of the autopilot unit, it was found that the engine experienced reliability problems due to a high amount of usage. The engine was replaced with a new O.S. Max 1.08 engine.

PMARC, a low order aerodynamic panel method, is used to determine linear stability derivatives that are used along with moment of inertia to conduct dynamic stability analysis. These stability derivatives, moment of inertia, and weight of stingray UAV are shown in the following tables.

Table 2.2 – Stingray UAV stability derivatives

Longitudinal Derivative	Per radian	Lateral Derivatives	Per radian
C_{Lo}	0.2773	$C_{y\beta}$	-0.311
C_{Do}	0.0474	$C_{l\beta}$	-0.085
$C_{L\alpha}$	5.1310	$C_{n\beta}$	0.072
$C_{m\alpha}$	-0.479	C_{yp}	-0.0545
C_{Lq}	8.965	C_{lp}	-0.450
C_{mq}	-12.622	C_{np}	-0.034
$C_{x\alpha dot}$	0.0	C_{yr}	0.242
$C_{m\alpha dot}$	-1.2523	C_{lr}	0.093
$C_{z\alpha dot}$	-0.4203	C_{nr}	-0.089
$C_{L\delta e}$	0.555	$C_{y\delta r}$	0.158
$C_{m\delta e}$	-1.6535	$C_{l\delta r}$	0.006
$C_{m\delta edot}$	0.0	$C_{n\delta r}$	-0.068
$C_{x\delta edot}$	0.0	$C_{y\delta a}$	0.0
$C_{z\delta edot}$	0.0	$C_{l\delta a}$	-0.345
		$C_{n\delta a}$	0.0

Table 2.3 – Stingray UAV moments of inertia and weight

I_{xx}	0.533 slug ft^2
I_{yy}	0.699 slug ft^2
I_{zz}	0.980 slug ft^2
I_{xz}	0.021 slug ft^2
Weight	24.45 lbs

In the design process, a dynamic stability analysis is performed to determine the handling qualities of the aircraft following the MIL-specifications[15] listed by the Department of Defense and the FAA. An aircraft's flying qualities are described in terms of three levels. An aircraft is said to have Level 1 flying qualities, if the pilot does not have an increase in workload to complete the mission flight phase. Level 2 flying qualities occurs when the pilot workload is increased. Level 3 flying qualities indicates that the pilot's workload is excessive

and the mission effectiveness may be reduced. The specifications of flying qualities depend on the class of aircraft. There is not a specific classification for UAV's, however, since the UAV's are highly maneuverable airplanes, these aircrafts are considered Class IV aircraft performing Category A flight phase. Therefore the MIL-specifications used for the Stingray UAV are for unforced motion on the flying modes. These flying modes are two longitudinal modes (short-period mode and Phugoid mode), and three lateral modes (Dutch Roll mode, spiral mode, and roll convergence). The non-dimensional eigenvalues, period, time to half, and damping ratio are determined for each mode for different flying phases including cruise, landing approach, and take off/landing, and compared to the tabulated MIL-specifications tables for flying qualities. Table 2.4 and Table 2.5 summarize the handling qualities for each mode at cruise and landing conditions. Following the MIL-specifications at the different flight conditions, it was determined that Stingray UAV has good handling qualities (Level 1) at cruise and landing with the exception of a Dutch Roll mode (Level 2).

Table 2.4 – Stingray UAV handling quality at cruise

	Short-Period Mode	Phugoid Mode	Dutch Roll Mode	Spiral Mode	Roll Convergence
Eigenvalue	-0.032 ± 0.0334i	-0.0002 ± 0.002i	-0.021 ± 0.197i	-0.0002	-0.415
Period (sec)	0.923	18.14	0.99		
Time to half (sec)	0.11	17.08	1.03	87.8	0.05
Damping ratio	0.69	0.12	0.11		
LEVEL	1	1	2	1	1

Table 2.5 Stingray UAV handling quality at landing

	Short-Period Mode	Phugoid Mode	Dutch Roll Mode	Spiral Mode	Roll Convergence
Eigenvalue	-0.032 ± 0.033i	-0.0002 ± 0.004i	-0.019 ± 0.21i	-0.0005	-0.434
Period (sec)	1.44	11.64	1.50		
Time to half (sec)	0.17	22.27	1.79	63.84	0.08
Damping ratio	0.69	0.058	0.092		
LEVEL	1	1	2	1	1

2.1.2 Avionics

As a research aircraft, Stingray UAV is equipped with the avionic systems and instruments for in-flight measurements and data acquisition. The main compartment houses the on-board flight computer (the LIFT system), avionics battery, and pressure transducers used for measurements of static and dynamic pressure. This compartment has access to the tail boom where the radio/servo interface, compressed air tank with servo and valving for the brakes, receiver and servo isolation card are placed. In the forward compartment there are two Humphrey angular rate transducers, fuel tank, servo battery, front gear servo, throttle servo, and a glow driver. A Pitot tube is carried on the outboard portion of the wings.

The avionics system revolves around the flight computer, which receives and stores information from the receiver, GPS, angular rate transducers, and pressure transducers, then analyzes the data and sends the appropriate command signal to the servos. A schematic of the avionics system is shown in Figure 2.3.[16]

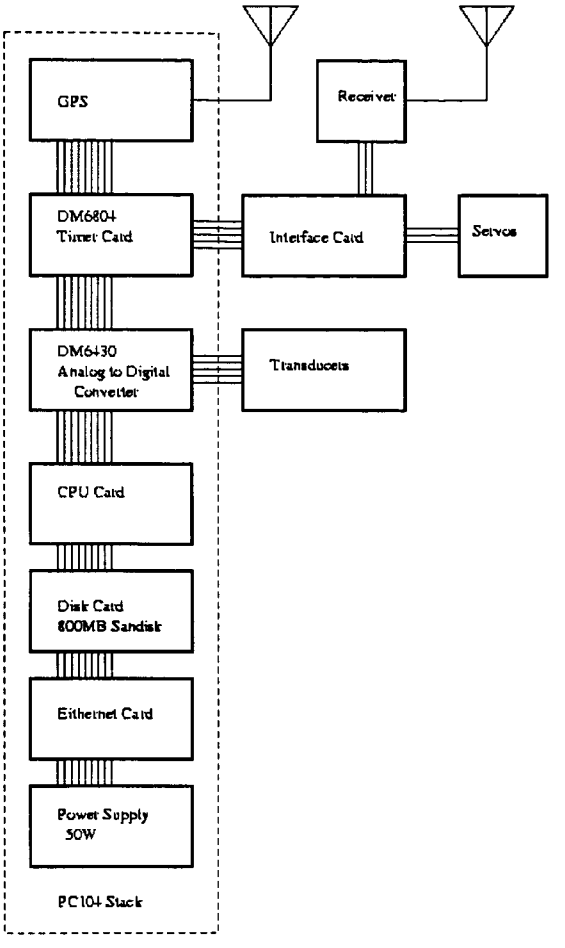


Figure 2.3 – Stingray UAV avionics schematic

2.1.2.1 LIFT system

The Linux In Flight Testing (LIFT) system that was developed at NC State is the on-board computer system. This system is capable of hard real-time control system implementation, data processing and in-flight data storage. The hard real-time capacity is achieved by the Real-Time Linux operating system and is used for data acquisition, control, and navigation.[16] The hard real time enables the implementation of the time critical tasks

such as attitude control. The system is built from seven commercial-off-the shelf (COTS) PC-104Plus cards. A CPU, Hard drive and power supply card form the basis of the LIFT system. A PC-104Plus GPS card provides the real-time position heading, and velocity information of the aircraft. The GPS card is connected to the GPS antenna, which is placed in the front compartment. The Ethernet connection on the LIFT system is used prior to and after flight to upload and download data. Before a flight, a laptop computer is connected to the LIFT system and is used to start the system, perform the preflight testing of the system, and download any necessary files such as Automatic Maneuvering Sequence (AMS) files. After the flight, the data acquired during flight is downloaded onto the laptop and the LIFT system is either shutdown or prepared for the next flight.

An analog to digital converter card on the LIFT system connects the transducer array. The LIFT system is connected to the radio/servo interface card by the timer card.

2.1.2.2 Radio/servo interface card

An in-house built radio/servo interface card is placed between the radio receiver, the LIFT system, and the servo isolation (ISO) card as shown in Figure 2.3. The radio/servo interface card has three functions. First, it multiplexes the radio control signals to two timers for the LIFT system. Second, it demultiplexes the servo signals from the two timers of the LIFT system. Third, it switches between actuator commands from the control receiver and the output of the LIFT system, by command from the pilot. Finally, it optically isolates the servos from the system.[16] The optical isolation is necessary to prevent the entry of inductive spikes from the servo motors into the electrical system.[16]

2.1.3 Instrumentation

The Stingray UAV is instrumented with angular rate transducers, a Pitot static probe, and pressure transducers.

2.1.3.1 Angular rate transducers

Two Humphrey Inc. angular rate transducers are installed in the UAV. A two-axis transducer is used to measure the pitch and roll rates, and a single-axis transducer to measure yaw rate. The two-axis transducer is mounted on the bulkhead that divides the main and forward compartments. The yaw rate transducer was initially placed on a plywood shelf in the forward compartment. However, this shelf is only attached to the right and aft side of the compartment; the results obtained in flight tests showed that the transducer was subject to a significant amount of vibration. The yaw rate transducer was therefore moved and attached, in an inverted position, on the top of the forward compartment. These transducers provide pitch, roll, and yaw rate information for feedback the closed-loop controllers described in section 2.2. The angular rate limits of the transducers are summarized in Table 2.6.

Table 2.6 - Measuring limits of the Humphrey angular rate transducers

Pitch rate	±120°/s
Roll rate	±180°/s
Yaw rate	±100°/s

2.1.3.2 Pitot static probe and Pressure transducers

The Pitot static probe that is used to obtain airspeed and altitude is mounted on the left wing. The total pressure and the static pressure ports are connected to a 10inH_2O pressure

transducer that is located in the outboard section of the wing. This transducer is configured to measure the pressure difference between the total and static pressures and generate a voltage in the range 1V (for 0in H_2O) and 6V (for 10in H_2O). The aircraft's velocity is then calculated using Bernoulli's equation:

$$V = \sqrt{\frac{2(P_o - P_s)}{\rho}} \tag{1}$$

For computational efficiency, a 4th order Taylor expansion is written for the square root function when equation (1) is implemented in kernel space. The reference point a is 10000. This expansion is:

$$\sqrt{x} = a^{\frac{1}{2}} + \frac{(x-a)}{2a^{\frac{1}{2}}} - \frac{(x-a)^2}{8a^{\frac{3}{2}}} + \frac{(x-a)^3}{16a^{\frac{5}{2}}} - \frac{5(x-a)^4}{128a^{\frac{7}{2}}} \tag{2}$$

The static port is connected to a second 10in H_2O pressure transducer to obtain an altitude reading. The reference port of this transducer is connected to a closed tube. Before flight, the closed tube is opened and then closed; thus the reference static pressure is set to the ambient pressure at the runway's elevation. In this manner, the altitude determined during flight is referenced to the runway. The following equation is used to convert the measured pressure difference to an altitude reading:

$$h = \ln\left(\frac{P_s}{P}\right)\left(\frac{RT}{-g_o}\right) \tag{3}$$

To decrease the computation time, equation (3) is simplified by using a slope factor for converting the differential altitude pressure to an altitude reading. This slope is computed by taking the derivative of equation (3) and using standard day conditions. Equation (4) with

the calculated slope is the equation implemented in the LIFT system to determine altitude reading.

$$\frac{dh}{d\Delta P} = 67.88\, ft/inH_2O \tag{4}$$

2.2 Aircraft Dynamics modeling

Traditionally, the aircraft dynamics are simulated using a linear state-space representation. In this state-space method, the eight equations of state are linearized using Bryan's approximation, small disturbance method, and the small angle approximation to yield two sets of four differential equations; one set for longitudinal dynamics and the other for lateral dynamics. In the present work the desired autonomous flight maneuvers, include large angle coordinated turns that do not satisfy the requirements of linearization. Therefore an accurate non-linear flight dynamics representation is required. In the non-linear representation, there are twelve ordinary nonlinear differential equations of state that are solved numerically to obtain the state of the aircraft. It is important to note that even though the equations of state are non-linear, the stability derivatives used in the aircraft representation are linear.

2.2.1 Flight Dynamic and Controls Toolbox description

The MATLAB Flight Dynamic and Controls (FDC) Toolbox[17] describes the aircraft dynamics with twelve non-linear ordinary differential equations. These equations apply for all rigid bodies, assuming a flat non-rotating Earth.[17] This toolbox was developed by Dr. Rauw from Delft University in Netherlands and is incorporated into a SIMULINK[18] block

diagram that can be reconfigured for different aircraft systems. The aircraft system that includes the aircraft aerodynamic coefficients, geometry, and mass parameters are loaded in an input file before the first simulation run. The input file with Stingray UAV's aircraft parameters is shown in Appendix A. The author added a subroutine to the FDC Toolbox to solve for trim conditions for a given velocity. The trim conditions consist of the trim angle of attack, elevator angle, and throttle. The external atmospheric disturbances (deterministic and stochastic) are also simulated in the aircraft dynamic model. A portion of the aircraft dynamic system is shown in Figure 2.4.

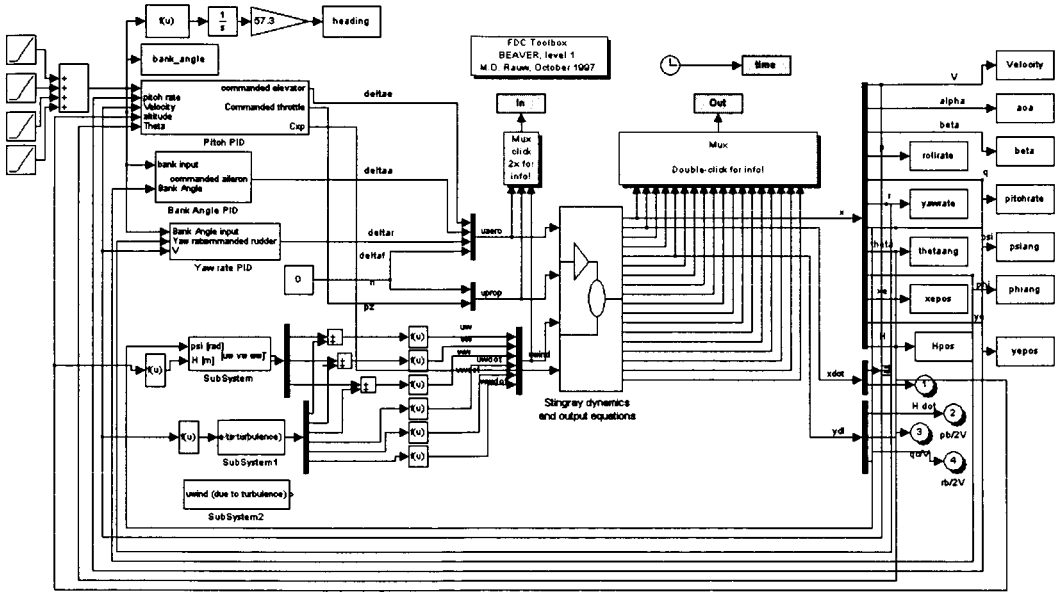


Figure 2.4 – Aircraft Dynamics Model

2.2.1.1 Non-linear equations of motion

The aircraft equations of motion are based on Newtonian mechanics on a rigid body:

$$\vec{F} = m\left(\left(\frac{d\vec{V}}{dt}\right)_{body} + \vec{\Omega}\times\vec{V}\right) \tag{5}$$

$$\vec{M} = \frac{d\left(\underline{I}\vec{\Omega}\right)_{body}}{dt} + \vec{\Omega}\times\left(\underline{I}\vec{\Omega}\right) \tag{6}$$

where $\vec{V}$ is the linear velocity vector, $\vec{\Omega}$ is the angular velocity vector, and $\underline{I}$ is the moment of inertia tensor.

Equations (5) and (6) for the force and moment on a rigid body are rearranged in a non-linear state-space configuration such that the linear and angular velocities are the state variables and the external forces and moments are the input of the state space. These external forces and moments are a combination of aerodynamic forces and moments, engine forces and moments, gravitational forces, and the forces and moments due to non-steady atmosphere.

In addition to the linear and angular velocities, the Euler's angles, and position coordinates relative to Earth are added to complete the state vector. The spatial orientation variables are needed to account for the gravitational force contribution.

The linear velocities u, v, w can be described by a combination of the true airspeed, the angle of attack and side-slip of the aircraft. Thus, for convenience the true airspeed, angle of attack and side-slip are used instead of the linear velocities.

The complexity of the equations emerges when the equations for external forces and moments also depend on the state variables and thus making the state-space equations implicit. To solve this problem, the force and moments equations are first computed without

the implicit variables (angle of attack and side-slip). Then, the state-space equation is solved to provide an initial value for the implicit variables. The true values are then computed using a correction factor. The non-linear equations of state are defined as:

$$X = mg\sin\theta + m(\dot{u}_e + qw_e - rv_e) \tag{7}$$

$$Y = -mg\cos\theta\sin\phi + m(\dot{v}_e + ru_e - pw_e) \tag{8}$$

$$Z = -mg\cos\theta\cos\phi + m(\dot{w}_e + pv_e - qu_e) \tag{9}$$

$$L = I_x\dot{p} - I_{zx}\dot{r} + qr(I_z - I_y) - I_{zx}pq + qh_z' - rh_y' \tag{10}$$

$$M = I_y\dot{q} + rp(I_x - I_z) + I_{zx}(p^2 - r^2) + rh_x' - ph_z' \tag{11}$$

$$N = I_z\dot{r} - I_{zx}\dot{p} + pq(I_y - I_x) + I_{zx}qr + ph_y' - qh_x' \tag{12}$$

$$\dot{\psi} = \frac{q\sin\varphi + r\cos\varphi}{\cos\theta} \tag{13}$$

$$\dot{\theta} = q\cos\varphi - r\sin\varphi \tag{14}$$

$$\dot{\varphi} = p + \dot{\psi}\sin\theta \tag{15}$$

$$\dot{x}_e = \{u_e\cos\theta + (v_e\sin\varphi + w_e\cos\varphi)\sin\theta\}\cos\psi - (v_e\cos\varphi - w_e\sin\varphi)\sin\psi \tag{16}$$

$$\dot{y}_e = \{u_e\cos\theta + (v_e\sin\varphi + w_e\cos\varphi)\sin\theta\}\sin\psi + (v_e\cos\varphi - w_e\sin\varphi)\cos\psi \tag{17}$$

$$\dot{z}_e = -u_e\sin\theta + (v_e\sin\varphi + w_e\cos\varphi)\cos\theta \tag{18}$$

The moment of inertia coefficients in Equation (10-12) are defined in Appendix B2 in reference [17].

2.2.1.2 External atmospheric disturbances

Wind shear and atmospheric turbulences are modeled in the FDC Toolbox to further simulate actual flight condition. Wind shear is encountered in flight operations such as final approach/landing and take-off/initial climb. There are different methods of modeling wind shear. The FDC Toolbox uses a deterministic disturbance method in which the wind shear profile is a function of altitude.

$$V_w = V_{w_{9.15}} \frac{H^{0.2545} - 0.4097}{1.3470} \quad (0 < \text{h} < 300\text{m}) \tag{19}$$

$$V_w = 2.86585 V_{w_{9.15}} \ (0 \geq 300m) \tag{20}$$

The components of the wind shear along the body axes are then defined as:

$$u_w = V_w \cos(\psi_w - \pi)\cos\psi + V_w \sin(\psi_w - \pi)\sin\psi \tag{21}$$

$$v_w = -V_w \cos(\psi_w - \pi)\sin\psi + V_w \sin(\psi_w - \pi)\cos\psi \tag{22}$$

The atmospheric turbulence is modeled using stochastic modeling theory. In the FDC Toolbox, turbulence is modeled as white noise that is passed through a linear Dryden filter as shown in Figure 2.5.

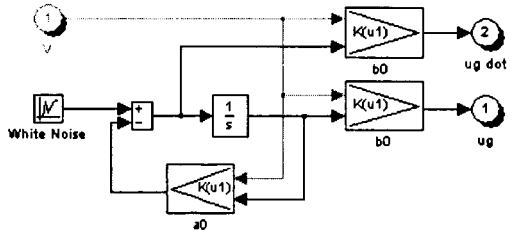


Figure 2.5 – Block diagram of the linear Dryden filter

2.2.2 Modifications

Three additional modifications were made to the FDC. First, the FDC toolbox was designed in the Netherlands, where the use of SI units is standard. Thus the SI units were converted into English units to be consistent with calculations and analysis already performed on Stingray UAV.

A second difference between the European and American standard is the non-dimensionalization of the pitch rate. In Europe, the pitch rate is non-dimensionalized with the factor: $\frac{\bar{c}}{V}$; whereas in the American standard, the pitch rate is non-dimensionalized with the factor: $\frac{\bar{c}}{2V}$. The Stingray UAV stability derivatives that depend on pitch rate such as C_{m_q} and C_{z_q}, follow the American standard for non-dimensionalized pitch rate. It was therefore decided to follow the American standard and change the definition of non-dimensionalized pitch rate in the FDC Toolbox. As a note, the non-dimensionalization of roll rate and yaw rate follows the same definition in Europe and America.

Finally, the power model in the FDC toolbox also needed to be redesigned to fit the power model of Stingray UAV. The representation of the engine power model was achieved by simplifying the definition of the force due to the engine defined for Stingray UAV as:

$$F = (C_{xp} dpt) q_{dyn} S \tag{23}$$

2.3 Autopilot Unit

The autopilot unit consists of five closed-loop controllers that are designed to command the appropriate deflection to the control surfaces and the throttle based on

commanded inputs. These five controllers are pitch rate PID, yaw damper, bank angle PID with turn coupler, altitude-hold PID, and velocity-hold PID. The autopilot unit was designed using MATLAB and SIMULINK software packages and utilized the FDC toolbox described in the previous section.

Initially, only pitch rate, yaw damper and bank angle autopilots were designed to perform the necessary maneuvers such as coordinated level turn. These controllers were closed-loop PID type controllers. When these autopilots were implemented on the non-linear aerodynamic model and their gains were determined, it was found that the altitude and velocity could not be held constant with only these three controllers due to increase of lift and drag in the turn. To resolve this problem, velocity and altitude hold closed-loop PID type autopilots were designed to perform a constant altitude and velocity coordinated turn.

The following approach is used in the design of the controllers. First, a linear continuous controller is designed. The gains of this controller are then adjusted to give the adequate performance. Next, the controller is discretized with a sampling time of 0.02 seconds, a sampling frequency of 50Hz. This corresponds to the sampling frequency of the LIFT system. The discretized controller is then implemented in the non-linear aircraft dynamics system; a final small adjustment of the gains is then made to account for the non-linearity.

2.3.1 Pitch rate autopilot

The pitch rate controller maps the pilot elevator input onto the aircraft pitch rate. The pitch rate controller is also used to vary the angle of attack and altitude of the aircraft. In the present task, the pitch rate controller is of particular interest because it functions well for the

desired coordinated turn maneuver. In such a maneuver, a pitch rate is commanded together with the bank angle to keep the aircraft in level flight during turn. It is important to note that the pitch rate controller does not change the elevator trim angle. In this application, the lift in the turn is not increased enough via elevator angle to hold altitude until the altitude hold is installed.

The pitch rate controller is a PID type closed-loop feedback controller and was designed using the MATLAB and SIMULINK packages. The controller is first designed in a linear aircraft dynamics system that is represented by a 4^{th} order linear longitude state space model:

$$\dot{\boldsymbol{x}}_{long} = A_{long}\boldsymbol{x}_{long} + B_{long}\delta_e \tag{24}$$

$$\boldsymbol{y}_{long} = C_{long}\boldsymbol{x}_{long} + D_{long}\delta_e \tag{25}$$

where,

$$\boldsymbol{x}_{long} = \begin{bmatrix} u \\ \alpha \\ \theta \\ q \end{bmatrix} \qquad \boldsymbol{y}_{long} = [q] \tag{26}$$

The matrix elements for this state space model are listed in Appendix B.

The elevator servo is modeled in the simulation by the following second order system.

$$\dot{\mathrm{x}}_{servo} = A_{servo}\mathrm{x}_{servo} + B_{servo}q_{command} \tag{27}$$

$$\mathrm{y}_{servo} = C_{servo}\mathrm{x}_{servo} + D_{servo}q_{command} \tag{28}$$

$$\boldsymbol{x}_{servo} = \begin{bmatrix} \dot{\delta}_e \\ \delta_e \end{bmatrix} \qquad \boldsymbol{y}_{servo} = [\delta_e] \tag{29}$$

The matrix elements for this servo model are also listed in Appendix B.

The block diagram of the pitch rate controller is shown in Figure 2.6, where the input is a commanded pitch rate.

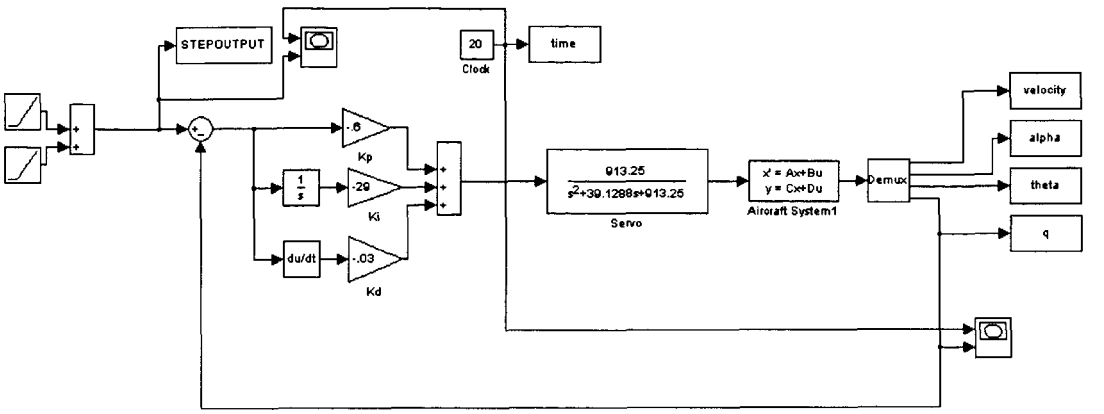


Figure 2.6 – Block diagram of pitch rate controller in linear continuous system

The PID gains are determined by examining the pitch rate response to a pitch rate input. The pitch rate response should compare well with the input and have adequate damping. The gains are adjusted until the aircraft pitch rate response compares well to the commanded pitch rate input. The effectiveness of the controller is shown in Figure 2.7 where the pitch rate response with and without the controller are compared.

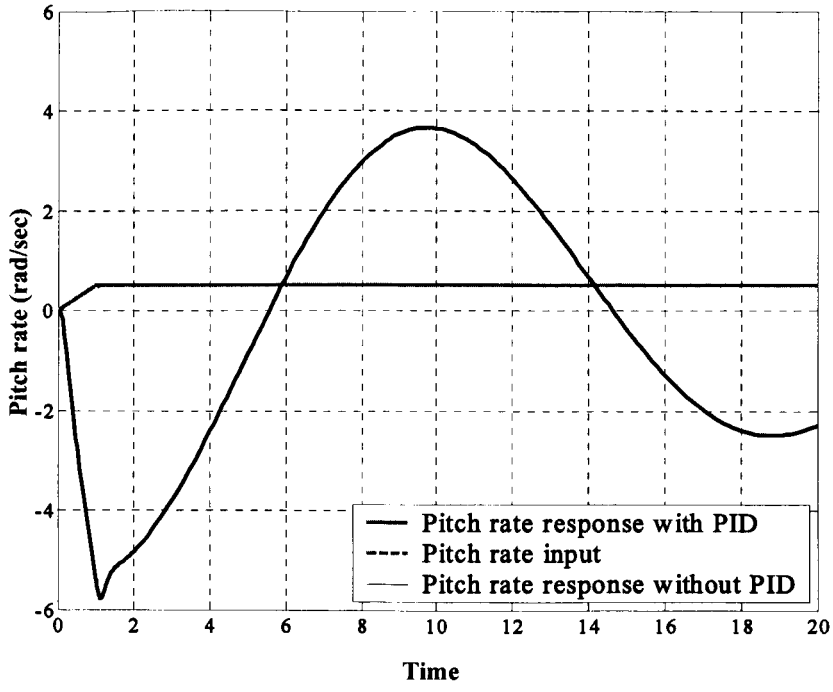


Figure 2.7 - Comparison of pitch rate response with and without PID controller

The eigenvalues of the longitude mode are altered by the presence of the pitch rate controller. Figure 2.8 shows the eigenvalues of the open loop and closed loop systems. Note that the complex conjugate eigenvalue pairs from the servo model are not shown.

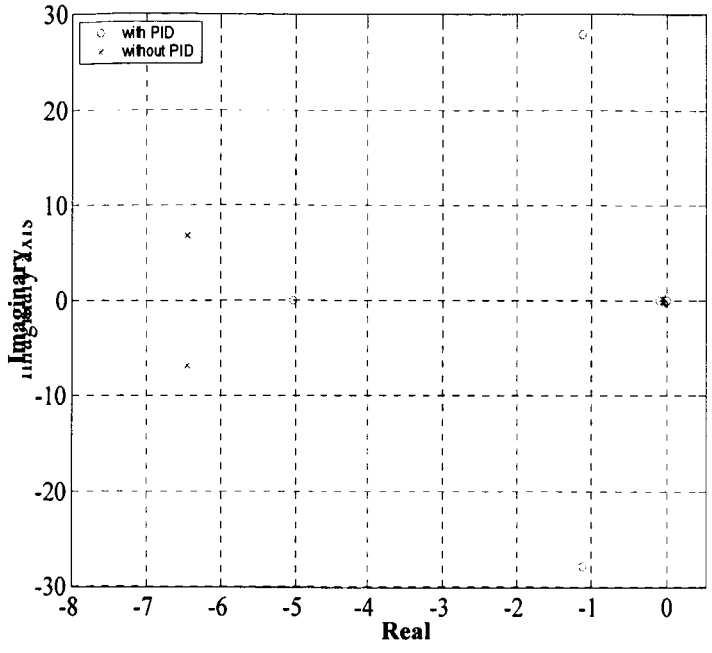


Figure 2.8 – Comparison of the pitch rate eigenvalues with and without the pitch rate PID controller

The open loop (without PID controller) short period eigenvalues are –6.46 ± 6.8i and the open loop Phugoid eigenvalues are –0.043 ± 0.35i. These eigenvalues yield Level 1 longitudinal handling qualities.

The closed loop (with PID controller) eigenvalues are:

$$-22.45 \pm 37.10i$$
$$-1.23 \pm 27.92i$$
$$-5.04$$
$$-0.011$$
$$8.95 \times 10^{-15}$$

The first complex conjugate eigenvalue pair in the above list arises from the 2nd order servo model and the last eigenvalue arises from the integrator of the PID controller. This eigenvalue is unstable but it has a low amplitude. The nature of the remaining eigenvalues is determined by analyzing the corresponding eigenvector. A Phugoid mode eigenvalue has a

corresponding eigenvector with a significant dependency on velocity, and low dependency on angle of attack and pitch rate. For example, the eigenvector of the open loop Phugoid eigenvalue shows this dependency:

$$\hat{E}_{openloop\_Phugoid} = \begin{bmatrix} -0.999 \\ 0.0005 - 0.000i \\ 0.0018 + 0.0108i \\ -0.0038 + 0.0002i \end{bmatrix} \tag{30}$$

The eigenvector of a short period eigenvalue has the reverse characteristic. That is, it has significant dependency on angle of attack and pitch rate and low dependency on velocity, as is illustrated with the eigenvector corresponding to the open loop short period eigenvalue:

$$\hat{E}_{openloop\_Short\_period} = \begin{bmatrix} 0.0026 - 0.3190i \\ -0.0208 - 0.1277i \\ -0.0684 - 0.0722i \\ 0.9336 \end{bmatrix} \tag{31}$$

If the eigenvector shows characteristics that are a combination of the short period and Phugoid modes, the eigenvalue is then classified as a third oscillatory mode.

The closed-loop eigenvector, defined as $\hat{E}^T = \left(x_{long}^T \middle| x_{servo}^T \middle| \delta_e \middle| \dot{\delta}_e \middle| \Omega\right)$, of the corresponding eigenvalue conjugate pair (-1.229 ± 27.9225i) is:

$$\hat{E}_{(-1.229\pm27.9225i)} = \begin{bmatrix} 0.009 + 0.001i \\ 0.004 - 0.007i \\ 0.002 - 0.008i \\ 0.231 + 0.050i \\ -0.001 - 0.037i \\ 0.971 \\ -0.002 + 0.008i \end{bmatrix} \tag{32}$$

This eigenvector does not show the distinct characteristics of either the short period mode or the Phugoid modes. Rather, it shows a combination of the characteristics of both modes. This eigenvalue pair is thus classified as third oscillatory.

The real eigenvalue (-5.0374) has the following corresponding eigenvector:

$$\hat{E}_{(-5.0374)} = \begin{bmatrix} -0.3401 \\ 0.5449 \\ -0.0152 \\ 0.0766 \\ -0.1484 \\ 0.7477 \\ 0.0152 \end{bmatrix} \tag{33}$$

This eigenvector shows that the corresponding eigenvalue is a short period-like eigenvalue.

Lastly, the eigenvector of the eigenvalue (-0.01046) is:

$$\hat{E}_{(-0.01046)} = \begin{bmatrix} -0.99999 \\ 0.0012 \\ 0 \\ 0 \\ -0.0004 \\ 0 \\ 0 \end{bmatrix} \tag{34}$$

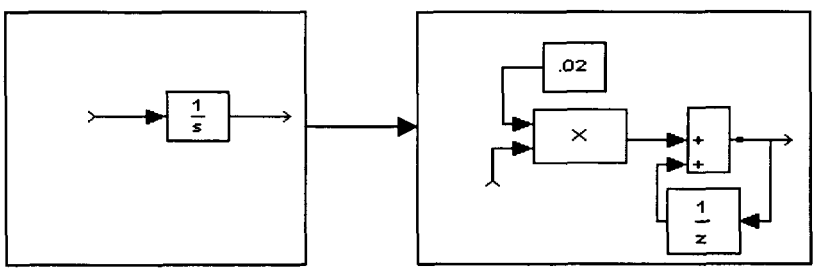
and shows that the corresponding eigenvalue is a Phugoid like mode eigenvalue.

The open-loop Phugoid mode has a greater damping than the closed-loop third oscillatory mode. However, the real part of the closed-loop third oscillatory eigenvalue is more negative than the real part of the open-loop Phugoid-like eigenvalue. Thus, the eigenvalue envelope of the closed-loop third oscillatory eigenvalue goes to zero faster than the open-loop system.

Once the PID gains are set, the pitch rate controller is discretized with a sampling time of 0.02 seconds. The transformation from continuous time to discrete time is performed by converting the PID controller from the Laplace domain to the z-domain. One method of conversion is the Tustin's or the Bilinear approximation which is based on the trapezoidal integration formula.[19] In this method a substitution factor for s is used to replace each occurrence of s. The substitution factor is:

$$s = \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \tag{35}$$

The method used in this application is based on Tustin's approximation. However, instead of using trapezoidal integration formula, the Laplace integrator $\frac{1}{s}$ was converted using a rectangular integration formula. This is illustrated in Figure 2.9.



Laplace space integrator Discrete time integrator

Figure 2.9 – Transformation of integrator from continuous to discrete time

The multiplication of a function by s in the frequency domain corresponds to a time derivative of the function in the time domain. The Laplace factor s is discretized using the definition of the slope of the line as illustrated in Figure 2.10.

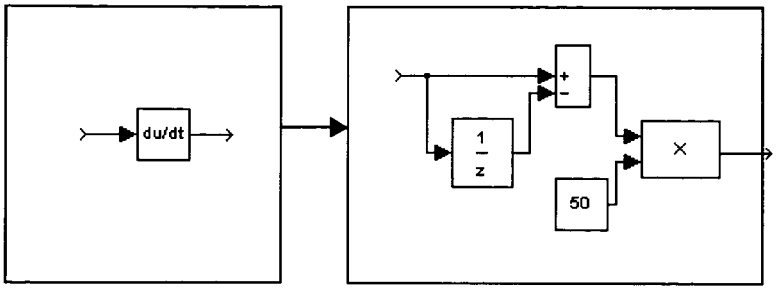


Figure 2.10 - Transformation of the time derivative from continuous to discrete time

The discretized pitch rate controller is shown in Figure 2.11. The transformation of the controller to discrete space did not cause any significant changes to the pitch rate response.

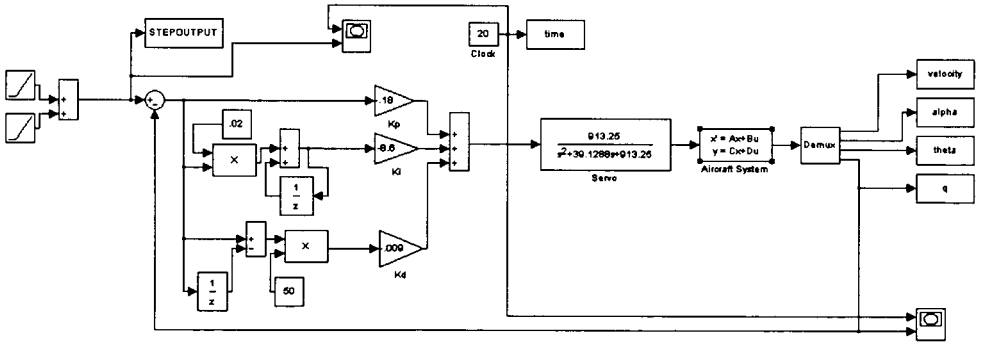


Figure 2.11 - Discretized pitch rate controller

Before the pitch rate controller was implemented into the non-linear system, it was run together in simulation with the discretized yaw damper and bank angle PID controllers with the turn coupler. As is described in section 2.3.3, the turn coupler maps bank angle and velocity to pitch and yaw rates. The pitch rate PID gains were adjusted to perform various

bank angle turns; for each bank angle input it was verified that the pitch rate response was satisfactory. Figure 2.12 shows the block diagram of the three discretized PID controllers and the turn coupler.

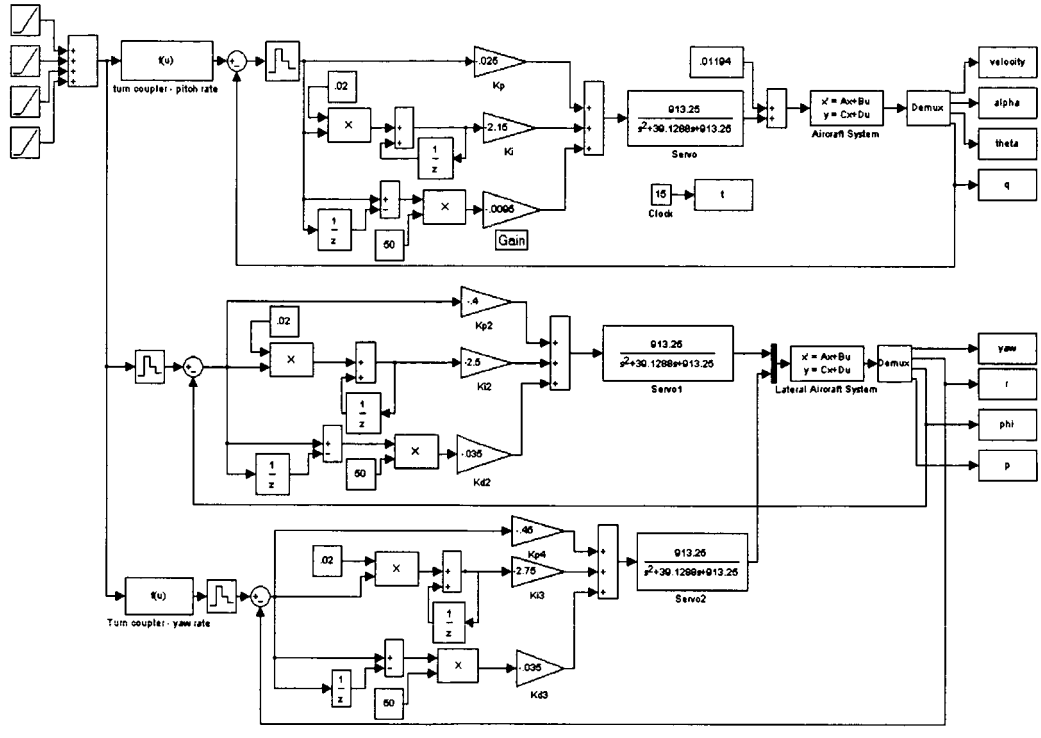


Figure 2.12 - Discretized pitch rate, yaw rate, and bank angle PID with turn coupler

The discretized pitch rate controller was then implemented into the non-linear aircraft dynamics system. A final adjustment to the gains was made to take into account the non-linearity. Figure 2.13 shows the pitch rate response to a 60° bank angle turn in the non-linear system. It is seen that the pitch rate response matches the input very well.

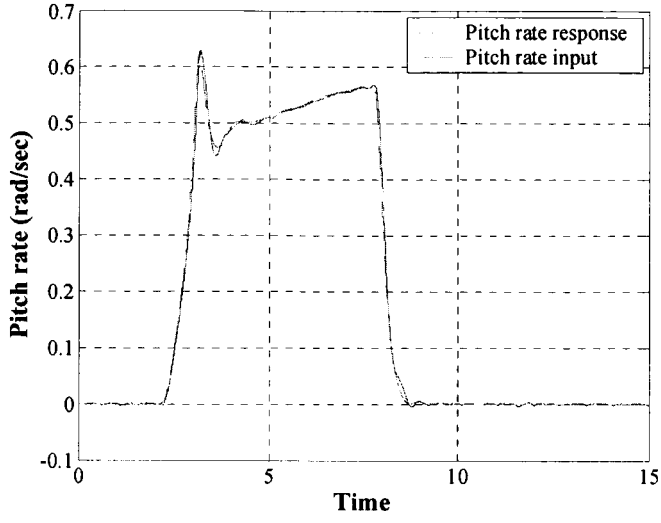


Figure 2.13 – Aircraft pitch rate response in the nonlinear system

However, it was subsequently found that the aircraft kept neither a constant altitude nor velocity in the turns, Figures 2.14 and 2.15. In a turn the velocity initially dropped abruptly, slowly recovered and then settled at a velocity that was greater than the initial velocity. The drop in velocity is due to the increase in lift (and thus also the drag) in the turn. The altitude dropped throughout and after the turn which accounts for this increased velocity.

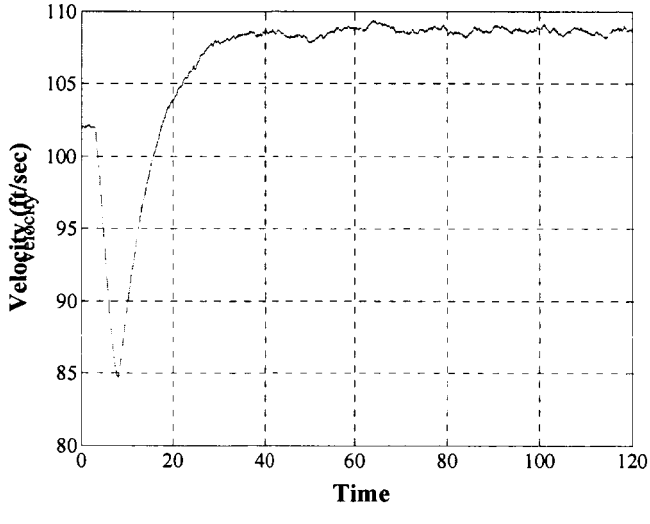


Figure 2.14 –Velocity response of the aircraft

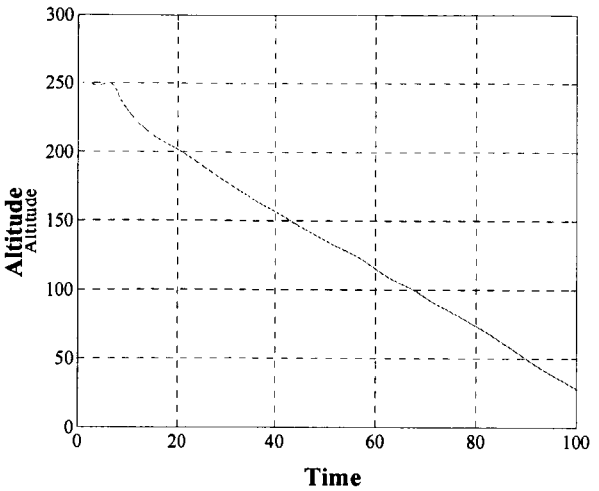


Figure 2.15 – Altitude position response of the aircraft

For this application, no combination of pitch rate, yaw rate, and bank angle PID controllers was found to solve the velocity and altitude problems. It was therefore decided to develop altitude-hold and velocity-hold controllers to correct these problems. The pitch rate response with altitude-hold and velocity-hold controllers is shown in Figure 2.16.

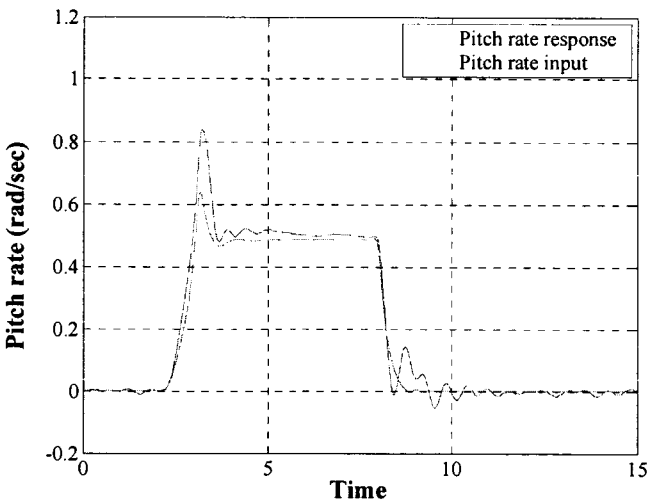


Figure 2.16 – Aircraft pitch rate response in the nonlinear system

The configuration of the final pitch rate PID controller is shown in Figure 2.17.

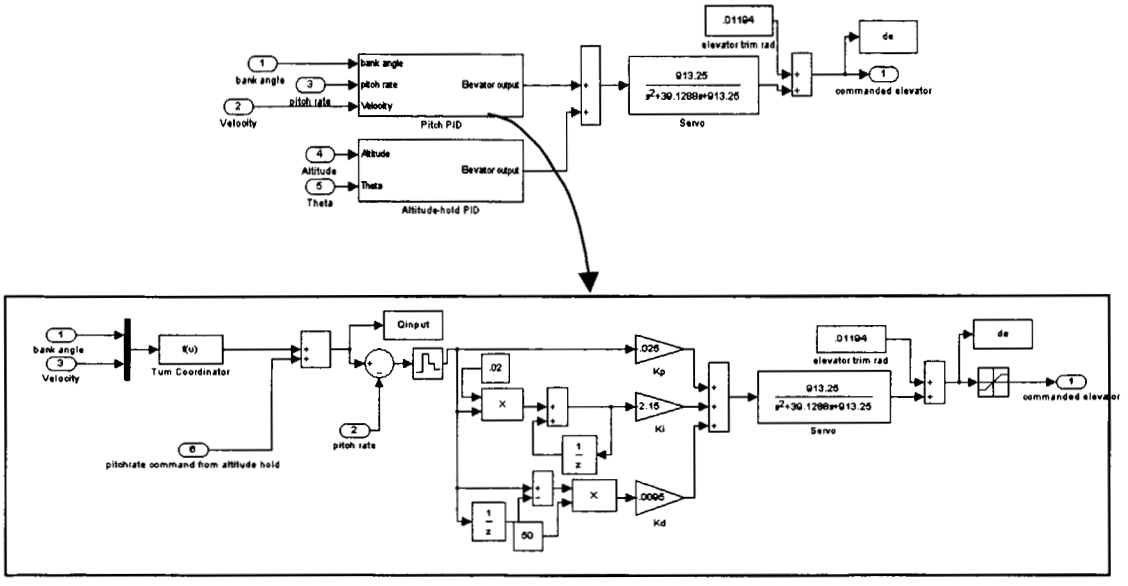


Figure 2.17 - Block Diagram of the final pitch rate controller

The pitch rate controller was written in the C programming language for implementation in the LIFT system. The pitch rate PID control law in block diagram form is:

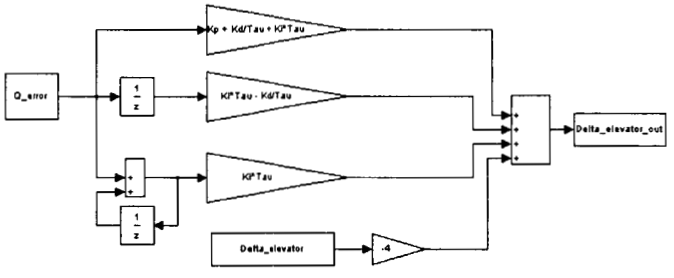


Figure 2.18 – Block diagram of the Pitch rate PID control law

2.3.2 Yaw damper

A yaw damper's function is to increase dampening of the Dutch Roll through rudder deflection and generate a commanding yaw rate. The yaw damper is a very important

controller for Stingray because the aircraft is known to have a slightly damped Dutch Roll mode. The yaw damper design went through several design iterations before the final design was obtained. The first yaw damper design was a yaw rate closed-loop PID controller developed in the same manner as the pitch rate PID. That is, the yaw rate PID was first designed in a linear aircraft dynamics system in continuous time. The yaw rate PID controls the lateral modes thus the aircraft dynamics were represented by a 4th order linear lateral state space model. The matrix elements for this state space model are:

$$\dot{x}_{lat} = A_{lat}x_{lat} + B_{lat}\delta_r \tag{36}$$

$$y_{lat} = C_{lat}x_{lat} + D_{lat}\delta_r \tag{37}$$

$$x_{lat} = \begin{bmatrix} \psi \\ r \\ \phi \\ p \end{bmatrix} \quad y_{lat} = [r] \tag{38}$$

The matrix elements for this aircraft dynamics model are shown in Appendix B.

The servo model was the same as in the pitch rate PID simulation. The block diagram for the PID yaw rate controller with the lateral aircraft dynamics and linear servo model is shown in Figure 2.19.

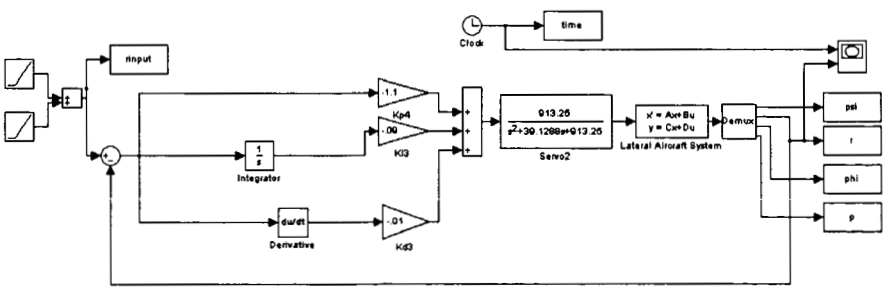


Figure 2.19 - Block diagram of the yaw rate PID in linear continuous system

The gains for the PID controller were adjusted following the method detailed in the previous section. The yaw rate response of the linear aircraft dynamics to a yaw rate input with and without the controller are shown in Figure 2.20. As shown on the figure, the yaw rate PID dampens the yaw rate response effectively.

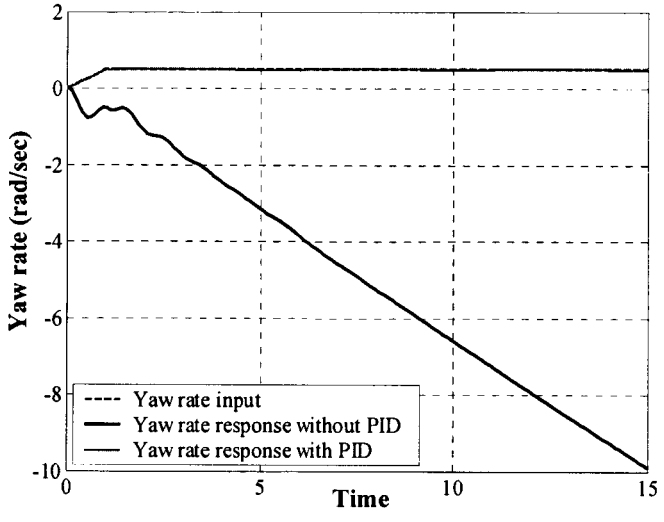


Figure 2.20 – Comparison of yaw rate response with and without controller

After the conversion to discrete space, the yaw rate PID was run together with the pitch rate PID and bank angle PID with turn coupler. The gains of the yaw rate PID were adjusted such that a 60° bank angle turn was performed. Finally, the yaw rate PID was implemented into the non-linear aircraft system and the gains were adjusted again to take into account the non-linear elements. The yaw rate response in the non linear simulation can be seen in Figure 2.21.

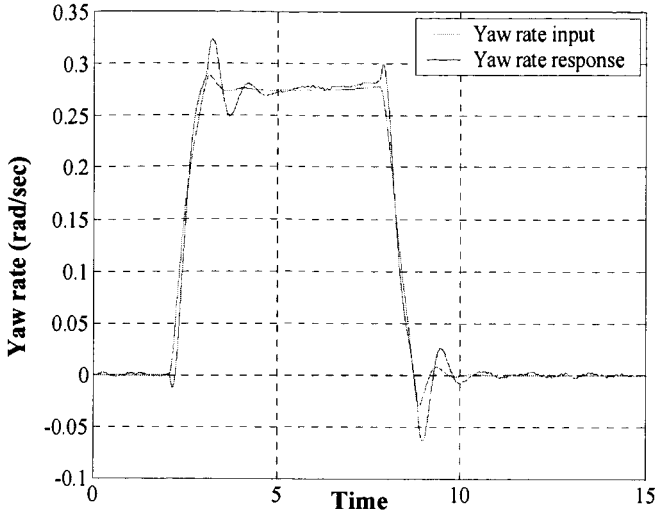


Figure 2.21 –Yaw rate response in the non-linear system

The final design of the yaw rate PID is shown in its block diagram form in the following figure:

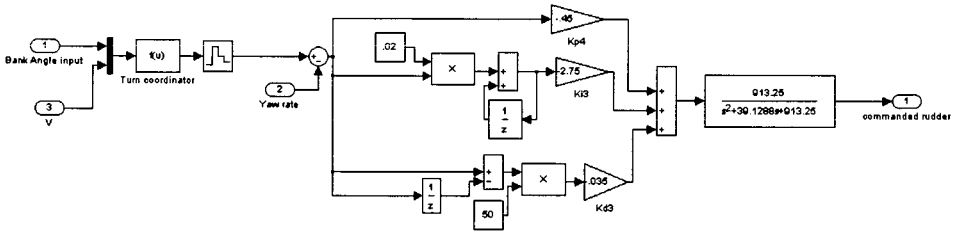


Figure 2.22 – Block diagram of the yaw rate PID in the non-linear system

The yaw rate PID controller was then rewritten into C programming language to be implemented into the LIFT system. The block diagram of this control law is:

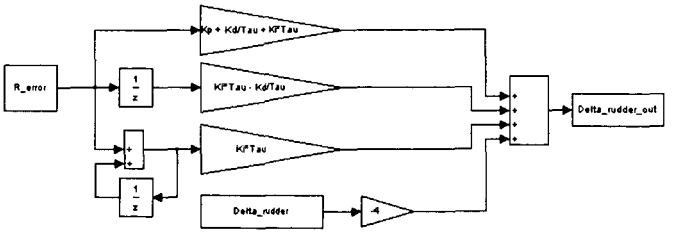


Figure 2.23– Block Diagram of the yaw rate PID control law

During flight testing, the yaw rate controller did not perform as desired. In this flight, the PID yaw damper counteracted the input of the pilot in the turns (see section 3.3.2). To solve this problem, a washout filter was introduced to eliminate the yaw rate feedback in a steady coordinated turn. This washout filter has the following form:

$$W_0(s) = \frac{1}{s + 1/\tau} \tag{39}$$

Knowing that the Stingray UAV experiences a level 2 Dutch Roll mode, the yaw damper was as a Dutch Roll damper was designed following the method in Blakelock.[20] The block diagram shown below in Figure 2.24, is the block diagram for the Dutch Roll damper and includes the washout circuit:

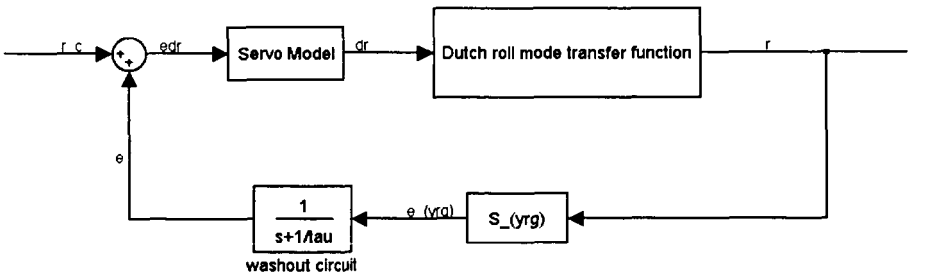


Figure 2.24 – Block diagram of a Dutch Roll damper

The design process consists of finding the time constant τ and the yaw rate gyro feedback gain S_{yrg} in a plot of the close loop root locus for different τ and S_{yrg}. The appropriate τ and S_{yrg} are found when their corresponding Dutch Roll mode eigenvalues yield the greatest damping while keeping the other eigenvalues stable.

Figure 2.25 illustrates the root locus for the Dutch Roll damper for time constant 0.75 with varying yaw rate gyro feedback gain. The root locus demonstrates how the eigenvalues of the Dutch Roll mode varies with respect to the time constant and S_{yrg}.

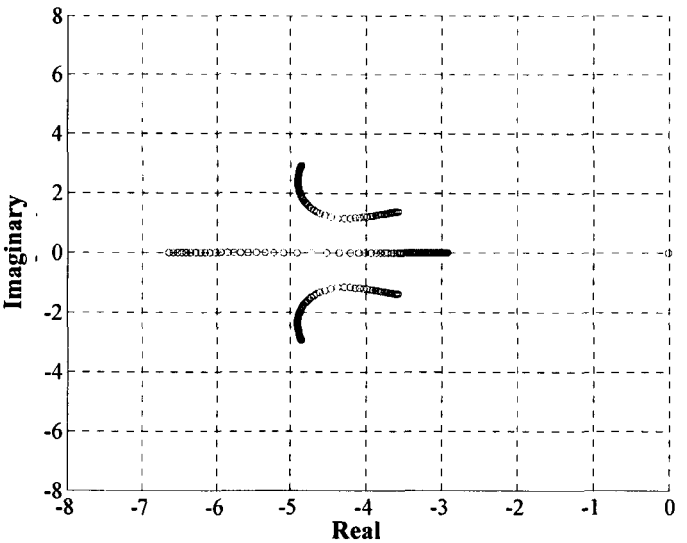


Figure 2.25 – Dutch Roll mode root locus for τ = 0.75sec. and varying S_{yrg}

The root locus plot also shows in red the time constant and S_{yrg} pair that yield the greatest Dutch Roll damping factor. Through several iterations, it was found that the appropriate time constant and yaw rate gyro sensitivity factor are 0.75 and 0.1908 respectively. The Dutch Roll root locus for the pair is shown in Figure 2.26. The plot illustrates that for the selected time constant and S_{yrg} combination, all the eigenvalues are stable and thus the washout circuit designed will not cause an instability.

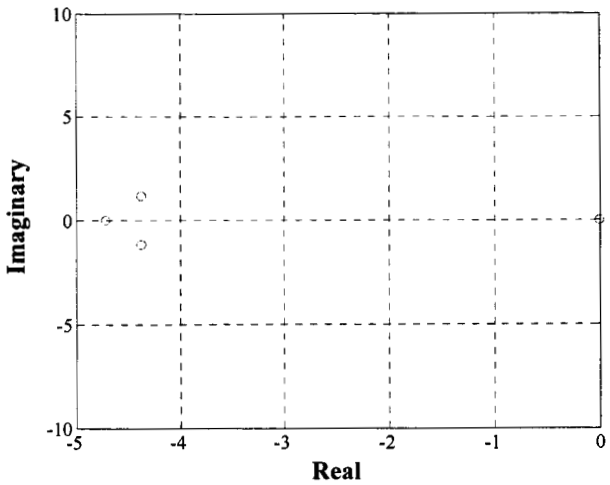


Figure 2.26 – Dutch Roll mode root locus for $\tau = 0.75$sec. and $S_{yrg} = 0.1908$

The continuous time washout circuit was then discretized and implemented into the aircraft simulation. Before the simulation is run, the mapping factor from yaw rate to rudder deflection, Kr_δr needs to be determined. The Kr_δr factor was determined by running the simulation and varying Kr_δr until the amplitude of the aircraft response corresponded well to the desired rudder and yaw rate input. Figure 2.27, shows the aircraft response to the input with a selected Kr_δr factor of –0.25.

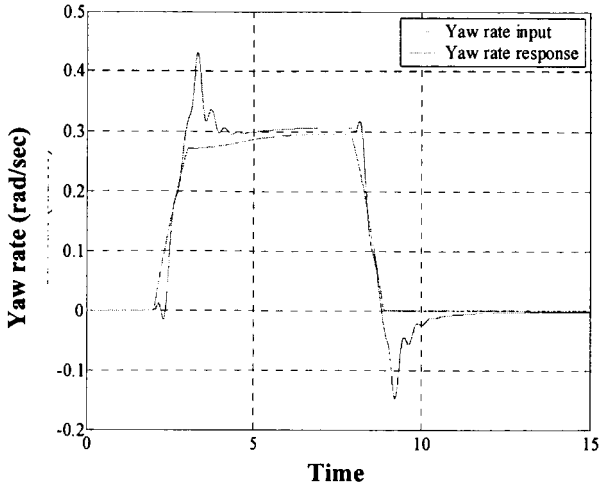


Figure 2.27 –Yaw rate response in the non-linear system with yaw damper

The final design parameters of the discrete time yaw damper washout circuit are:

$$W_o = \frac{z-1}{z-0.9733} \text{ and } Kr\_\delta r = -0.25 \tag{40}$$

The control law of the yaw damper was rewritten in C. The block diagram of this control law is:

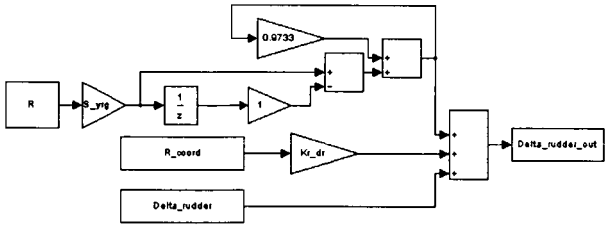


Figure 2.28– Block Diagram of the yaw damper control law

2.3.3 Bank angle PID with turn coupler

The bank angle controller is used to command a desired bank angle turn. To a coordinated turn, the pitch and yaw rates add vectorially to equal to the turn rate vector. The

commanded bank angle is therefore converted to pitch and yaw rates by the following coordinated turn coupler:

$$q = \frac{g_o \tan(\phi)\sin(\phi)}{V} \qquad r = \frac{g_o \sin(\phi)}{V} \tag{41}$$

The velocity in the above equations is the aircraft actual airspeed measured with the Pitot-static probe. The commanded yaw rate and pitch rate are then inputs to their corresponding controllers.

The bank angle PID was developed following the methods described in the pitch rate PID section. The bank angle PID was first designed with linear lateral aircraft dynamics in continuous time. The gains were adjusted to obtain a desire bank angle response. The block diagram of the controller is shown in the follow figure:

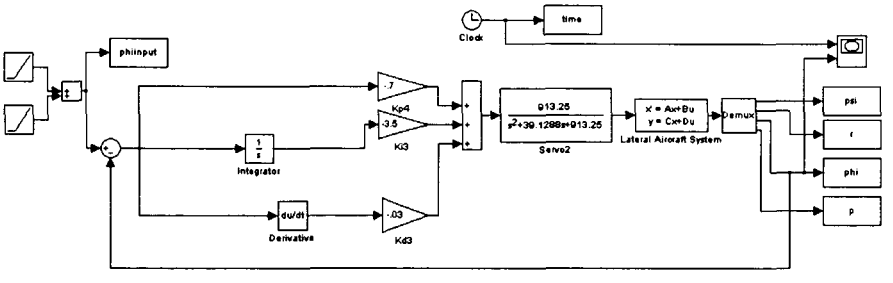


Figure 2.29 – Block diagram of the bank angle PID in linear continuous system

After the bank angle PID was discretized, it was linked with the pitch rate and yaw damper PID through the turn coordinator. The gains were once again adjusted such that the bank angle PID would function as desired for a 60° bank angle turn.

The bank angle PID was then implemented into the non-linear simulation and its gains were adjusted. The final configuration of the bank angle PID is shown in the following block diagram:

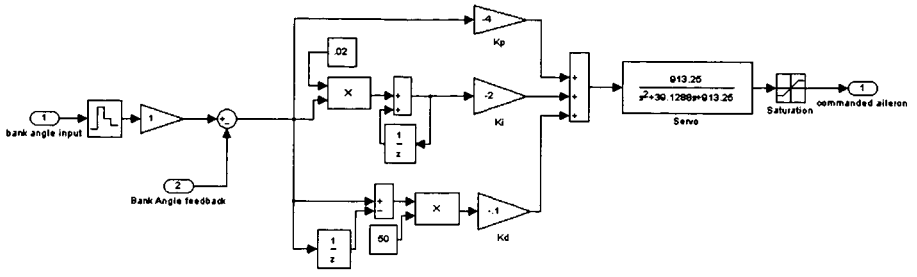


Figure 2.30 – Block diagram of the bank angle PID in the non-linear system

The aircraft bank angle response to a 60° bank angle turn is shown in the following figure.

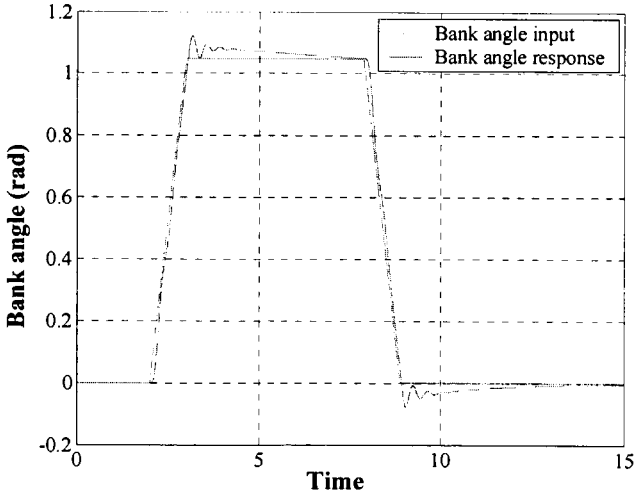


Figure 2.31 – Bank angle response in the non-linear system to a 60° bank angle turn

The block diagram of the bank angle PID control law is:

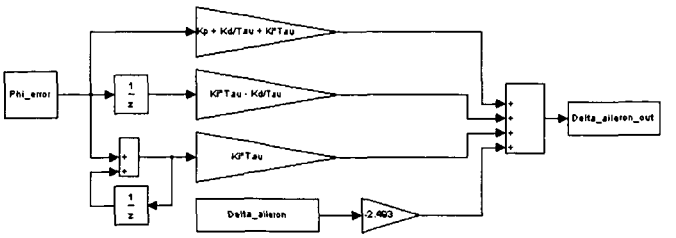


Figure 2.32– Block diagram of the bank angle PID control law

2.3.4 Altitude-hold PID

The altitude-hold autopilot is a PID type controller that was designed to hold a desired altitude during straight flights and in turns. The development of the altitude hold was made directly in the non-linear aircraft dynamics system after it was determined that the aircraft could not keep altitude with the pitch rate, yaw damper, and bank angle controllers.

The altitude hold controller was designed by evaluating the altitude and pitch angle errors and mapping these errors to a pitch rate command. The block diagram of the altitude hold controller is shown in Figure 2.33.

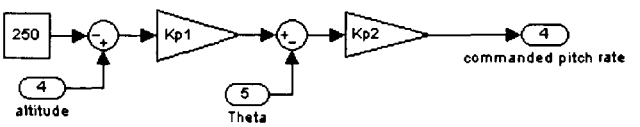


Figure 2.33 – Block diagram of an altitude-hold controller

The input into the autopilot is a preprogrammed altitude. The autopilot outputs a pitch rate command that is sent to the pitch rate controller. There are two feedbacks for the altitude-hold. The first feedback is the altitude measured by the pressure transducer. The

second feedback is the flight path angle. This angle is obtained by solving the Euler's angles equations (Equations (13), (14)) simultaneously using the angular rates measured by the transducers.

The gain Kp_1 in Figure 2.33 represents the mapping of an altitude error to a pitch angle correction. The value of this gain is determined by setting the desired altitude correction of 10 ft in 1 second for the aircraft at cruise. From this parameter the corresponding pitch angle correction can be determined as follows:

$$\theta_{corr} = -\arcsin\frac{\Delta h}{1\,\text{sec}\cdot V_{cruise}} \cong -0.1 rad \tag{42}$$

Thus for each altitude error of 10 ft, the aircraft should have a pitch angle correction of 0.1 radians. Thus the gain Kp_1 is determined to be:

$$Kp_1 = \frac{\theta_{corr}}{\Delta h} = -0.01 \tag{43}$$

The gain Kp_2 in Figure 2.33 is the mapping factor from the pitch angle error to the pitch rate command that is necessary to eliminate the pitch angle error. This gain is determined by setting the pitch rate necessary to correct a pitch angle error of 0.1 radians is 0.5 rad/sec. Thus Kp_2 is set to:

$$Kp_2 = 0.1 \tag{44}$$

The gains Kp_1 and Kp_2 are then implemented and adjusted by running the non-linear simulation such that the altitude remains constant. The block diagram of the final altitude-hold controller is shown in Figure 2.34.

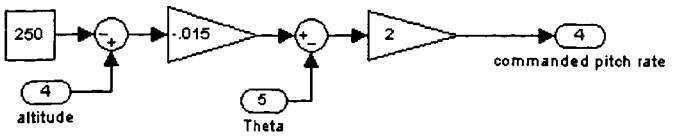


Figure 2.34 – Block Diagram of the altitude-hold controller

The altitude response of the aircraft to a 60° bank angle turn is shown in Figure 2.35. The altitude is kept constant with a ± 5 ft. error. This was deemed acceptable.

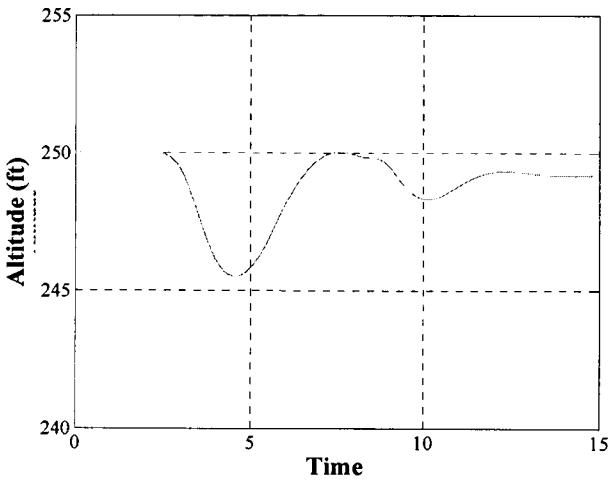


Figure 2.35 - Altitude response in the non-linear system

The block diagram of the altitude-hold control law is shown in Figure 2.36.

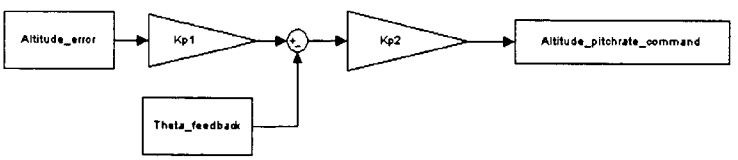


Figure 2.36 - Altitude-hold control law

2.3.5 Velocity-hold PI

The velocity-hold was added to the autopilot unit to keep the velocity constant. The velocity-hold contains, in addition to the proportional and integrator gains, a first order

model of the engine and the throttle servo. This first order model was required due to the fact that the output of the velocity-hold controller is a throttle command and the engine has a certain response time unlike a control surface. An experiment was set up to measure the time response of the engine to a throttle command. From the experiment, a first order linear servo and engine time response model was obtained. The transfer function of the model is:

$$E(s) = \frac{5.399}{s + 5.399} \tag{45}$$

The input to the autopilot is the commanded cruise velocity from the navigator and the feedback is the velocity obtained from the Pitot static probe. The gains of the velocity-hold are determined by running the non-linear simulation and adjusting the gains such that the velocity remained constant. According to Blakelock[20], the velocity settling time of an aircraft with a velocity hold controller is in the order of 30 seconds. This desired large settling time causes the velocity gains to be fairly low and not over-sensitive to velocity changes. Furthermore, to adjust for the aircraft's loss of velocity in the turn, a function based on bank angle is introduced to increase the desired velocity in the turn (Equation (43)). In this manner, the loss of velocity in the turn is effectively reduced while keeping the throttle change smooth.

$$V = \frac{15}{\pi}\phi_{command} + V_{cruise} \tag{46}$$

The block diagram of the final velocity-hold is shown in Figure 2.37.

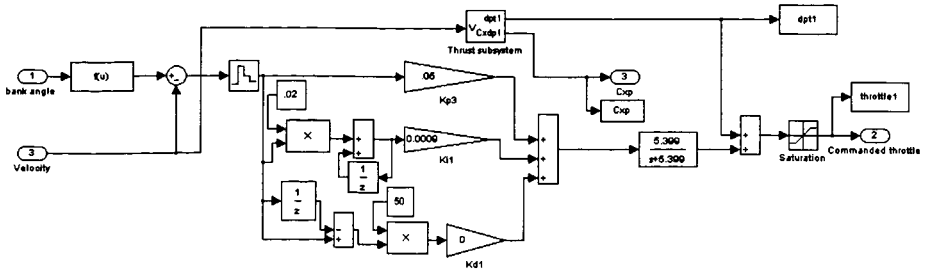


Figure 2.37 – Block Diagram of the velocity-hold PI

The velocity response of the aircraft to a 60° bank angle turn is shown in Figure 2.38. The velocity was kept constant with a ± 5 ft/sec error. This was deemed acceptable.

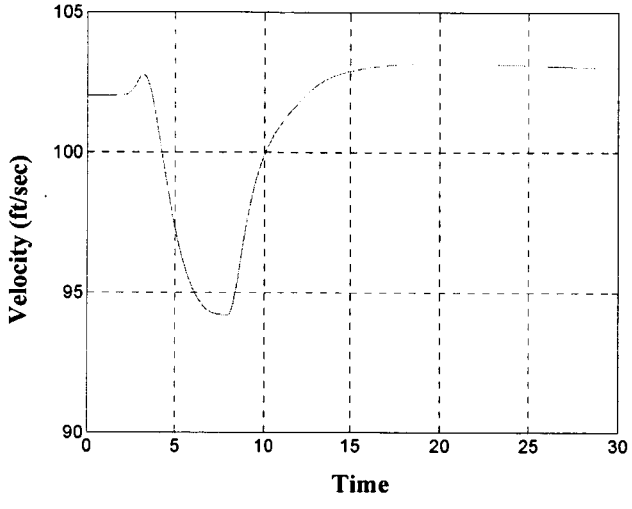


Figure 2.38 - Velocity response in the non-linear system

The block diagram of the velocity-hold control law is shown in Figure 2.39.

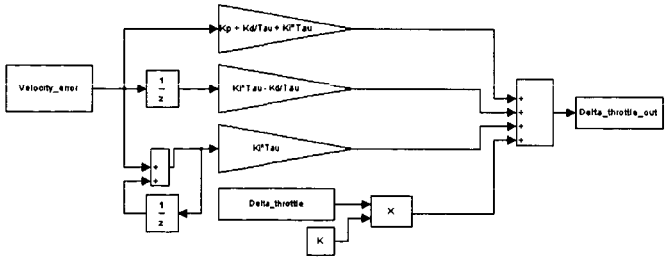


Figure 2.39 - Velocity-hold control law

To implement the autopilot unit into the LIFT system, each controller was converted into a control law written in the C programming language. The control laws in C format for the autopilot unit can be found in Appendix C.

2.3.6 Robustness of the Autopilot unit

A series of simulations in MATLAB were conducted to verify the operation of the autopilots and the aircraft system. The toolbox allows for different initial settings of the variables such as velocity, angle of attack, and angular rates. This was a useful tool to check the integrity and the range of the function of the autopilots. In these simulations, different bank angles, initial trim conditions and velocities are the inputs. The results showed that the autopilot unit performed well with various initial trim conditions and bank angles. The operation of the autopilot, however, is limited to the velocity of the aircraft and was observed to perform effectively within the velocity range of 80-115 ft/sec. The velocity range was limited to the maximum aircraft velocity of 115 ft/sec according to design[14].

The robustness of the controllers was tested by analyzing the eigenvalues of the system when the aerodynamic coefficients were varied at +/- 20%. The eigenvalues from the different aerodynamic coefficients are plotted in Figure 2.40a-c.

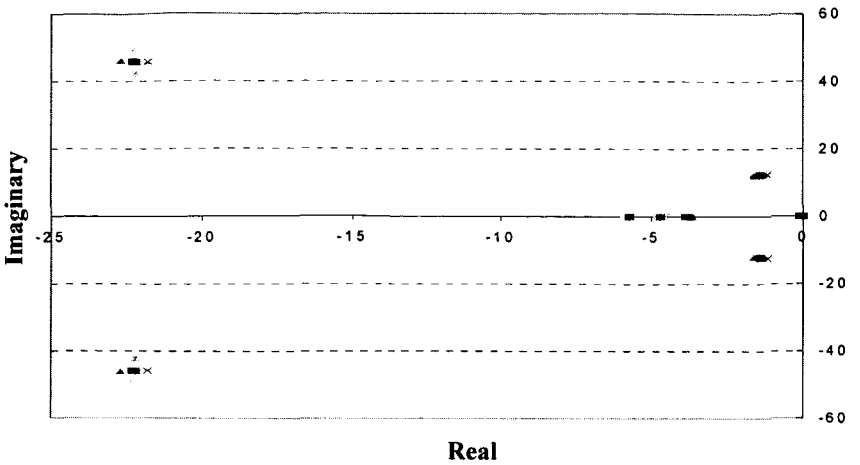


Figure 2.40a – Eigenvalues for the longitudinal mode

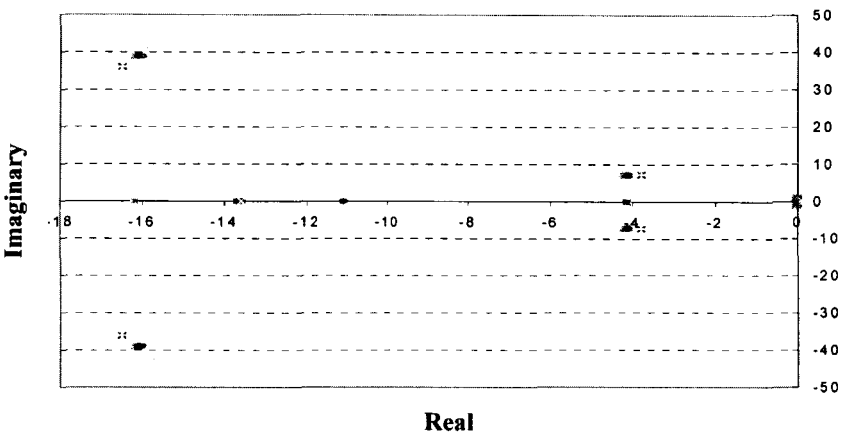


Figure 2.40b – Eigenvalues for the lateral mode (Yaw rate)

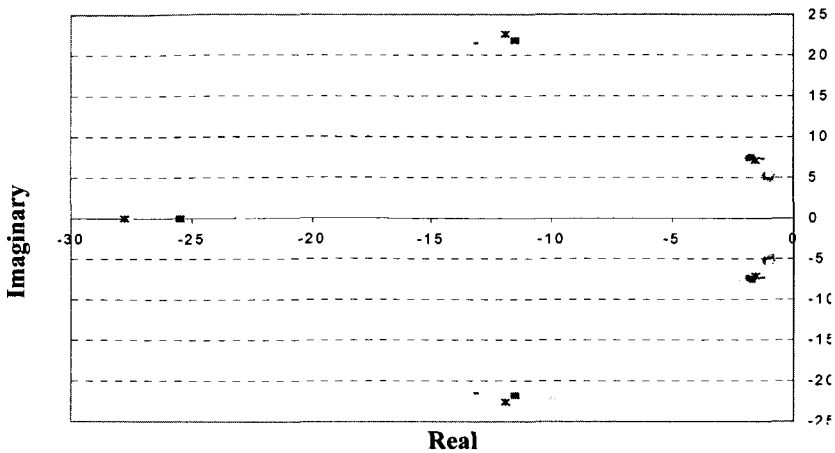


Figure 2.40c – Eigenvalues for the lateral mode (Roll rate)

The test showed that overall the eigenvalues did not significantly shift with varying aerodynamic coefficients values and thus demonstrated that the controllers are efficient for a range of stability derivatives. The test did also highlight the dominating aerodynamic coefficient, $C_{y\beta}$, which causes the greatest change in eigenvalues placement. This coefficient is part of the lateral aerodynamic coefficients and enhances the Dutch Roll mode.

2.4 Navigator

The navigator is designed to guide the aircraft through a pattern of predetermined waypoints. These waypoints consist of latitude, longitude, and altitude positions and are listed in an input file that is downloaded to the LIFT system prior to flight.

A schematic of the navigation system is shown in Figure 2.41. The navigator program processes the following steps. First, the navigator reads the GPS information (latitude, longitude, and heading), velocity, and altitude of the aircraft. Next, this information is compared to the desired waypoint. The navigator then determines whether the waypoint has

been reached, or whether a heading correction is necessary. In the case of the latter, the navigator computes the desired heading correction in terms of a bank angle command. The last step is to output the bank angle command in the form of an AMS command to the autopilot unit. The desired altitude and velocity are also sent to the autopilot unit. In the rest of this section, the algorithm of the navigator is explained in further details.

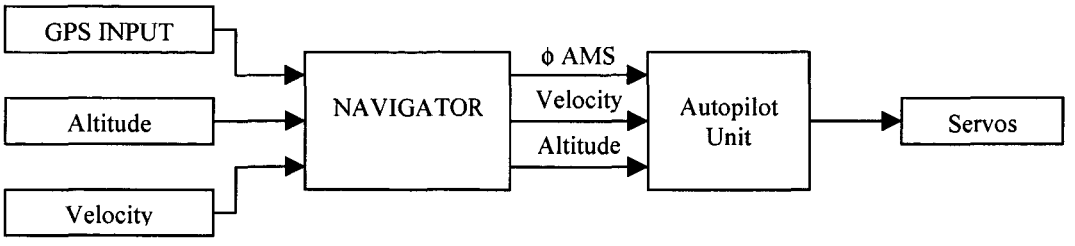


Figure 2.41 – Block diagram of the navigation system

The desired waypoint pattern consists of target waypoints and intermediate waypoints. The intermediate waypoints are used to lead the aircraft along a desired path towards the target waypoints; they are also used to set up a climbing or descending path towards the target waypoints.

The aircraft is said to have reached a waypoint if it is located within a determined circle around the waypoint. The radius of this circle, also termed the capture radius, is the turn radius that the aircraft is capable of achieving with a 2g turn, that is a 60° bank angle turn. This capture radius is defined by the following equation:

$$R_o = \frac{V^2}{g_o \tan(\pi/3)} \tag{47}$$

The diagram shown in Figure 2.42 illustrates the relationship between the capture radius, distance AB, and the heading angles Ψ_{err} and $\Delta\Psi$. At each sampling time, the position of the aircraft (A) and heading ($\vec{A}$) are compared with the destination point (B). If distance AB is less than the minimum turning radius, then the waypoint has been reached and the aircraft can proceed to the next GPS waypoint. On the other hand if distance AB is greater than the capture radius, the navigator continues its analysis by computing the heading error, Ψ_{err}, and the angle $\Delta\Psi$ formed by the line (AB) and the tangent line (AC) of the capture circle passing through point (A).

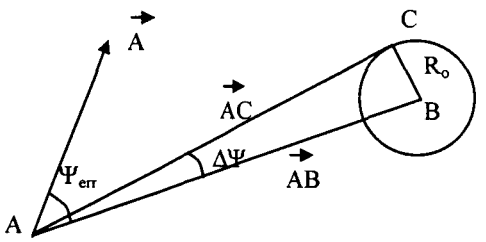


Figure 2.42 - Navigation geometry

One of the difficulties in designing the navigator lies in determining when a commanded turn is necessary. It is not desirable to correct the heading of the aircraft every time the aircraft is only slightly off its desired course. This could lead to the aircraft making many little corrections, or oscillating about the desired course. Likewise it is not desirable to only command large bank angle turns. For these reasons, a comparison ground between Ψ_{err} and $\Delta\Psi$ was set. It was decided that an appropriate comparison is to compare $\Delta\Psi$ with half the value of Ψ_{err}; a correction turn is only commanded if $\Delta\Psi$ is less than $\frac{\Psi_{err}}{2}$. If $\Delta\Psi$ is greater than $\frac{\Psi_{err}}{2}$, the navigation system waits for the next sampling time to read a new GPS data.

The next step is to compute the correction heading angle Ψ_{corr}. The turn process can be described as a ramp input, Figure 2.43. The aircraft is gradually banked at a rate of $\pi/3$ rad/sec until it reaches a peak bank angle. The aircraft is held at the peak bank angle for a determined time, and then gradually banked down to its original position. The ramp input is designed to reach a maximum bank angle of $\pi/3$ rad. The heading correction created by ramping the bank angle up to $\pi/3$ rad is Ψ_1. From symmetry, the ramp down portion also generates a correction angle of Ψ_1. The constant $\pi/3$ rad bank angle is only necessary if the needed correction is greater than $2\Psi_1$.

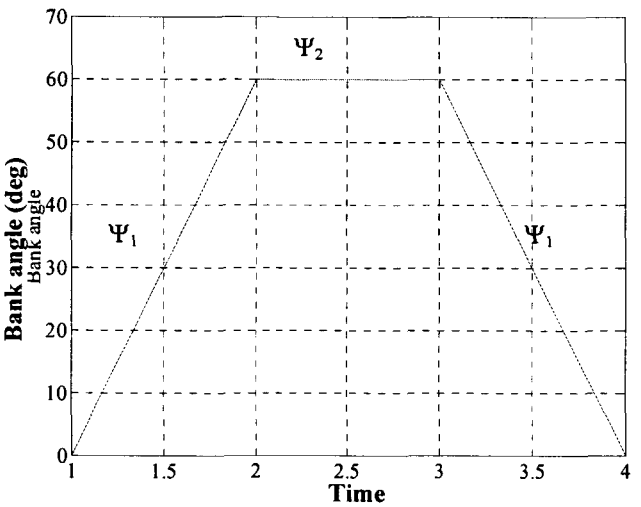


Figure 2.43 – Bank angle ramp command

The correction angle Ψ_1 is defined in Equation (48):

$$\psi = \int \omega \cdot dt \tag{48}$$

where

$$\omega = \frac{g_o}{V}\tan(\phi) \tag{49}$$

$$\phi = f(t) = ct = \frac{\pi}{3}t \tag{50}$$

and thus

$$\psi = -\frac{g_o}{Vc}\ln(\cos(ct))\Big|_{t_o}^{t_1} \tag{51}$$

where t_o and t_1 are respectively the starting and finishing time of the bank angle ramp input. Furthermore, solving equation (51) for the ramp time of 1 second, will yield the correction angle Ψ_1:

$$\psi_1 = \frac{-3}{\pi}\frac{g_o}{V}\ln(\cos(\pi/3)) \tag{52}$$

Comparing Ψ_{err} with $2\Psi_1$ yields two situations: one where $\Psi_{err} \leq 2\Psi_1$ and a second where $\Psi_{err} \geq 2\ \Psi_1$. For the first case, $\frac{\psi_{err}}{2}$ is substituted for Ψ in equation (51) to solve for the ramping time, t_1. Assuming t_o is zero, t_1 is then defined as:

$$t_1 = \frac{3}{\pi}\cos^{-1}\left(e^{\left(\frac{-\pi V}{6\ g_o}\psi_{err}\right)}\right) \tag{53}$$

The maximum bank angle, ϕ_{max} , for the required turn is then calculated using equation (54):

$$\phi_{max} = \frac{\pi}{3}t_1 \tag{54}$$

Finally, the time, t_f, to complete the entire turn is defined as:

$$t_f = 2t_1 \tag{55}$$

In the second case where $\Psi_{err} \geq 2\ \Psi_1$, the $\frac{\pi}{3}$ rad bank angle is held for a time, t_2, and the correction angle, Ψ_2, that is generated when the aircraft is banked 60° is defined as:

$$\psi_2 = \psi_{err} - 2\psi_1 \tag{56}$$

The duration, t_2, when the aircraft is kept at a $\pi/3$ rad bank angle is determined from:

$$t_2 = \frac{V}{g\tan(\pi/3)}\psi_2 \tag{57}$$

Finally, the time to complete the turn for the second case is defined as:

$$t_f = 2t_1 + t_2 \tag{58}$$

The bank angle ramp input command defined by the navigator is sent to the autopilot unit in the form of a bank angle AMS input. The navigator follows this process at each sampling time. The flow chart of the navigation program is summarized in Figure 2.44 and a detailed algorithm of the navigation program is shown in Appendix D.

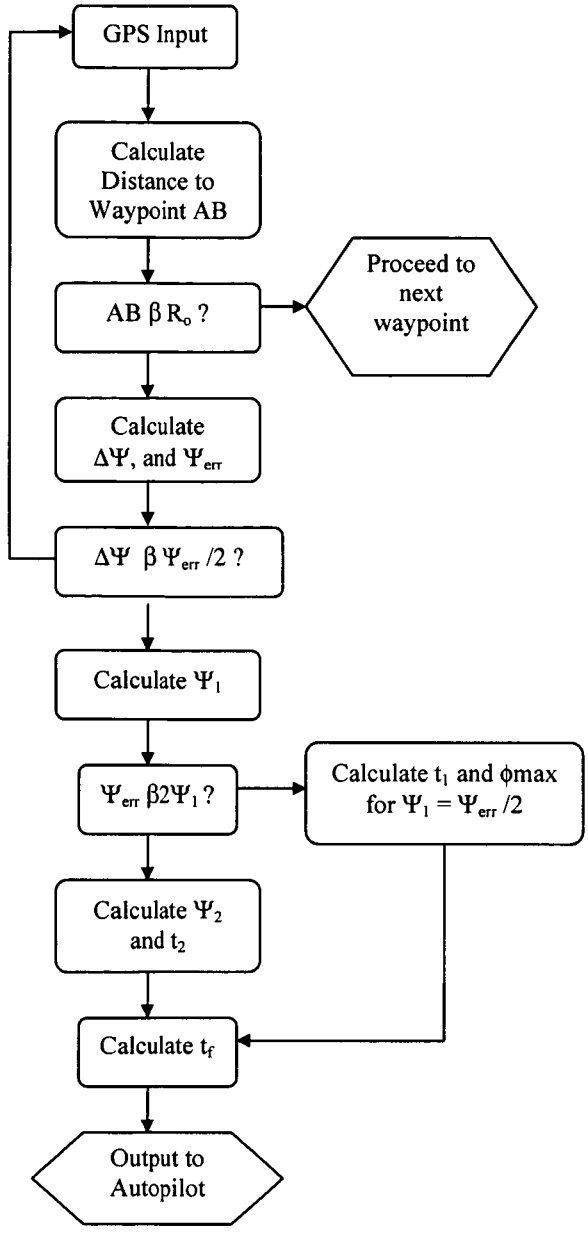


Figure 2.44 – Navigation System Flow Chart

3 Flight Results and Discussions

Stingray UAV is flown at the NC State UAV flight test facility named Perkins Field in Butner, NC. The asphalt runway is 450 ft long and 50 ft wide.

To ensure that the preparation of the aircraft is performed correctly, a set of checklists is used on the day prior to flight and on the day of the flight. A checklist of the required equipment is also used. These checklists are found in Appendix E.

During flights, a flight log is written to note the pilot's radio commands, comments, and the visual observation of the aircraft (Appendix F). The data collected during flight is analyzed and compared to the calculations and simulations following the flights.

3.1 Flight test sequence

The flight test schedule was divided into three parts: GPS, autopilot unit, and navigation test flights. The flight test of the GPS system consisted of checking the integrity of the GPS system with the LIFT system on the UAV. The series of autopilot flight verifications was a build-up program in the order of the autopilots that are described above.

3.2 GPS

The first phase consisted of static and dynamic tests of the position estimates with the LIFT system. During the static tests, the GPS software and hardware were checked for functionality. The GPS data indicated that typically 3 satellites were locked, and at a fixed location the acquired geographic latitude and longitude acquired did not fluctuate. Initially in the static test the GPS antenna was positioned in the main cargo compartment. During the

test it was found that it was difficult to gain and lock onto a satellite with the antenna placed in the main cargo bay. When the antenna was moved to the forward compartment, the satellite contact improved greatly. The antenna was thus placed in the front compartment for the remainder of the tests.

The dynamic tests consisted of a series of UAV flights at the NC State UAV flight test facility, Perkins Field. During these flights the UAV was flown in a series of oval circuits along the runway. Figure 3.1 shows a typical flight test trajectory in terms of the GPS latitude and longitude.

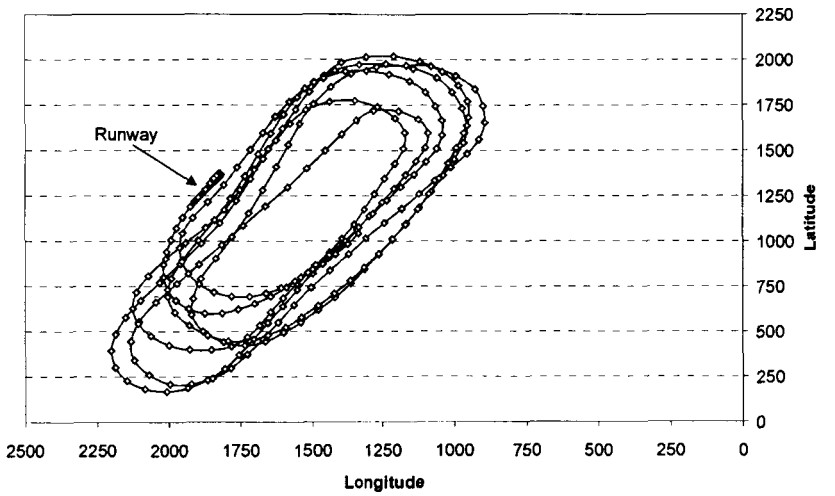


Figure 3.1 - Latitude and longitude GPS position

In addition to the position, the GPS card in the LIFT system provides the aircraft's heading and velocity. The heading and velocity during the cruising part of the flight are shown in Figure 3.2. The acceleration of the aircraft during the straight flight (constant heading) portion of the flight paths can be seen. Also during the 180-degree turns, from a 30° to 210° heading and vice-versa, the deceleration of the aircraft is also seen.

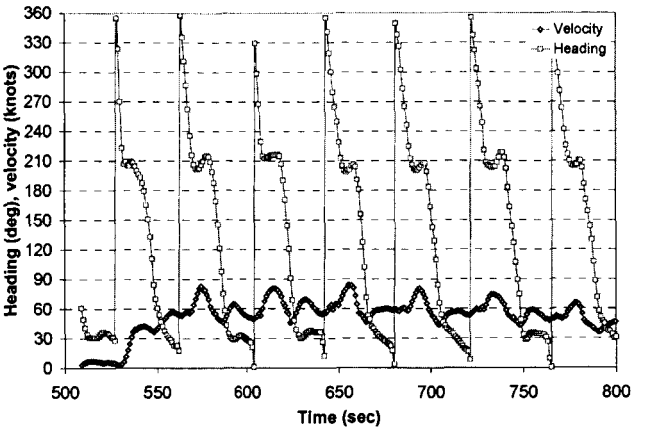


Figure 3.2 - GPS Heading and velocity data

The slope of the heading plot during the turn provides information about the bank angle, the load factor and the increase in drag. Indeed, the slope of the heading is the turn rate, ω, of the aircraft. From the turn rate, the load factor is determined from:

$$n = \left(\frac{\omega V}{g}\right)^2 + 1 \tag{59}$$

The aircraft's bank angle in the turn is then computed as:

$$\phi = \arccos\left(\frac{1}{n}\right) \tag{60}$$

Lastly, the increase in the aircraft's drag is computed by calculating the increase in the induced drag coefficient using the following equation:

$$C_{D_i} = \frac{4W^2\left(n^2 - 1\right)}{\rho^2 V^4 S^2 \pi e A R} \tag{61}$$

In Figure 3.3, the change in C_{Di} is plotted for one of the turns. This figure indicates the significant increase in C_{Di} in the turn, which leads to a decrease in altitude and velocity as demonstrated in the simulation run presented in Section 2.3.1.

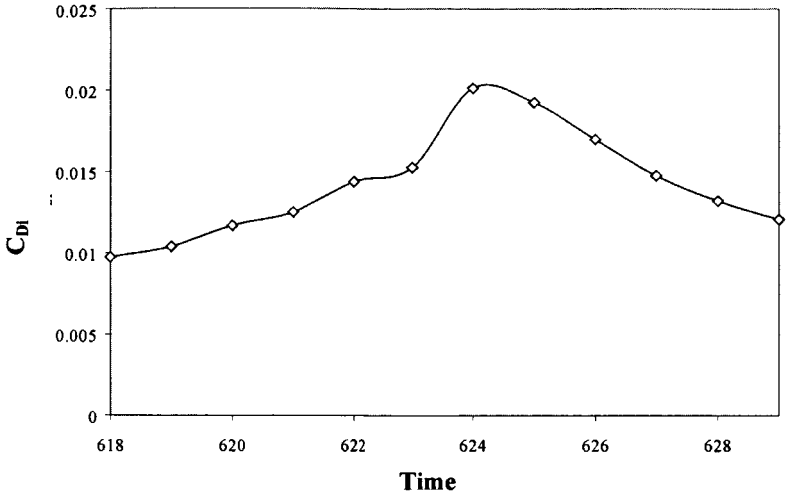


Figure 3.3 – C_{Di} variation in a turn

Overall the static and dynamic tests verified the reliability and accuracy of the GPS unit in the LIFT avionics system.

3.3 Autopilot Unit

The autopilot unit was test flown in a systematic manner. First the pitch rate controller was flown, then the yaw damper. Next, the bank angle controller with the turn coordinator and altitude hold were flown. Due to time constraints, the velocity-hold was not flown.

During the flights, an Automatic Maneuvering Sequence (AMS) was engaged to study the response of the aircraft and the efficiency of the controllers. The AMS is a preprogrammed command input to a chosen control surface. A rudder AMS was engaged without the yaw damper to study the Dutch Roll mode of the aircraft. Then a rudder AMS was engaged with the yaw damper to examine the effect of the yaw damper on the Dutch

Roll mode. In the turn coordinate flights, an aileron AMS was used to perform a 180° turn at a 60° bank angle.

In the following sections, the flight test results of the pitch rate, yaw damper, coordinated turn flights are presented.

3.3.1 Pitch rate controller

The pitch rate autopilot was the first controller tested. The aircraft was flown in an oval pattern, as shown in Figure 3.1. The aircraft was trimmed at less than full throttle. From visual observation, it was noted that the aircraft remained trimmed and maintained its flight path when the autopilot was engaged. Stingray UAV also performed well in the turns. During the flight test, the autopilot was engaged several times. A pilot commanded elevator doublet was performed when the autopilot was engaged. From observation, the aircraft responded well to the doublet.

The flight test data stored in the LIFT system was analyzed and compared to the performance of the designed pitch rate controller and aircraft system. The pilot input was compared with the aircraft pitch rate response. The two plots, shown in Figure 3.4, compare very well.

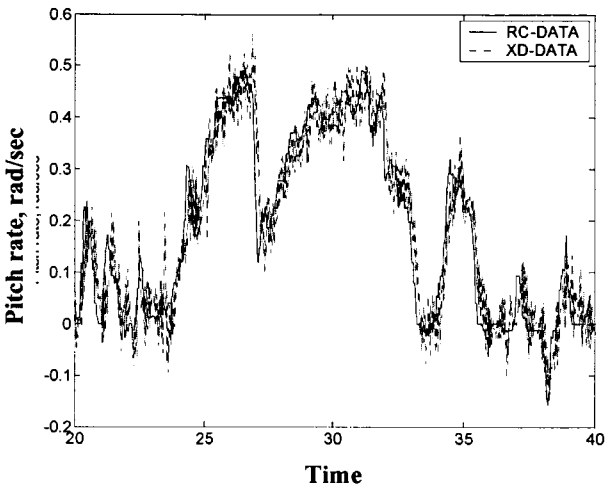


Figure 3.4 – Comparison of piloted pitch rate command and flight pitch rate data

The flight test result was then compared to the SIMULINK designed aircraft system. To verify the aircraft system result, a few modifications in the SIMULINK model were necessary. The reason for the modifications arose due to the non-linearity of the aircraft model. The state vector is modified to isolate the effect of the other autopilots. Hence, in the state vector, the velocity is set to the design cruise speed of 102ft/sec, and the altitude to the estimated cruise altitude of 250ft above ground level. Additionally, the roll rate and yaw rate from flight are used to substitute the feedback roll and yaw rates of the model. In this manner, only the effect of the pitch rate autopilot is seen.

The pilot elevator input is used as the input to the pitch rate autopilot in SIMULINK. Fontenrose[21] showed that the pilot commanded elevator input is mapped to pitch rate by a factor of - 4/sec. Thus, when inputting the pilot commanded pitch rate into the designed aircraft system, the input is multiplied by - 4/sec. The pitch rate response from the simulation is compared with the flight pitch rate data in Figure 3.5. As seen in the figure the system and

flight pitch rate compare very well. This result shows that the SIMULINK aircraft model is a good representation of the aircraft.

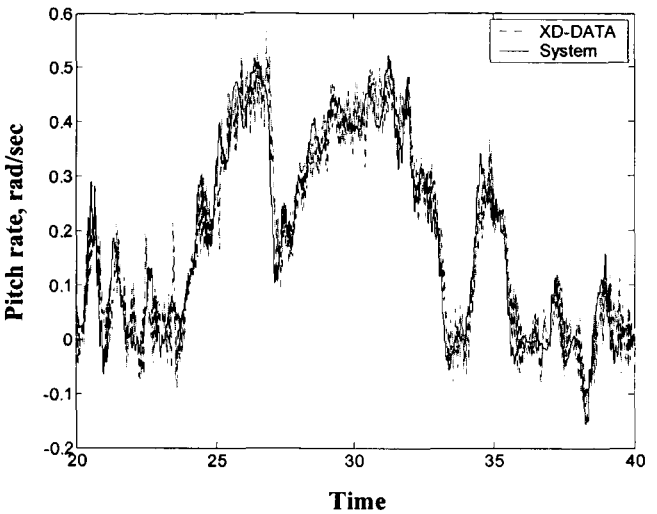


Figure 3.5 – Comparison of designed pitch rate response and flight pitch rate data

A Fast Fourier Transform (FFT) of the measured pitch rate response from the flight data and the designed aircraft system are compared in Figure 3.6. It can be seen that the peaks of the two FFTs occur at the same frequency. This further emphasizes that the modeled aircraft is a good representation of the actual aircraft system.

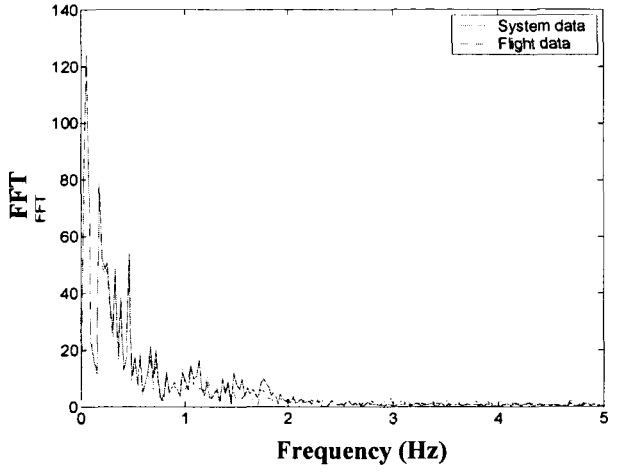


Figure 3.6– Pitch rate FFT of transducer data and aircraft system data

3.3.2 Yaw damper

In a flight with the yaw damper deactivated, the rudder Automatic Maneuvering Sequence (AMS) was engaged to study the response of the aircraft. The rudder AMS was executed with the pilot's "hands off" the transmitter to enable the aircraft's inherent Dutch Roll mode to be studied. In Figure 3.7, the roll rate, and yaw rate response are plotted together with the rudder AMS input. The results confirmed that Stingray UAV has a noticeable Dutch Roll mode.

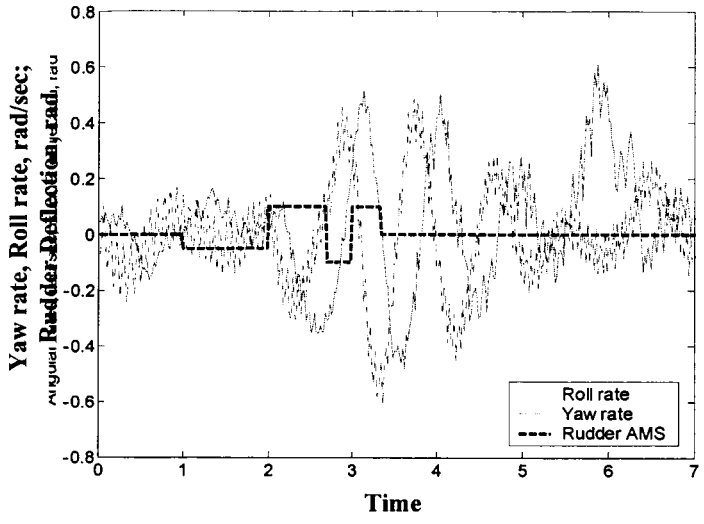


Figure 3.7 – Rudder AMS input with yaw and roll rate aircraft response

Further analysis was conducted to predict the effect of the yaw damper autopilot. The rudder AMS was run in the SIMULINK model. The yaw rate data from the simulation along with the flight yaw rate data and rudder AMS input are plotted in Figure 3.8. As shown in the figure, the yaw rate dampens faster with the addition of the yaw damper controller, as is expected.

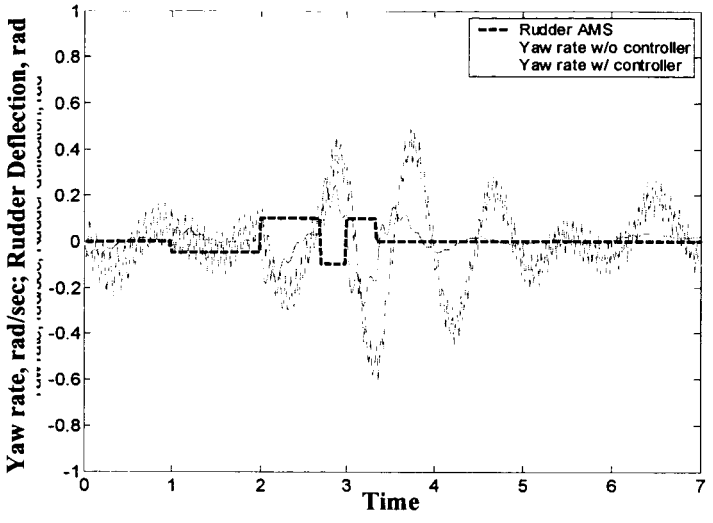


Figure 3.8 – Rudder AMS input with system yaw rate response with yaw damper

Three flights were conducted to arrive to the final design of the yaw damper. The PID yaw damper was first yaw damper test flown. The visual observation of the flight showed that the yaw damper performed well when a rudder pulse was engaged in the straight path portions of the flight. However, in the turns, the yaw damper seemed to be "fighting" the pilot input causing the aircraft to yaw out of the turn. This was observed each time the aircraft was in a turn. It was found in literature (Etkin[22]) that when the aircraft is in a steady state situation the yaw rate is not zero and the yaw damper has a tendency to command the opposite rudder deflection angle. This phenomenon is also evident in the flight data shown in Figure 3.9, where the pilot input is compared with the aircraft response. In the straight path, the response of the aircraft follows the pilot input. However, in the turns the aircraft response deviates from the pilot command. To overcome this problem, the yaw damper was redesigned to include a washout circuit. The design of this circuit is detailed in Section 2.3.3.

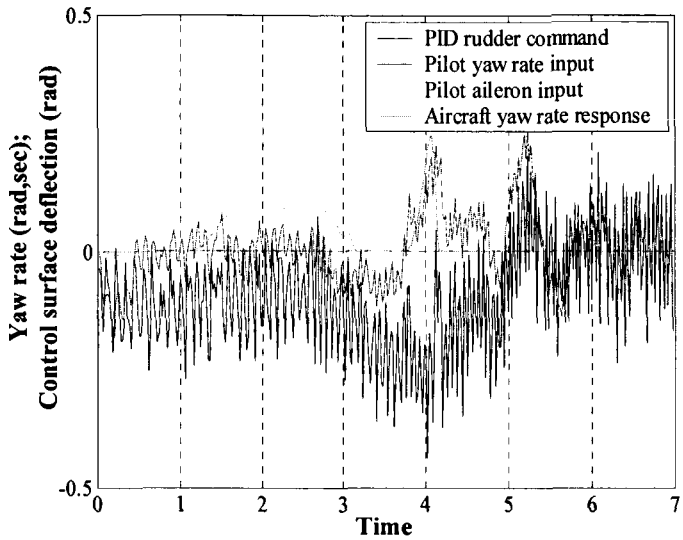


Figure 3.9 – Comparison of piloted yaw rate command and flight yaw rate data

The flight test results of the second yaw damper design (yaw damper with washout circuit) showed a net improvement in the turns when compared to the PID yaw damper. The washout circuit seemed to eliminate the steady-state effect on the yaw rate and the aircraft no longer counteracted the pilot's command. However, additional design problems emerged when the flight data was analyzed. The AMS rudder input is compared with the aircraft response, Figure 3.10. This comparison shows that the aircraft responds well to the rudder AMS, however, in the turns the yaw damper still attempts to correct the yaw rate measured by the transducer. Thus the yaw damper does not eliminate the steady-state effect in the turns. Therefore, even though the yaw damper design clearly improved the aircraft response, it was decided that a more effective yaw damper could be redesigned.

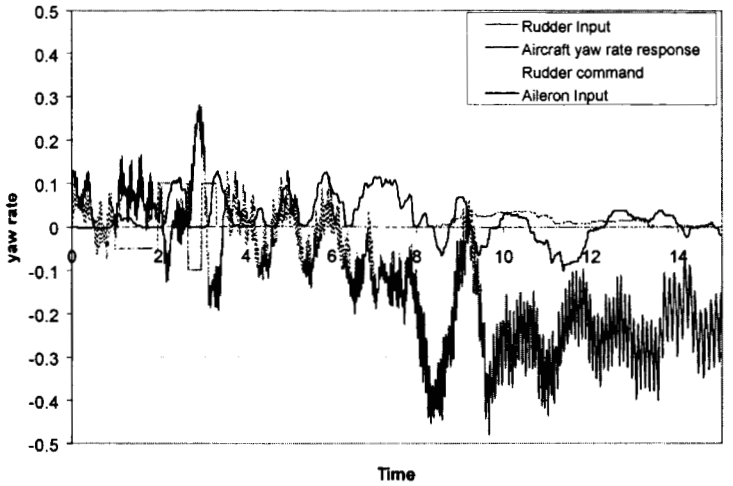


Figure 3.10 - Comparison of Rudder AMS input and aircraft yaw rate response

The flight test results of the third yaw damper showed that its design was successful. From visual observation, the aircraft responded well to the Rudder AMS and to the pilot's inputs in the turns. The analysis of the flight data showed a significant number of improvements in the dampening of the Dutch roll mode. In Figure 3.11, the pilot yaw rate input is compared to the aircraft yaw rate response measured by the angular rate transducer. The aileron input is also plotted to show when the aircraft is in a turn. The output rudder command is also plotted on the figure to show when a rudder deflection was commanded. The results show that the yaw rate response follows the rudder AMS input well and is dampened within an adequate time period. In the turns, the angular rate transducer measures a yaw rate, however, there is no rudder deflection commanded from the LIFT system unless the pilot commands a yaw rate. This shows that the yaw damper effectively filters out the steady state yaw rate in the turns.

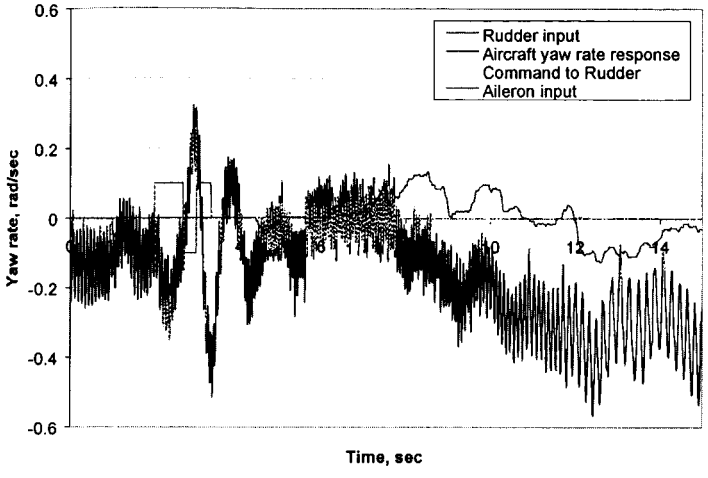


Figure 3.11 - Comparison of Rudder AMS input and aircraft yaw rate response

Next, the pilot's input was run in the simulation. The SIMULINK autopilot unit contains the five designed controllers. However, during this flight, only the PID pitch rate and yaw damper controller were engaged, therefore a few modifications were made on the model such that only these two controllers were active. The state vector was modified such that the velocity remains constant at cruise speed (102ft/sec), the altitude is constant, and the roll rate feedback comes from the aircraft transducer data. The pilot input is used as the input to the autopilot unit in SIMULINK. The pitch rate input and rudder input were directed respectively to the pitch rate controller and yaw damper. The aileron input bypassed the bank angle controller and turn coupler and was linked to the aileron servo model. After the simulation run, the yaw rate feedback from the system is then compared with the yaw rate measured by the transducer, as shown in Figure 3.12. This figure shows that the yaw rates compare very well and thus confirms that the simulation model is a good representation of the actual aircraft.

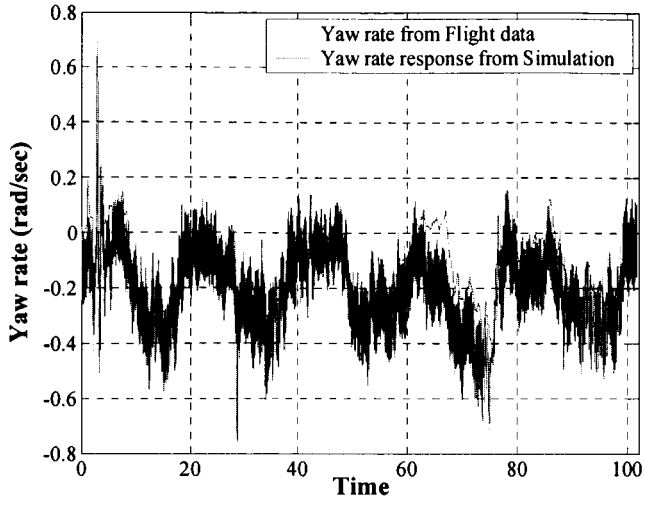


Figure 3.12 - Comparison of the yaw rate response from simulation and flight data

3.3.3 Coordinated turn flight

The bank angle PID controller and turn coupler were flown with the PID pitch rate and PID yaw damper. As described in Section 3.3.2, the PID yaw damper was not an adequate design. Therefore, the results from the first flight with the bank angle controller yielded similar results to the flight with the PID yaw damper. The visual observation during the flight was that the aircraft's altitude increased significantly while its velocity decreased causing the aircraft to approach stall. The aircraft also seemed to tend to yaw out of the turn due to the effect of the PID yaw damper.

For the coordinated turn flight, in addition to the pitch rate PID and yaw damper controllers, the bank angle controller with the turn coordinate coupler, and altitude-hold were activated. The throttle was set at 90% to insure sufficient velocity in the turn.

An aileron AMS was developed to perform a 180° turn. The aileron is mapped to a bank angle by a factor of –2.5. This factor is determined by setting the maximum aileron

deflection to a corresponding 60° bank. The AMS is run every time the controllers are engaged.

In the controller, the bank angle command was limited to 1.4 radians to avoid the situation where the bank angle is 90° causing the turn coupler equation (Equation (41)) to be undefined.

During the first coordinated turn flight, the aircraft banked heavily on the onset of the AMS causing the aircraft to turn sharply and loose velocity and altitude. The aircraft went into a stall and rolled. The aircraft was then brought down for landing. And the flight data was downloaded for analysis. Figure 3.13 shows the bank angle input plotted with the bank angle response of the aircraft. The figure indicates that there is a high overshoot in the bank angle causing the aircraft to over bank as observed in flight. Towards the end of the AMS run the aircraft starts to oscillate and then went into a roll. The bank angle data suggest that either the bank angle controller was not designed properly or there was an error in the roll model of the aircraft in the simulation.

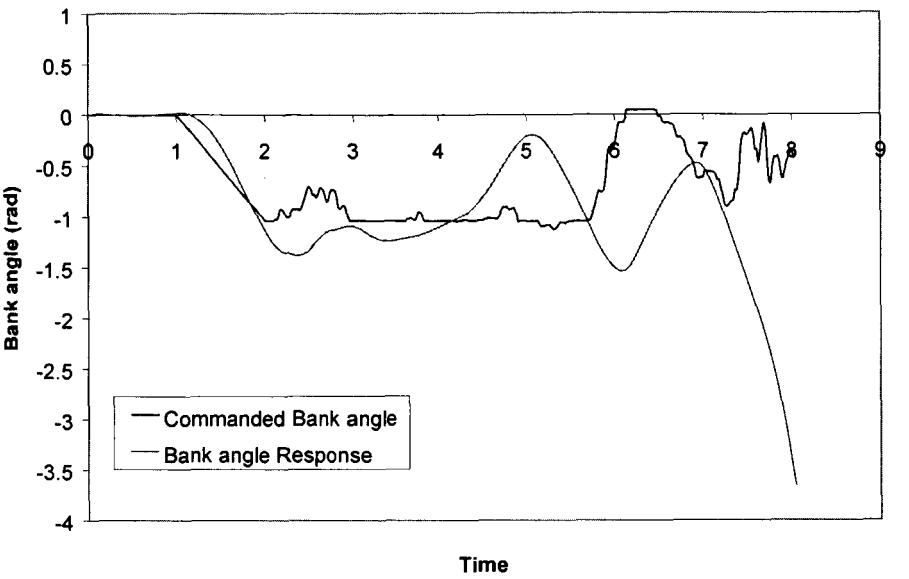


Figure 3.13 – Comparison of bank angle input and response flight data

The flight data was then compared to the simulation. To run the simulation with the flight data input, the pilot input and AMS input were implemented into the simulation. The velocity-hold was deactivated because this controller was not flown in the test flight. Therefore, the velocity data from the flight was also implemented into the simulation as the velocity feedback of the aircraft.

The comparison between the simulation and flight data yielded very interesting result about the aircraft model. The comparison of the roll rate data (Figure 3.14) shows that the slope of the flight roll rate at the onset of the turn is less steep than the slope of the simulation roll rate. This difference in the slope indicates that the aircraft roll model is different than the actual aircraft.

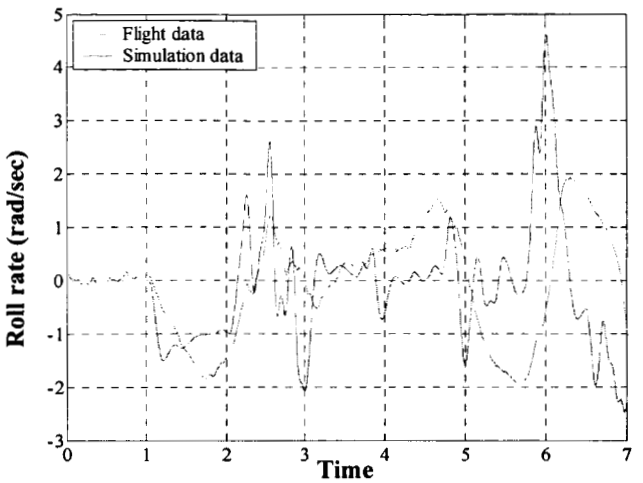


Figure 3.14 – Comparison of the flight and simulation roll rate data

PMARC is known to have difficulties in predicting forces and moments due to control surface deflections. The stability derivatives in roll were therefore studied. It was found that $C_{y\delta a}$, and $C_{n\delta a}$ appeared high. $C_{y\delta a}$ was one sixth the value of $C_{y\delta r}$. It was decided to set $C_{y\delta a}$, and $C_{n\delta a}$ to zero. C_{lp} computed by PMARC is assumed to be correct because this stability derivative is a function of the local angle of attack along the wing, which PMARC can accurately compute. The rolling moment due to aileron deflection, $C_{l\delta a}$, was recalculated using the flight data. Looking at the roll rate flight data (Figure 3.14), there is a period of steady state roll rate between the time period 5.50 and 5.76 seconds. During this time period, the data shows that the aircraft's aileron are fully deflected at 0.65 rad and the aircraft is flying at an average velocity of 65 ft/s. Knowing this information, $C_{l\delta a}$ can be calculated using the following steady state roll rate equation:

$$C_{l_{\delta a}} = -\frac{p_{ss}}{C_{lp}} \frac{b}{2V} \frac{1}{\delta a} = -0.0632 \tag{62}$$

Equation (59) yields a $C_{l\delta a}$ of –0.0632, which is significantly lower than that predicted by PMARC.

The moment of inertia, I_{xx} was also recalculated and compared with the original I_{xx} using the following equation for roll rate:

$$I_{xx} = \frac{1}{2}\rho V^2 Sb\left(C_{lp}\frac{bp}{2V} + C_{l_{\delta a}}\delta a\right)\frac{1}{dp/dt} \tag{63}$$

Using the flight data between the time period 1.10 and 1.66 seconds in the onset of the turn, I_{xx} is determined at each sampling point. The average I_{xx} over the chosen time period is 0.522 slugft^2, which corresponds well to the original value of 0.533 slugft^2. While Stingray UAV's weight has increased due to the additional equipment, the increase in weight is mainly in the centerline section of the aircraft and not outboard in the wings. Thus, the new I_{xx} value is reasonable.

The simulation was run with the new $C_{l\delta a}$, I_{xx}, $C_{y\delta a}$ and $C_{n\delta a}$ and compared to the flight data. Figure 3.15 shows the new and old simulation roll rate data with the roll rate from the flight data. It is seen that the change in stability derivative and I_{xx} clearly improves the comparison of the simulation with the flight data especially in the onset of the turn. This result also indicated a net improvement in the lateral mode representation of the aircraft.

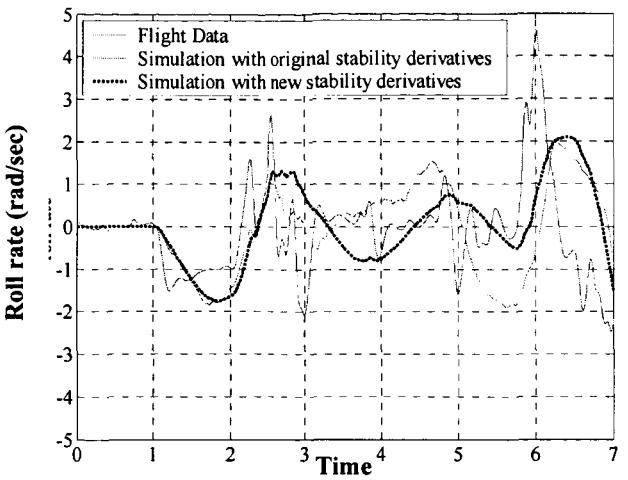


Figure 3.15 – Effect of the calculated $C_{l\delta a}$, I_{xx}, $C_{y\delta a}$ and $C_{n\delta a}$ on the roll rate response of the simulation compared with the flight data

To prepare for the next coordinated turn flight, the bank angle controller's gains were readjusted to take into account the effect of the new stability derivatives and I_{xx}. The final bank angle controller configuration is described in Section 2.3.3.

During the second flight, the same aileron AMS was used to command a 60° bank angle 180° turn. The controller was engaged twice during the flight. The first time, the aircraft banked to 60°, lost altitude in the turn, and then started to oscillate from one wing to the other in the middle of the turn. In the second turn, the pilot increased the velocity going into the turn. In this turn, the aircraft kept a steady bank angle throughout the turn while loosing some altitude, which the pilot attempted to compensate with an elevator input. The aircraft completed a 180° turn before the end of the AMS, and the pilot input aileron to roll the aircraft out of the turn. At this time, the aircraft went into a roll ratchet and then

consequently went into a roll. Figure 3.16 shows the longitude and latitude position of the aircraft in the turn.

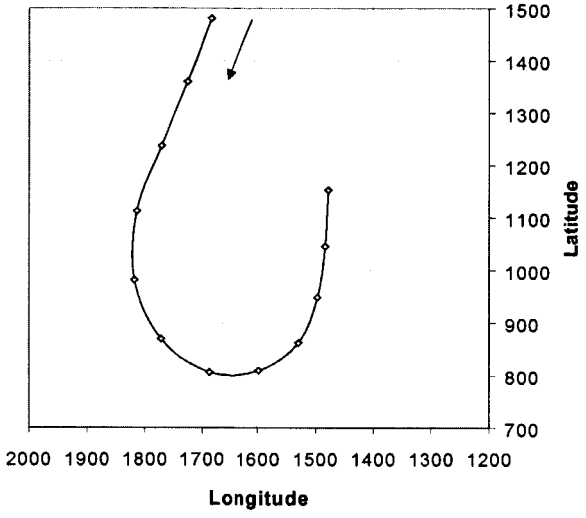


Figure 3.16 – GPS Position of the aircraft in the turn

The analysis of the flight data showed that the bank angle response compares well with the commanded bank angle input for the main portion of the turn (Figure 3.17). Around the time of 7.5 seconds, the aircraft's bank angle response starts to increasingly lag behind the bank angle input. This observed tendency is due to the decrease in airspeed causing the response to get slower.

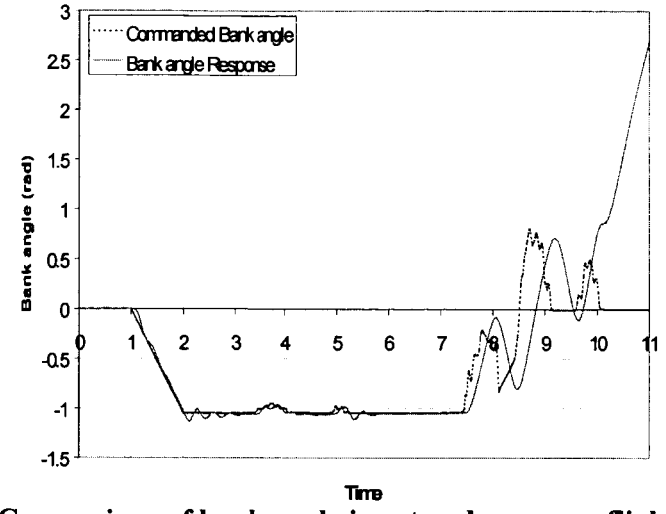


Figure 3.17 – Comparison of bank angle input and response flight data

The phenomena of roll ratcheting occurs when a system's roll dampening is too large[23] and the pilot's sudden input to the roll control system results in an oscillatory motion in roll. This motion has a high frequency between (1.8-3Hz)[23]. Figure 3.18 shows the roll rate of the aircraft in the turn. A roll oscillation is also noticed starting at time 7.5 seconds. The frequency of the oscillation, the high roll rate value, and evidence of pilot aileron input (Figure 3.17) indicate that the phenomena of roll ratcheting is evident starting at time 7.5 seconds. There is also a roll oscillation in the time intervals 1-2 seconds and 2 to 3 seconds with a frequency of 3 Hz indicating a presence of roll ratcheting.

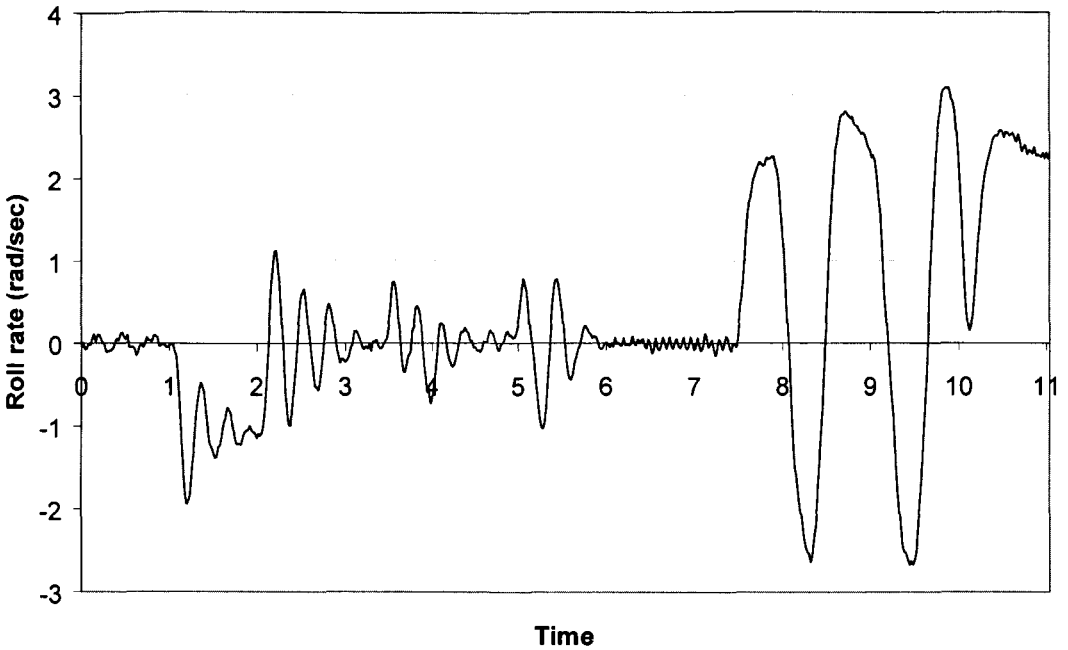


Figure 3.18 – Aircraft's roll rate data from flight

A simulation with input data of the flight was run to compare the flight data with the simulation output data. Figure 3.19 shows the roll rate data of the flight and the simulation. It is noticed that the roll data follows the same tendency as the roll rate from the flight with a slight lag and lower amplitude. Overall the figure shows that there is a net improvement in the roll model of the aircraft in the simulation, but it also suggests that the value for the roll stability derivatives are not the right value and can be improved.

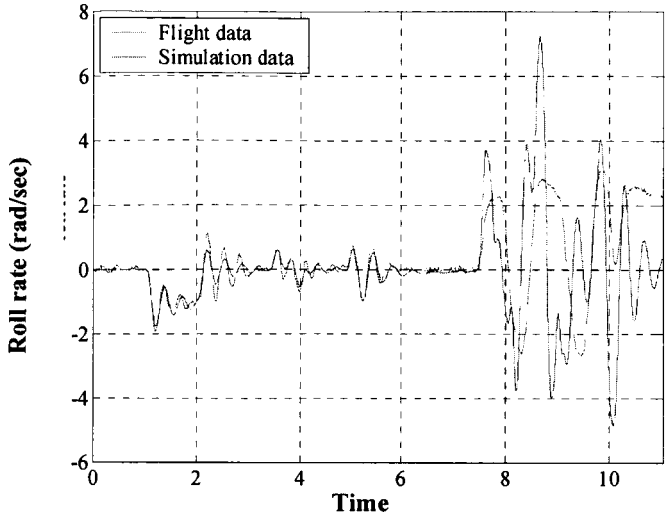


Figure 3.19 – Comparison of flight and simulation roll rate data

The bank angle responses of the flight and simulation data are compared in Figure 3.20. The figure indicates that the data compare very well for most of the turn until the start of the roll ratcheting that was not predicted by the simulation.

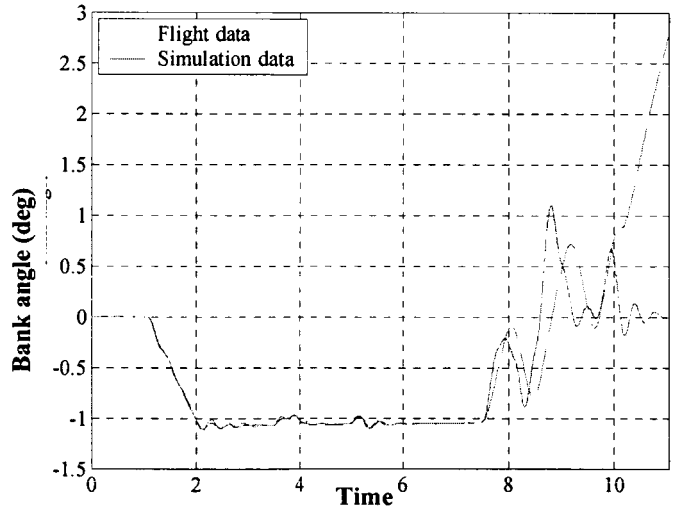


Figure 3.20 – Comparison of flight and simulation bank angle response

3.3.4 Altitude-hold

The altitude measurement during the second coordinated turn flight gave indication a measurement problem existed. Figure 3.21 illustrates the altitude reading during flight. The altitude decreases in the beginning of the turn as expected. However, the amplitude of the descent seemed higher than observed during flight. The altitude then dramatically increases from time 3 to 7 seconds. According to the measurement the aircraft ascended 130 ft in 4 seconds. This ascent was not observed in flight. Furthermore the simulation run of the flight data did not predict such an increase in altitude as shown in Figure 3.22.

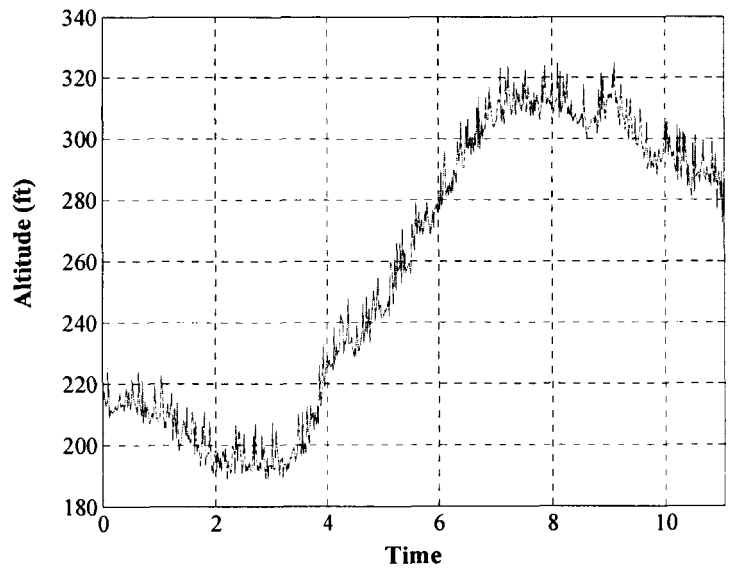


Figure 3.21 – Altitude measurement in flight

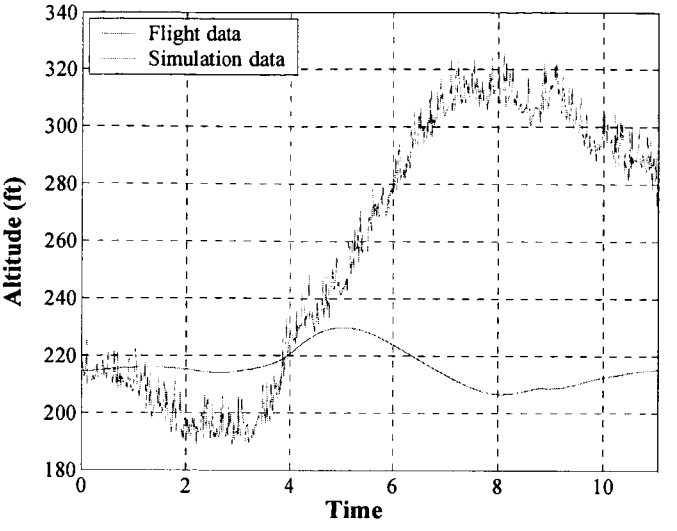


Figure 3.22 – Comparison of the flight and simulation altitude data

The altitude transducer was checked after to flight to examine if it had sustained damage. The examination showed that the transducer was in good condition. A probable cause of the measurement error is the fact that the static port lies too close to the leading edge of the wing and thus potentially located in the wing's upwash area. The 1 foot long pitot static probe is mounted directly to the leading edge of the wing instead of on an extended boom due to the aircraft's weight constraint.

4 Concluding Remarks

4.1 Summary of Results

The design of a GPS auto-navigation system for unmanned air vehicles has been completed. The navigation system is composed of an avionics system, an autopilot unit consisting of five controllers for attitude control, and a navigator that is programmed to fly through a pre-determined waypoint pattern.

Stingray UAV is equipped with the avionics system and instruments for in-flight measurements and data acquisition. The avionics system is based on the on-board computer system named Linux In Flight Testing (LIFT) system that was developed at NC State. This system is capable of hard real-time control system implementation, data processing and in-flight data storage. The efficiency and capability of the avionics system was demonstrated in flight.

The navigator is designed to guide the aircraft through a predetermined waypoints pattern. These waypoints consist of a latitude, longitude, and altitude position and are listed in an input file that is downloaded to LIFT system prior to flight.

The navigator performs the following steps. First, the navigator reads the GPS information velocity, and altitude of the aircraft and compares it to the desired waypoint. The navigator then determines whether a heading correction is necessary and then computes the desired heading correction in terms of a bank angle, velocity and altitude command to the autopilot unit.

The autopilot unit is composed of five controllers: pitch rate PID, yaw damper, bank angle PID with turn coupler, altitude-hold, and velocity-hold. The controllers were designed

in discrete time using MATLAB, SIMULINK and the Flight Dynamic and Controls (FDC) Toolbox. The design of the pitch rate PID, yaw damper, bank angle PID, and altitude-hold controllers was demonstrated in flight by visual observation and in the post-flight analysis of the flight data. The pitch rate controller proved to work very efficiently. The yaw damper was capable of efficiently dampening the Stingray UAV's inherent Dutch roll mode. The aircraft was flown with the pitch rate PID, yaw damper, bank angle PID, and altitude-hold controllers to perform a coordinated turn. The aircraft performed the turn well but exhibited a roll ratchet due to the over damping of the bank angle controller.

The FDC Toolbox describes the aircraft dynamics with twelve non-linear ordinary differential equations. This toolbox proved to be very useful in not only designing the attitude controllers but also to verify the accuracy of the aircraft dynamics model using the flight data. For example, the original roll stability derivatives were found to be incorrect when the flight data of the coordinated flight was analyzed using FDC Toolbox.

Overall, the comparison of the flight data and simulation data showed that the aircraft dynamics modeled in SIMULINK compared very well to the actual aircraft dynamics.

This research demonstrated the feasibility of designing an auto-navigation system for unmanned air vehicles. Autonomous flight opens up many applications for UAVs. An autonomous UAV can be used in applications that include pollution sensing, border and fishery surveillance, and search and rescue operations. It can also be used as a research testbed to validate designed systems such as flow control systems where flight test repeatability is an important issue.

This research also demonstrated the ability to perform high quality research on a UAV at a low cost and fast turnaround time. The availability of low-cost, light-weight, and reliable

avionics system to collect and store flight data has shown to yield quality research using UAVs.

4.2 Recommendations for future work

This research opens up many opportunities for further flight testing. The last coordinated turn flight left room for improvement in the roll model of the aircraft. The flight data can be used to find the actual roll stability derivatives.

The inaccuracy in the altitude measurement should be resolved before the next test flight. This could be achieved by mounting the Pitot static probe on a boom to increase the distance between the static port and the leading edge of the wing.

The design of a new engine model in the aircraft model should be investigated. The current engine model is represented by an approximation of the force due to the engine. Once a reliable engine model is obtained, the velocity-hold controller should be modified and test flown.

The navigator should be test flown first in-line-of sight using GPS data from previous for the waypoint coordinates. Then, the aircraft should be flown autonomously out of line of sight.

5 References

1. Lax, M., Sutherland, B., "An Extended Role for Unmanned Aerial Vehicle in the Royal Australian Air Force," APSC Paper Number 46, Australia, 1995.
2. Wong, K.C., Bil, C., "UAVs over Australia – *Market and Opportunities*," Bristol RPV/UAV Systems Conference, Bristol, UK, 30 March - 1 April 1998.
3. Insitu Press Release, October 12th 1998
4. Nilsson, C.C.A., Hall, C.E., Heinzen, S.N., Chokani, N., "GPS Auto-Navigation Design for Unmanned Air Vehicles," AIAA 2002-0389, 40th Aerospace Sciences Meeting & Exhibit, Reno NV, January 2002.
5. United States Air Force, http://www.af.mil/news/factsheets/global.html
6. McMinn, J.D., Jackson, E.B., "Autoreturn Function for a Remotely Piloted Vehicle," AIAA 2002-4673, AIAA Guidance, Navigation & Control Conference, August 2002.
7. Evans, J., Inalhan, G., Jang, J.S., Teo, R., Tomlin, C.J., "DragonFly: A Versatile UAV Platform for the Advancement of Aircraft Navigation and Control," Proc. IEEE 20th Digital Avionics System Conference, October 2001.
8. Johnson, E.N., Fontaine S.G., "Minimum Complexity Uninhabited Air vehicle Guidance and Flight Control System," Proc. IEEE 20th Digital Avionics System Conference, October 2001.
9. MAE 478/479, Senior Design group, "MAE 478 Aircraft Design Proposal Fall 1997," North Carolina State University, December 1997.
10. Unigraphics®, Electronic Data systems Corporation

11. Dale L. Ashby, Michael R. Dudley, Steve K. Iguchi, Lindsey Browne, and Joseph Katz: "Potential Flow Theory and Operation Guide for the Panel Code PMARC", NASA TM-102851, January 1991.

12. The MathWorks, Inc., Using Matlab, Natick MA, The MathWorks, Inc. 1997.

13. ANSYS *Theory Reference*, ver. 000855, Eigth, SAS IP, Inc.

14. MAE 478/479, Senior Design Group, "MAE 479 Senior Design Final Report Spring 1998," North Carolina State University, May 1998.

15. Nelson, R.C., *Flight Stability and Automatic Control*, McGraw-Hill Companies, Inc., 1998

16. Hall, C.E., "A Real-Time Linux based System for Flight Testing of Remotely Piloted Vehicles," Proc. IEEE 19th Digital Avionics System Conference, October 2000.

17. Rauw, M.O., FDC1.3 SR2 – A SIMULINK Toolbox for Flight Dynamics and Control Analysis, Haarlem, The Netherlands, June 2000.

18. The MathWorks, Inc., Using Simulink, Natick MA, The MathWorks, Inc. 2000.

19. Franklin, G.F., Powell, J.D., Emani-Naeini, A., *Feedback Control of Dynamic Systems*, Addison Wesley Publishing Company, Inc., 1995.

20. Blakelock, J.H., *Automatic control of Aircraft and Missiles*, John Wiley & Son, Inc., Canada, 1991.

21. Fontenrose, P.L., Hall, C.E., "Development and Flight Testing of Quantitative Feedback Theory Pitch Rate Stability Augmentation System," Journal of Guidance, Control, and Dynamics, Vol. 19, No.5, September-October, 1996, pp.1109-1115.

22. Etkin, B., Reid, L.D., *Dynamics of Flight stability and Control,* 3rd ed., John Wiley & Sons, Inc., New York, 1996

23. McClean, D., *Automated Flight Control Systems,* Prentice Hall International, United Kingdom, 1990.

6 Appendices

Appendix A - Stingray Characteristics

```
disp('Modified by Caroline Nilsson for Stingray UAV spec. 05/18/01');


%------------------------------------------------------------------
% The FDC toolbox - MODBUILD
% ==========================
% MODBUILD is a Matlab program, used to build a datafile with
% parameters for the aircraft model. This version of MODBUILD can
% be applied to the DHC-2 'Beaver' model (the Simulink system
% BEAVER) only, but some of the subroutines which are called are
% more general (see comment lines in MODBUILD.M).
%
% MODBUILD needs to be used every time the model parameters are
% changed. This includes changes in mass properties, since these
% are considered to be constant during the short time-intervals
% over which the aircraft responses are considered. It is not
% necessary to run MODBUILD more than once for the same set of
% model parameters if you use the 'save' option.
%
% There is only one way to change parameters, namely: by edi-
% ting the program! Since in practice, MODBUILD will not be
% applied very often, this is not really as inconvenient as it
% looks. Moreover, it is straightforward to add user-input lines
% if required. So, in other words, MODBUILD functions as a simple
% batch-file. The user is informed about the construction of the
% datafile by means of cleverly placed ECHO ON and ECHO OFF com-
% mands.
%
% Change the aircraft-dependent parts of this program if you want
% to implement models of other aircraft.
%------------------------------------------------------------------

% First, we'll save the data which is currently available in the workspace
% to a temporaty file. This is to make sure that no important data will be
% deleted during execution of MODBUILD.
% ---------------------------------------------------------------------
save modbuild.tmp
clear

% MODBUILD HEADER.
%=================
clc;
disp('The FDC toolbox - MODBUILD');
disp('==========================');
disp(' ');
disp('Build datafiles with parameters for non-linear aircraft model.');
disp(' ');
disp('Results are valid for the DHC-2 "Beaver" aircraft. Change the');
disp('program MODBUILD.M if models of other aircraft are to be used.');
disp(' ');
disp(' ');
```

```
disp('<< Press key to continue >>');
pause
clc

% DEFINE PARAMETERS FOR THE AERODYNAMIC MODEL
% ===========================================

% CURRENT CONFIGURATION: Standard nonlinear model of the DHC-2 'Beaver',
% according to [Tjee and Mulder, 1988]. The aerodynamic force and moment
% coefficients for this aircraft can be written as nonlinear polynomial
% functions of the state and input variables. This model description is
% very compact, because only one set of stability and control derivatives
% (= polynomial coefficients) is needed.
%
% For other aircraft, it is more customary to determine the aerodynamic
% forces and moments by interpolating in large multi-dimensional tables.
% Although such models use significantly more parameter matrices, it is
% really quite straightforward to replace the following matrix definitions
% by similar definitions for other aircraft.
%
% Here, the aerodynamic stability and control derivatives are stored in
% the matrix AM. For aerodynamic models, consisting of multiple tables,
% it is recommended to use variable names like AM1, AM2, etc. to maintain
% commonality with the current 'Beaver' model BEAVER.

echo on

 % Define stability and control derivatives of the DHC-2 'Beaver'
 % for the nonlinear aerodynamic model, which is valid within the
 % 35-55 m/s TAS range (see [Tjee & Mulder, 1988]).
 % ------------------------------------------------------------
 CX0  = -0.0474;  CZ0  = -0.2773;  Cm0  = 0.02185;
 CXa  = 0.0188;   CZa  = -5.1310;  Cma  = -0.4790;
 CXa2 = 0;        CZa3 = 0;        Cma2 = 0;
 CXa3 = 0;        CZq  = -8.965;   Cmq  = -12.622;
 CXq  = 0;        CZde = -0.5550;  Cmde = -1.6535;
 CXdr = 0;        CZdeb2 = 0;      Cmb2 = 0;
 CXdf = 0;        CZdf = 0;        Cmr  = 0;
 CXadf = 0;       CZadf = 0;       Cmdf = 0;
 CXadot= 0;       CZadot = -0.4203; Cmadot =-1.2523;


 CY0  = 0;        Cl0  = 0;        Cn0  = 0;
 CYb  = -0.311;   Clb  = -0.0850;  Cnb  = 0.0720;
 CYp  = -0.0545;  Clp  = -0.4500;  Cnp  = -0.0340;
 CYr  = 0.2420;   Clr  = 0.0930;   Cnr  = -0.0890;
 CYda = 0.0000;   Clda = -0.0632;  Cnda = 0.00;
 CYdr = 0.1580;   Cldr = 0.006000; Cndr = -0.0680;
 CYdra = 0;       Cldaa = 0;       Cnq  = 0;
 CYbdot= 0;                        Cnb3 = 0;

%old Clda = -.345
echo off
```

```
AM = [ CX0    CY0    CZ0    Cl0    Cm0    Cn0 ;
    CXa   0     CZa   0     Cma   0   ;
    CXa2  0     0     0     Cma2  0   ;
    CXa3  0     CZa3  0     0     0   ;
    0     CYb   0     Clb   0     Cnb ;
    0     0     0     0     Cmb2  0   ;
    0     0     0     0     0     Cnb3;
    0     CYp   0     Clp   0     Cnp ;
    CXq   0     CZq   0     Cmq   Cnq ;
    0     CYr   0     Clr   Cmr   Cnr ;
    0     0     CZde  0     Cmde  0   ;
    CXdf  0     CZdf  0     Cmdf  0   ;
    0     CYda  0     Clda  0     Cnda;
    CXdr  CYdr  0     Cldr  0     Cndr;
    CXadf 0     CZadf 0     0     0   ;
    0     CYdra 0     0     0     0   ;
    0     0     0     Cldaa 0     0   ;
    0     0     CZdeb2 0    0     0   ;
    0     CYbdot 0    0     0     0   ;
    0     0     CZadot 0    Cmadot 0  ];


AM = AM';

disp(' ');
disp('<< Press key to continue >>');
pause
clc


% DEFINE PARAMETERS FOR ENGINE FORCES & MOMENTS MODEL.
% ===================================================

% CURRENT CONFIGURATION: Standard nonlinear model of the DHC-2 'Beaver',
% according to [Tjee and Mulder, 1988].
%
% The engine forces and moments model of the DHC-2 'Beaver' also expresses
% the force and moment coefficients as nonlinear polynomial functions of
% external inputs and state variables.
%
% Here, the engine stability and control derivatives are stored in the
% matrix EM. For engine models, consisting of multiple tables, it is re-
% commended to use variable names like EM1, EM2, etc. to maintain com-
% monality with the current 'Beaver' model BEAVER.

echo on

 % The nonlinear engine model of the DHC-2 "BEAVER" aircraft
 % valid within the 35-55 m/sec TAS-range (see [Tjee & Mulder, 1988]).
 % ------------------------------------------------------------------
 CXdpt  = 0.04511 %0.06138 %0.1161
 CXadpt2 = 0;  %.1453
 CZdpt  = 0;  %-.1563
 Cla2dpt = 0;  %-0.01406
```

```
 Cmdpt  = 0;  %-0.07895
 Cndpt3 = 0;  %-0.003026

echo off

EM = [ CXdpt   0     CZdpt   0     Cmdpt   0    ;
    0     0     0     0     0     Cndpt3 ;
    CXadpt2 0     0     0     0     0     ;
    0     0     0     Cla2dpt 0     0    ];

EM = EM';

disp(' ');
disp('<< Press key to continue >>');
pause
clc


% DEFINITION OF RELEVANT AIRCRAFT GEOMETRY AND MASS-DISTRIBUTION PARAMETERS.
%
% (Note: geometry and mass properties are assumed to be constant during the
% motions of interest! If it is required to compute the influence of changes
% in these properties on-line during the simulation, many blocks in the sys-
% tems BEAVER or BEAVER1 have to be changed. First of all, a block which
% computes the geometrical and/or mass properties must be added to the sys-
% tem. Then use a text editor to find the blocks in which the parameter ma-
% trices GM1 and GM2 are used and replace these parameters by variables.
% Add corresponding Inport blocks to these blocks, and make the proper
% connections to the block in which the geometry and mass properties are
% calculated.)
% ================================================================================

echo on

 % Aircraft data on which the aerodynamic model is based. CHANGE THE
 % VARIABLES ACCORDING TO YOUR OWN WISHES BY EDITING MODBUILD.
 % ------------------------------------------------------------------

 % Mass and mass-distribution.
 % --------------------------
 Ix    = 0.522;          % kgm^2 in Fr slug*ft^2
 Iy    = 0.699;          % kgm^2 in Fr slug*ft^2
 Iz    = 0.980;          % kgm^2 in Fr slug*ft^2
 Jxy   =   0.0;          % kgm^2 in Fr slug*ft^2
 Jxz   =  0.021;          % kgm^2 in Fr slug*ft^2
 Jyz   =   0.0;          % kgm^2 in Fr slug*ft^2
 m     = 24/32.17405;          % kg       slug

 % geometric data.
 % ---------------
 cbar  =   1;            % m   ft
 b     =  77/12;          % m   ft
 S     =  6.33;          % m^2 ft^2
```

```
echo off

% THE FOLLOWING EQUATIONS ARE VALID FOR ANY RIGID AIRCRAFT
% (ALSO VALID FOR NON-SYMMETRIC AIRCRAFT).
%
% Calculate inertia parameters (see NASA TP 2768). The formula's
% are valid for symmetric and asymmetric aircraft.
% ------------------------------------------------------------
detI = Ix*Iy*Iz - 2*Jxy*Jxz*Jyz - Ix*Jyz^2 - Iy*Jxz^2 - Iz*Jxy^2;
I1   = Iy*Iz - Jyz^2;
I2   = Jxy*Iz + Jyz*Jxz;
I3   = Jxy*Jyz + Iy*Jxz;
I4   = Ix*Iz - Jxz^2;
I5   = Ix*Jyz + Jxy*Jxz;
I6   = Ix*Iy - Jxy^2;

Pl  = I1/detI; Pm = I2/detI; Pn = I3/detI;
Ppp = -(Jxz*I2 - Jxy*I3)/detI;
Ppq = (Jxz*I1 - Jyz*I2 - (Iy-Ix)*I3)/detI;
Ppr = -(Jxy*I1 + (Ix-Iz)*I2 - Jyz*I3)/detI;
Pqq = (Jyz*I1 - Jxy*I3)/detI;
Pqr = -((Iz-Iy)*I1 - Jxy*I2 + Jxz*I3)/detI;
Prr = -(Jyz*I1 - Jxz*I2)/detI;

Ql  = I2/detI;
Qm = I4/detI;
Qn = I5/detI;
Qpp = -(Jxz*I4 - Jxy*I5)/detI;
Qpq = (Jxz*I2 - Jyz*I4 - (Iy-Ix)*I5)/detI;
Qpr = -(Jxy*I2 + (Ix-Iz)*I4 - Jyz*I5)/detI;
Qqq = (Jyz*I2 - Jxy*I5)/detI;
Qqr = -((Iz-Iy)*I2 - Jxy*I4 + Jxz*I5)/detI;
Qrr = -(Jyz*I2 - Jxz*I4)/detI;

Rl  = I3/detI; Rm = I5/detI; Rn = I6/detI;
Rpp = -(Jxz*I5 - Jxy*I6)/detI;
Rpq = (Jxz*I3 - Jyz*I5 - (Iy-Ix)*I6)/detI;
Rpr = -(Jxy*I3 + (Ix-Iz)*I5 - Jyz*I6)/detI;
Rqq = (Jyz*I3 - Jxy*I6)/detI;
Rqr = -((Iz-Iy)*I3 - Jxy*I5 + Jxz*I6)/detI;
Rrr = -(Jyz*I3 - Jxz*I5)/detI;

% Summarizing results in aircraft parameter matrices GM1 and GM2.
% ------------------------------------------------------------
GM1 = [cbar b   S   Ix  Iy  Iz  Jxy Jxz Jyz m];

GM2 = [ Pl  Pm  Pn  Ppp Ppq Ppr Pqq Pqr Prr ;
     Ql  Qm  Qn  Qpp Qpq Qpr Qqq Qqr Qrr ;
     Rl  Rm  Rn  Rpp Rpq Rpr Rqq Rqr Rrr ];

disp(' ');
disp('<< Press key to continue >>');
pause
clc
```

```
% Save the aircraft parameters to file. The default folder for this
% action is the subfolder DATA of the FDC toolbox, which is put in
% the variable defdir. The current directory is stored in the
% variable currentdir, which is used if the default directory can't
% be found.
% ================================================================

disp('Hello!');
defdir = datadir;
currentdir = chdir;
disp('Hello!');
% Go to default directory if that directory exists (if not, start
% save-routine from the current directory).
% ----------------------------------------------------------------
eval(['chdir ',defdir,';'],['chdir ',currentdir,';']);

% Obtain path (use default filename AIRCRAFT.DAT).
% ------------------------------------------------
[filename,dirname] = uiputfile('aircraft.dat','Save aircraft model parameters');

% Build string variable which specifies what to save in which file.
% ----------------------------------------------------------------
savecmmnd=['save ',dirname,filename,' AM EM GM1 GM2'];

% Evaluate the save command-string.
% ---------------------------------
eval(savecmmnd);


% clear used variables
% --------------------
clear savecmmnd defdir currentdir

% CONCLUDING REMARKS.
%====================
clc
disp('Ready. The data can now be loaded into the Matlab workspace');
disp('by typing: LOAD AIRCRAFT.DAT -MAT');
disp(' ');
disp('Aerodynamic data is contained in the matrix AM, engine data is');
disp('contained in EM, aircraft geometry, mass, and mass distribution');
disp('data is contained in GM1 and GM2.');
disp(' ');
disp('You may also use the macro LOADER.M to load the data matrices');
disp('in the Matlab workspace. MODBUILD doesn"t have to be used again,');
disp('unless you want to make changes to the model parameters. In that');
disp('case, MODBUILD.M needs to be edited.');
disp(' ');

% Clear workspace, retrieve variables from temporary MAT-file and delete
% temporary MAT-file. Workspace will now contain the same variables as
% before running MODBUILD.
```

```
% ------------------------------------------------------------------
clear
load modbuild.tmp -mat
delete('modbuild.tmp');


% ------------------------------------------------------------
% References:
%
% R.T.H. Tjee and J.A. Mulder. Stability and Control Deriva-
% tives of the De Havilland DHC-2 "Beaver" aircraft. Report
% LR-556, Delft University of Technology, 1988.
%
% Duke, E.L., Patterson, B.P. and Antoniewicz, R.F. User's
% manual for LINEAR, a Fortran Program to derive Linear Air-
% craft Models. NASA TP 2768, 1987.
%
% Ruijgrok: Elements of Airplane Performance, Delft University
% Press, DUT (L&R), 1990.
%
% Rauw, M.O.: A Simulink environment for Aircraft Dynamics and
% Control analysis - Application to the DHC-2 'Beaver',
% Graduate's thesis, DUT (L&R), 1993.
%
% Rauw, M.O.: FDC 1.2 - A Simulink Toolbox for Flight Dynamics
% and Control Analysis, 1997.
% ------------------------------------------------------------


%-----------------------------------------------------
% The FDC toolbox. Copyright Marc Rauw, 1994-2000.
% Last revision of this program: June 12, 2000. (SR2 fix)
kadjksadjadjadj
```

Appendix B - Linear State-Space Matrices

Longitudinal mode

$$A_{long} = \begin{bmatrix} -0.10 & 2.18 & -32.2 & 0 \\ -0.0062 & -5.33 & 0 & 0.957 \\ 0 & 0 & 0 & 1 \\ 0.0043 & -49.85 & 0 & -7.57 \end{bmatrix}$$

$$B_{long} = \begin{bmatrix} 0 \\ -0.57 \\ 0 \\ -184.3 \end{bmatrix}$$

$$C_{long} = [0 \quad 0 \quad 0 \quad 1]$$

$$D_{long} = [0]$$

Servo

$$A_{servo} = \begin{bmatrix} -39.13 & -913.25 \\ 1 & 0 \end{bmatrix}$$

$$B_{servo} = \begin{bmatrix} 913.25 \\ 0 \end{bmatrix}$$

$$C_{servo} = [0 \quad 1]$$

$$D_{servo} = [0]$$

Lateral mode

$$A_{lat} = \begin{bmatrix} -0.3194 & -0.992 & 0.316 & -0.002 \\ 35.13 & -1.37 & 0 & -0.83 \\ 0 & 0 & 0 & 1 \\ -78.6 & 2.70 & 0 & -13.34 \end{bmatrix}$$

$$B_{lat} = \begin{bmatrix} 0.18 \\ -34.67 \\ 0 \\ 4.27 \end{bmatrix}$$

$$C_{lat} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

$$D_{lat} = [0]$$

Appendix C - Autopilot Control Law

Pitch rate control law

```
***Control law for Pitch rate
**Coefficients
Kp = -0.025
Ki = -2.15
Kd = -.0095
K = -4
tau = .02

K_1 = 1/Mde*(Kp+Kd/tau+Ki*tau)*Mtau*Ma2_d
K_2 = 1/Mde*(Ki*tau-Kd/tau)*Mtau*Ma2_d
K_3 = 1/Mde*Ki*tau*Mtau*Ma2_d

*** Subroutine delta_elevator_out

delta_elev_out = K_1*Q_INT + K_2*Q_INT_OLD + K_3*SUMQ_INT_OLD +
                 K*delta_elev
SUMQ_INT_OLD_ERROR = SUMQ_INT_OLD + Q_INT_OLD
 Q_INT_OLD = Q_INT
```

Yaw damper control law

```
For rt_data4.h file

/* Yaw Rate damper */

Krr = -.25
a_r = .9733
b_r = 1

------------------------------------------------------------
For control_test4.c file


/*
* Yaw rate damper
*/


Y_INT = ( Mr_xad) / Mdelta_r * ( R_INT - b_r * R_INT_OLD )
                  + a_r *Y_INT_OLD
```

```
delta_rudder_out = (s16)(int) (delta_rudder * Krr +
                    r_coord / Mdelta_r ) +
                    Y_INT

R_INT_OLD = R_INT
Y_INT_OLD = Y_INT

/*
* End Yaw damper
*/
```

Bank angle controller control law

```
***Control law for bank_angle

**Coefficients
Kp = -4
Ki = -2
Kd = -.1
K = -2.4933
tau = .02

K_1_p = ( 1./Mdelta_p ) * ( Kp_p + Kd_p/Tau + Ki_p*Tau) * Mpxd * Mpa2d;
K_2_p = ( 1./Mdelta_p ) * ( Ki_p*Tau - Kd_p/Tau) * Mpxd * Mpa2d;
K_3_p = (1./Mdelta_p ) * Ki_p*Tau*Mpxd * Mpa2d;

Mp_err_int = K * Mdelta_aileron / (Mpxd * Mpa2d );

*** Subroutine delta_aileron_out

P_ERR_INT = (int) (Mp_err_int * delta_aileron) - P_INT;

delta_aileron_out = (s16)( K_1_p * (float)P_ERR_INT +
                    K_2_p * (float)P_ERR_INT_OLD +
                    K_3_p * (float)SUM_P_ERR_INT );

SUM_P_ERR_INT = P_ERR_INT_OLD;
P_ERR_INT_OLD = P_ERR_INT;
```

Altitude-hold control law

```
***Control law for altitude hold

**Coefficients
Kp1_alt = -0.015
Kp2_alt = 2
Tau = .02

*** Subroutine alt_delta_elevator_out

ALT_ERR =  ALT_measured - ALT_NAV;

Theta_ERR = ( Kp1_alt * ALT_ERR ) - Theta;

Q_INPUT_ALT = Theta_ERR * Kp2_alt;

Q_ERR_INT = (int) (Mq_err_int * delta_elevator +
            q_coord / Mq_xad +
            Q_INPUT_ALT / Mq_xad) -
            Q_INT;
```

Velocity-hold control law

```
***Control law for Velocity hold
**Coefficients
Kp_vel = .05
Ki_vel = .0009
Kd_vel = .03
Tau = .02

K_1_p = ( 1./Mdelta_t ) * ( Kp_vel + Kd_vel/Tau + Ki_vel*Tau);
K_2_p = ( 1./Mdelta_t ) * ( Ki_vel*Tau - Kd_vel/Tau);
K_3_p = (1./Mdelta_t ) * Ki_vel*Tau;

Mp_err_int = K * Mdelta_aileron / (Mpxd * Mpa2d );

*** Subroutine delta_aileron_out
VEL_NAV = 102 (or set when controller is engaged)
VEL_COORD = ((15 * Phi) / PI) + VEL_NAV
Velocity_ERR = VEL_COORD - velocity;

delta_throttle_1 = (s16)( K_1_p * Velocity_ERR +
                   K_2_p * Velocity_ERR_OLD +
```

```
                K_3_p * Velocity_ERR_INTEGRATE);

Velocity_ERR_INTEGRATE += Velocity_ERR_OLD;
Velocity_ERR_OLD = Velocity_ERR_;
```

Appendix D – Navigator Algorithm

- **Parameter setting:**
  - Read-in way point:
    - Waypt 1.
    - Waypt 2.
    - Waypt 3.
    - Waypt 4.
  - Equator Radius: a
    - $a = 6378137$ in meters
  - Polar Radius: b
    - $b = 6356752.314$ in meters
  - Ramp input slope: c
    - $c = \pi/3$
- **GPS Input:**
  - READ present latitude, longitude, heading
  - READ velocity (V) , altitude (h) from Pitot static tube

- **Minimum turning radius Calculation**
  - $R_o = \dfrac{V^2}{g\tan(\pi/3)}$

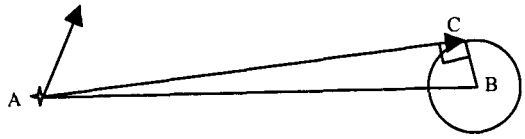
- **Subroutine for GPS pt conversion**
  - Degrees and minute to Degrees
    - Alatdeg = Latdeg. + Latmin./60
    - Alongdeg = Longdeg. + longmin./60
  - Latitude and Longitude Conversion factors:
    - $Flat = \dfrac{\pi}{180}\left(\dfrac{a^2b^2}{\left(a^2\cos^2(Latitude)+b^2\sin^2(Latitude)\right)^{3/2}}+h\right)\cdot\dfrac{1}{.3048}$
    - $Flon = \dfrac{\pi}{180}\left(\dfrac{a^2}{\sqrt{a^2\cos^2(Latitude)+b^2\sin^2(Latitude)}}+h\right)\cos(Latitude)\cdot\dfrac{1}{.3048}$

    where h is in meters, Flon=ft/deg. , Flat = ft/deg

End of subroutine

Every GPS Pt and Waypoint have to go through the above subroutine before being used in calculations.; i.e. GPS point A (latitude in minute and degrees, longitude in minute and degrees) becomes A(Alatdeg,Alongdeg) in deg and Flat and Flon in ft/deg.

In the next section, A is the present position of the aircraft, B is the desired waypoint, and C is point on the capture circle around B that the aircraft can reach with minimum heading correction.



- **Waypoint capture determination**
  - CALCULATE distance AB from present pt (A) to way point (B)
    - A(Alatdeg, Alongdeg) and B(Blatdeg,Blongdeg)
    - $|AB| = \sqrt{(Flat(Blat\,\text{deg} - Alat\,\text{deg}))^2 + (Flon(Blong\,\text{deg} - Along\,\text{deg}))^2}$
    - If |AB| < or = Ro, Way pt is reached, go to next way point

- **Subroutine: Heading Calculation**

  Want to find heading using points A(Alatdeg,Alongdeg) and B(Blatdeg,Blongdeg)

  If Alongdeg = Blongdeg then HeadingAB = 360° or 180°
  If Alongdeg < Blongdeg then

$$HeadingAB = 90^\circ + \frac{180}{\pi}\arctan\left(\frac{Flat(Blat\,\text{deg} - Alat\,\text{deg})}{Flon(Blong\,\text{deg} - Along\,\text{deg})}\right)$$

  If Alongdeg > Blongdeg then

$$HeadingAB = 270^\circ + \frac{180}{\pi}\arctan\left(\frac{Flat(Blat\,\text{deg} - Alat\,\text{deg})}{Flon(Blong\,\text{deg} - Along\,\text{deg})}\right)$$

  End of Subroutine for heading calculation

- **Determine angle $\Delta\psi$**

$\Delta\psi$ is defined as the angle $\angle CAB$

$$\Delta\psi = \sin^{-1}\left(\frac{R_o}{|AB|}\right) \qquad \text{in rad}$$

- **Heading Error Calculation**

Need to calculate heading error by comparing present heading of the aircraft to heading desired (point A to Point B)

  - Use heading calc subroutine to obtain Heading AB.
  - Use the following heading error subroutine

$$\beta_1 = HeadingAB - HeadingA$$

If $|\beta_1| < 180$ then

$\Psi_{err} = \beta_1$ convert to rad.

Else

$\Psi_{err} = 360 - |\beta_1|$ convert to rad.

End if

- **Heading Correction Calculation**

If $\Delta\Psi \geq \frac{\Psi_{err}}{2}$ then

no heading correction needed, READ next GPS input

Else $\Psi_{corr} = \Psi_{err}$

- **For intermediate point B': Calculate angle ABB'**
  - A(Alatdeg, Alongdeg), B(Blatdeg,Blongdeg), B'(B'latdeg,B'longdeg)
  - $$|BB'| = \sqrt{(Flat(B'lat\deg - Blat\deg))^2 + Flon((B'long\deg - Blong\deg))^2}$$
  - $$\overrightarrow{AB} = (Flat(Blat\deg - Alat\deg))i + (Flon(Blong\deg - Along\deg))j$$
  - $$\overrightarrow{BB'} = (Flat(B'lat\deg - Blat\deg))i + (Flon(B'long\deg - Blong\deg))j$$
  - $$\angle ABB' = \cos^{-1}\left(\frac{\overrightarrow{AB}\cdot\overrightarrow{BB'}}{|AB||BB'|}\right)$$
  - If $\angle ABB' \leq 90^\circ$, Way pt is reached, go to next way pt

- **Heading correction command calculation:**

- **Calculate** Ψ_1 (Heading Correction due to 1sec. ramping to 60deg. bank angle):
  - $\Psi_1 = \frac{-g\ln(\cos(c))}{cV}$ in rad
- **Comparison** of Ψ_1 and Ψerr/2:
  - If $\Psi_1 \leq \Psi$err/2:
    - Calculate t_1 and ϕ_{max} for $\Psi_1 = \Psi$err/2
      - $t_1 = \frac{1}{c}\cos^{-1}\left(\exp\left(\frac{-\Psi_{err}}{2}\frac{Vc}{g}\right)\right)$
      - $\phi_{max} = ct_1$ (in radians)
    - Calculate t_f
      - $t_f = 2t_1$
    - set AMS matrix
      - in = [0 t_1 t_f;
        0 ϕ_{max}*180/π 0];
  - If $\Psi_1 > \Psi$err/2:
    - Calculate Ψ_2 and t_2
      - $\Psi_2 = \Psi_{err} - 2\Psi_1$
      - $t_2 = \Psi_2 \frac{V}{g\tan(\pi/3)}$
    - Calculate t_f:
      - $t_f = 2t_1 + t_2$
    - set AMS matrix
      - in = [0 t_1 t_1 t_f;
        0 60 60 0];

- **Output to autopilot unit**
  - **Bank angle AMS input**

```
T = 0.02;                          % sampling time
m = 0.0004;                        % calibration factor for aileron
channel   = 5;                     % channel for aileron
filename  = 'test_in.txt';

min_time = min(in(1,:));
max_time = max(in(1,:));
t = min_time : T : max_time;       % sets up a row vector with
                                   the time steps of the flight
                                   computer
```

```
delta_in = interp1(in(1,:),in(2,:),t);      % interpolation of the values
                                              to the flight computer time steps
delta_in = delta_in / m;                    % conversion to the values, the
                                              flight computer needs as
                                              input
y = round(delta_in);                        % conversion to integer values
[dummy,length] = size(delta_in);            % size of the vector is written to
                                              the variable 'length'
totallength = length + round(1/T);

fid = fopen(filename,'w');                  % write the values to a text file
                                              'test_in.txt'
for c = 1 : length,
  fprintf(fid,'%08i\n',y(c));
end
for c = 1 : round(1/T),                     % adds an additional second of zeros
  fprintf(fid,'%08i\n',0);
end
```

- **Velocity**
  - V = 102 ft/sec or measured when controller is engaged
- **Altitude**
  - From way points, or assuming cruising altitude of 250ft.

Appendix E -Stingray Check lists

STINGRAY PACKING CHECKLIST

Preparation Date:
Packing Date:
Flight Date:

Departure Time:
Arrival Time:
Flight No:

STINGRAY:
- ❑ Plugs in inlets/exhaust
- ❑ Clips on control surfaces

TOOLS:

TAPE:
- ❑ Duck tape
- ❑ Clear tape
- ❑ Electrical tape

BALL DRIVERS:
- ❑ 2-56 (flap/aileron hatch)
- ❑ ¼-20 (landing gear)
- ❑ Muffler/engine Allen Wrench

SCREW DRIVERS:
- ❑ Small, medium, large flat head
- ❑ Small, medium, large Phillips head

PLIERS:
- ❑ Needle-nose
- ❑ Adjustable wrench
- ❑ Glow plug wrench
- ❑ Scissors

ENGINE:
- ❑ Fuel-in can with pump
- ❑ Electric torque starter
- ❑ Starter battery
- ❑ External glow plug battery
- ❑ Fuel tubing
- ❑ Extra propellers

LANDING GEAR:
- ❑ Bicycle pump
- ❑ Brake pistons (2)
- ❑ Brake lines
- ❑ 4-40 retention screws

BATTERIES:
- ❑ Servo (1)
- ❑ Avionics (2)

REPAIR EQUIPMENT:
- ❑ 5 min. epoxy
- ❑ CA
- ❑ Mixing sticks
- ❑ Exacto knife/blades
- ❑ Light ply scraps

MISCEALLENOUS:
- ❑ Control surface protractors
- ❑ Extra skids
- ❑ Extra hatch screws
- ❑ Extra empennage nylon bolt
- ❑ Extra hinges
- ❑ Extra clevises
- ❑ Extra control horns
- ❑ Extra Velcro straps
- ❑ Extra foam rubber
- ❑ Flashlight

TRANSMITTER:
- ❑ Primary TX
- ❑ Secondary TX
- ❑ External amplifier battery and case
- ❑ Neck Strap
- ❑ JRPCM 10SX manual
- ❑ TX program Hard Copy

GROUND VIDEO:
- ❑ Video Camera
- ❑ Batteries
- ❑ Tapes
- ❑ Digital Camera

- ❑ C-clams
- ❑ Floppy Disk

MISC:

- ❑ Digital multi-meter
- ❑ Electronic scale
- ❑ Stop watches (2)
- ❑ Clip-board, pen
- ❑ Windex
- ❑ Trash bags
- ❑ Fire Extinguisher
- ❑ First Aid kit
- ❑ Hearing protectors

STINGRAY CHECKLIST – DAY PRIOR TO FLIGHT

Preparation Date:
Packing Date:
Flight Date:

Departure Time:
Arrival Time:
Flight No:

- ❑ Put vertical on plane
- ❑ Fill brake air cylinder
- ❑ General airframe inspection
- ❑ Attach wing onto plane
- ❑ Place internal components
- ❑ Visually check avionics and servo wires connection
- ❑ Tape over holes and hatch screws
- ❑ Remove foam from flight surfaces
- ❑ Attach transmitter antenna
- ❑ Turn on transmitter
- ❑ Verify transmitter program
- ❑ Turn on receiver
- ❑ Turn on servos
- ❑ Perform operation check on all servos
- ❑ Turn off servos
- ❑ Turn off receiver
- ❑ Turn off transmitter
- ❑ Remove antenna from transmitter
- ❑ Place hatches on Aircraft
- ❑ Place C.G. rig on aircraft
- ❑ Balance and weight aircraft; W=
- ❑ Remove C.G. rig
- ❑ Open hatch
- ❑ Remove wing
- ❑ Remove batteries (servo, avionics)

<u>CHARGE BATTERIES:</u>

- ❑ Avionics battery
- ❑ Servo battery
- ❑ Starter battery
- ❑ Transmitter
- ❑ Glow-driver battery
- ❑ Video battery

STINGRAY FLIGHT-LINE PROCEDURE

Flight Date:
Overcast:

Flight No:
Wind:

- ❑ Remove plugs from inlets
- ❑ Remove plugs from exhaust
- ❑ Remove quick access hatch
- ❑ Inspect aircraft interior
- ❑ Check security of airspeed probe and bird
- ❑ Inspect all pushrods and linkages
- ❑ Check hinges
- ❑ Check clevis retainers
- ❑ Remove foam from flight surfaces
- ❑ Attach wings
- ❑ Tape Wing seam
- ❑ Fuel Aircraft
- ❑ Place GPS antenna outside aircraft
- ❑ Attach transmitter antenna
- ❑ Turn on transmitter
- ❑ Verify transmitter program
- ❑ Turn on receiver
- ❑ Turn on Avionics
- ❑ Link with Laptop, activate LIFT system
- ❑ Turn on servos
- ❑ Perform operation check on all servos
- ❑ Transmitter Range Check
- ❑ Pump in air for brake
- ❑ Make sure, have satellite contact
- ❑ Tape GPS antenna to the front hatch
- ❑ Close the pressure tube circuit on altitude pressure transducer
- ❑ Close all hatches
- ❑ Bring aircraft to Runway
- ❑ Check that glow driver is off of engine
- ❑ Prime engine
- ❑ Start glow driver
- ❑ Start engine
- ❑ Start stopwatch
- ❑ Clear takeoff area of all personal
- ❑ Clear runway

POST FLIGHT:

- ❑ Shut off glow driver
- ❑ Open First hatch
- ❑ Open second hatch
- ❑ Data retrieval:
- ❑ Shut off servos
- ❑ Shut off receiver
- ❑ Shut off avionics
- ❑ Shut off transmitter
- ❑ Empty fuel tank
- ❑ Clean plane with Windex
- ❑ Detach wings
- ❑ Pack all items and tool

Appendix F - Stingray Flight log

STINGRAY IN-FLIGHT NOTES

Flight No: 2 GPS FLIGHT I
Initial Conditions: Mid-rate on all control surfaces.

Flight Time	Event	Comment
0:42	Take Off	
1:06	Trim attempt	Hard to see if trimmed b/c wind
1:19	1 circle completed	
2:17	2 circles completed	
3:17	3 circles completed	
3:25	Engine strange sound	
3:33	4 circles completed	
4:00	5 circles completed	
4:36	Set up for landing	
5:00	Low fly by	
5:51	Touch and go	
6:43	Landing	Touched wing tip skid on slow down
7:15	Engine cut	

Engine Run Time:~8:30
Total Flight Time:7:18
FDR Run Time:__:__
Temperature:__

Final Weight:21.65 lb + Fuel
Final Tank Pressure:___
ATM Pressure:__

Additional Comments:
- Sensitive noise gear
- Aileron sluggish

STINGRAY IN-FLIGHT NOTES

Flight No: 2 GPS FLIGHT II
Initial Conditions: Mid-rate on all control surfaces.

Flight Time	Event	Comment
0:19	Take Off	
0:48	1st Round	
1:19	Trim	
1:27	2nd round	
1:50		Slight Breeze
2:08	3rd round	
2:56	4th round	
2:59		Slowing down
3:35	5th round	Low pass
4:22	6th round	Low pass
5:02	7th round	Low pass
5:37	8th round	Ready to land
6:13	Idling	
6:26	9th round attempt landing	
7:07	Touch + go	
7:36	Landing	

Engine Run Time
Total Flight Time:
FDR Run Time:__:__
Temperature:__

Final Weight:21.65 lb + Fuel
Final Tank Pressure:___
ATM Pressure:__

STINGRAY IN-FLIGHT NOTES

Flight No: III 10/27/01
Initial Conditions:

Flight Time	Event	Comment
2.17	Take off	
	Attempt to trim	Engine went in to idle (fail safe mode)
	Need to bring it back	
3.37	Touch Down	

Engine Run Time 3.37
Total Flight Time:
FDR Run Time:__:__
Temperature:__

Final Weight:
Final Tank Pressure:___
ATM Pressure:__

Flight Card: Trim Aileron Doublet
Controller ON Controller OFF
Rudder Doublet Land

Additional Comments:

Repair:

- Right Blend: epoxy, filling
- Test engine
- Full check of internal components
- Check spar
- Landing gear: order new ones + install

For next Flight:

- Antenna on Vertical
- Counterpoise placement
- Range Check!!

STINGRAY IN-FLIGHT NOTES

Flight No: Pitch rate controller
Initial Conditions:

Flight Time	Event	Comment
2.29	Take off w/ 10 deg. flap – uphill	
3.12	Flap up	
3.53	Controller ON	
5.02	Controller OFF	
5.21	Controller ON	
5.55	Elevator Pulse	
6.15	Controller OFF	
6.21	10 deg. Flap	
6.50	Full Flap	
8.13	Approach to land	
8.27	Landing	

Engine Run Time9:28
Total Flight Time:
FDR Run Time:__:__
Temperature:__

Final Weight:
Final Tank Pressure:___
ATM Pressure:__

Flight Card: Throttle back
Controller ON
Maneuver Elevator
Turn execution
Controller OFF
10deg. Flap
Landing pattern
Full Flap
Landing

STINGRAY IN-FLIGHT NOTES

Flight No: Pitch rate controller
Initial Conditions:

Flight Time	Event	Comment
0.33	10 deg. flap	
1.00	Low rate rudder	
1.07	Take off	
1.34	Flaps up	
2.20	Controller ON w/ AMS	No take back – Hands-off
3.04	Controller OFF	After 1 circuit
3.22	Controller ON w/ AMS	Hands-off
3.30	Controller OFF	
3.45	Controller ON w/ AMS	Hands-off
3.54	Controller OFF	Slight breeze from N.
4.27	Controller ON w/ AMS	Full Throttle
4.33	Controller OFF	
4.57	10 deg. Flap	
5.21	Throttle Back	Prepare to land
6.05	Full Flap	
6.21	Landing	
6.43	Engine off	

Engine Run Time 6.43
Total Flight Time:
FDR Run Time:__:__
Temperature:__

Final Weight:
Final Tank Pressure:___
ATM Pressure:__

Flight Card: Throttle back
Controller ON
Maneuver Elevator
Turn execution
Controller OFF
10deg. Flap
Landing pattern
Full Flap
Landing

STINGRAY IN-FLIGHT NOTES

Flight No: Turn coordinator 03/07/02
Initial Conditions:

Flight Time	Event	Comment
1.25	Take off	
1.50	Flaps up	
2.15	Engine died	
2.46	Landing	

Engine Run Time 2.15
Total Flight Time:
FDR Run Time:__:__
Temperature:__

Final Weight:
Final Tank Pressure:___
ATM Pressure:__

Flight Card:	Take off 10deg. flap	Controller ON	Controller OFF
	Up flap	Easy turns	Landing w/ Full flap
	Trim	small doublet	
	Small doublet	Large doublet/pulse	

Additional Comments: Replaced Glow plugs for next flight

STINGRAY IN-FLIGHT NOTES

Flight No: Turn coordinator 03/07/02
Initial Conditions:

Flight Time	Event	Comment
2.11	Flap 10deg. Down	
2.30	Take off	
2.57	Flap up	
3.50	Controller ON	Gaining altitude in turn
4.01	Controller OFF	Seem to fight controller
		Just Aileron Input
4.30	Controller ON	
4.39	Controller OFF	
5.04	Controller ON	
5.10	Controller OFF	OFF in turn
5.41	Controller ON	
5.53	Controller OFF	Gaining altitude in turn
6.08	Preparation for landing	
6.32	10 deg. Flap	
6.45	Full flap	
7.15	Fly by low	
7.57	Landing	Bouncy

Engine Run Time 7.87
Total Flight Time:
FDR Run Time:__:__
Temperature:__

Final Weight:
Final Tank Pressure:___
ATM Pressure:__

Flight Card: Take off 10deg. flap
Up flap
Trim
Small doublet

Controller ON
Easy turns
small doublet
Large doublet/pulse

Controller OFF
Landing w/ Full flap

Additional Comments: Stingray seemed to fight controller in the turn
Stingray kept gaining altitude and loosing velocity
Front gear bent upon landing

STINGRAY IN-FLIGHT NOTES

Flight No: New Yaw damper 05/07/02
Initial Conditions: a little windy

Flight Time	Event	Comment
.43	Flap 10deg. Down	
1.34	Full Flap, High rate rudder	
2.24	Take off	
2.42	Flap up	
3.18	1 click down right	
3.42	Controller ON	Controller On in straight path, Hands-off, rudder AMS Performed a full circuit with controller ON
4.24	Controller OFF	
4.25	1 click left A.	
5.02	Controller ON	Controller On in straight off-off, rudder AMS Performed 2 full circuit with controller ON
6.25	Controller OFF	
6.50	10 deg. Flap	
7.05	Fly by	
7.40	Landing	Ran off the runway, left wing hit starter battery

Engine Run Time 7.40
Total Flight Time:
FDR Run Time:__:__
Temperature:__

Final Weight:
Final Tank Pressure:___
ATM Pressure:__

Flight Card: Take off 10deg. flap
Up flap
Trim
Controller ON
Easy turns
Controller OFF
Landing w/ Full flap

Additional Comments: Stingray seemed to fly well with controller ON

Damage upon landing:

- Right wing (outer spar broken)
- Right blend
- Right Landing gear
- Cone + prop

STINGRAY IN-FLIGHT NOTES

Flight No: Yaw Damper III 07/02/02
Initial Conditions: New Engine

Flight Time	Event	Comment
3:20	Low rate Rudder	
3.27	Take off	No flap.
4.27	Trimming of the aircraft	
5.44	Controller ON	Controller On in straight off-off, rudder AMS Performed 2.5 circuit with controller ON
7.25	Controller OFF	Aircraft performed well
7.50	10° Flap	
8.08	Full Flap	
8.36	Low pass	Setting up for landing
9.01	Throttling down	
9.21	Low pass	
9.50	Throttling down	
10.12	Low pass	Velocity too high
10.20	Click down on throttle	
	Cutting back on power	Over tree? Yes
	Low pass	Throttle clicks needed
11.47	Low pass	
12.19	Kill throttle	Stearns helping with throttle
	Low pass	
	Low pass	
15.15	Low pass	Alerted Stearns about engine time
16	Low pass	
18.03	Engine died	Attempt to land in the field, Hit tree

Engine Run Time 18.03
Total Flight Time:18.20
FDR Run Time:__:__

Final Weight: 24.45
Final Tank Pressure:___
ATM Pressure:__

Flight Card: Take off 10deg. flap
Up flap
Trim

Controller ON
Easy turns

Controller OFF
Landing w/ Full flap

Additional Comments: Stingray seemed to fly well with controller ON
Damage upon landing:

- Right and Left wing spar
- Left blend

STINGRAY IN-FLIGHT NOTES

Flight No: Turn Coordinator Flight I 07/27/02
Initial Conditions:

Flight Time	Event	Comment
	10 deg Flap	
1.30	Low rate rudder	
1.34	Take Off	Before mid-runway
2.04	Couple click Right Aileron	
2.16	1 Click Right Aileron	Engine noise/vibration
2.44	1 Click Left Aileron	
3.16		Setting up for Controller
3.45	Controller ON	Aircraft turned right instead of left, and stalled
3.48	Controller Off	
4.28	10 deg. Flap	
5.10		Setting up for landing, plenty of altitude
5.45	Low pass	
6.10		Stearns: "Above tree"
6.25	Low throttle	
6.50	Low pass	
7.23	Low throttle	Idling
7.53	Landing	

Engine Run Time: 7.53
Total Flight Time:
FDR Run Time:__:__
Temperature:__

Final Weight: 24.45
Final Tank Pressure:___
ATM Pressure:__

Flight Card: Take off 10deg. flap
Up flap
Trim

Controller ON
Controller OFF
Landing w/ Full flap

STINGRAY IN-FLIGHT NOTES

Flight No: Turn Coordinator Flight II 07/29/02
Initial Conditions:

Flight Time	Event	Comment
1.40	10 deg Flap	All Control surfaces Mid rate, Rudder high rate
1.50	Low rate rudder	
2.28	Take Off	Before mid-runway
2.50	Flaps up	
3.57	Controller ON	Aircraft banked heavily, gained altitude, rolled
4.06	Controller OFF	
4.53	10 deg. Flap	
4.58	Full Flap	
5.54	Landing	

Engine Run Time: 6.58
Total Flight Time:
FDR Run Time:__:__
Temperature:__

Final Weight: 24.45
Final Tank Pressure:___
ATM Pressure:__

Flight Card: Take off 10deg. flap
Up flap
Trim

Controller ON
Controller OFF
Landing w/ Full flap

STINGRAY IN-FLIGHT NOTES

Flight No: Turn Coordinator Flight III - 08/02/02
Initial Conditions:

Flight Time	Event	Comment
1.15	10 deg Flap	All Control surfaces Mid rate, Rudder high rate
1.53	Low rate rudder	
1.58	Take Off	Before mid-runway
2.18	Flaps up	
3.00	1 Click Aileron	
3.17	Controller ON	Aircraft banked, a little wobbly
	1 click throttle	
3.25	Controller OFF	
3.46	Controller ON	Pilot added elevator, aircraft performed a better turn, aircraft rolled at the end of the turn, stalled
4.00	Controller OFF	
5.23	10 deg. Flap	
5.50	Full Flap	
6.15	Touch and go	
7.12	Landing	

Engine Run Time: 7.32
Total Flight Time:
FDR Run Time:__:__
Temperature:__

Final Weight: 24.45
Final Tank Pressure:___
ATM Pressure:__

Flight Card: Take off 10deg. flap
Up flap
Trim

Controller ON
Controller OFF
Landing w/ Full flap