# Autonomous Flight, Fault, and Energy Management of the Flying Fish Solar-Powered Seaplane

by

Ryan D. Eubank

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
2012

Doctoral Committee:

Associate Professor Ella M. Atkins, Chair
Professor Dennis S. Bernstein
Professor Ilya V. Kolmanovsky
Professor Guy A. Meadows
Associate Professor Luis P. Bernal

Experience is what you get when you do not get what you expect.

To my wife, Ilana, who both put up with and supported the monumental effort required to reach this point.

# ACKNOWLEDGEMENTS

First I would like to thank Dr. Atkins for taking a chance and offering a research position to an awkward and untested "freshman Ryan" all those years ago... and for not subsequently cutting her losses in all the years since. It was a long and sometimes bumpy road, but also a fun and memorable journey. I'm sure that I would not be where I am today without her guidance and support.

I must also extend a massive and heartfelt Thank You to the entire Flying Fish Team led by Prof. Guy Meadows who with Dr. Ella Aktins, Dr. Luis Bernal, Dr. Brian Gilchrist, and Dr. Pete Washabaugh together forged a multidisciplinary team from the departments of Aerospace Engineering, Naval Architecture & Marine Engineering, Electrical Engineering & Computer Science, and Atmospheric Oceanic & Space Sciences. I am also indebted to the students and staff who contributed to the monumental effort required for the development, construction, frequent repair and modification, and flight testing of two generations of Flying Fish flight test vehicles. Over the years this team has included: Nick Wild, Dan Macy, Curt Cooper, Keith Shaw, Stephanie Ogura, Derrick Yeo, Jonathan Brown, Gerald Giarmo, Lorelle Meadows, Brittany Smith, Hans Van Sumeren, Deano Smith, Dave Parsons, Joe Wild, and Tim Peters. Thank you all for the application of your expertise towards making the dream of Flying Fish into a reality.

Additional thanks are, of course, due to all of the teachers, professors, and peers who taught, challenged, and assisted me along the way. The complete list of those who qualify for thanks is really much too long to be recorder here, so I can only name

a few. Thank you Derrick for helping to support flight tests with humor as well as piloting skill, for late nights spent working on miscellaneous problems, and for help corralling UROP students. Thank you to Justin for being a sounding board for ideas, assisting with the final software challenges, and for helping to review and submit my final manuscript. Thank you to Cat for joining me in regular writing sweatshops and for providing feedback on journal papers and my thesis. Thank you to Andy for you early support of the avionics effort and for all your great ideas and assistance in the years that followed. Thank you to Eric for supporting ground station software, for good humor, and for general help in problem solving. Thank you to Ali for bringing both a good sense of humor and the will to ask the hard questions to presentations and meetings. I also would like to offer special thanks to these individuals for their friendship and the immeasurable help it offered in keeping me sane.

Thank you also to my family for their support and for "minimal" inquiries about how much longer this "whole Pd.D. thing" was going to take. I told you it would take a while, but not even I realized it would take as long as it did. Time really flies when you are periodically having fun.

Finally, I want to thank my father for helping to put dreams of flight into my head. Those dreams have brought me quite a long way.

# TABLE OF CONTENTS

**DEDICATION** . . . . . ii

**ACKNOWLEDGEMENTS** . . . . . iii

**LIST OF FIGURES** . . . . . ix

**LIST OF ABBREVIATIONS** . . . . . xi

**ABSTRACT** . . . . . xvii

**CHAPTER**

**I. Introduction** . . . . . 1

1.1 Motivation - The Modern Seaplane . . . . . 1
1.2 Flying Fish Platform . . . . . 7
1.3 Problem Statement . . . . . 9
1.4 Research Objectives . . . . . 10
1.5 Contributions . . . . . 11
1.6 Innovations . . . . . 12
1.7 Outline . . . . . 14

**II. Avionics Development and Integration** . . . . . 15

2.1 Avionics Hardware . . . . . 18
2.1.1 Power Systems and Solar Energy Harvesting . . . . 18
2.1.2 Central Avionics . . . . . 24

2.1.3 Sensor Systems . . . . . 27
2.2 Flight Management System . . . . . 32

**III. Dynamics and Kinematics of a Seaplane** . . . . . 38

3.1 Aerodynamic Forces and Moments . . . . . 38
3.1.1 Aerodynamic Forces . . . . . 40
3.1.2 Aerodynamic Moments . . . . . 46
3.2 Aircraft Dynamics and Equilibrium (Trim) States . . . . . 49
3.3 Aircraft Kinematics . . . . . 57
3.4 Seaplane Dynamics . . . . . 60
3.4.1 Hydrodynamics . . . . . 60
3.4.2 Ocean Waves . . . . . 62
3.4.3 Ocean Currents . . . . . 63

**IV. Autonomous Control and Guidance of a Seaplane** . . . . . 67

4.1 Trim-State Determination . . . . . 69
4.2 Control Strategy . . . . . 77
4.2.1 Longitudinal Control Law . . . . . 77
4.2.2 Lateral Control Law . . . . . 81
4.2.3 Mixed Open/Closed Loop Control for Takeoff . . . . 83
4.3 Guidance . . . . . 84
4.3.1 Sequential Guidance Modes . . . . . 86
4.3.2 Guidance Mode Transitions . . . . . 92
4.4 Results . . . . . 94
4.4.1 Gain Tuning Strategy . . . . . 95
4.4.2 Longitudinal Control Performance . . . . . 97
4.4.3 Lateral Control Performance . . . . . 99
4.4.4 Guidance Sequence Testing . . . . . 101
4.4.5 Mode Transition Performance . . . . . 104
4.4.6 Watch Circle Crossing Performance . . . . . 105
4.4.7 Unattended Deployment Requirements . . . . . 108

**V. Fault-Tolerant Air Data System** . . . . . 110

5.1 Background . . . . . 114
5.2 Implementation . . . . . 116
5.2.1 Signal Fault Detection . . . . . 117
5.2.2 Confidence-Discriminate Data Fusion . . . . . 122
5.2.3 Wind Estimation . . . . . 126
5.2.4 Signal Fault Detection Algorithm . . . . . 129
5.3 Results . . . . . 133

**VI. Energy-Aware Flight Management** . . . . . 141

6.1 Energy and Environment Models . . . . . 143
6.1.1 Environment Models . . . . . 144
6.1.2 Vehicle Energy Dynamics . . . . . 148
6.2 Mission Planner . . . . . 155
6.3 Search Strategies . . . . . 169
6.4 Case Study Results . . . . . 176
6.4.1 Greedy Search . . . . . 176
6.4.2 Midnight Missions: Non-Energy-Harvesting Optimal Planning . . . . . 183
6.4.3 Midday Missions: Admissible Planning with Uncertain Optimality . . . . . 190
6.4.4 Dawn/Dusk Missions: Planning Near Minimum Charging Conditions . . . . . 193
6.4.5 Summary . . . . . 195

**VII. Conclusions and Future Work** . . . . . 197

7.1 Major Findings . . . . . 198
7.2 Future Work . . . . . 200

**APPENDIX** . . . . . 202
A.1 Multi-Protocol Device Interface Board with 12-bit Analog/Digital Converter . . . . . 203
A.2 Remote Analog/Digital Conversion Nodes . . . . . 206
A.3 High-Power Low-Loss Current Limiter . . . . . 209

**BIBLIOGRAPHY** . . . . . 212

# LIST OF FIGURES

**Figure**


1.1 Seaplane Examples: a) Float Plane, b) Flying Boat . . . . . 2
1.2 Flying Fish Vehicles . . . . . 7
1.3 Phase I & II Flying Fish Vehicle Planforms . . . . . 9
2.1 Phase I Flying Fish Flight System . . . . . 16
2.2 Flying Fish Avionics Overview . . . . . 17
2.3 Installed Solar Array . . . . . 24
2.4 Flying Fish Central Avionics . . . . . 25
2.5 Wolverine FMS Architecture . . . . . 32
3.1 Peak Propeller Efficiency . . . . . 46
3.2 Steady Helical Flight, Perspective View[16] . . . . . 51
3.3 Steady Helical Flight, Detail Views[16] . . . . . 51
3.4 Small-Angle Approximation Error . . . . . 53
3.5 Dubins Path Waypoint-Based Flight Plan . . . . . 60
3.6 Linear Drift Simulation . . . . . 66
4.1 Piloted Flight Profile . . . . . 70
4.2 Pilot Control Inputs . . . . . 70
4.3 Piloted Takeoff . . . . . 71
4.4 Racetrack-Pattern Flight Test . . . . . 72
4.5 Racetrack-Pattern Flight Profile . . . . . 73
4.6 Trimmed Descent Example # 1 . . . . . 74
4.7 Trimmed Descent Example # 2 . . . . . 75
4.8 Trimmed Descent Example # 3 . . . . . 76
4.9 Longitudinal Control Law . . . . . 78

4.10 Lateral Control Law . . . . . 81
4.11 Mixed Open/Closed-Loop Takeoff Commands . . . . . 84
4.12 Cyclic Mission Profile . . . . . 85
4.13 Flying Fish w/Dolphins . . . . . 95
4.14 Longitudinal Control - Phase II, Pitch . . . . . 98
4.15 Longitudinal Control - Phase II, Altitude . . . . . 99
4.16 Longitudinal Control - Phase II, Airspeed . . . . . 99
4.17 Lateral Control - Phase, Yaw/Heading . . . . . 101
4.18 Lateral Control - Phase, Roll/Bank . . . . . 101
4.19 Autonomous Flight Control, Phase I . . . . . 102
4.20 Autonomous Flight Profile, Phase I . . . . . 103
4.21 Autonomous Flight Attitude, Phase I . . . . . 103
4.22 Autonomous Flight, Phase I - Monterey 2007 . . . . . 104
4.23 Smoothed Pitch Response, Phase I . . . . . 105
4.24 Watch Crossing, Phase I . . . . . 106
4.25 Accurate Watch Crossing, Phase I . . . . . 107
4.26 Improved Circle Crossing, Phase II . . . . . 108
5.1 System Configuration with ADS Confidence Filter . . . . . 117
5.2 Variance Analysis . . . . . 134
5.3 Failure Detection Results . . . . . 134
5.4 Nominal Operation Case with Wind Model Estimate . . . . . 135
5.5 Single Probe Failure Case . . . . . 136
5.6 Double Probe Failure Case . . . . . 136
5.7 High Speed Taxi, Double Failure Case . . . . . 138
5.8 Flight Test, Single Failure Case . . . . . 138
5.9 Initially Failed Probe Case . . . . . 139
5.10 Flight Through Rain . . . . . 140
6.1 Daily Solar Energy Model . . . . . 146
6.2 NREL-Saft Battery Model[71] . . . . . 149
6.3 Battery Simulation: Two Flights . . . . . 152
6.4 Empirically Determined Curve-Fit Motor Power . . . . . 155
6.5 Planning Environment . . . . . 157

6.6 Example Missions . . . . . 163
6.7 Energy Required to Takeoff and Fly to an Aerial Goal (Altitude = 25.0m) . . . . . 164
6.8 Energy Required to Takeoff and Fly to an Aerial Goal (Altitude = 25.0m) . . . . . 165
6.9 Expansion of All Possible Mission Trajectories Between Goal Points 169
6.10 Symmetric Graph Problem with Four Nodes . . . . . 170
6.11 Example Search Tree for Four Node Graph . . . . . 172
6.12 Example Search Tree for Four Node Graph . . . . . 173
6.13 Douglas Lake Operating Region . . . . . 178
6.14 Boundary Maintenance Mission . . . . . 179
6.15 Boundary Maintenance + Surface Goal Plan . . . . . 180
6.16 Greedy Multi-Goal Planning . . . . . 182
6.17 Midnight Uniform-Cost Planning . . . . . 185
6.18 Comparison of Actual Cost and Heuristic Estimates of Landing . . . 190
6.19 Cruise Duration Required to Guarantee Admissibility . . . . . 190
6.20 Midday Uniform Cost Planning . . . . . 192
6.21 Midday A* Planning . . . . . 193
6.22 Early Dusk Uniform-Cost Planning . . . . . 194
6.23 Early Dusk A* Planning . . . . . 194
6.24 Dusk/Dawn Mission Planning . . . . . 195
A.1 Logic-Level Interface & ADC Circuit Board . . . . . 203
A.2 Fully Assembled 3rd Generation Interface-ADC Board . . . . . 204
A.3 Remote ADC With Integral Pressure Transducers . . . . . 207
A.4 Current Limiting Circuit Design . . . . . 210

# LIST OF ABBREVIATIONS

| | |
|---|---|
| $V$,$\check{V}_{\mathrm{i}}$ | Forward airspeed and trim/reference airspeed for mode $i$ (see mode) |
| $h$,$\check{h}_{\mathrm{i}}$ | Altitude and trim/reference altitude in mode $i$ (see mode) |
| $\alpha$,$\check{\alpha}_{\mathrm{i}}$ | Angle of attack and trim/reference angle of attack in mode $i$, see also: AOA |
| $\mathcal{A}_{\mathrm{i}}$ | Area of the shape or object indicated by the subscript $i$ |
| **AR** | Wing aspect ratio: $(\frac{b^2}{S})$ |
| $\mu$,$\check{\mu}_{\mathrm{i}}$ | Aircraft bank and trim/reference bank angles in mode $i$ (see mode) |
| $\bar{p}$ | Satisfaction priority of mission goals or constraints |
| $\mathbb{B}_k$ | Mission completion benefit of the $k^{th}$ flight plan |
| $\check{U}$ | The set of trim control inputs for all vehicle operational modes |
| $\check{X}$ | The set of vehicle trim states used by the outer-loop guidance laws |
| $c$ | Mean wing chord |
| $\mathcal{C}_{BH}$ | Flight planning constraint: Hard boundary (inviolable containment region) |
| $\mathcal{C}_{BS}$ | Flight planning constraint: Soft boundary (advisory containment region) |
| $\mathcal{C}_{OH}$ | Flight planning constraint: Hard obstacle (inviolable exclusion region) |
| $\mathcal{C}_{OS}$ | Flight planning constraint: Soft obstacle (advisory exclusion region) |
| $\mathbb{C}_k$ | Mission completion cost of the $k^{th}$ flight plan |
| $\Delta\theta_s$ | Trajectory interpolation step size |
| $\rho$ | The density of air under given atmospheric conditions (nominally: $1.225^{\mathrm{kg}})/_{\mathrm{m}^3}$ |
| $\eta_{\mathrm{i}}$ | Efficienc of the system/process that is indicated by the subscript $i$ |
| $\mathcal{E}_{\mathrm{cost}}$ | Energy expenditure required to complete a flight plan |

$\mathcal{E}_{\mathrm{i}}$ Energy of the system or process indicated by the subscript $i$

$\mathcal{E}_{\max}$ Maximum energy storage capacity of vehicle

$\mathcal{E}_{\mathrm{rcvr}}$ Energy that can be recovered as a result of flight plan completion

$\mathcal{E}_{\mathrm{sys}}$ Energy currently stored by vehicle

$\gamma$,$\check{\gamma}_{\mathrm{i}}$ Flight path and trim/reference flight path angles in mode $i$ (see mode)

$\mathrm{v}_{0,i}$ Intial goal completion value of the $i^{th}$ goal

$\mathrm{v}_i$ Goal completion value of the $i^{th}$ goal

$\mathrm{v}_{t,i}$ Rate of increasing goal exploration value of the $i^{th}$ goal

$\omega_S$,$\omega_A$ Goal waypoint: Surface/Landed and Airborne/Flying, respectively

$g$ Acceleration due to gravity ( $9.806^{\mathrm{m}}/_{\mathrm{s}^2}$)

$\sigma$,$\check{\sigma}_{\mathrm{i}}$ Aircraft heading trim/reference heading angles in mode $i$ (see mode)

$L$,$D$,$Y$ Aerodynamics lift, drag, and sideforce forces, respectively

$\mathcal{S}$ The 4-tuple that defines the guidance and control trim state and gain database

$\mathbb{P}_k$ Planner data for the $k^{th}$ goal or constraint

$S$ Wing planiform area

$\mathcal{P}_{\mathrm{i}}$ Power of the system or process indicated by the subscript $i$

$\mathcal{P}_{\mathrm{spec}}$ Power, per unit area, available from incident solar radiation

$d$ Propeller diameter

$\mathcal{L}$,$\mathcal{M}$,$\mathcal{N}$ Aircraft rolling, pitching, and yawing moments, respectively

$\beta$,$\check{\beta}_{\mathrm{i}}$ Sideslip and trim/reference sideslip in trim/mode $i$, see also: SS

$\theta_{array}$ Solar array incidence (in vehicle frame, pitch axis)

$T$ Aircraft thrust

$R$ Turning radius

$\mathbb{U}_k$ Total mission utility of the $k^{th}$ flight plan

$\mathbb{V}_k$ Mission completion (goal achievement) value for the $k^{th}$ flight plan

$\mathbb{V}$,$\mathbb{B}$,$\mathbb{U}$ Planner utility weighting values for value-to-utility, benefit-to-utility, and cost-to-utility scaling

$\nu$ Variance of a set of measurements

$\vec{s}_*$ Solar incidence vector in the *coordinate frame

$W$ Aircraft weight

$\tau_i$ Weighting parameter related to the process/variable given by the subscript $i$

$b$ Aircraft wingspan

$a_{sun}$ Solar azimuth angle

$C_L$,$C_D$,$C_Y$ Coefficients of aerodynamic lift, drag, and sideforce force, respectively

$C_{\mathcal{L}}$,$C_{\mathcal{M}}$,$C_{\mathcal{N}}$ Coefficients of aerodynamic roll, pitch, and yaw moments, respectively

$e$ Oswald efficiency factor

$e_{sun}$ Solar elevation angle

$f$ Frequency of digital guidance and control execution

$f(\cdot)$ Abbreviated notation for variables required to describe Sun-relative configuration of a solar array, includes: position, attitude, and atmospheric conditions (See: Sec.6.1.1).

$G_{i,j}$ Guidance gain of type $i$ in mode $j$ (see mode)

$K_{ij,k}$ Control gain of type $i$ acting on control axis $j$ in mode $k$ (see mode)

$R_e$ Reynolds number

$T_s$ Time duration of mode-switch trajectory interpolation

$\check{\delta}_{i,j}$ Trim control inputs of type $i$ in mode $j$ (see mode)

$\delta$,$\Delta\delta_i$ Absolute and offset control inputs of type $i$

$\theta$,$\check{\theta}_i$ Estimated pitch and trim pitch in mode $i$ (see mode)

$\phi$,$\check{\phi}_i$ Estimated roll and trim roll in mode $i$ (see mode)

$\psi$,$\check{\psi}_i$ Estimated yaw and trim yaw in mode $i$ (see mode)

ZN Zielger-Nichols method for gain determination

AOA Vertical Angle Of Attack ($\alpha$) between the airfoil datum and relative wind

ACAT Autonomous Cargo Amphibious Transport

ACPU Avioincs Central Processing Unit or, simply, the avioincs computer

ADC Analog to digital converter

ADS-B Automatic Dependent Surveillance-Broadcast

$C$ Confidence level [0,1] associated with a sensor signal

CG Center of gravity

COTS Commecially available, Off-The-Shelf equipment and materials

DARPA Defense Advanced Research Projects Agency

ESC Electronic speed control

FFSP Frequent Flier Salesman Problem, a proposed varient of the classic TSP

FMS Flight Management System

GA General Aviation, as in General Aviation aircraft

GNC Guidance, Navigation, and Control

IMU Inertial measurement unit

INS Inertial navigation system

$\Lambda_{type}$ Sensor failure vote for a signal ($s_n$) subject to test of type: *type*

MPPT Maximum power point tracker for optimal power exctraction from solar cells

NREL National Renewable Energy Laboratory

OOP Object-oriented programming

OR Outlier rejection

OS Operating System

$P_{type}$ Probability that a given sensor/signal has an active failure of type: *type*

RC Radio-Control

ROI Region of interest

$s_n$ The $n^{\text{th}}$ sensor signal

S-UAS Seaplane (U)npiloted (A)erial (S)ystem

SSC Servo swtich controller, manages pilot and autopilot control inputs

SS Sideslip ($\beta$): Horizontal angle between aircraft centerline and relative wind

$\mathbb{T}_{type}$ Tolerance for comparison of signal performance for a test of type: *type*

TCAS Traffic Collision Avoidance System

TSP Traveling Salesman Problem

UAS Unpiloted Aerial System

UAV Unpiloted Aerial Vehicle

$\vec{v}_F$ Vehicle velocity vector in reference frame: $F$

$\vec{w}_F$ Wind vector in reference frame: $F$

$\xi$ Weighted-average measurement age decay parameter [0,1]

$\phi$,$\theta$,$\psi$ Control axis: roll ($\phi$), pitch ($\theta$), and yaw/heading ($\psi$), respectively

$a$,$r$,$e$,$t$ Input types: $a$ileron, $r$udder, $e$levator, and $t$hrottle

$d$ Desired value

mode Mode-specific variable, possible values: accel, takeoff, climb, cruise, descent, landing, drift, and turn

$P$,$I$,$D$,$C$ Gain types: $P$roportional, $I$ntegral, $D$erivative, and $C$ritical, respectively

# ABSTRACT

Autonomous Flight, Fault, and Energy Management of the Flying Fish Solar-Powered Seaplane

by

Ryan D. Eubank

Chair: Ella M. Atkins

The Flying Fish autonomous unmanned seaplane is designed and built for persistent ocean surveillance. Solar energy harvesting and always-on autonomous control and guidance are required to achieve unattended long-term operation. This thesis describes the Flying Fish avionics and software systems that enable the system to plan, self-initiate, and autonomously execute drift-flight cycles necessary to maintain a designated watch circle subject to environmentally influenced drift. We first present the avionics and flight software architecture developed for the unique challenges of an autonomous energy-harvesting seaplane requiring the system to be: waterproof, robust over a variety of sea states, and lightweight for flight. Seaplane kinematics and dynamics are developed based on conventional aircraft and watercraft and upon empirical flight test data. These models serve as the basis for development of flight control and guidance strategies which take the form of a cyclic multi-mode guidance protocol that smoothly transitions between nested gain-scheduled proportional-derivative feedback control laws tuned for the trim conditions of each flight mode. A fault-tolerant airspeed sensing system is developed in response to elevated failure

rates arising from pitot probe water ingestion in the test environment. The fault-tolerance strategy utilizes sensor characteristics and signal energy to combine redundant sensor measurements in a weighted voting strategy, handling repeated failures, sensor recovery, non-homogenous sensors, and periods of complete sensing failure. Finally, a graph-based mission planner combines models of global solar energy, local ocean-currents, and wind with flight-verified/derived aircraft models to provide an energy-aware flight planning tool. An NP-hard asymmetric multi-visit traveling salesman planning problem is posed that integrates vehicle performance and environment models using energy as the primary cost metric. A novel A* search heuristic is presented to improve search efficiency relative to uniform cost search. A series of cases studies are conducted with surface and airborne goals for various times of day and for multi-day scenarios. Energy-optimal solutions are identified except in cases where energy harvesting produces multiple comparable-cost plans via negative-cost cycles. The always-on cyclic guidance/control system, airspeed sensor fault management algorithm, and the nested-TSP heuristic for A* are all critical innovation required to solve the posed research challenges.

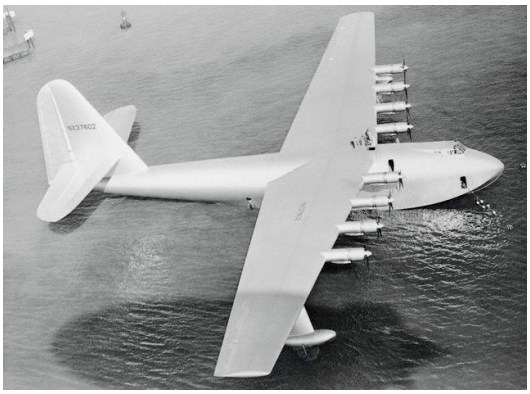
# CHAPTER I

# Introduction

## 1.1 Motivation - The Modern Seaplane

A decade after the turn of the twentieth century, on the heels of the first successful heavier-than-air vehicles, early aviators began to focus considerable energy on the development of aircraft that operated from water rather than from land.[1] These aircraft, which would serve a key role in the developing aviation industry, fell into two broad categories: (1) floatplanes and (2) flying boats, based on the nature of their buoyant structures. The former is characterized by the use of slender floats attached to the fuselage for buoyancy (Fig. 1.1a) and the latter by having the fuselage serve as the primary buoyant structure (Fig. 1.1b). These vehicles are often collectively known as seaplanes and they saw extensive commercial and military application in the earliest years of modern aviation before fading into near obscurity in the twenty-first century.

(a) British Supermarine S.6B, Circa 1931



(b) Hughes H-4 Hercules "Spruce Goose," Circa 1947

Figure 1.1: Seaplane Examples: a) Float Plane, b) Flying Boat

Early seaplanes possessed two critical advantages over their land-based counterparts that would greatly aid the developing global aviation industry: (1) a seaplane can operate beyond developed runways, requiring only a sufficiently large body of water for takeoff and landing and, (2) a seaplane can be landed with little-or-no crosswind as the average body of water does not have the inherent landing-direction constraints of a narrow runway. These factors imparted a high level of perceived safety to early seaplanes, at least amongst pilots, and were key to the logistical proliferation of aircraft in an era that had yet to see the development of widespread airport infrastructure. Water deployment made the seaplane well suited for a range of military applications, especially in support of the world's navies. The seaplane also played a role in early commercial aviation when, in the wake of the first World War, the surplus of military seaplanes provided a ready-built fleet of cheap minimal-infrastructure aircraft to jump-start the air-carrier industries. Seaplane popularity continued through two world wars, particularly in military search and rescue, submarine warfare, and reconnaissance roles. However, by the end of the second World War, advances in aircraft design began to significantly widen the performance gap between land-based aircraft, which did not face the weight and drag penalties incurred by buoyant structures, and

the slower heavier seaplane. Given the continuous expansion of airports with developed runways, the growing performance disparity between land and sea aircraft led to a steady decline in the overall popularity of the seaplane. Military land-based and carrier-based aircraft, along with a growing contingent of military rotorcraft, steadily filled what had become traditional seaplane roles. The U.S. Navy eventually retired its last flying boat squadron in 1967.[2] Today the commercial aviation industry has largely abandoned seaplane development and there are only a handful of general aviation (GA) seaplanes on the market. Seaplanes persist now almost exclusively in a small niche market that services the needs of floatplane pilots and specialty aviation operations beyond the reach of prepared runways.

While the latter half of the last century saw the marked decline of the seaplane, that same fifty year span saw explosive development in the area of unmanned aerial vehicles (UAV). Owing to the complexity of these systems, the modern UAV is more properly classified as an unmanned aerial system (UAS) which encompasses the requisite operational elements of the UAV airframe, principle and auxiliary avionics, payloads, and ground-station/user-interface. Modern UAS are often designed for and assigned to missions considered too dull, dirty, or dangerous for a piloted aircraft. Common UAS missions include a wide range of surveillance and inspection tasks but have more recently begun to include broader mission profiles with defensive and offensive military action. As continued technological advancements have given rise to ever-increasing system capabilities the number of missions that can be flown by UAS, and the number of UAS developed to fly those missions, has increased at a steady pace; Department of Defense spending on UAS projects has nearly doubled every year for the last decade.[3] UAS have clearly become an established and prominent element of modern aviation. In 2005 the United States Department of Defense estimated that more than thirty nations were developing or manufacturing in excess of 250 different UAS platforms; the number of participants and vehicles is certainly higher today.[3]

Given the chronological disparity between the eras of seaplane aviation and the modern UAS, the potential for synergistic intersection of the core technologies has remained largely unexplored. The revival and application of seaplane technology to a UAS would enable an array of previously unachievable open-water unmanned surveillance missions with an unparalleled deployment footprint over the largely water-covered surface of the earth. A deployed seaplane-UAS would have the ability to intercept, chase, or rendezvous at flight speeds that exceed those of most surface and sub-surface watercraft. Furthermore, unlike traditional land-based UAS, the dynamic landing footprint of the seaplane-UAS (S-UAS) provides more flexibility to pursue a single mission over multiple flights and the unique potential to pursue goals both while aloft and while on the surface. An autonomous seaplane could, for example, incorporate a landing site as a strategic waypoint within the scheme of a longer unattended mission profile. The flexibility to continue a mission beyond flight also allows for the reallocation of on-board resources, that would otherwise be committed to flight activities, to further mission goals; a landed UAS can direct the bulk of its computational, energy, and storage resources towards data collection and processing tasks and to flight and mission planning. Moreover, a seaplane-UAS equipped with energy harvesting technologies, such as solar or wave energy collection, could effectively self-refuel in its landed state before continuing to a new mission goal. This set of capabilities can be extended to a wide variety of scientific, commercial, and military goals. Specific applications might include oil-spill or algae-bloom identification/boundary-tracking, coastal observation and protection, water-quality analysis, or communication assistance between remote airborne, surface, and submerged assets.

A research concept vehicle known as the Autonomous Cargo Amphibious Transport (ACAT), fielded by Pisanich and Morris at the NASA Ames research center in 2002, was perhaps the first attempt to develop a modern seaplane-UAS.[4] The ACAT project produced a proof-of-concept prototype for an unmanned amphibious-landing

cargo/supply vehicle. The novel UAS design, built in hobby scale, was capable of autonomous takeoff, waypoint flight, and auto-landing to paved runways and smooth controlled water surfaces. The ACAT's autonomous flight capabilities were derived primarily from predefined trajectory following and the project seems to have terminated at the conceptual stage. After ACAT the seaplane-UAS concept largely disappeared from public view until, in the last half decade, the appearance of the Flying Fish concept and a number of commercial projects announced a new era of seaplane-UAS development. The three commercial contemporaries of the Flying Fish project are the Oregon Iron Works' Sea-Scout,[1] Warrior Ltd. Gull,[2] and the DRS Technologies RQ-15 Neptune[3] UAS.[3] As commercial products, very little has been formally published regarding the design and development of these systems and, beyond the manufacturer's advertisements, there are no records of these system's tested operational capabilities and performance. Promotional information suggests that Sea-Scout successfully demonstrated at least one autonomous landing as early as 2006, that the Gull has been able to achieve some form of autonomous takeoff/landing and waypoint trajectory following, and that the Neptune has a DRS Technologies autopilot with waypoint-based and user-interactive flight control. The Sea Scout and Gull are high-wing single engine floatplanes that operate strictly from the water. The RQ-15 Neptune, reportedly delivered to the Navy in 2005, is a blended-wing-body design but is not strictly a seaplane. Rather the Neptune appears to be a shipboard pneumatically-launched UAS capable of water landings for recovery purposes. These system are all notably of a different scale and performance-class than Flying Fish with heavy-lifting combustion-engine propulsion and without energy harvesting components.

The Flying Fish UAS was built as a proof-of-concept entry into this new class of

[1]Oregon Iron Works, Inc., Sea Scout Unmanned Seaplane: http://www.oregoniron.com/sea-scout.htm

[2]Warrior (Aero-Marine), Ltd., Gull UAV: http://www.warrioraero.com/GULL/index.htm

[3]DRS Technologies, Neptune UAS: http://www.drs-ds.com/Products/UAS/PDF/neptune.pdf

UAS. The concept had its genesis in a Defense Advanced Research Projects Agency (DARPA) project that called for persistent sub/surface surveillance of predefined circular regions, known as "watch circles," on the open ocean. Flying Fish was conceived as a flexible alternative to solving the persistent surveillance problem with a surface vehicle. The concept of applying a flight vehicle to this type of problem gives rise to the fundamental engineering questions that motivate the Flying Fish project, including:

1. Will the reduced energy required for low-drag flight (compared to high-drag water transit) be worth the energy required for takeoff?

2. Will such potential energy advantages be sufficient for the development of a viable aircraft given a reduced energy storage density (versus a surface vehicle) due to weight restrictions?

3. What will be the impact of environmental conditions and watch-circle size on the relative efficacy of a flight vehicle (e.g., what effect will the size/location of the watch-circle have on the efficiency of system operations)?

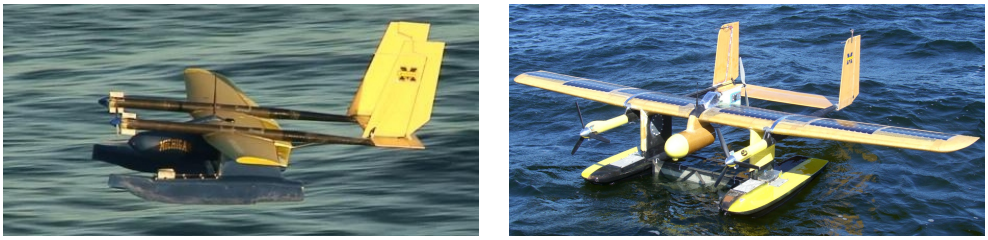
4. Will repositioning speed/flexibility outweigh the overall reduction in payload capacity of a flight vehicle over a surface vehicle?

5. Can a flight vehicle be made sufficiently robust to survive in the ocean over the long term?

The multidisciplinary Flying Fish team designed the system with a focus on unattended long-term operation, requiring efficient energy harvesting capabilities, fully-autonomous always-on operation, and a vehicle design that balanced airframe robustness with weight and efficiency. The Phase I Flying Fish was flight tested within only eight months and over the span of the project two flight vehicles were developed and field-tested, demonstrating robust self-managed autonomous takeoff, flight, and

landing from the open ocean and inland lakes. In addition to sequential self-initiated autonomous flight operations the Phase II Flying Fish has demonstrated solar-energy recovery and autonomous charging during live field tests.

## 1.2 Flying Fish Platform



(a) Phase I (b) Phase II

Figure 1.2: Flying Fish Vehicles

The Flying Fish airframe is a twin-tail, twin-boom fixed-wing dual-pontoon floatplane based on a conventional aircraft configuration (Fig. 1.2). Both phases of Flying Fish vehicle development have similar aerodynamic configurations differing primarily by scale (Fig. 1.3) and modifications to the wing and stabilizers of the Phase II vehicle to improve handling, structural integrity, and to accommodate solar panels. Table 1.1 presents an overview of key characteristics for both flight vehicles. In both phases vehicle structures were made unusually strong for a UAS to maximize resilience to the harsh ocean environment and are, as a result, relatively heavy. The addition of the twin pontoon flotation system provides hydrodynamic stability and minimizes drag during takeoff, but adds appreciable drag when airborne. The implementation of counter-rotating main motors for cancellation of the net propulsive torques increases aerodynamic symmetry and improves handling over single and co-rotating propellers. The propulsion system of the Phase II flight vehicle adds a third "boost" pusher-configuration motor implemented to add thrust for takeoff/climb under adverse flight

conditions. The set of three motors can deliver over 5kW of propulsive power to the propellers. Differential thrust was also added in the Phase II vehicle and has proven invaluable to maneuvering on the water. The central avionics are housed in the water-tight fuselage below the main wing, batteries and power systems are sealed under water-tight hatches in the pontoons, and the the solar energy system is concentrated in the vertical leg stanchions with solar-cells distributed over the upper surface of the main wing.

Table 1.1: Phase I & II Vehicle Specifications

| Attribute \ Vehicle | Phase I | Phase II |
|---|---|---|
| Wing Span ($b$) | 2.23 m | 3.76 m |
| Wing Area ($S$) | 0.84 $m^2$ | 1.88 $m^2$ |
| Mean Chord ($c$) | 35.7 cm | 53.7 cm |
| Airfoil | NACA 4414 | NACA 2414 |
| Dihedral | 4 deg | 3 deg |
| Pontoon Length | 1.0 m | 1.15 m |
| Flight Weight ($W$) | 18 kg | 30 kg |
| Cruise Speed ($V$) | 13 $^m/_s$ | 17 $^m/_s$ |
| Primary Voltage | 16.8 V | 21 V |
| Battery Capacity | 8.8 Ah | 48.6Ah |

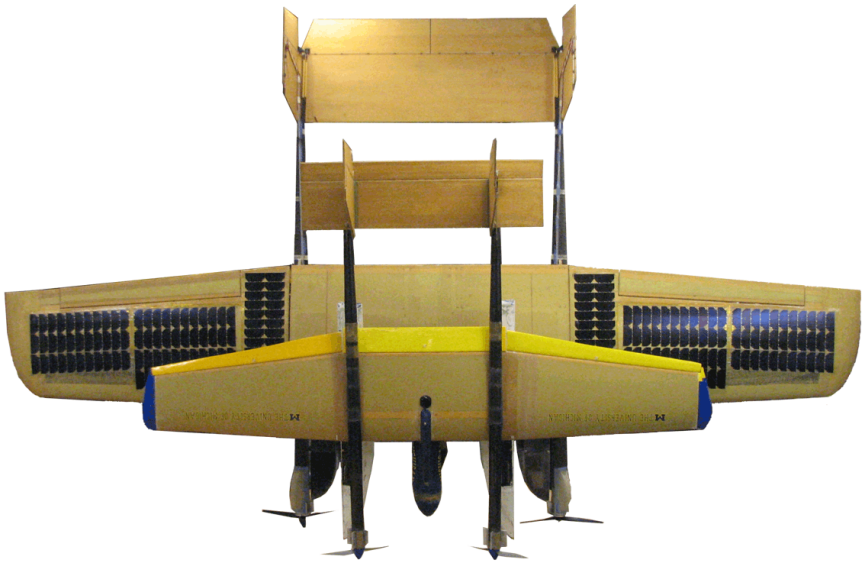


Figure 1.3: Phase I & II Flying Fish Vehicle Planforms

## 1.3 Problem Statement

***Develop and deploy an avionics system for unattended mission and flight management of the energy-harvesting Flying Fish seaplane for the purpose of conducting persistent ocean surveillance.***

The Flying Fish must possess flexible, rugged avionics and software capable of autonomously planning and executing the persistent ocean surveillance mission. The avionics must carry and interface all of the sensors and actuators required for state determination and flight control as well as supporting any mission payloads. The avionics computer must possess the processing power to host a flight management system (FMS) capable of robustly achieving the range of mission goals.

Flight control and stabilization requires vehicle trim state characterization and a control/guidance strategy capable of meeting flight goals. The autonomous flight control system must be capable of guiding takeoff subject to complex hydrodynamic interactions without destabilizing the vehicle. The range of flight modes must be robust to achieve the desired mission with a high level of safety and repeatability. The

system will, however, benefit from a deployment in open-water operating regions thus have the freedom to takeoff/land without the typical constraints of runway location and direction (e.g., Flying Fish can always land into the wind whenever necessary subject to obstacle avoidance constraints).

The system must survive the generally hostile and highly-variable ocean environment; the water that provides a near infinite landing surface is also corrosive, continuously changing shape, and shifting with weather and currents. The system must be robust to the effects of potentially adverse wind, current, and solar insolation conditions. The system must be able to handle the loss of redundant sensors due to fouling in this harsh environment. This fault tolerance capability must be able to diagnose and, if possible, recover from the failure-critical sensor systems.

Continuous deployment requires always-on mission monitoring, system management, and flight planning capability. The Flying Fish mission does not end at landing but continues through every subsequent drift and flight sequence until all mission goals are achieved or the system is recalled. For unattended operation (no ground station operator(s) directly supervising the system), the system must plan and execute both immediate flight operations and must plan for long-term survivability subject to the range of environmental (e.g. solar, current, wave, wind) conditions. The system must also be able to detect and handle system failures and unexpected environmental conditions and to re-plan operations to provide maximum survivability and goal satisfaction.

## 1.4 Research Objectives

The task of developing an operationally-robust seaplane-UAS capable of self-initiated autonomous flight and solar energy recovery is subject to a variety of engineering challenges. The first practical requirement is to develop, implement, and test the

avionics system with which the solutions to these engineering challenges (e.g., flight control, flight management, fault detection, mission planning) can be executed. Once the avionics system (hardware and software) is prepared and validated the objectives for this research are as follows:

- Develop models of flight kinematics and dynamics, energy harvesting and expenditure dynamics, and pertinent environmental conditions for the purpose of performance estimation in mission planning.
- Implement a fully-autonomous guidance, navigation, and control system capable of reliably directing all flight operations from self-initiated takeoff through self-initiated landing. Validate performance through flight testing.
- Maintain robust operations of the flight vehicle subject to the ocean environment. Specifically develop and flight validate filters/algorithms capable of discerning and recovering from common failures, such as the repeatedly-observed air data port water blockage.
- Assemble a planning and estimation system that includes vehicle, environmental, and energy collection/expenditure models and develop mission/flight planning techniques that are capable of devising and executing plans for the long-term deployment of an unattended energy-harvesting seaplane for persistent ocean surveillance missions.

## 1.5 Contributions

The contributions of this work are summarized below.

- A complex avionics system including custom electronics (Appx. A) has been developed concurrently with a customized flight software system and has been

shown, in real-world flight testing, to successfully interface and autonomously manage all vehicle subsystems including data collection, command and control, and communication (Ch. II).

- Guidance and control strategies were developed which achieve always-on fully autonomous vehicle-initiated flight, from takeoff through landing, over drift-fly cycles executed to maintain a continuous presence within a designated watch-circle region (Ch. IV).

- A confidence filter system was developed that leverages a signal-level fault-rejection/recovery schema to maintain flight operations in adverse environments despite high failure rates of critical air-data sensors in non-homogeneous redundant sensor networks (Ch. V).

- A search-based mission planner incorporating vehicle performance and energy collection and expenditure models was developed and evaluated to provide a framework for autonomous short and long-term mission planning subject to the full energy dynamics of a solar-regenerative seaplane-UAS (Ch. VI).

- In this thesis, a series of case studies are presented to study the efficacy and performance of a complex solar-regenerative seaplane-UAS energy-based flight planner. A novel heuristic is developed to improve search efficiency under adverse conditions and optimal plans under circumstances for which a global optimal is guaranteed to exist (Ch. VI).

## 1.6 Innovations

The innovations required to realize the contributions of this work are described below.

- Evaluation metrics were developed and appropriately combined to accurately

and robustly assess the fitness of failure-prone redundant pressure-based air-data-system sensors and to support data-fusion for accurate and feedback-safe flight speed determination (5.2.1-5.2.3). The proposed algorithm was validated by flight test data.

- A unique linear drift model was developed from empirical data (3.4.3) that is capable of providing reasonable estimates of the cumulative free drift behavior of a floating seaplane that arises from underlying interdependent nonlinear vehicle and environmental dynamics (e.g. coupled highly-variable complex-form aerodynamics and multi-hull hydrodynamics on a constrained free boundary surface subject to liquid transport dynamics).

- A cyclical always-on control and guidance strategy was implemented and successfully flight-tested. While the algorithm is itself simple, it is the first of its kind, enabling the system to repeatedly self-initiate and guide takeoff, flight, and landing activities based on watch circle boundaries.

- Physics-based trajectory planning, vehicle performance, and models of energy harvesting, usage, and storage were all integrated into a discrete search engine for the determination of energy-optimal paths subject to solar energy recovery dynamics. While individual models are themselves adapted from the literature, the combination of vehicle performance, environment, and solar energy harvesting and usage models have never before been integrated into a system that optimizes energy use over multiple flights with surface and airborne targets visited by the same platform.

- The Flying Fish mission planning problem is an asymmetric, non-metric, negative-cost version of the NP-hard Traveling Salesman Problem (TSP). An innovative heuristic is devised based on the concept of solving a simplified TSP problem

as the inner-loop of the top-level planner. This heuristic is demonstrated to substantially reduce search time.

## 1.7 Outline

In the following chapters this thesis will explore the development of the Flying Fish flight avionics and control systems and the specific challenges that were faced and overcome during the course of fielding this operational flight system. In Chapter II we describe the flight hardware and software focusing on avionics and sensor systems and the details of implementation, application to system goals, and inherent capabilities and limitations as determined through flight testing. In Chapter III we present a summary of seaplane dynamics and kinematics, develop steady flight equations and trim state definitions, and present simplified hydrodynamic and kinematic models to be used within the flight planning algorithms. Chapter IV presents the method used to determine vehicle trim states from flight data and describes the control and guidance laws that have stabilized and directed two phases of field-tested flight vehicles. Chapter Vdescribes a sensor fusion algorithm with which we overcame the specific challenges associated with accurate and reliable airspeed determination for the Flying Fish seaplane. Chapter VI presents the system models and planning architecture that enable the energy-aware flight management system. Conclusions will be presented in Chapter VII along with discussion of the current and future research topics related to the central concepts in this thesis. Appendices provide additional details on custom avionics systems developed in support of this research.

# CHAPTER II

# Avionics Development and Integration

The Flying Fish avionics were required to be robust but efficient by balancing processing capabilities, flexible interfaces, and communication capabilities, against power consumption, mass, and volume. Given the complex tasks that must be accomplished by the avionics, the available development time, and the desire to focus research on the stated challenges it was desirable to leverage the strength of mature commercial off-the-shelf (COTS) electronics but the specialized nature of the Flying Fish system also meant that no all-in-one solution was commercially available. As such, the Flying Fish avionics were assembled from a predominance of carefully selected COTS systems interfaced by custom sub-systems. Adaptable hardware interfaces, both physical and software-based, enabled interoperability between the broad range of devices required to meet the Flying Fish mission goals. The flight management system (FMS) software running on the avionics was developed concurrent to the earliest Flying Fish design phase as part of an open-source flight software initiative in the Aerospace department's Autonomous Aerospace Systems Laboratory (A2SYS). The resulting Wolverine-FMS was customized to the Flying Fish mission, environment, vehicle requirements, and the specific hardware in the avionics system. Refined over two flight-tested builds, the Flying Fish avionics has accomplished its design goals.

The Phase I Flying Fish avionics architecture was originally developed by Dr. Ella

Atkins of the University of Michigan's Aerospace Engineering Department (Fig. 2.1). The high-level architecture, while matured and expanded in the second phase design, has been consistent over the course of the program. This chapter will focus on the development of the Phase II avionics but an overview of the Phase I avionics is presented here for context. The avionics central processing unit (ACPU) in the Phase I Flying Fish, a Linux-based Gumstix Verdex embedded computer system, is responsible for executing the integrated Flight Management System. The Phase I ACPU interfaces with a Digi-Xtend wireless radio-modem for remote communication and a Microbotics MIDG-II inertial navigation system for Kalman-filtered 6DOF state measurements. A servo interface board accepts commands from both the test pilot and the ACPU for control actuation. The ACPU also interfaces analog pressure transducers for airspeed and an ultrasonic surface-ranging system added near the end of the Phase I vehicle test cycle.

(a) Airframe (b) Avionics

Figure 2.1: Phase I Flying Fish Flight System

The Phase II Flying Fish avionics system developed for the work presented in this dissertation was, much like the Phase I avionics, based on a Linux-based Gumstix ACPU (from the updated Overo line), integrated MIDG-IIC inertial navigation system, ultrasonic ranging unit, and analog air data system. The Phase II design updates the air data system with five-hole airspeed/directional probes and probe heating. The updated avionics also incorporate temperature, current, and voltage sensing, and interfaces with the complex solar energy harvesting system and a high-powered tri-

motor propulsion system. The avionics are organized into two mirrored (left/right) propulsion, power, and sensor interface subsystems connected by the central processing and communication pod (Fig. 2.2). The primary twin-electric propulsion system, mirrored primary power systems, remote analog sensing nodes, and system actuation components are concentrated in and on the pontoons, vertical wing stanchions, and motor booms. The central avionics houses the inertial navigation and ultrasonic sensors, communication systems, and flight management computer in proximity to the boost propulsion system. A solar array is distributed over the upper surface of the main wing with associated energy collection hardware concentrated in the vertical wing stanchions. In this chapter we describe the development and implementation of power distribution and harvesting systems as well as the central avionics and sensor suite. Challenges and solutions associated with the development and integration of these systems are highlighted along with the multipurpose application of these system to both flight and science goals.

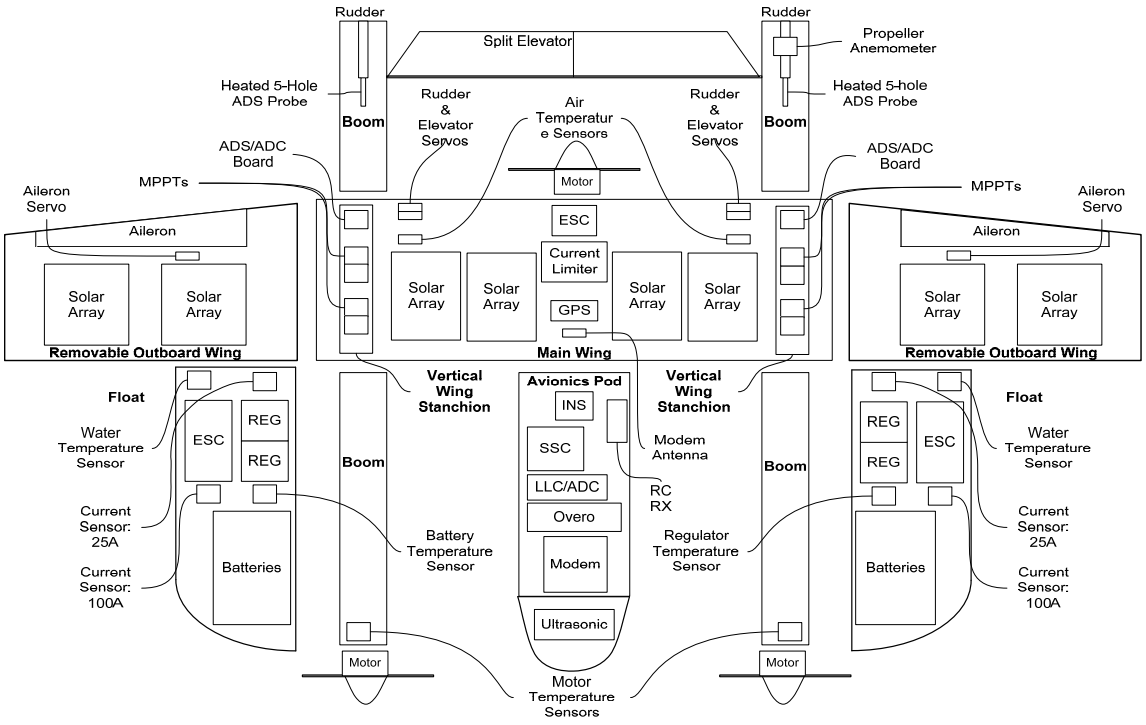



Figure 2.2: Flying Fish Avionics Overview

## 2.1 Avionics Hardware

### 2.1.1 Power Systems and Solar Energy Harvesting

The Flying Fish power system is based on two symmetric regulated lithium-battery power subsystems with multiple power busses and the solar energy harvesting systems. Water-tight compartments in each float house banks of 5-cell lithium-polymer batteries. Each battery compartment can accommodate a maximum of 1800kJ of lithium batteries, but may carry less to accommodate weight/balance and payload requirement. An additional 360kJ battery is housed on the center-line of the aircraft below the main flight computer. Configured for maximum endurance Flying Fish carries 3240kJ of battery capacity. Two major challenges facing the battery system are low tolerance for water exposure and temperature control. As part of the weight savings over a sealed battery system, multi-cell wrapped lithium battery packs have multiple contact points that could be shorted by a water leak. To combat water-ingress issues Flying Fish employs primary-secondary water protection throughout its electrical systems; every subsystem is customized for a maximum level of water resistance (secondary containment) while all of the electronics are sealed into water-tight compartments (primary containment). The team also implemented a program of custom-sealing each battery pack and cell contact. Nevertheless, we rely primarily on a waterproofed battery enclosure. The temperature control problem is two-fold as the batteries are susceptible to both extreme highs and lows in temperature. A problem facing the batteries in a solar-energy harvesting system is low battery down-time. If rapid flight-drift cycles are indicated for mission satisfaction the batteries will repeatedly be subject to cycles of heating under discharge conditions and heating under charge conditions. Battery temperature drops during the lower-energy charge process but the application of charge current reduces cooling, sustaining higher temperatures into the next flight cycle. Sufficiently rapid flight cycles, particularly in

warm weather, can produce a cumulative heating condition that, if left unchecked can quickly lead to lithium cell damage or destruction. For this reason, a heat-sink-integral metallic battery-compartment hatch is employed for warm weather operation. Conversely, operations in cold northern climates can hinder the system by dramatically decreasing the charge/discharge capabilities of the lithium-chemistry batteries. For cold weather operations Flying Fish utilizes PVC hatches to reduce heat dissipation from the battery enclosure. To achieve sustained operation in temperatures below ~10°C the battery compartment would require a heating system to maintain sufficiently high temperatures during extended drift cycles.

A high-capacity power bus provides direct connections for the batteries, avionics power busses, primary motors, and solar-energy harvesting system to minimize the distance, and hence losses, between the system's major power sources and loads. The avionics voltage regulation and power distribution busses are housed within separate waterproof compartments to the rear of the battery boxes. Flying Fish employs three regulated low-voltage (5V) power busses and one battery-level (16-21V) power bus. Low-noise voltage regulation for the primary avionics computer and sensor systems is handled by isolated and shielded 5V-10W regulators. The primary avionics power bus is supplied by the parallel combination of the regulated 5V output from both floats providing failsafe power to the low-voltage avionics in the event of a failure in either outboard power system. Unfortunately, stiffness in the output of the regulators prevents them from balancing the avionics load across both banks of batteries. Over the range of battery supply voltages one of the two regulators will always have a slightly higher output and thus carry the full avionics load. Separate 5V-25W hobby-class regulators provide power to the higher-demand control surface servos on each side of the aircraft. The noise characteristics and total rated efficiency of hobby-class regulators are often less favorable than those of scientific instruments but hobby regulators usually offer considerably better power-to-weight and power-

to-volume ratios than their heavily shielded and filtered precision counterparts. The servo regulators remain isolated from one another, forming the other two low-voltage busses, to help ensure that any single power-system failure can affect no more than half of the control-surface actuators. Three servos on each side of the aircraft provide actuation for the ailerons, dual rudders, and a split elevator.

Adjacent to the avionics regulator compartments, separate sealed enclosures house the two 160A high-voltage brushless speed controllers required to drive the main motors. Each electronic speed control (ESC) unit drives one of the main 2.8kW-rated electric out-runner motors which generate in excess of 5.5kg of thrust at ~7000rpm turning a 400mm 3-bladed propeller. The key challenge facing the high power system is the implementation of appropriate isolation between high-power systems and between high and low-power components. In the above configuration the motors can draw over 90A at ~20V. Such high power-handling necessitates the isolation of the motor buses within their respective halves of the vehicle. This is important for several reasons. With a direct connection a failure in any single primary power system could result in power draws in excess of 2.0kW across the aircraft to supply the opposing motor. The gauge and length of wiring required to sustain such loading would weigh nearly as much as the motor drawing the power. Furthermore, any such connection between the two power systems would increase the likelihood of cascading failures across the aircraft. Additionally, the interconnection between the two high power systems would carry the high-energy motor-commutation electromagnetic (EM) noise near the system sensors, actuators, and communication systems in the center of the aircraft. Finally, slight variations in loading, charging, and component efficiency are likely to produce unequal charge states between the main battery banks. A direct connection between the high power busses would allow unregulated cross-charging as the battery banks continuously seek equilibrium. While charge balancing is required, the mechanism of cross-charging is inefficient as the continuous energy transfer is

subject to parasitic losses in the wires, connectors, and batteries. In order to achieve efficient and safe charge balancing the two primary battery banks are joined via MOSFET-based "ideal" steering diodes at the avionics pod. The diodes provide the requisite return path isolation, only allowing power to be drawn towards the central avionics pod, with a low voltage drop of approximately 20mV. In this manner battery-level avionics loads will always draw from the more highly-charged battery bank, balancing system energy with a minimum impact on overall efficiency. The minimization of voltage drops and efficiency losses is paramount for the development of a solar-powered sustainable flight system.

During initial flight testing of the Phase II Flying Fish it was discovered that, while the vehicle could takeoff under ideal conditions and fly safely with the two main motors, additional thrust was often required for liftoff. Subject to the vehicle scale and configuration it was determined that the primary propulsion system was already near the maximum achievable thrust. Subsequently, a pusher-configuration boost motor was added between the tail booms to ensure robust takeoff performance. The specific challenges of such a system, outside of its physical addition to the vehicle, are primarily in power distribution. The high power handling ESC must be located in close proximity to a power supply as inductive effects on the input lines combined with MOSFET-driven electronic commutation of the brushless motor can induce destructive voltage spikes and ringing. Commercial ESCs have sufficient built-in capacitance to damp spikes and ringing for factory-length input wires (plus typical battery wiring), but substantially more external capacitance would be needed for long wire lengths. Unfortunately, the diode-steered link between the battery banks could not support the >1kW required by the boost motor. This posed a significant problem as the primary battery compartments are more than a meter (wire path) from the center line of the aircraft. The external capacitance required to supply an ESC at the center of the aircraft with power from a primary battery bank would more than triple the in-

stalled ESC volume. Moreover, direct access to the primary batteries by high-current wire would be structurally invasive and add considerable mass. The added weight, structural impact, capacitance requirements, and cross-charging problem meant that the ESC could not be connected to both battery banks, but drawing power from a single bank would result in a load imbalance exceeding the charge-balancing capabilities of the avionics load sharing system. Finally, it was important to consider that the high-frequency speed-variable EM noise induced over the ESC wiring could prove problematic if the wires were installed in proximity (and in parallel) to the signal wires running through the main wing and vertical stanchions. The only viable strategy to supply the high-demand boost motor, subject to these installation and configuration challenges, was to add an auxiliary battery in the bottom of the avionics pod. With this addition, a pair of relatively short (shielded and twisted) supply lines could be run directly away from the central avionics to an ESC installed within the boost motor support structure. The primary remaining challenge was to enable charging of the reserve battery from the main power system without exceeding the power limits of the installed wiring. A custom designed current-limiting circuit was installed to manage the charge of the reserve battery, charging the central battery to the voltage of the battery-level bus while limiting the current draw to a 1C charge rate (~5.4A) for the center battery (Appendix: A.3). The current limiter operates primarily (>90% of deployment) below the 5.4A clamping threshold where the circuit acts as a low-loss shunt to between the reserve and primary batteries for the purpose of charging. Above the current threshold a high-power MOSFET clamps the output current which, while not highly efficient, ensures the safety of the avionics bus during brief boost-motor applications. After capacitors were selected to stabilize the output of the current limiter over the full range of battery/ESC operating conditions the final circuit was installed and has performed robustly throughout the flight testing program.

The solar energy harvesting system is comprised of two primary elements: a solar array distributed over the wing (Fig. 2.3) and a set of microprocessor-driven power management devices know as maximum power-point trackers (MPPTs).[5] With an area of approximately 1.3m$^2$ the solar array is comprised of 352 semi-flexible polymer-encapsulated gallium arsenide solar cells. Solar panels are subject to varying power output characteristics as a function of the intensity, spectrum, and angle of incident sunlight; the function of the MPPT is to dynamically vary the load across the solar panel in order to extract the maximum energy under the given conditions. For a given set of conditions the supply voltage from a solar panel slowly decreases as the current draw increases until, at some threshold, the voltage begins to fall off sharply for higher current draw. A point near that threshold will yield the maximum available power to be drawn from the cells. The MPPTs vary their internal load and sample the current-voltage curve to find and follow the maximum power point and subsequently regulate the peak extracted voltage down to 21V. The cells are grouped into eight sub-arrays, each independently managed by a single MPPT. Having multiple sub-arrays and MPPTs allows the system to better tolerate partial shading, geometric variations, and other physical and electrical differences across the array by individually, and continuously, optimizing each section. Two MPPT boards are housed in each leg stanchion, each with two separate MPPT channels. The four MPPT channels on each side of the aircraft are combined in parallel with the adjacent battery bank. At peak output voltage the MPPTs acts as an array of constant-voltage chargers, delivering as much current as the avionics and batteries draw without exceeding the voltage set point. The MPPTs control electronics source power directly from the solar inputs, requiring less than ~75mW, and automatically shut down when insufficient solar energy is available. When inactive the diode-isolated MPPTs draw less than 30$\mu$A from the batteries. The MPPT are 98% efficient under typical operating conditions and have an absolute maximum rated output of about 400W. With the

Flying Fish array the total system output is expected to peak closer to 270W; an output of 250W has been observed under sunny conditions in the northernmost latitudes of Michigan during mid-June. The major concerns associated with the MPPT are related to EM noise. The high frequency switching of the internal regulators and power-point tracking electronics result in electrical oscillations with significant associated power over a wide range of the EM spectrum common to many communication devices. Preliminary testing suggested that these effects could dramatically impact the range of the wireless communication systems used in the Flying Fish. Physical proximity was a key factor in signal loss/attenuation but less so than actual electrical contact. In order to mitigate these effects the MPPTs were located in each wing stanchion, separated from most of the other avionics. The input and output lines were wrapped through ferrite beads to attenuate high frequency noise transmission. Physical integration of the solar power system to the vehicle and avionics system was conducted by the author.

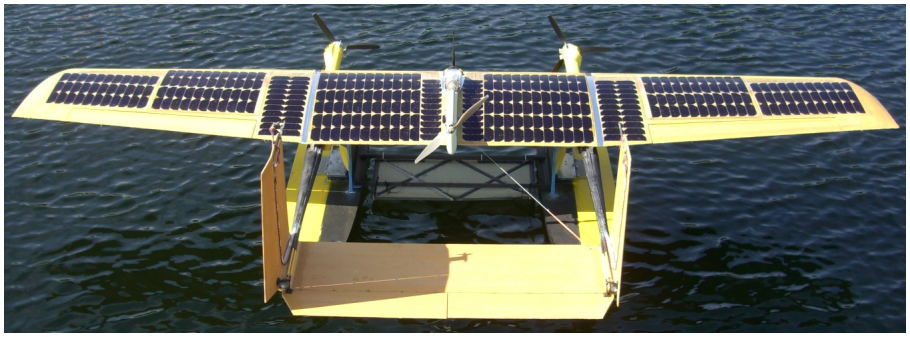


Figure 2.3: Installed Solar Array

### 2.1.2 Central Avionics

The primary Flying Fish avionics are organized in a waterproof avionics pod under the center of the wing (Fig. 2.4). The shelf carries the flight computer, inertial and ultrasonic sensors, communication systems, and actuator signal generation electronics. An external power switch, radio-control (RC) receiver antenna, GPS antenna,

and water-proof connector penetrate the rear hatch.

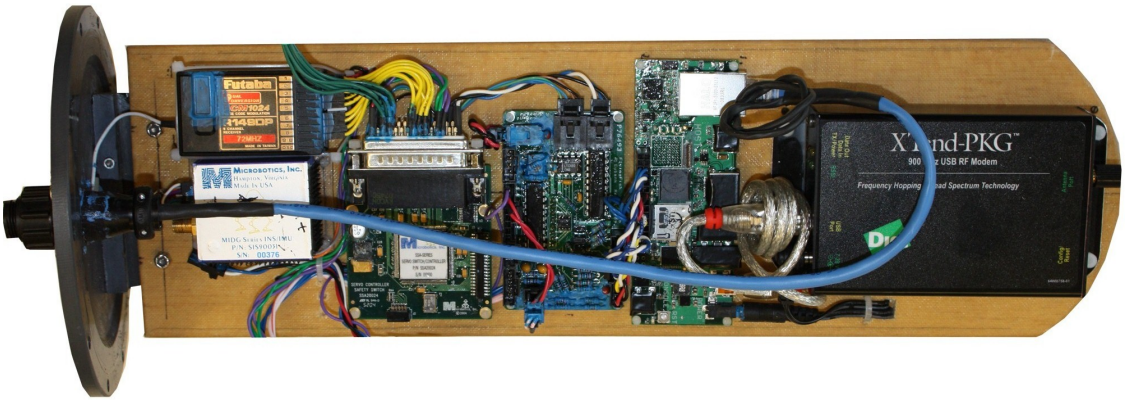



Figure 2.4: Flying Fish Central Avionics

A single Gumstix Overo 600MHz embedded Linux computer is the main processor, weighing only 6g with typical power draw under one watt. The Overo offers two 3-wire serial ports, six 10-bit A/D inputs, six PWM outputs, USB, and interfaces for i2c/SPI and Texas-Instruments 1-wire protocols. The addition of a Gumstix Tobi expansion board increases the footprint and weight of the Overo but eases integration by breaking out key pins to accessible headers as well as by adding USB-to-serial conversion for the console, 10/100baseT Ethernet, audio/video connection, and on-board voltage regulation. It is possible through manipulation of the bootloader to obtain access to an additional serial port (critical for Flying Fish) and to re-direct the system console to USB, freeing another serial port. The Overo system uses LVCMOS (0.0-1.8V) signal levels, necessitating logic level shifting (LLS) to interface with most other devices.

A custom interface board was developed (Appendix:A) to shift from LVCMOS to TTL (0-5V) logic levels. This logic interface board also provides secondary conversion to RS-232, RS-422, and RS-485 to interface Overo serial ports to the Microbotics, Inc. Servo Switch Controller (SSC) and MIDG-IIC inertial navigation system (INS) as well as the custom maximum power point trackers (MPPT). The off-the-shelf Microbotics SSC was used for our research-oriented flight tests to provide an interface for switching

servo actuation commands between a traditional radio-controlled (RC) receiver and the computer autopilot. The SSC provides a failsafe that allows a standby pilot to take control of the airframe using an RC transmitter. The Microbotics MIDG IIC provides the primary navigation capabilities of the avionics system as discussed below. The MPPTs, also discussed below, have a command and reporting interface that allows the avionics computer to set output limits and collect power data from the solar energy harvesting system. The interface board also provides single-line bi-directional LLS to interface the Overo to an i2c network of three 8-channel 10-bit Analog/Digital converters (ADC) distributed in the airframe. The primary ADC is housed on the logic-interface card; the other two ADCs are located on custom remote-sensing boards in the tops of the vertical wing stanchions. External ADCs were selected over the Overo ADC inputs which, again due to LVCMOS voltage level limits, had too small an input range (0-2.4V) to be useful for the sensors in use on Flying Fish.

External communications are handled through wireless transmission and a wired Ethernet connection for pre/post-flight high-speed data transfer. The primary method of communication between the ground station and flight computer is a Digi XTend 900MHz USB radio-frequency (RF) modem. Providing up to 60km line-of-sight range at full power with a high gain antenna, the 115.2kBd modem provides long-range RF communication. The on-board modem is set to 100mW transmission power to conserve battery life but the ground station operates at a full 1W to help ensure flight commands are received by the aircraft. The RC control system provides a secondary unidirectional wireless interface at 72MHz that allows failsafe recovery via piloted flight. The system also has an active 1.575GHz GPS system. While on the ground, a waterproof connector on the avionics pod provides access to the 10/100baseT Ethernet of the Tobi Gumstix expansion board. This umbilical also carries 5V power and servo signals from the SSC which, in combination with Ethernet, enable modu-

lar external payloads to be interfaced. The Overo supports both 802.11g Wi-Fi and Bluetooth; for Flying Fish they are deactivated. The primary concerns with wireless communication are interference, range, and EM noise. The RC receiver must be isolated to the extent possible from EM noise, as well as range-tested prior to flight. We prioritized in-flight range over water surface range and thus mounted the antenna under the airframe. While we had sufficient range for flight tests, all wireless signals were sharply attenuated by multipath effects induced by radio reflection off the water surface as well as by the composite airframe. For Flying Fish, the RC receiver was mounted with maximum separation from the computer and radio-modem, but still needed to reside on the common avionics shelf given the requirement for a waterproof compartment with minimal penetrations. Wire lengths were minimized and wires that would come near or directly connect to the RC receiver were run in physically-separate conduits from all computer and power wires. The RC antenna exits the rear of the avionics compartment away from both avionics and aircraft structure. This choice minimizes radio signal reflections and attenuation from wing solar panels and carbon composite structure. The radio-modem was mounted with its OEM aluminum case, despite a weight penalty, also to minimize EM interaction with the RC receiver and servos.[1]

### 2.1.3 Sensor Systems

Sensors systems are distributed throughout Flying Fish in waterproof enclosures. Air data sensors are concentrated on the tail of the aircraft, power sensing is concentrated in the floats and vertical wing stanchions, and the central avionics pod houses, as previously discussed, an inertial navigation system and ultrasonic altimeter. Flying

[1] Although Flying Fish could operate unattended given its autonomous flight and energy harvesting capabilities, it remains a University research platform for which backup pilot control is essential for safety. Long-term validation, and reliable identification/avoidance of surface vessels if deployed in an open water environment, remain future work.

Fish also has internal and external temperature sensors distributed throughout the vehicle. In this section we will present the details of the INS, ultrasonic altimeter, air data system, MPPT, and power sensors.

Flying Fish utilizes the Microbotics MIDG IIC commercial off-the-shelf INS which provides a Kalman-filtered vehicle state estimate from a three-axis gyroscope, three orthogonal linear accelerometers, a three-axis magnetometer, and a GPS. The filtered output, raw sensor measurements, and additional data are available at up to 50Hz over the 115.2kBaud differential-signaling RS-422 serial port protocol. The MIDG IIC includes a shielded shock-resistant housing, weighs 55g, and draws less than 1.2W with an active GPS antenna. It operates over a wide range of battery-level voltages by using internal regulation. To minimize errors and the need to apply transformations or rotations to the INS state vector the MIDG IIC has been located on the aircraft center line at the rear of the avionics compartment on the longitudinal center of gravity (CG) aligned with the primary vehicle axes. The primary challenges associated with integration of the MIDG IIC are the unknown quality of the internal sensor signals and the closed nature of the proprietary internal Kalman filter software. In terms of the sensor quality of the MIDG IIC the authors have observed that, while the INS has proven reliable in the field, the 3-axis magnetometer does not produce accurate or repeatable results. The team has therefore had some difficulty measuring and attempting to compensate for magnetic variations at distant testing locations. The lack of information and control of the internal filter precludes any attempt to properly diagnose or correct the problem or its potential effect on the filtered state estimate. A further issue is erroneous altitude estimates upon landing; this perhaps is due to integration of accelerometer signals rather than reliance on GPS data. Subject to the dynamics of a moderately hard water landing the altitude of the vehicle always propagates to well below the current water level and remains incorrect over a nontrivial period of time (on the order of minutes). It is generally

the case that, except for gross measurement and rate of change, Flying Fish does not rely on the INS for altitude estimation.

Accurate low-altitude height-above-water estimation is made using a Senix TSPC-15S waterproof ultrasonic distance measurement system. The Flying Fish team purchased the TSPC-15S without the typical industrial stainless-steel housing, thus reducing its weight sufficiently to mount in a small-medium scale UAS. The TSPC-15S has an optimal range of 0.25m-6m and a maximum range of just over 9m. Since serial interfaces to the avionics computer were required for the MIDG, MPPTs, SSC, and modem the TSPC-15S's analog output is sampled using the 10bit ADC in the main avionics pod. The downward-looking ultrasonic altimeter is installed in the tip of the avionics pod. The major challenges with an ultrasonic altimeter are handling out-of-range measurement signals, dealing with the limited measurement range of the sensor, and measurement degradation subject to misalignment between the sensor and landing surface. Outside of its operating range the sensor cannot distinguish between a delayed pulse echo and the pulse timing of in-range measurements. It therefore is important to distinguish true in-range from false echo measurements. To address this problem we use median filtering for impulsive echo-responses, analysis of the rate of change of the sensor value, and comparison with other altitude sensing systems. For a moderately fast approach to landing the ultrasonic altimeter might only be in its valid operating region a few seconds before touchdown. For Flying Fish we therefore follow a shallow approach to landing, improving our in-range confidence estimation, and responding to valid ultrasonic data with a moderate flare only within the last two meters of descent. Another issue is sensor orientation with respect to the measurement surface. The nature of reflected-signal sensors is that they require, subject to material properties, approximately orthogonal incidence of the signal and the measured surface. For an aircraft this means the ultrasonic altimeter is unreliable during banked turning, high-pitch takeoff climb-out, and under other circumstances

requiring high pitch or roll angles with respect to the landing surface. Shape of the surface being imaged may also be a factor. Over any ocean surface with significant wave structure the normal plane of the water can change dramatically over time. Our tests have demonstrated that, while the signal may be degraded, sufficient locally-level reflective surface is present in most wave structure to return suitable power from an ultrasonic pulse to make a valid measurement.

Additional physical measurements of the environment are made by the Flying Fish air data system (ADS) which includes redundant 5-hole air-data probes located on the dual vertical tails of the aircraft and a propeller-anemometer above the left vertical stabilizer. The 5-hole probes combine a pitot-static measurement for airspeed ($V$) with differential pressure measurements for angle-of-attack ($\alpha$) and angle-of-sideslip ($\beta$) determination. The aerodynamic effects of air passing over the body and hemispherical tip of the 5-hole probes are measured via custom remote ADC boards referenced in the previous section. The ADC boards (Appendix: A) provide a digital interface to the main computer system for eight analog input channels, the first four of which are utilized by onboard ADS pressure sensors including: one gauge sensor for airspeed, two differential sensors for $\alpha$ and $\beta$, and a barometric-range sensor for altitude. The propeller-anemometer produces a hall-effect-based digital pulse train measured via an auxiliary digital input to the SSC. A major challenge faced by Flying Fish is water-fouling of the 5-hole probe ports. Forcible water ingress to pressure ports due to the energetic transit of the vehicle through the water is a major problem resulting in erroneous airspeed input to the flight controller. To address this problem, we added a second pitot probe and probe heating for evaporative water-evacuation. Additionally, an air-data fault detection and mitigation filter combines airspeed measurements from the redundant sensors with vehicle motion and wind estimates to produce higher-confidence airspeeds for the flight controller.[6] Additional challenges facing this system include calibrating the 5-hole probe given its sensitivity to minor

manufacturing variations and the desire for the propeller anemometer to measure full flight speed (20m/s) while remaining sensitive to low-speed wind (<1m/s). Wind-tunnel tests were conducted with the complete ADS probe systems to develop basic calibrations; calibration refinements were made from flight test data.

The MPPTs provide sensing capabilities in addition to regulating power point. Each of the eight MPPTs reports the solar-array voltage, output voltage (battery voltage), output current (distributed charge rate, less avionics load), and internal and external temperatures. The external temperature sensors include: 2 air temperatures (under the wing), 2 water temperatures (under the port and starboard floats), 2 motor temperatures (in the port and starboard motor booms), 1 battery temperature in the starboard float, and 1 voltage regulator temperature in the port float. The temperature probes are thermistors that require waterproofing. For Flying Fish the probes were potted into narrow tubular housings with integral strain relief using thermally-conductive epoxy. Each probe was flush-mounted to the composite structure.

The final sensors are battery voltage and current draw. The i2c ADC network reads battery voltage with calibrated voltage division; it also measures four current sensors in the aircraft floats. The hall-effect current sensors monitor current to the motors (up to 100A) and to the servos (up to 25A) but have proven noisy given their proximity to the motor control and avionics bus regulation devices. The team is exploring shielding for these sensors; the addition of capacitive filtering near the sensors has reduced but not eliminated noise issues.

## 2.2 Flight Management System

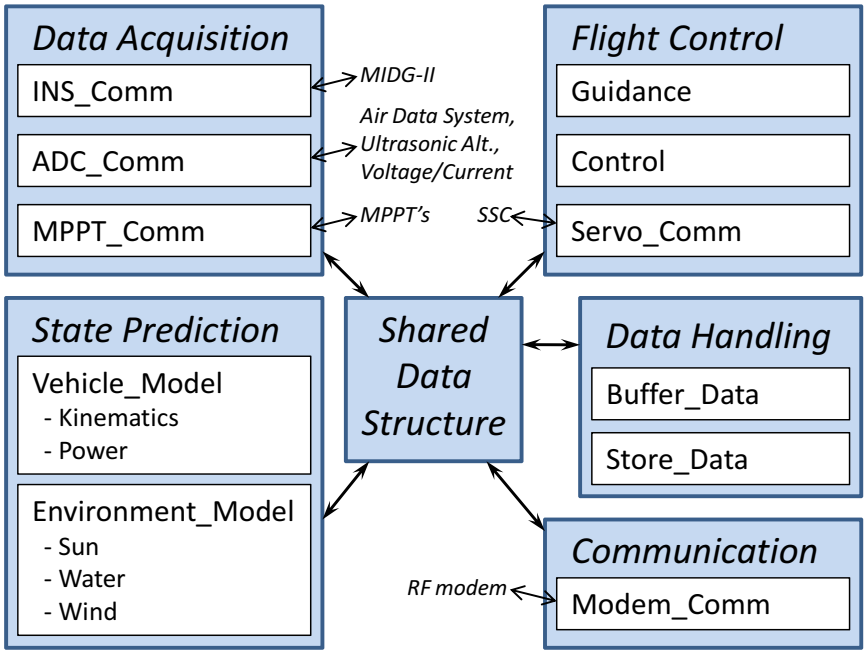



Figure 2.5: Wolverine FMS Architecture

Flight management systems (FMS) first became standard equipment on major commercial aircraft in 1982 where they were used to offload vehicle management tasks from the pilot and, as a result of spiking aviation fuel costs in the preceding decade, to plan and guide fuel-optimized trajectories.[7] The proliferation of these systems marked a major turning point for flight software as the FMS was, at that time, the most software-intensive system onboard a commercial aircraft. The purpose of the FMS goes beyond that of an autopilot to include: flight planning, navigation, guidance, performance prediction and optimization, communication, control, and the management of user interfaces.[7, 8] In the age of the glass cockpit with fuel costs rising again the FMS plays an even greater role in system management and flight optimization. Flight management systems are generally comprised of one or more flight management computers executing the collected FMS software, various communication equipment, and a user control/display unit that serves as an interface for

the aircraft pilot/operator. In the 21st century, modern FMS have begun to evolve into networked collections of processing, sensing, and actuation sub-systems that are distributed throughout an airframe. The Full Authority Digital Engine Control (FADEC) exemplifies this concept; being itself a distributed sensing and control system, the FADEC provides complete computer control of engine operating parameters as an integrated element of the larger distributed sensing and command systems of tan aircraft.[9] Regardless of their design commercial FMS traditionally employ triple-redundancy for critical system [10, 11] to meet safety certification requirements; FMS for UAS may not be triply redundant, favoring reduced mass and cost over increased risk of failure given that no humans are onboard.

The Flying Fish Flight Management System manages sensors, communications, data acquisition, and guidance, navigation, and control to achieve energy and mission goals. The Flying Fish FMS is based on the Wolverine FMS, an open-source flight management system development in the Autonomous Aerospace Systems Laboratory of the University of Michigan's Department of Aerospace Engineering.[12] In this section we will overview Wolverine FMS for the Phase II Flying Fish with focus on prediction and optimization augmentations to manage energy.

Wolverine FMS is a multi-threaded C-based flight management architecture that employs shared memory with common data structures and a standardized modular framework to support fully-autonomous UAS flight. It has been adapted to three platforms to-date, including an aerobatic Funtana,[13] a highly-flexible flying-wing research UAS (X-HALE),[14], and the first[15] and Phase II Flying Fish.[6] The goal of the Wolverine FMS is a suitably-complete but efficient set of data objects and related methods to make basic FMS implementation possible with few hardware-specific interface modules. While approximating some features of object-oriented programming (OOP), such as modularity and data encapsulation, Wolverine FMS is ultimately procedurally-driven given its C implementation to maximize speed and

applicability to different processors. Figure 2.5 shows Wolverine FMS as it has been adapted to Flying Fish. The primary FMS tasks can be grouped into five categories: data acquisition, state prediction, autopilot, communication, and data handling.

Data acquisition tasks involve polling, sampling, or decoding sensor data and depositing the resulting information in the correct shared data structure(s). Hardware-specific customization of the Wolverine FMS for Flying Fish is concentrated in the data acquisition system as sensors and interfaces are system-specific. The data acquisition module executes the following procedures: INS_Comm, ADC_Comm, and MPPT_Comm. INS_Comm buffers and parses scheduled messages for the filtered vehicle state, acceleration, and raw GPS from the MIDG IIC. ADC_Comm polls the A/D converter network via i2c collecting battery voltages and currents, ADS pressures, and ultrasonic altimeter measurements. MPPT_Comm polls each MPPT sequentially on the multi-slave RS/422 serial network for six parameters: voltage in, voltage out, current out, internal temperature, external temperature, and MPPT status. Taking advantage of the MPPT input buffer to minimize response delay, the executive sends all six requests at once. Accommodating the potential maximum response delay of 0.06s makes the MPPT bus the slowest in the system. With eight MPPTs on the same bus the per-MPPT update rate is limited to 2Hz, sufficient for energy monitoring purposes. The MPPT module implements an independent timer to avoid collisions on the multi-drop master/slave bus by automatically throttling message rates to the MPPT regardless of the execution rate of the master program.

State estimation tasks are concerned with the development of flight performance estimates subject to predicted environmental conditions. The state estimation portion of the Wolverine FMS executes the following procedures: Vehicle_Model and Environment_Model. These modules highlight the two broad categories of prediction required for energy-aware flight management and planning. Vehicle_Model, divided into flight and energy models, provides a mechanism for estimating the state of the

vehicle over time. Flight prediction estimates flight performance for a given plan while energy prediction estimates the energy collected and expended during plan execution. Environment_Model, divided into solar, wind, and water models, provides estimates of short and long term environmental conditions. The solar model provides an estimate of the sun's position and solar spectral power density. The wind model provides an estimate of wind speed and direction. Finally, the wave model provides estimates of wave structure and the speed and direction of ocean currents. Details of these models are below.

The autopilot converts state data, mission parameters, and queued operator commands into valid guidance and control actions. The autopilot executes procedures Guidance, Control, and Servo_Comm. Guidance is responsible for converting navigation data into desired reference commands that follow the desired flight path. Control is responsible for determining appropriate actuator responses to track Guidance outputs. Flying Fish guidance and control are based on mode-switched, gain-scheduled guidance and control laws that provide appropriate outputs for each phase of flight.[15] Servo_Comm is responsible for converting and communicating the desired actuator commands to the Microbotics SSC for Flying Fish.

Communication is responsible for the collection and delivery of telemetry between onboard FMS and the ground station (GS). For Flying Fish, a serial RS/232 wireless modem link is driven by the FMS Modem_Comm function. To minimize impact on critical guidance navigation and control (GNC) processes and to avoid overflow, Modem_Comm manages a message queue that divides available bandwidth between messages based on relative size. Under changing telemetry requirements every full set of messages can be broadcast on a customizable schedule. Otherwise, if the messages and timings remain fixed the same schedule is executed repeatedly. To develop a schedule Modem_Comm calculates the total message size of the current telemetry payload and allocates time slices to each message based on their percentage of the

entire telemetry payload. Any backups due, for example, to wireless communication errors are either eliminated during idle time at the end of the time-slice or in subsequent time slices provided sufficient bandwidth is available. This system provides the FMS with real-time feedback of the modem's current performance given the transmission schedule under current conditions. While not implemented in this version, this capability could be extended to negotiating telemetry in real-time with the GS.

The final task, data handling, is concerned with the short-term buffering of flight data and long term storage of mission data. Data handling employs two cooperative procedures: Buffer_Data and Store_Data. Buffer_Data samples pertinent shared data at a fixed frequency and records it into circular buffers in system memory. Store_Data draws data from the circular buffers and commits it to non-volatile storage. This design addresses challenges of delay management with an operating-system-based (OS) computer system. Within a Linux disk manager it is the system kernel and not the user software that arbitrates the physical transfer of cached data to disk. With disk writing commonly amongst the slowest processes, kernel-arbitrated disk write events can give rise to non-negligible asynchronous delays in the software execution rate. In a flight system such delays can lead to destabilization of a digital control law through time delay or even periodic loss of control. Our threaded Flying Fish data manager employs the Buffer_Data and Store_Data procedures to buffer data to memory during flight, when execution rates are critical, and push/store data to disk during drift, when control rates are less important. Executing at 30Hz the data acquisition code can sample and buffer shared data to memory, free from disk delays, for over 2hrs. At the termination of each flight, before each takeoff, and prior to a buffer overflow the data manager compacts and dumps the buffer to disk. If the buffer is filled during flight FMS settings allow for either a temporary suspension of data recording or overwriting the beginning of the circular buffer. Since the data is stored in a compressed binary format a new set of data files is created whenever buffering resumes in order

to minimize the risk for collateral data loss in the event of file corruption. The data manager can store data to disk for every data structure at a full 30Hz for over 40hrs without an off-board storage device.

# CHAPTER III

# Dynamics and Kinematics of a Seaplane

This chapter presents the fundamental dynamics models applicable to the Flying Fish seaplane. These models provide a framework for the development and analysis of flight control and guidance laws, and support vehicle performance estimates in flight and mission planning algorithms. This chapter will first summarize the dominant forces and moments effecting a flight vehicle. The equations of motion will be analyzed for equilibrium conditions from which the conditions and equations of stable, steady, trimmed flight will be developed. Vehicle kinematics will also be developed and simplified kinematic models will be presented for later use in guidance and mission planning tasks. Finally we will present estimates of combined aero-hydrodynamic phenomena affecting an operational seaplane. Throughout the chapter we will discuss the specific considerations and impact of seaplane design on aircraft dynamics.

## 3.1 Aerodynamic Forces and Moments

The fundamental aerodynamic forces acting on an aircraft are most commonly expressed as the collected set of three orthogonal wind-axes forces: aerodynamic lift ($L$), drag ($D$), and side-force ($Y$) as well as weight ($W$) and thrust ($T$). Rotational dynamics are typically expressed as three orthogonal aerodynamic body-axis moments

named for the associated rotational behavior, specifically: rolling ($\mathcal{L}$), pitching ($\mathcal{M}$), and yawing ($\mathcal{N}$) moments. In this section we will expand upon the fundamental forces and torques acting on a flight vehicle following the nomenclature, style, and conventions used by [16] to enable the development of aircraft equations of motions in the following section.

Consider a flight vehicle moving at steady sub-sonic speeds in a stationary low-stratosphere standard atmosphere. We assume non-aerobatic flight is prescribed, that the wing, tail, and control surfaces remain un-stalled, and that control deflections remain within command limits. These assumptions are generally acceptable for Flying Fish as the mission profiles do not indicate aerobatic flight maneuvers and the vehicle is expected to operate at relatively low speeds and altitudes ($V < 20^{\mathrm{m}}/_{\mathrm{s}}$, $h < 100\mathrm{m}$) with relatively high command authority in all control surfaces. We assume flights will be conducted well under stall angles of attack thus air flow is expected to be approximately laminar. The wing flow Reynolds number starts near $R_e = 4 \times 10^5$ at the stall speed of the vehicle and extends upwards through the nominal transitional range to approximately $R_e = 7 \times 10^5$ at the maximum speed of the vehicle. Anecdotally the light uniform texture of the Kevlar composite wing built for Flying Fish appears to have a positive effect on flow and boundary layer properties as observed during flight segments with high angles of attack, providing overall flow properties that would tend to keep the predominance of flow laminar for the majority of flight conditions. The effects of the semi-conformal faceted solar array on wing airflow have not been fully characterized but the most likely impact, a slight increase in drag and perhaps a degradation of stall characteristics, does not effect the mathematical development in this section.

### 3.1.1 Aerodynamic Forces

Aerodynamic lift, the most basic requirement of fixed-wing flight, is directed vertically or longitudinally in the plane of symmetry of the aircraft, perpendicular to aircraft velocity, with a magnitude given by:

$$L = \frac{1}{2}\rho V^2 S C_L \tag{3.1}$$

In this flight regime lift is reasonably assumed to be linearly proportional to the dimensionless lift coefficient $C_L$ scaled by the dynamic pressure $\frac{1}{2}\rho V^2$ exerted by the the atmosphere at the flight speed $V$. Lift is scaled, by convention, to the characteristic dimension of the wing planform area $S$. Below stall the dimensionless lift coefficient is reasonably approximated by a linear function of the aircraft angle of attack as:

$$C_L = C_{L_0} + C_{L_\alpha}\alpha \tag{3.2}$$

The lift coefficient is a combination of the zero angle of attack lift coefficient $C_{L_0}$ and the angle of attack lift slope $C_{L_\alpha} > 0$. A symmetric airfoil, for example, has a $C_{L_0} = 0$. The NACA 2414 wing profile was used on the Phase II Flying Fish. If we consider a perfect infinite wing we find, from wind-tunnel testing, that the NACA 2414 has $C_{L_0} \approxeq 0.23$ and $C_{L_\alpha} \approxeq 0.12$ for an angle of attack expressed in degrees. In practice it is often necessary to use a wind-tunnel to determine accurate vehicle lift coefficients which must be corrected for wing geometry variations due to chord and thickness taper, wing sweep, twist, tips effects of a finite wing, and the influence of connected aerodynamic structures.

The aerodynamic drag force is directed opposite the velocity vector with magnitude expressed as:

$$D = \frac{1}{2}\rho V^2 S C_D \tag{3.3}$$

Much like the lift force, drag is a linear function of a dimensionless coefficient and the dynamic pressure exerted by the atmosphere. Unlike lift, however, the drag coefficient is a quadratic expression of the coefficient of lift, and hence of the angle of attack:

$$C_D = C_{D_0} + KC_L^2,\ K = \frac{1}{\pi e \mathbf{AR}} \tag{3.4}$$

$$C_D = C_{D_0} + K(C_{L_0} + C_{L_\alpha}\alpha)^2 \tag{3.5}$$

where $\mathbf{AR} = \frac{b^2}{S}$ is the aspect ratio of the wing, determined from the wing area $S$ and wingspan $b$, and the empirical Oswald efficiency factor of the wing, $e$. The drag coefficient equation 3.5 expresses the cumulative drag coefficient as a zero-lift drag due to viscous drag effects $C_{D_0} > 0$ and an angle-of-attack-dependent drag term. The latter term, known as the induced drag, indicates that a portion of drag is due to the generation of lift by the wing. The former term is of particular importance for seaplane dynamics as the addition of hydrodynamic structures tend to result in higher vehicle form drag. This term is particularly important for the floatplane-style configuration as the buoyant structures represent sizable fixed external drag bodies. Considering only surface area, the skin-friction viscous drag of symmetric floats can be 5-10 times that of a fixed tricycle gear aircraft with aerodynamic fairings (at the scale of Flying Fish). Float structures also tend to have a significant weight penalty as they have to be able to withstand high hydrodynamic loading during takeoff and descent.

For flight conditions where the velocity vector does not lay in the plane of symmetry the vehicle is said to be slipping or skidding and an orthogonal side-force develops. The aerodynamic side-force exerted on a slipping aircraft is directed perpendicular to the velocity and lift vectors in the direction of the aircraft's nose relative to the wind vector with magnitude:

$$Y = \frac{1}{2}\rho V^2 S C_Y \tag{3.6}$$

The form of the side-force follows that of the lift and drag forces and, just as lift is proportional to angle of attack, the side-force coefficient is linearly dependent on the sideslip angle $\beta$. Notably the sideslip coefficient has no zero-slip bias term as side-force is zero when the velocity vector is in the plane of symmetry:

$$C_Y = C_{Y_\beta}\beta \tag{3.7}$$

Positive sideslip gives rise to side-force directed towards the negative side of the plane of symmetry and as such one common convention for the sideslip coefficient is to take $C_{Y_\beta} < 0$ such that positive $\beta$ produces a negative force. It is generally undesirable for an aircraft to fly subject to non-zero side force. Given that the Flying Fish need not satisfy runway-constrained landing trajectories and can be reasonably assumed capable of coordinated turns over its commanded flight envelope (sufficient rudder is available to cancel aerodynamic slipping at all mission-commanded bank angles) we can subsequently assume that under nominal flight conditions the side-force is approximately zero.

The two additional forces of note are vehicle weight $W$ and installed thrust $T$. For an all-electric flight vehicle with no mission-separable payloads the flight weight remains constant and is directed downward, in the direction of gravity, where gravitational acceleration constant $g = 9.806^{\mathrm{m}}/_{\mathrm{s}^2}$. The mass of the aircraft is then given by: $m = {}^{W}/_{g}$. The thrust force is assumed to be directed along the aircraft's body $x$-axis, out the nose of the vehicle. To develop an expression for thrust we first write an equation for the power output of the propeller which, neglecting rotation, is simply the product of the thrust $T$ and flight speed $V$:

$$\mathcal{P}_{\text{prop}} = T \cdot V \tag{3.8}$$

We can relate the unknown power output of the propeller to the known input power

(i.e., motor shaft power) using the definition of power efficiency:

$$\eta_{\text{prop}} = \frac{\mathcal{P}_{\text{out}}}{\mathcal{P}_{\text{in}}} = \frac{\mathcal{P}_{\text{prop}}}{\mathcal{P}_{\text{motor}}} \tag{3.9}$$

Given that the power of an electric motor is not a function of altitude we can express thrust as:

$$T = \frac{\eta_{\text{prop}} \cdot \mathcal{P}_{\text{motor}}}{V} \tag{3.10}$$

Unfortunately, while the propeller input power $\mathcal{P}_{\text{motor}}$ (i.e., motor output power) can be characterized using the motor manufacturer's specifications and also estimated, in situ, from power consumption during flight, the propeller efficiency $\eta_{\text{prop}}$ is a complex function of air density, airspeed, and propeller configuration. Alternate expressions for thrust and power can be derived to formulate expression for propeller efficiency in terms of known system variables.[17] Thrust from a propeller can be expressed using first-principle momentum by treating the propeller as a device that increases the velocity of an incoming mass of air to some exit velocity $V_e$:

$$T = \frac{1}{2}\mathcal{A}_{\text{prop}} \cdot \rho(V_e^2 - V^2) \tag{3.11}$$

where $\mathcal{A}_{\text{prop}}$ is the area swept by the blades of the propeller and $\rho$ is the density of the incoming air. If we express the exit velocity as delta function of input velocity we can derive an alternative formulation of the fundamental momentum-theory thrust equation:

$$V_{out} = V + \Delta V \tag{3.12}$$

$$T = \mathcal{A}_{\text{prop}} \cdot \rho \left( V + \frac{\Delta V}{2} \right) \Delta V \tag{3.13}$$

If we can formulate an expression for $\Delta V$ independent of thrust we can equate (3.10) to (3.13) to develop a relationship between velocity, power, and efficiency. We first

write an expression for motor input power in terms of the velocity, ignoring rotational losses:

$$\mathcal{P}_{\text{motor}} = T\left(V + \frac{\Delta V}{2}\right) = TV + T\frac{\Delta V}{2} \tag{3.14}$$

into which we substitute the relationships in Eqs. (3.8), (3.9), and (3.10) to obtain:

$$\mathcal{P}_{\text{motor}} = \mathcal{P}_{\text{prop}} + T\frac{\Delta V}{2} = \eta_{\text{prop}}\mathcal{P}_{\text{motor}} + \frac{\eta_{\text{prop}} \cdot \mathcal{P}_{\text{motor}}\Delta V}{2V} \tag{3.15}$$

The motor power term, $\mathcal{P}_{\text{motor}}$, can be divided out of Eq. (3.15) and Eq. (3.15) can be subsequently be reformulated and solved for $\Delta V$:

$$\eta_{\text{prop}} + \frac{\eta_{\text{prop}}\Delta V}{2V} = 1 \tag{3.16}$$

$$\Delta V = \frac{2V(1-\eta_{\text{prop}})}{\eta_{\text{prop}}} \tag{3.17}$$

We next divide Eq. (3.14) by thrust, $T$, to obtain:

$$\left(V + \frac{\Delta V}{2}\right) = \frac{\mathcal{P}_{\text{motor}}}{T} = \frac{\mathcal{P}_{\text{motor}}}{\left(\frac{\eta_{\text{prop}}\mathcal{P}_{\text{motor}}}{V}\right)} = \frac{V}{\eta_{\text{prop}}} \tag{3.18}$$

Substituting Eqs. (3.17) and (3.18) into Eq. (3.13) yields:

$$\frac{\eta_{\text{prop}}\mathcal{P}_{\text{motor}}}{V} = \mathcal{A}_{\text{prop}} \cdot \rho\left(\frac{V}{\eta_{\text{prop}}}\right)\frac{2V(1-\eta_{\text{prop}})}{\eta_{\text{prop}}} \tag{3.19}$$

which cannot be solved for the efficiency term $\eta_{\text{prop}}$ but can still be reduced to expression relating efficiency to known quantities:

$$\frac{\eta_{\text{prop}}^3}{(1-\eta_{\text{prop}})} = 2\mathcal{A}_{\text{prop}} \cdot \rho\left(\frac{V^3}{\mathcal{P}_{\text{motor}}}\right) \tag{3.20}$$

We can replace the area swept by the propeller with a function of the propeller diameter $d$, namely: $\mathcal{A}_{\text{prop}} = (\pi/4)\, d^2$ and reorganize the result to write an expression

that is well suited to a numeric solution:

$$V = \eta_{\text{prop}} \left( \frac{2}{\pi \cdot \rho(1 - \eta_{\text{prop}})} \left( \frac{\mathcal{P}_{\text{motor}}}{d^2} \right) \right)^{\frac{1}{3}} \tag{3.21}$$

If we characterize the motor and propeller combination by the ratio of power to squared diameter $\mathcal{P}_{\text{motor}}/d^2$ we can develop efficiency curves for a propulsive system as a function of velocity for a given air density (Fig. 3.1, $\rho = 1.225^{\text{kg}}/_{\text{m}^3}$). Note that these curves represent an upper limit on efficiency for an optimal propeller without friction or rotational losses; actual performance will generally be reduced by 10% or more with the greatest inaccuracy at the extremities of performance. At the maximum cruise velocity $V \approx 20^{\text{m}}/_{\text{s}}$ we find that the 1.8kW main motors have a peak efficiency of $\eta_{\text{prop}} = 0.83$ and the 1.2kW boost motor has a peak efficiency of $\eta_{\text{prop}} = 0.88$. The main and boost motors have efficiencies of $\eta_{\text{prop}} = 0.73$ and $\eta_{\text{prop}} = 0.79$ at the liftoff speed $V \approx 15^{\text{m}}/_{\text{s}}$, respectively. If we de-rate the efficiencies by 20%, the resulting first principle estimate of peak liftoff thrust is $T = 66\text{N} + 66\text{N} + 56\text{N} = 178\text{N}$ from the two main motors plus the boost motor. Similarly, the high-speed cruise peak thrust is $T = 56\text{N} + 56\text{N} = 112\text{N}$ from the two main motors. These numbers agree with the bench-tested static thrust. Using this information we can use a curve-fit of the propulsive efficiency per motor, or an average efficiency for a trimmed flight condition, and rewrite Eq. (3.10) to include a throttle input $0 < \delta_t < 1$ as:

$$T = \delta_t \frac{\eta_{\text{prop}}(V, \mathcal{P}_{\text{motor}}, d) \cdot \mathcal{P}_{\text{motor}}}{V} \tag{3.22}$$

This model will serve two purposes. In this chapter, when we begin to discuss steady flight conditions, it provides insight to the conditions required for a trimmed throttle setting and in Ch. VI we will be able to use this model to estimate thrust performance within the flight planning module.

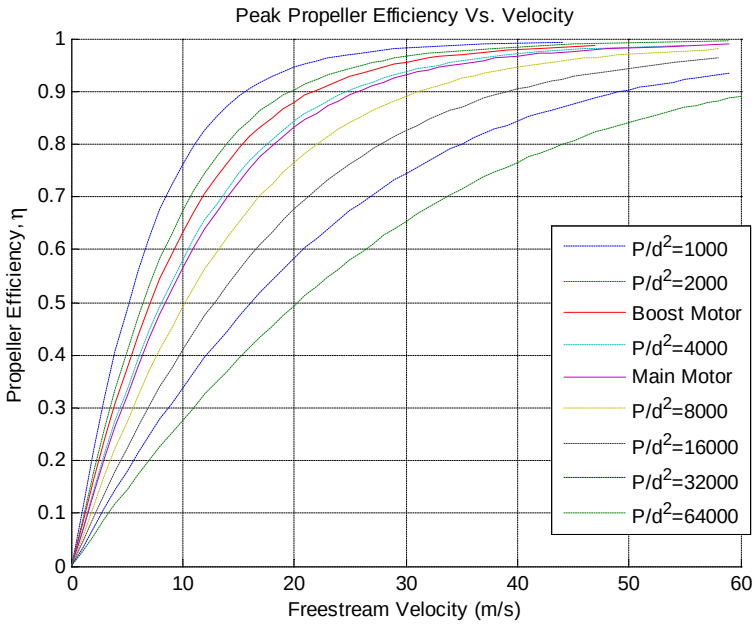



Figure 3.1: Peak Propeller Efficiency

### 3.1.2 Aerodynamic Moments

The aerodynamic rolling moment $\mathcal{L}$ gives rise to rotation about the aircraft body $x$-axis, which is directed by common convention, out the nose of the aircraft. Positive rolling moment generates a right roll tendency. Aircraft moments are typically functions of vehicle configuration and control surface deflections and are typically expressed in aircraft body axes. The standard control surfaces on an aircraft are the rudder, elevator, and ailerons which primarily induce yawing, pitching, and rolling moment, respectively, although the moments induced by control inputs are cross-coupled to some degree in most aircraft. The control vector of the standard inputs: $\delta_t$ (throttle), $\delta_a$ (aileron), $\delta_e$ (elevator), and $\delta_r$ (rudder) is defined as:

$$\vec{\delta} = \begin{bmatrix} \delta_t & \delta_e & \delta_a & \delta_e \end{bmatrix}^T \tag{3.23}$$

The aircraft moment equations are also typically scaled to the characteristic dimension of the wing planform area $S$ and a moment arm, chosen by common convention for each axis as provided in the reference text for this material.[16] The convention for rolling moment is to take the wingspan $b$ as the moment-arm to yield the following expression for the magnitude of the rolling moment:

$$\mathcal{L} = \frac{1}{2}\rho V^2 S b C_{\mathcal{L}} \tag{3.24}$$

Like the orthogonal aerodynamic forces the aerodynamic moments scale to the dynamic pressure and a dimensionless coefficient. The coefficient of the rolling moment is weakly influenced by rudder input and strongly influenced by differential aileron input and sideslip:

$$C_{\mathcal{L}} = C_{\mathcal{L}_\beta}\beta + C_{\mathcal{L}_{\delta_a}}\delta_a + C_{\mathcal{L}_{\delta_r}}\delta_r \tag{3.25}$$

Generally we can say that $C_{\mathcal{L}_{\delta_a}} > C_{\mathcal{L}_\beta}, C_{\mathcal{L}_{\delta_r}}$. One accepted convention for rudder deflection is to define positive rudder deflection as the direction that induces positive moment about the vehicle $z$-axis, directed out the belly of the aircraft, or more succinctly: $C_{\mathcal{L}_{\delta_r}} > 0$. For an aircraft in the class of Flying Fish for which stability is paramount we require $C_{\mathcal{L}_\beta} \leq 0$ to provide roll stability. This condition requires that for a positive roll deflection, the resulting positive sideslip will give rise to either a restoring negative roll moment (dihedral effect) or else no additional roll to precipitate roll divergence. We will select a similar convention for aileron deflection: a positive rotation of the right aileron about the aircraft body $x$-axis will be considered a positive aileron deflection and (assuming opposite deflection of the left aileron) give rise to a negative rolling moment, so $C_{\mathcal{L}_{\delta_a}} < 0$.

By convention a positive pitching moment about the aircraft $y$-axis, directed out the

right wing of the aircraft, produces a nose-up tendency with magnitude:

$$\mathcal{M} = \frac{1}{2}\rho V^2 S c C_{\mathcal{M}} \tag{3.26}$$

Here the moment-arm is selected by convention as the mean wing chord $c$. The pitching moment coefficient is dependent on the angle of attack and elevator deflection and typically has a zero angle of attack term $C_{\mathcal{M}_0}$ that indicates a non-zero pitching moment even at zero angle of attack:

$$C_{\mathcal{M}} = C_{\mathcal{M}_0} + C_{\mathcal{M}_\alpha}\alpha + C_{\theta_{\delta_e}}\delta_e \tag{3.27}$$

One convention for elevator deflection states that positive deflection is given by positive rotation about the aircraft $y$-axis which produces a downward elevator deflection and negative pitching moment, therefore $C_{\mathcal{M}_{\delta_e}} < 0$. Stability requires that $C_{\mathcal{M}_\alpha} < 0$ for the aircraft else increasing angle of attack would give rise to increasing moment which would subsequently increase the angle of attack to the point of stall. The average cambered airfoil, which is to say specifically not a symmetric or reflex airfoil, has at least a small negative pitching moment at zero angle of attack which is usually balanced by selection of the installed incidence of the horizontal tail to give downward lift to balance the pitching moments. Pitching moment offsets are otherwise balanced by elevator trim under given flight conditions characterized by the aircraft velocity and angle of attack.

The aerodynamic yawing moment, positive for positive rotation about the aircraft $z$-axis (directed out the belly of the aircraft) produces a nose-right yaw tendency with a torque magnitude of:

$$\mathcal{N} = \frac{1}{2}\rho V^2 S b C_{\mathcal{N}} \tag{3.28}$$

Like the roll moment the characteristic moment arm of the yaw moment is selected

by convention as the wingspan of the vehicle. The yaw force coefficient is strongly dependent on sideslip and rudder input and weakly dependent on (and preferably independent of) aileron input:

$$C_{\mathcal{N}} = C_{\mathcal{N}_\beta}\beta + C_{\mathcal{N}_{\delta_a}}\delta_a + C_{\mathcal{N}_{\delta_r}}\delta_r \tag{3.29}$$

Almost all aircraft are subject to $C_{\mathcal{N}_\beta} > 0$ as the side-force generated by slipping flight acts on the vertical tail to produce restoring moment. That is, for a positive sideslip, wherein the velocity vector is to the right of the aircraft nose, the side-force (Eq. 3.6) on the vertical tail (which is behind the aircraft center of gravity) gives rise to a positive yaw moment directing the nose back towards the velocity vector. Following the stated convention for rudder deflection, positive deflection for positive rotation about the $y$-axis, we find that positive rudder yields negative yawing moment, therefore: $C_{\mathcal{N}_{\delta_r}} < 0$. We desire no yawing moment from aileron deflection but in practice the differential deflection of the ailerons yields differential effective camber/lift on the left and right wings and subsequently creates a drag differential. During a positive right roll, for example, the left aileron is deflected downward increasing effective camber and lift and induced drag on the left wing (by the square of the coefficient of lift as given in Eq. 3.3) to produces a negative yawing moment, commonly referred to as adverse yaw. Following the stated convention for aileron deflection we find $C_{\mathcal{N}_{\delta_a}} > 0$ as a positive aileron deflection generates a negative roll, increased right wing drag, and positive yawing moment.

## 3.2 Aircraft Dynamics and Equilibrium (Trim) States

The aerodynamic forces and moments can now be collected into first-order equations of motion for the aircraft from which we will develop equilibrium equations and trim conditions. For clarity we will again adopt the notation and conventions and follow

a similar line of derivation to that present in McClamroch's text on steady flight.[16] Starting from Newton's second law of motion ($F = ma$) we can set the product of the aircraft mass and acceleration along the velocity vector equal to the sum the forces also in the direction of the velocity vector:

$$\frac{W}{g}\frac{dV}{dt} = -W\sin(\gamma) - D + T\cos(\alpha) \tag{3.30}$$

This equation relates the velocity-direction acceleration to the component of aircraft weight directed along the velocity vector at the given flight path angle $\gamma$, the drag $D$ (aligned with the velocity vector), and the component of thrust $T$ in the velocity-vector direction given an offset by the aircraft angle of attack. We can also write the force equations in the lateral (alternately denoted as the *radial* direction for a turning vehicle) direction as:

$$\frac{W}{g}V\cos(\gamma)\frac{d\sigma}{dt} = L\sin(\mu) + T\sin(\alpha)\sin(\mu) \tag{3.31}$$

This equation relates the rotating aircraft rate of heading change to the lift component directed into the turn for the given bank angle $\mu$ and the thrust component directed into the turn for the given bank and angle of attack. Finally we write the vertical flight equation relating the rate of change of the flight path angle to the weight, lift, and thrust directed perpendicular to the velocity vector.

$$\frac{W}{g}V\frac{d\gamma}{dt} = -W\cos(\gamma) + L\cos(\mu) + T\sin(\alpha)\cos(\mu) \tag{3.32}$$

Subsequently we can write the collected translational dynamics of the vehicle, given the vehicle velocity, flight path angle and heading as:

$$\frac{dx}{dt} = V\cos(\sigma)\cos(\gamma) \tag{3.33}$$

$$\frac{dy}{dt} = V\sin(\sigma)\cos(\gamma) \tag{3.34}$$

$$\frac{dz}{dt} = V\sin(\gamma) \tag{3.35}$$

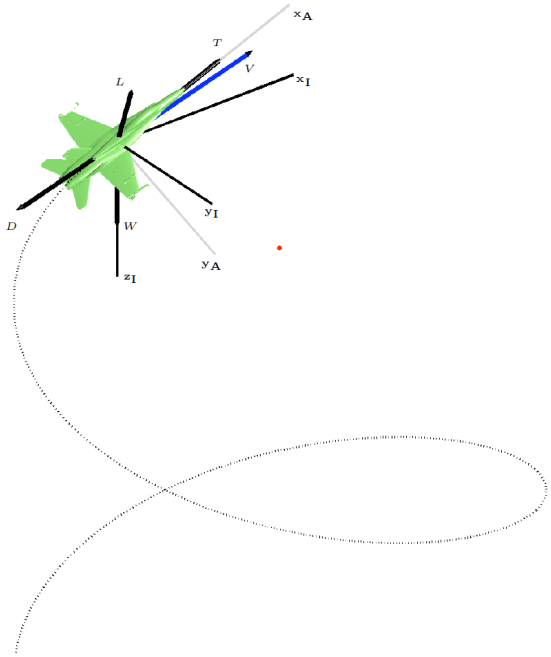
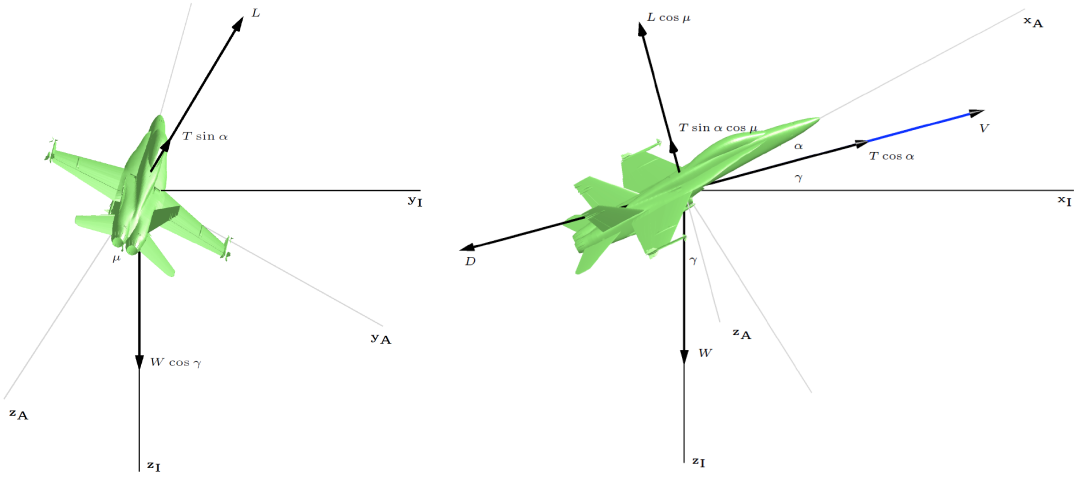



Figure 3.2: Steady Helical Flight, Perspective View[16]




(a) Rear View

(b) Lateral View

Figure 3.3: Steady Helical Flight, Detail Views[16]

By analyzing the unaccelerated cases of Eqs. (3.30)-(3.32) we can begin to develop steady flight equations. If we hold velocity and flight path angle steady and fix the radial acceleration we recover the most generic case of steady flight: steady climbing/descending constant velocity flight with a continuous fixed turn rate. These conditions prescribe a helical flight path (Fig. 3.2-3.3).

Formally, steady helical flight with a fixed radius $R$ requires: non-accelerated flight, constant flight path angle, constant centrifugal acceleration, and zero aerodynamic pitching moment.

$$\frac{dV}{dt} = \frac{d\gamma}{dt} = 0 \tag{3.36}$$

$$\mathcal{M} = 0 \tag{3.37}$$

$$\frac{d\sigma}{dt} = \frac{V\cos(\gamma)}{R} \tag{3.38}$$

From these equations the equations of steady helical flight can be written, keeping in mind that vehicle weight is constant for our electric seaplane:

$$-W\sin(\gamma) - D + T\cos(\alpha) = 0 \tag{3.39}$$

$$L\sin(\mu) + T\sin(\alpha)\sin(\mu) = \frac{W}{g}\frac{V^2\cos^2(\gamma)}{R} \tag{3.40}$$

$$W\cos(\gamma) - L\cos(\mu) - T\sin(\alpha)\cos(\mu) = 0 \tag{3.41}$$

If we assume that the angle of attack and flight path angles are small we can simplify these equations further by replacing sine and cosine with linear approximations based on a Taylor Series expansion of the trigonometric functions:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots \tag{3.42}$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots \tag{3.43}$$

Given that $x$ is small, $x$ raised to any power $n \geq 2$ can be considered small and subsequently negligible such that the Taylor series approximations can be reasonably truncated to their linear terms:

$$\sin(x) \approxeq x \tag{3.44}$$

$$\cos(x) \approxeq 1, \tag{3.45}$$

The percent error of the approximations is presented in Fig. 3.4. Applying the small-angle assumption to Eqs. (3.39)-(3.41) produces the following expressions for steady helical flight:

$$T = W \cdot \gamma + D \tag{3.46}$$

$$\frac{W}{g}\frac{V^2}{R} = L\sin(\mu) + T \cdot \alpha \sin(\mu) \tag{3.47}$$

$$W = L\cos(\mu) - T \cdot \alpha \cos(\mu) \tag{3.48}$$

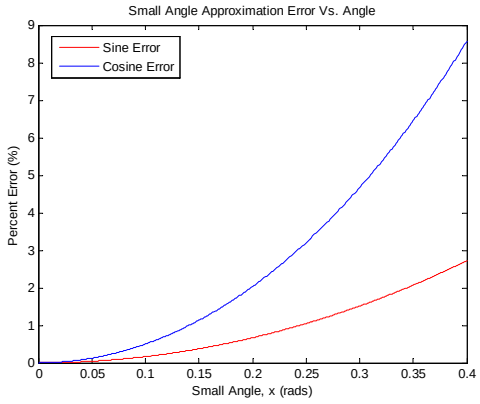



Figure 3.4: Small-Angle Approximation Error

Note that we do not apply the small-angle assumption to the bank angle $\mu$ which may be appreciable for turning flight. It is also common to assume that the thrust is small compared to lift, which is the major benefit of banked turning, and that the terms that involve thrust $T$ multiplying a small angle of attack and trigonometric bank-angle relationship (also less than unity) can be deleted to produce the most

compact form of steady helical flight:

$$T = W \cdot \gamma + D \tag{3.49}$$

$$\frac{W}{g}\frac{V^2}{R} = L\sin(\mu) \tag{3.50}$$

$$W = L\cos(\mu) \tag{3.51}$$

These equations, along with the requirement for zero pitching moment, define an equilibrium flight condition for which the vehicle remains unaccelerated (except for constant centrifugal acceleration) if not subject to disturbances. Let us briefly consider the range of potential steady equilibrium flight conditions before analyzing the trim requirements. If the flight path angle is set to zero we obtain the steady flight equations for level turning flight:

$$T = D \tag{3.52}$$

$$\frac{W}{g}\frac{V^2}{r} = L\sin(\mu) \tag{3.53}$$

$$W = L\cos(\mu) \tag{3.54}$$

If, instead, the centrifugal acceleration and bank angle are set identically to zero we obtain steady flight equations for straight climbing flight:

$$T = W \cdot \gamma + D \tag{3.55}$$

$$W = L \tag{3.56}$$

Finally, if we set the the bank angle, centrifugal acceleration, and flight path angle to zero we recover the most elementary steady flight conditions, straight steady level flight:

$$T = D \tag{3.57}$$

$$W = L \tag{3.58}$$

Recalling Eqs. (3.1) and (3.2) we can determine from Eq. (3.49) that steady helical flight requires:

$$L = \frac{W}{\cos(\mu)} = \frac{1}{2}\rho V^2 S(C_{L_0} + C_{L_\alpha}\alpha) \tag{3.59}$$

$$V^2(C_{L_0} + C_{L_\alpha}\alpha) = \frac{2W}{\rho S\cos(\mu)} \tag{3.60}$$

Equation (3.60) describes a constraint relationship between the matched trim velocity and angle of attack for a given vehicle weight and bank angle. Recalling the thrust Eq. (3.22) and the drag Eqs. (3.3)-(3.5) we can determine the required thrust (and throttle):

$$T = \delta_t \frac{\eta_{\text{prop}}(V, \mathcal{P}_{\text{motor}}, d) \cdot \mathcal{P}_{\text{motor}}}{V} = W \cdot \gamma + \frac{1}{2}\rho V^2 S(C_{D_0} + K(C_{L_0} + C_{L_\alpha}\alpha)^2) \tag{3.61}$$

$$\delta_t = \frac{W \cdot \gamma + \frac{1}{2}\rho V^3 S(C_{D_0} + K(C_{L_0} + C_{L_\alpha}\alpha)^2)}{\eta_{\text{prop}}(V, \mathcal{P}_{\text{motor}}, d) \cdot \mathcal{P}_{\text{motor}}} \tag{3.62}$$

Here trim angle of attack $\alpha$ and velocity $V$ are specified by the relationship in Eq. (3.59) indicating a required thrust $T$ and throttle $\delta_t$ as a function of the flight path angle $\gamma$. Recalling the standing requirement for zero pitching moment we can write an additional constraint expression for steady helical flight:

$$\mathcal{M} = 0 = \frac{1}{2}\rho V^2 Sc(C_{\mathcal{M}_0} + C_{\mathcal{M}_\alpha}\alpha + C_{\mathcal{M}_{\delta_e}}\delta_e) \tag{3.63}$$

which implies:

$$C_{\mathcal{M}} = 0 = C_{\mathcal{M}_0} + C_{\mathcal{M}_\alpha}\alpha + C_{\mathcal{M}_{\delta_e}}\delta_e \tag{3.64}$$

This final equation specifies an elevator requirement for trim subject to the previously

constrained trim angle of attack:

$$\delta_e = \frac{C_{\mathcal{M}_0} + C_{\mathcal{M}_\alpha}\alpha}{C_{\mathcal{M}_{\delta_e}}} \tag{3.65}$$

Let us now introduce a new notation for trim states, hereafter trim conditions will be denoted with a caron or "check" character over the trim-condition variable and a roman subscript denoting the name of the trim condition or mode on all control variables. Using this notation we formally define the set of equations that govern the equilibrium conditions of helical flight as:

$$\text{Trimmed Steady Helical Flight} \leftrightarrow \begin{cases} \check{V}^2_{\text{helix}}(C_{L_0} + C_{L_\alpha}\check{\alpha}_{\text{helix}}) = \frac{2W}{\rho S \cos(\check{\mu}_{\text{helix}})} \\ \check{\delta}_{a,\text{helix}} = \check{\delta}_{r,\text{helix}=}0 \\ \check{\delta}_{t,\text{helix}} = \frac{W\cdot\check{\gamma}_{\text{helix}} + \frac{1}{2}\rho\check{V}^3_{\text{helix}} S(C_{D_0} + K(C_{L_0} + C_{L_\alpha}\check{\alpha}_{\text{helix}})^2)}{\eta_{\text{prop}}(\check{V}_{\text{helix}}, \mathcal{P}_{\text{motor}}, d)\cdot\mathcal{P}_{\text{motor}}} \\ \check{\delta}_{e,\text{helix}} = \frac{C_{\mathcal{M}_0} + C_{\mathcal{M}_\alpha}\check{\alpha}_{\text{helix}}}{C_{\mathcal{M}_{\delta_e}}} \end{cases} \tag{3.66}$$

Deriving the same relationships for level turning flight requires only the deletion of flight path angle terms which:

$$\text{Trimmed Steady Level Turning Flight} \leftrightarrow \begin{cases} V^2(C_{L_0} + C_{L_\alpha}\alpha) = \frac{2W}{\rho S \cos(\mu)} \\ \delta_a = \delta_r = 0 \\ \delta_t = \frac{\frac{1}{2}\rho V^3 S(C_{D_0} + K(C_{L_0} + C_{L_\alpha}\alpha)^2)}{\eta_{\text{prop}}(V, \mathcal{P}_{\text{motor}}, d)\cdot\mathcal{P}_{\text{motor}}} \\ \delta_e = \frac{C_{\mathcal{M}_0} + C_{\mathcal{M}_\alpha}\alpha}{C_{\mathcal{M}_{\delta_e}}} \end{cases} \tag{3.67}$$

Similarly, trimmed straight climbing flight requires the deletion of bank angle terms:

$$\text{Trimmed Steady Straight Climbing Flight} \leftrightarrow \begin{cases} V^2(C_{L_0} + C_{L_\alpha}\alpha) = \frac{2W}{\rho S} \\ \delta_a = \delta_r = 0 \\ \delta_t = \frac{W\cdot\gamma + \frac{1}{2}\rho V^3 S(C_{D_0} + K(C_{L_0} + C_{L_\alpha}\alpha)^2)}{\eta_{\text{prop}}(V, \mathcal{P}_{\text{motor}}, d)\cdot\mathcal{P}_{\text{motor}}} \\ \delta_e = \frac{C_{\mathcal{M}_0} + C_{\mathcal{M}_\alpha}\alpha}{C_{\mathcal{M}_{\delta_e}}} \end{cases} \tag{3.68}$$

Straight steady straight level flight trim conditions are given by:

$$\text{Trimmed Steady Straight Level Flight} \leftrightarrow \begin{cases} V^2(C_{L_0} + C_{L_\alpha}\alpha) = \frac{2W}{\rho S} \\ \delta_a = \delta_r = 0 \\ \delta_t = \frac{\frac{1}{2}\rho V^3 S(C_{D_0} + K(C_{L_0} + C_{L_\alpha}\alpha)^2)}{\eta_{\text{prop}}(V, \mathcal{P}_{\text{motor}}, d)\cdot\mathcal{P}_{\text{motor}}} \\ \delta_e = \frac{C_{\mathcal{M}_0} + C_{\mathcal{M}_\alpha}\alpha}{C_{\mathcal{M}_{\delta_e}}} \end{cases} \tag{3.69}$$

## 3.3 Aircraft Kinematics

To estimate flight performance, subject to wind, a kinematic model for vehicle motion is required. For our purposes flight is subdivided into four discrete segments: takeoff, climb, cruise, and descent. Takeoff is defined as the portion of time between the application of full propulsive power and separation from the water. This portion of the flight is dominated by complex hydrodynamic interactions but testing has borne out that the forward acceleration, at least over relatively calm water, is approximately constant. Climb is defined as the portion of time from first leaving the water to achieving cruise altitude and is characterized by a trimmed climb rate (e.g., best climb, maximum climb, etc.) to the cruise altitude. At the termination of cruise the landing phase begins, and is defined as the portion of flight descending from cruise to the water surface.

Takeoff, climb, and descent are assumed to be directed into the wind. This assump-

tion has guided flight testing and has proven to be both reasonable and practical as non-slipping/non-skidding flight is aerodynamically efficient and up-wind flight is naturally adopted by a seaplane in unconstrained transit over water or, when airborne, during stabilized stick-free flight. Primary maneuvering is conducted during cruise for which vehicle motion is modeled by a bank-to-turn unicycle-model subject to environmental wind, $\vec{w}_I$, of magnitude $\|\vec{w}_I\|$ and heading $\angle\vec{w}_I$, flying at a constant airspeed $V$, and constant altitude $h$. A model for this system is given by:

$$\dot{x} = V\cos(\sigma) + \|\vec{w}_I\|\cos(\angle\vec{w}_I) \tag{3.70}$$

$$\dot{y} = V\sin(\sigma) + \|\vec{w}_I\|\sin(\angle\vec{w}_I) \tag{3.71}$$

$$\dot{\sigma} = \frac{g\tan(\mu)}{V} \tag{3.72}$$

The model relates the Cartesian coordinates of the aircraft in navigation axis (locally-level inertial frame with $x$-axis North, $y$-axis East, and $z$-axis down) to gravitational acceleration $g$, bank angle $\mu$, and heading angle $\sigma$.. Variations of this model that omit wind effects have been used successfully in similar aircraft path planning tasks.[18]

Cruise trajectories are developed by the connection of critical mission waypoints by trimmed flight segments (Fig. 3.5). To optimize the flight plan length a Dubins path is constructed through the set of waypoints. In his 1957 treatise on the subject of minimum length curves L. E. Dubins proved that a combination of (at most) three minimum-radius turns and straight-line segments could describe the minimum length path between any two points subject to any initial and final direction requirements and turn radius constraint.[19] Generally a Dubins path is composed of a fixed-radius arc (typically at the minimum possible radius) from the initial heading to some new

heading, a straight-line segment on the new heading, and a terminal arc (again, typically at the minimum turn radius) from the intermediate heading to the prescribed terminal position and heading. There exist, however, a range of special configurations for which fewer segments and different combinations of segments are required. For example, given the appropriate set of heading constraints the minimum length path between two points that both lie on a single circle, with a radius equal to the minimum turn radius, is the arc of the circle between the two points. Figure 3.5 provides an example Dubins watch-circle crossing. The flight sequence is as follows:

1. At the termination of takeoff and climb the vehicle reaches cruise altitude on an upwind heading.
2. The vehicle flies the initial arc of the Dubins cruise path on arc segment {w2,w3}.
3. The straight-line cruise segment {w3,w4} connects the initial and final Dubins arcs.
4. The terminal Dubins arc segment {w4,w5} brings the aircraft to an upwind heading on approach to landing.
5. The vehicle descends into the wind to the water's surface.

The Dubins path is particularly well suited to aircraft trajectory determination as it produces a minimum length sequence of trimmed turns and trimmed cruise segments through an arbitrary set of waypoint subject to arbitrary heading constraints. While flight vehicles cannot generally follow the exact Dubins path, due to the implicit requirement for instantaneous acceleration changes, Dubins provides a useful, smooth, and piece-wise continuous baseline path that can be approximated during flight with a suitable controller and guidance strategy.

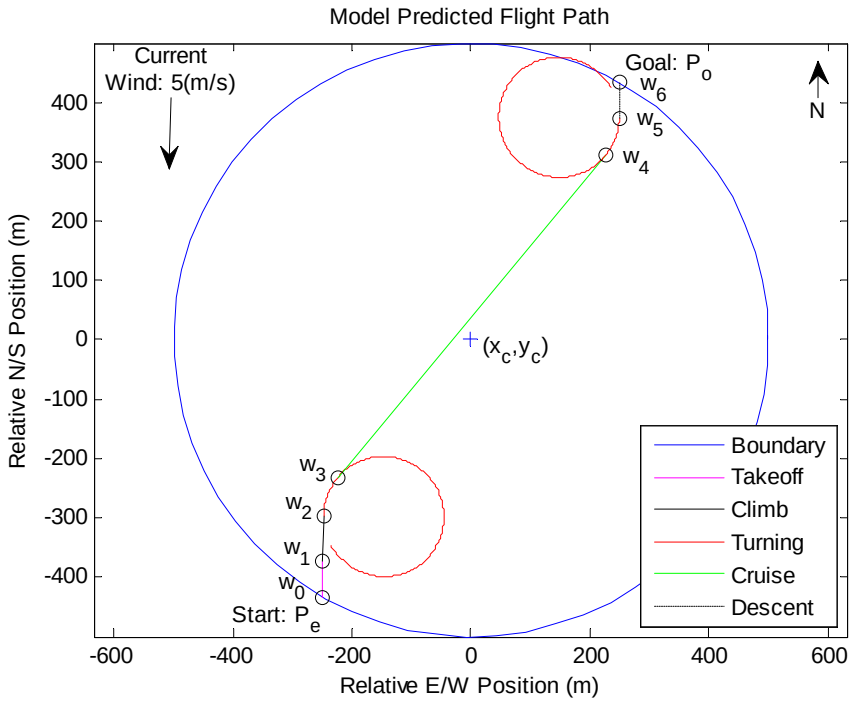



Figure 3.5: Dubins Path Waypoint-Based Flight Plan

## 3.4 Seaplane Dynamics

Up to this point we have presented dynamics and kinematics that can, with the exceptions of the implementation details, be applied to a wide variety of aircraft. The more interesting and challenging dynamics and kinematics problems facing the Flying Fish deal with its nature as a seaplane and the challenges of the its hydrodynamic interactions with the water surface. In this section we will provide some background on the complex hydrodynamics problem, discuss the limitations, simplifications, and assumptions that were adopted during system development, and present simplified models for critical hydrodynamic phenomena.

### 3.4.1 Hydrodynamics

Flying Fish hydrodynamics are fairly complex as we must consider the buoyant and viscous interaction of a dual-asymmetric displacement hull with a free-boundary sur-

face subject to noisy broad-spectrum-periodic three-dimensional deformation. Takeoff is subject to all the complexity of drift with the addition of significant translational motion relative to wind and current. This relative motion induces large-magnitude aerodynamic and hydrodynamic forces and moments.

The takeoff process is actually comprised of two hydrodynamic regimes and a nonlinear transition region. Initially the pontoon motions are governed by displacement-hull dynamics. As the vehicle advances towards liftoff speeds the pontoons transition from displacement to hydroplaning (skimming) over the surface of the water. Solving the equations of motion for both displacement-hull phenomena and hydroplane dynamics generally requires the assumption of fixed translation speed with a traditionally-shaped symmetric single-hull that is longer than all but the long-period ocean "swell" wavelength.[20, 21] With twin asymmetric hulls, continuously changing speed, characteristic length much small than the average surface wave-length, and the addition of aerodynamic forcing the Flying Fish airframe is subject to complex hydrodynamic interactions that are not addressed by common analytical methods. Furthermore, even if the full equations are developed, limitations in the sensing and actuation apparatus will further complicate the implementation of the hydrodynamic model in a control system, as will be discussed in the next chapter.

The richness of the ocean wave spectra is difficult to characterize with a lightweight embedded sensing system. As a result the ocean forcing terms input to a hydrodynamic model will tend to have low signal to noise ratios. For example, the broad frequency content and low energy of chop dynamics make them almost indistinguishable from sensor noise to the average embedded inertial sensor despite their impact on vehicle dynamics on approach to takeoff speeds. Furthermore the limited response rates and control deflections, in combination with reduced aerodynamic control effectiveness below flight speeds, will severely limit control accuracy and effectiveness in response to hydrodynamic feedback.

Ultimately the development of hydrodynamic equations of motion is neither practical nor useful for the deployment of Flying Fish. It is instead more practical to develop approximate models for critical environmental and system processes that can be leveraged by the flight planning and control system to diagnose and robustly respond to nominal and off-nominal conditions.

### 3.4.2 Ocean Waves

During takeoff from the ocean's surface the vehicle will traverse a set of superimposed wave forms with amplitudes and periods that potentially differ by orders of magnitude. The longest fastest waves, defined as "swell," can have amplitudes on the order of several meters or more and periods approaching 10-30 seconds or longer. Swell dynamics are longer-period than the vehicle dynamics, and swell magnitudes may be larger than the vehicle without compromising its ability to operate as long as the slope of the swells is not unmanageably steep. More difficult to model are the moderate and small amplitude "waves" and "chop" which range in amplitude from a few centimeters to a meter with very short periods and fast dynamics. The relatively high slope and frequency of chop induces the vehicle to "drive through" rather than travel with the surface as would be typical for swell traversal.

The most basic wave model is a single sinusoid; the superposition of multiple sinusoid terms can by used to express a reasonably realistic ocean environment. The vertical displacement $n(t)$ of a point in such a sinusoidal wave structure can be written:[20]

$$n(t) = \sum_{i=1}^{M} \frac{H_i}{2} \cos(f_i t - \epsilon_i) \tag{3.73}$$

for wave heights $H_i$, frequencies $f_i$, and phase shifts $\epsilon_i$. Realistic ocean swell structure will tend to have a narrow range of frequency content forming so-called narrow-banded seas. If the summed sinusoid terms of Equation 3.73 are also given direction with

respect to one another we can develop the directional wave spectrum which provides a moderately high-fidelity bulk simulation for ocean wave dynamics. This model can subsequently be used by the FMS to estimate the crude ocean shape after wave frequencies and amplitudes are extracted from wave motion via spectral analysis of sensor data (e.g., fast Fourier-transform of INS-determined vehicle motion). The development of the directional wave spectrum and the extension of wave shapes to dynamic models and wave energy estimates are provided in Reference [20].

### 3.4.3 Ocean Currents

Ocean currents can be difficult to accurately determine on the water, or more precisely the current can be difficult to differentiate from other drift-inducing phenomena on the water as a complex interplay of ocean wave dynamics, vehicle displacement-hull dynamics, and wind combine with ocean currents to influence cumulative drift behavior. Rather than attempting to extract and treat the current value separately it is easier to consider the bulk drift phenomena. The dominant phenomena governing vehicle drift dynamics are wind and water drag on the vehicle; the vehicle is forced by aerodynamic drag in the direction of the local wind and by hydrodynamic drag in the direction of local current. Since the dynamics are of fairly low speed with low Reynold's numbers it is reasonable to approximate both the aerodynamic and hydrodynamic drag forces as Stokes' viscous drag proportional to the inertial wind $\vec{w}_I$ and drift speed ($\vec{v}_{drift}$), less the speed of the local current ($\vec{v}_{h2o}$), as follows:

$$\vec{F}_{wind} = \tau_{wind} \cdot \vec{w}_I \tag{3.74}$$

$$\vec{F}_{h2o} = -\tau_{h20} \left( \vec{v}_{drift} - \vec{v}_{h2o} \right) \tag{3.75}$$

$$\vec{F}_{drift} \equiv \sum \vec{F} = \vec{F}_{wind} + \vec{F}_{h2o} = \tau_{wind} \cdot \vec{w}_I - \tau_{h20} \left( \vec{v}_{drift} - \vec{v}_{h2o} \right) \tag{3.76}$$

Organizing the forces, and creating fictitious inputs windspeed $\vec{u}_{wind}$ and current speed $\vec{u}_{h2o}$, we can write the equations of motion for a drifting point-mass and collect the terms into a second-order state space model as follows:

$$m\ddot{\vec{x}} = -\tau_{h20} \cdot \dot{\vec{x}} + b \cdot \vec{u}_{h2o} + \tau_{wind} \cdot \vec{u}_{wind} \tag{3.77}$$

$$\begin{bmatrix} \dot{\vec{x}} \\ \ddot{\vec{x}} \end{bmatrix} = \begin{bmatrix} \dot{\vec{x}} \\ \ddot{\vec{x}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-\tau_{h20}}{m} \end{bmatrix} \begin{bmatrix} \vec{x} \\ \dot{\vec{x}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{\tau_{h20}}{m} & \frac{\tau_{wind}}{m} \end{bmatrix} \begin{bmatrix} \vec{u}_{h2o} \\ \vec{u}_{wind} \end{bmatrix} \tag{3.78}$$

A major challenge facing the implementation of this model is in the extraction of accurate drag coefficients from either basic physical principles or test data. In the former case we find that the physical form of Flying Fish, particularly its unconventional aero-hydro structures and structural interfaces, make first-principle drag estimation problematic. In the latter case we find that an equilibrium drift condition serves to mask the absolute coefficient magnitude and, given the small expected magnitude of the coefficients, it is numerically difficult to extract reliable estimates. Experimental data has revealed that an equilibrium between wind and water drag is achieved in still water when the vehicle is drifting at ~3.5% of the speed of the wind. Using this observation an equilibrium expression can be written for the drag dynamics in order to develop a relationship between the drag coefficients. Given that drift forces are in balance for when the vehicle drift speed is 3.5% of wind speed:

$$\vec{v}_{drift} = 0.035\,\vec{w}_I,\ \vec{v}_{h2o} = 0 \rightarrow \vec{F}_{drift} = 0 \tag{3.79}$$

The expression for drift force can be written as:

$$\vec{F}_{drift} = \tau_{wind} \cdot \vec{w}_I - \tau_{h20}(0.035\,\vec{w}_I - \vec{v}_{h2o}) = (\tau_{wind} - 0.035\,\tau_{h20})\vec{w}_I + \vec{v}_{h2o} = 0 \tag{3.80}$$

For which, the still-water equilibrium requires:

$$\vec{w}_I \neq 0,\, v_{h2o} = 0 \rightarrow (\tau_{wind} - 0.035\,\tau_{h20}) = 0 \leftrightarrow \tau_{wind} = 0.035\,\tau_{h20} \tag{3.81}$$

In this manner the model is reduced to a single unknown parameter. We subsequently develop a reasonable initial estimate of the unknown parameter from the standard aerodynamic drag Eq. (3.3) using an approximation of the cross section of the vehicle ($\mathcal{A}_{\mathrm{FF}} \approx 0.5\mathrm{m}^2$), the density of air ($\rho = 1.22521\frac{\mathrm{kg}}{\mathrm{m}^3}$), an approximate coefficient of drag for a semi-streamlined body ($C_D \approx 0.1$), and a test-averaged wind velocity. With this information we can write an initial guess for the Stokes' drag coefficient and the state space equations for drift:

$$\tau_{wind} = \vec{F}_{wind}^{T} \cdot \vec{w}_I^{-1} \approx \frac{1}{2}\rho \left\|\vec{w}_I\right\| SC_D = 0.1225 \tag{3.82}$$

$$\begin{bmatrix} \dot{\vec{x}} \\ \ddot{\vec{x}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -0.1094 \end{bmatrix} \begin{bmatrix} \vec{x} \\ \dot{\vec{x}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0.1094 & 0.0038 \end{bmatrix} \begin{bmatrix} \vec{u}_{h2o} \\ \vec{u}_{wind} \end{bmatrix} \tag{3.83}$$

A linear simulation of drift is presented for moderate winds ($4 \pm 2\frac{\mathrm{m}}{\mathrm{s}}$) and ocean currents ($0.25 \pm 0.25\frac{\mathrm{m}}{\mathrm{s}}$) in Fig. 3.6. Given that a deployed vehicle will have a reasonable measurement of drift position, velocity, and direction from the inertial navigation system and GPS sensors it is reasonable to expect that the flight management system can periodically update the estimated drag value to accommodate for environmental factors and maintain accurate drift forecasting.

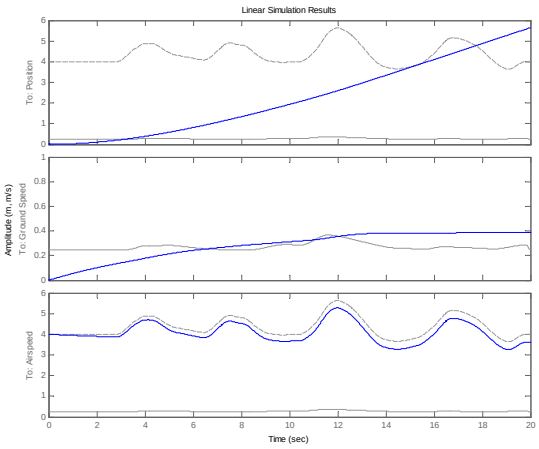



Figure 3.6: Linear Drift Simulation

# CHAPTER IV

# Autonomous Control and Guidance of a Seaplane

Robust guidance, navigation, and control are core requirements for successful repeatable autonomous flight. The Flying Fish flight test program required stable flight operations and safe and efficient transitions to/from the water with guidance to follow proposed mission (e.g., watch circle crossing) profiles. Both Flying Fish airframes have a similar fixed-wing configuration with the Phase II vehicle designed to be somewhat more docile via adjustment of the wing and stabilizer characteristics during re-design. The twin-engine dual vertical tail Flying Fish vehicles have stable flight performance subject to well understood dynamics, and we required only docile maneuvers that we have found can be robustly executed with linear feedback control laws.

During initial flight tests of the Phase I vehicle, the aircraft was validated to be naturally stable with relatively docile response to small-magnitude control surface inputs. The Phase II vehicle, leveraging the experience gained during the flight testing program of its predecessor, demonstrated comparable performance and stability characteristics, albeit at higher flight speeds due to its increased weight. Both airframes have generously-sized control surfaces, and appreciable aerodynamic forces can be applied if required. Notably, the travel range and servo torque limitations in combination with higher mass and inertia result in somewhat slower response rates

for the Phase II vehicle. Flying Fish structures are notably constructed for higher sustained loading than other vehicles of the same scale in order to increase survivability in the harsh ocean environment; the aircraft is therefore heavier than comparable vehicles and with the addition of hydrodynamic structures is subject to additional drag penalties. Specifically, the addition of the twin pontoon flotation system provides hydrodynamic stability and minimizes hydrodynamic drag during takeoff, but adds appreciable drag during flight.

Given that the conventional aircraft dynamics are governed by well-understood principles, that initial testing demonstrated stable/controllable flight, and that the Flying Fish mission profile did not require aerobatic maneuvers, we implemented a traditional gain-scheduled flight controller based on steady trimmed flight conditions. The guidance system commands a sequence of equilibrium flight conditions, for which the linearization of conventional aircraft dynamics subject to small disturbances yields decoupled lateral-longitudinal dynamic responses.[22] Decoupled dynamics allows independent control development and execution for steady level, climbing, descending, and turning behaviors. The literature[23] provides extensive precedent for this treatment of flight control. Under these assumptions a set of proportional-derivative (PD) control laws were developed for the decoupled continuous dynamics and applied to the portion of the flight envelope expected to be traversed using a gain-scheduling scheme. As is also convention, we assumed the digital control system updates would occur at a sufficiently high frequency to allow continuous-time dynamics and control equations to approximate the digital system performance; note that our Phase II Gumstix-based autopilot loop is executed at a rate of ~50Hz, a figure obtained through benchtop and flight tests. The high-frequency INS filter operates independent of the control system so that even for slower control updates the sensor feedback suffers minimal aliasing and provides the control system with undistorted and timely state reports. The decoupled controllers were tuned for each of several flight modes,

with the addition of specific open-loop behaviors and rule-based guidance logic, as discussed in Sec. 4.3. In this section the individual longitudinal and lateral inner-loop control laws are described along with adaptations required to handle the more complex takeoff aero/hydrodynamics.

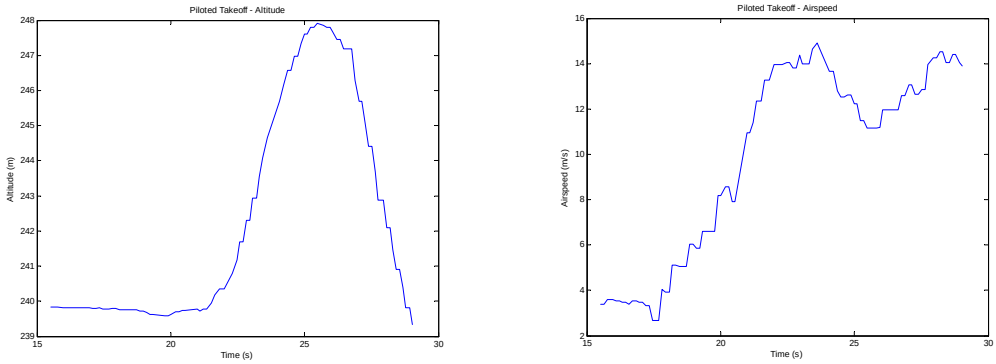
The decoupled control model was extended to the takeoff dynamics through the additional of open-loop behaviors and rule-based guidance logic, as discussed below. For our analysis, it was assumed that the digital control system operated at a sufficiently high frequency to allow continuous time dynamics and control equations to approximate system performance. Notably, the high frequency INS filter operates independent of the control system so that even for slower control updates the sensor feedback suffers minimal aliasing and provides the control system with undistorted and timely state reports.

Below we first present our initial work to define approximate equilibrium or trim states about which our gain-scheduled controllers were designed. Flight tests for this initial effort were manually piloted and were conducted for both the first and Phase II vehicles. The flight-tested control and guidance laws are then presented, followed by results from a series of open-water flight tests that validate the guidance and control law design and implementation.

## 4.1 Trim-State Determination

In the first series of Phase I and II flight tests a pilot manually controlled the aircraft to establish vehicle airworthiness and to qualitatively explore vehicle performance capabilities.[24] These initial flights were conducted without the costly avionics on-board to eliminate any risk to those systems. After the pilot established airworthiness and reliability the Linux-based avionics were installed to acquire sufficient data to characterize flight performance during subsequent manually-piloted flight tests. The

altitude and airspeed profile for an example piloted flight are presented in Fig. 4.1. The pilot's control inputs were recorded to aid with the characterization of vehicle and control performance (Fig. 4.2). Note that control inputs are always presented in microseconds of PWM signal, the units output to the servos/motor-controllers, to provide a common scaling for the different control ranges. Vehicle attitude and the rates of change of the attitude variables over the example piloted flight are presented in Fig. 4.3. Using such flight data, trim states were subsequently extracted for a variety of flight conditions following the procedures described below.




(a) Altitude (b) Airspeed

Figure 4.1: Piloted Flight Profile

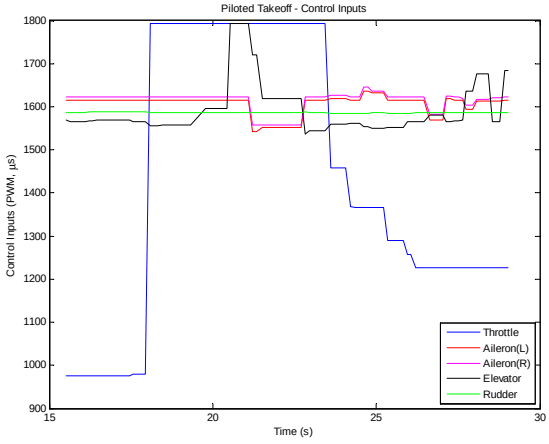
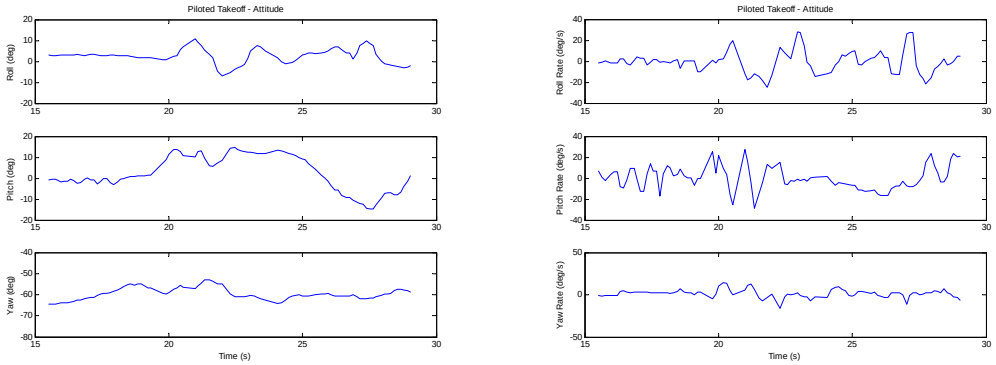



Figure 4.2: Pilot Control Inputs




(a) Attitude (b) Attitude Rates

Figure 4.3: Piloted Takeoff

For trim state determination we targeted a specific trim condition based on an expected fixed-wing steady flight condition (Eqs. 3.66-3.69), designed a flight profile with sufficient persistent flight time in the desired trim state, executed the flight and captured data, and then extracted trim estimates from isolated segments of flight data for which the trim state was held. The order and priority of trim state determination were governed by the watch circle crossing mission. Hence, of the expected set of trim conditions, characterization of steady level flight and steady climb/descent were the primary goals. Although we did not specifically capture steady turning flight data for Flying Fish, we approximated the helical trim conditions from the other trim states. The algorithm for trimmed flight is:

1. *Select* desired steady flight condition (e.g., steady level, steady climb/descent, steady level turn)

2. *Identify* flight profile with significant trim state content (e.g., extended leg racetrack)

3. *Instruct* the pilot concerning the characteristics of the desired trim condition and flight pattern

4. *Execute* flight test plan (with preference given to low/steady-wind conditions to reduce uncertainty; headwind maximizes steady flight time)

5. *Document* flight conditions and maneuver details/times to aid identification of trim flight segments in post-processed data

6. *Segment* flight data into collections of like trim states

7. *Analyze/Average* flight conditions and pilot inputs over trim flight segments to determine central values, variations, and trends

For straight-line trim state determination we are primarily concerned with the trim airspeed ($\check{V}$), angle of attack ($\check{\alpha}$), flight path angle ($\check{\gamma}$), elevator ($\check{\delta}_e$), and throttle ($\check{\delta}_t$). The bank angle ($\check{\mu}$), heading ($\check{\sigma}$), and ailerons/rudder deflections ($\check{\delta}_a, \check{\delta}_r$) are either zero or negligible for non-side-slipping flight. Because calibrated angle of attack measurements are unavailable for the flight vehicles, we estimated the relative longitudinal flight angles based on the pitch angle and flight path angle relationship which is given for straight level non-slipping flight by:

$$\theta = \gamma + \alpha \tag{4.1}$$

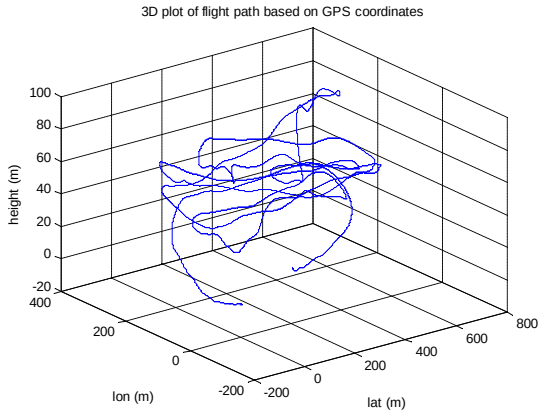
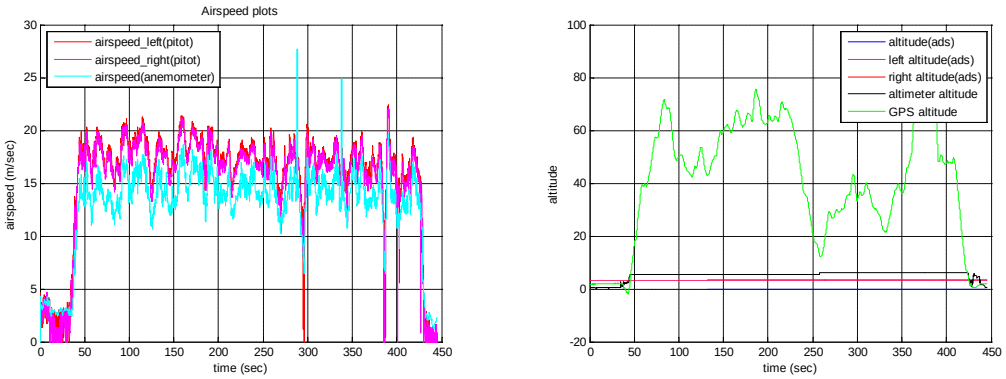



Figure 4.4: Racetrack-Pattern Flight Test




(a) Airspeed

(b) Altitude

Figure 4.5: Racetrack-Pattern Flight Profile

An example racetrack-pattern flight test used to determine steady-level flight trim conditions is presented in Figures 4.4-4.5. From this flight data we see average airspeed in the pertinent flight regimes and by segmenting the flight into climb, level-cruise, and descent segments we can identify the average trim conditions during each phase of flight. From the segmented flight data we find average INS pitch values and can generate average flight path angles by performing a linear fit to the altitude versus distance traveled. We can subsequently estimate the trim angle of attack. A detailed example of descent trim analysis is presented in Figures 4.6-4.8. In each case, after the throttle is reduced and the vehicle settles into a steady descent with little change in servo commands from the pilot, the average pitch angle, flight speed, and flight path angle are extracted. After processing the breadth of several flight tests a full set of average trim states is collected; the pertinent trim states are presented in Table 4.1 for straight-line flight trim conditions.

Table 4.1: Steady/Straight Flight Trim Conditions

| Control \ Mode | Climb | Level | Descent |
|---|---|---|---|
| $\check{\delta}_{e,\text{mode}}$ (deg) | 6.02 | 3.54 | 4.77 |
| $\check{\delta}_{t,\text{mode}}$ (%) | 100.0 | 70.0 | 30.0 |
| $\check{V}_{\text{mode}}(^{\text{m}}/_{\text{s}})$ | 15.2 | 18.2 | 14.7 |
| $\check{\gamma}_{\text{mode}}$ (deg) | ~ 12 | ~ 0 | ~ -3 |
| $\check{\alpha}_{\text{mode}}$ (deg) | ~ 2 | ~ 5 | ~ 6 |
| $\check{\theta}_{mode} = \check{\gamma}_{\text{mode}} + \check{\alpha}_{\text{mode}}$(deg) | 13.4 | 5.3 | 3.2 |

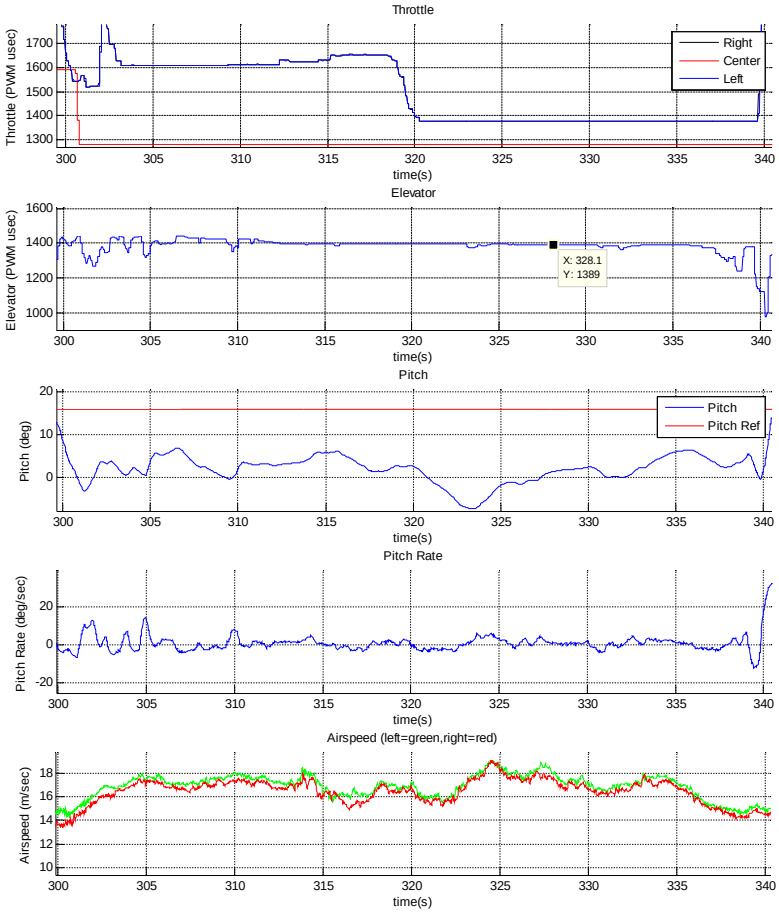



Figure 4.6: Trimmed Descent Example # 1

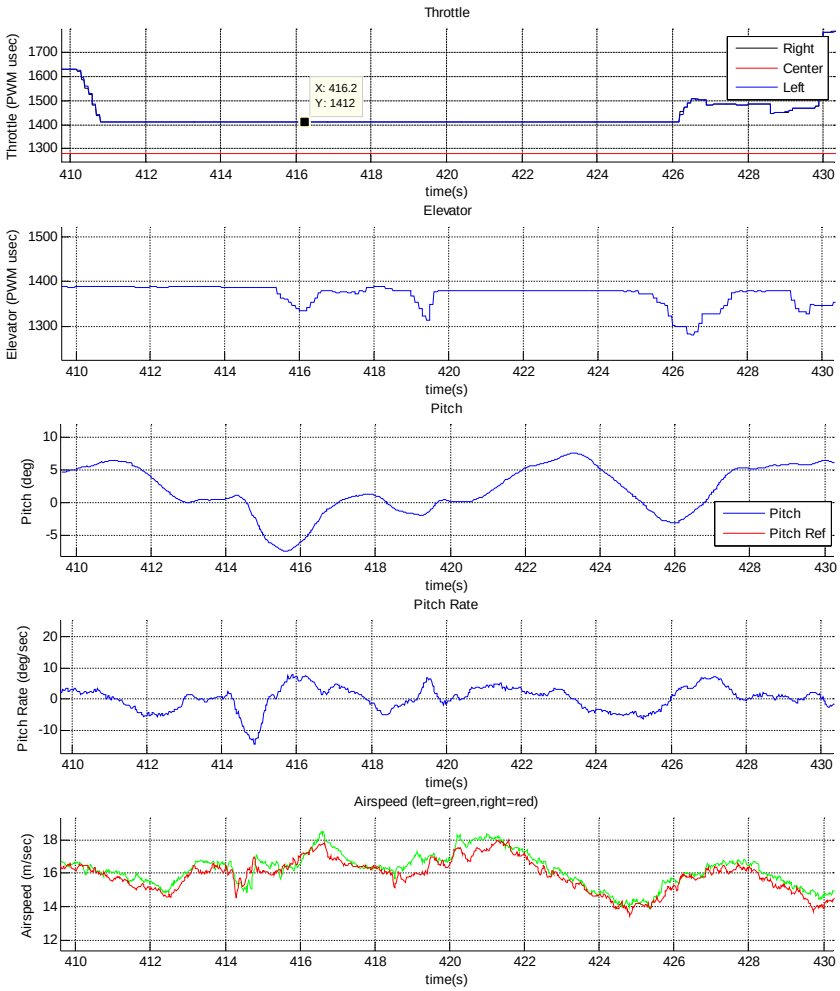



Figure 4.7: Trimmed Descent Example # 2

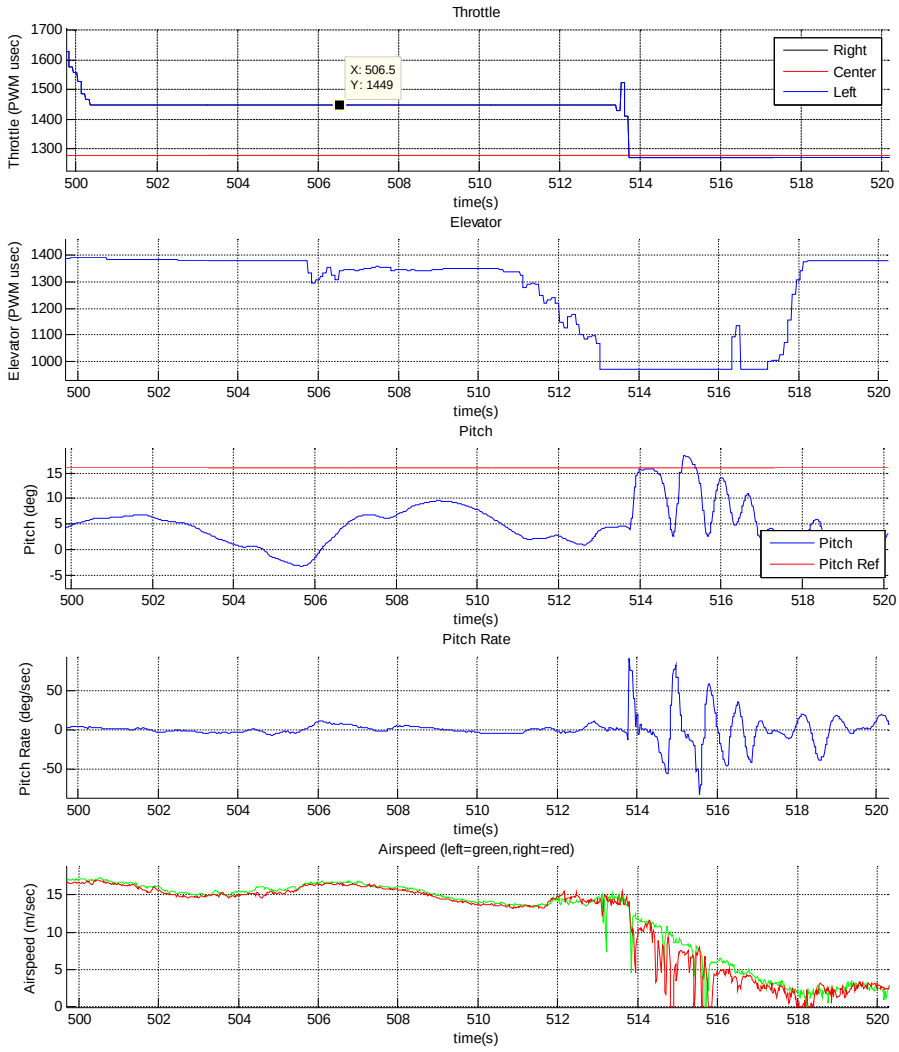



Figure 4.8: Trimmed Descent Example # 3

Once the trim states were determined, testing transitioned to the water's surface for taxi and takeoff development for both vehicle phases. Analysis of manually-piloted takeoffs revealed that pilot inputs for a well-trimmed takeoff were steady between the initiation of the takeoff sequence (throttle up) and the execution of the elevator pulse that breaks the vehicle free from the water (Fig. 4.2). The observation that steady inputs, augmented by impulsive takeoff initiating commands, could achieve takeoff subject to the complex water surface and vehicle interaction dynamics was critical for the guidance and control strategy described below.

## 4.2 Control Strategy

A set of proportional-derivative (PD) control laws were written for the decoupled continuous dynamics and applied to the full flight regime using a gain-scheduling scheme. The controllers were tuned for each of several guidance modes discussed subsequently. In this section the individual longitudinal and lateral control laws will be described as will the open-loop pitch controller for takeoff.

In the first phase of Flying Fish development dual requirements for fast development and robust flight stabilization led to the development of a multi-mode control scheme. This control system utilized straightforward control laws tuned for stability in each trimmed phase of flight with fast transitions between controls. Given suitably-docile trajectories and small differences between trimmed control points non-linear transitions could be ignored. Crucial to the success of the early-stage controller were the inherent stability of the airframes, high control authorities, and docile maneuvering requirements. The phase one controller was divided, much like the phase two controller, into lateral and longitudinal dynamics and included specific open-loop control elements.

### 4.2.1 Longitudinal Control Law

Flying Fish flight plans are defined as sequences of climb, cruise, and descent steady flight segments supplemented by autonomous takeoff and landing sequences. Total control outputs are defined as mode-specific trim commands for each effector summed with PD regulation terms to account for imprecision in the trim command as well as external disturbances. The longitudinal inner-loop PD controller (Fig. 4.9) output is the change in elevator position $\Delta\delta_e$ relative to the elevator trim for that mode based

on desired pitch angle $\theta_d$ and pitch rate $\dot{\theta}_d$ inputs:

$$\Delta\delta_e = K_{P\theta}(\theta_d(V,h) - \theta) + K_{D\theta}(\dot{\theta}_d - \dot{\theta}) \tag{4.2}$$

Desired pitch rate $\dot{\theta}_d$ is always set to zero to damp the rate of control reaction. Note that angle of attack is not used in this control law because it is not assumed observable from the sensors. To simplify the tuning process, the Flying Fish throttle command is strictly set by trim state thus is not subject to feedback control. As such we specify the longitudinal throttle PD control law as:

$$\Delta\delta_t = 0 \tag{4.3}$$

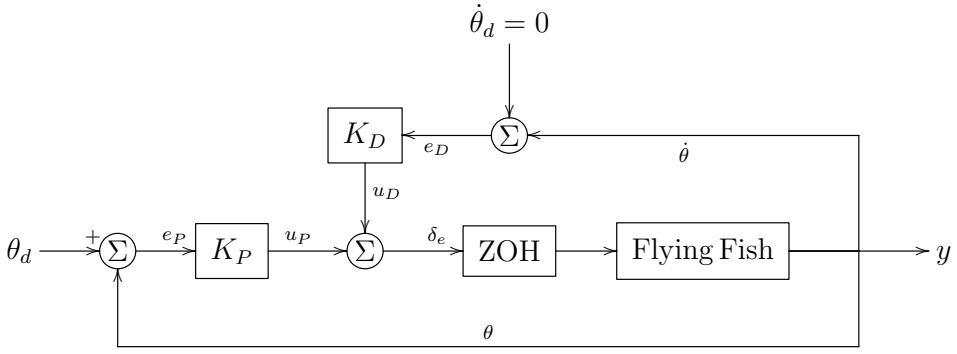



Figure 4.9: Longitudinal Control Law

The throttle and elevator trim states for each flight mode were determined from a series of open-loop manually-piloted flights for each vehicle. The gains $K_{P\theta}$ and $K_{D\theta}$ in Eq. (4.2) are determined via tuning of flight performance and by analysis of flight test data. Initial PD gain estimates were based on a modified Ziegler-Nichols (ZN) method.[25] Generally the ZN method, and related variants, suggest zeroing derivative and integral gains and increasing the proportional gain to a critical gain $K_C$ at which the system oscillates at a constant amplitude. With no desire to push a flight vehicle to sustained oscillation we instead increased the proportional gain until reasonable

initial control performance was observed and then recovered the critical gain from the ZN PID gain relationships:

$$K_P = 0.6K_C \Rightarrow K_C = \frac{K_P}{0.6} \tag{4.4}$$

Flight test data and field observations provided an estimate of the critical oscillatory period $T_C$ from which the previous analysis could be used to determine the derivative and integral control gain estimates:

$$K_I = \frac{2K_P}{T_C} \tag{4.5}$$

$$K_D = \frac{K_P T_C}{8} \tag{4.6}$$

The integral control terms are omitted through the control laws to prevent integrator wind-up from saturated guidance or control responses and any potential destabilization of the flight control system from related issues. Following this procedure, gains can be selected and tuned for control performance about a given trim state. As discussed in much greater detail below, the Flying Fish guidance system decomposes each flight into a number of segments, or modes, for which equilibrium or trim conditions and closed-loop control gains can be separately determined. There are seven sequential operational modes (drift, acceleration (to takeoff), takeoff, climb, cruise, descent, and landing) for which there are seven possible trim states and seven possible longitudinal gain sets. However, only three of these modes (climb, cruise, and descent) are long-term flight modes requiring the definition of equilibrium conditions and gain sets. For simplicity, the "climb" gains are applied through the full flight-initiation process, from the start of acceleration to the termination of climb, except for specific scripted behaviors during takeoff as discussed below. Similarly the "descent" gains are applied to the completion of the landing except for the open-loop flare within

the landing phase. The cruise equilibrium and gain set are applied only through cruise. The trim/default control deflections are specified for every operational mode as, even during drift, the control surfaces must be directed to a reasonable position (e.g. neutralized for drift). For clarity we introduce a simple subscript notation for mode-dependent variables wherein the mode name is appended to the last subscript. The longitudinal mode-dependent gains are:

$$K_{P\theta,\text{mode}} \ , \ K_{D\theta,\text{mode}} \tag{4.7}$$

We must also specify a trim elevator $\check{\delta}_e$ and throttle $\check{\delta}_t$, also mode dependent, such that the absolute elevator and throttle commands are given by:

$$\delta_e = \check{\delta}_{e,\text{mode}} + \Delta\delta_e \tag{4.8}$$

$$\delta_t = \check{\delta}_{t,\text{mode}} + \Delta\delta_t \tag{4.9}$$

The desired pitch angle $\theta_d(u, z)$ is determined in real-time by the guidance system to satisfy the requirements of each operational mode based on either vehicle airspeed $V$ or altitude $h$, or both. It is important to note here that while we categorize the trajectory input to the longitudinal control as a guidance law the system is structurally identical to the traditional inner-loop/outer-loop control scheme commonly applied to small-scale UAS.[23] This applies also to the lateral control laws presented below. The distinction is largely semantic but using this convention, and classifying the reference generation loops as intercept-guidance laws, conveniently aligns almost identically with the actual software implementation onboard the vehicle.

### 4.2.2 Lateral Control Law

The Flying Fish lateral control laws (Fig. 4.10) take a similar form to those of the longitudinal controller, with effectors computed as the sum of trim commands plus feedback control corrections. The lateral PD controller specifies both aileron and rudder output perturbations:

$$\Delta\delta_a = K_{P\phi}(\phi_d - \phi) + K_{D\phi}(\dot{\phi}_d - \dot{\phi}) \tag{4.10}$$

$$\Delta\delta_r = K_{P\psi}(\psi_d - \psi) + K_{D\psi}(\dot{\psi}_d - \dot{\psi}) \tag{4.11}$$

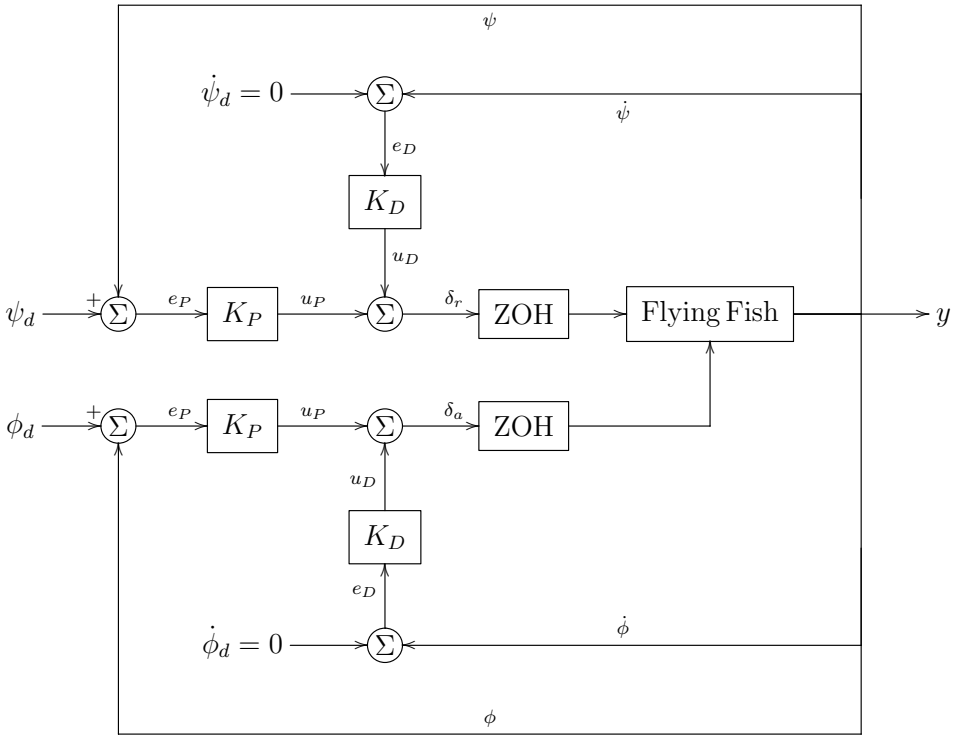



Figure 4.10: Lateral Control Law

The roll control loop actuates the aileron, taking the desired roll angle $\phi_d$ and desired roll rate $\dot{\phi}_d$ and generating an aileron command change $\Delta\delta_a$. The yaw control loop actuates the rudder, taking the desired heading angle $\psi_d$ and desired yaw rate $\dot{\psi}_d$ and generating an output rudder offset $\Delta\delta_r$. Again the guidance system provides the

real-time attitude reference state via an intercept-guidance law and the reference rates are set to zero to damp control response speed. The side-slip angle is not included in this control law because it is not observed from sensor data. A schedule of gains sets was developed for the lateral controller trim states of the three primary flight modes (climb, cruise, and descent) employing the previously outlined methods. The lateral mode-dependent gains are:

$$K_{P\phi,\text{mode}},\ K_{D\phi,\text{mode}},\ K_{P\psi,\text{mode}},\ K_{D\psi,\text{mode}} \tag{4.12}$$

and the trim rudder $\check{\delta}_r$ and aileron $\check{\delta}_a$ for each mode are specified to yield the absolute rudder $\delta_r$ and aileron $\delta_a$ as:

$$\delta_r = \check{\delta}_{r,\text{mode}} + \Delta\delta_r \tag{4.13}$$

$$\delta_a = \check{\delta}_{a,\text{mode}} + \Delta\delta_a \tag{4.14}$$

For Flying Fish, the large twin-rudder surfaces created the potential for non-negligible adverse roll moments. This problem is more pronounced in the Phase I vehicle so the vertical stabilizer's aspect ratio was reduced for Phase II, but in both vehicles comparably large aileron surfaces provide sufficient control authority to compensate the adverse roll thus meet roll performance goals without explicit coupling in the control laws shown above. The decoupled control law structure simplifies guidance commands with minimal trajectory error accumulated during response to damp adverse roll motion. For our control laws we assume that vehicle attitudes and wind-relative motion are of sufficiently small angles that yaw and heading are approximately equal in the inertial reference frame, although the intercept guidance law will drive the vehicle back to the desired flight track if wind-induced offsets are appreciable.

### 4.2.3 Mixed Open/Closed Loop Control for Takeoff

As previously discussed a major challenge for flight control of an ocean-based UAS is the complex hydrodynamic interactions that dominate takeoff. In practice the design of Flying Fish exhibits lateral aero/hydrodynamics that are slow, wind-constrained, and heavily damped on the water thanks to hydrodynamic drag and a low/wide vehicle stance. The wide twin pontoons provide buoyant opposing torque to rolling moments, and hull asymmetry produces a dynamic restorative force in response to downward float pressure from rolling moments at hydroplaning speeds, further stabilizing motion through the water. Lateral control authority, particularly rudder effectiveness, is greatly diminished below flight speeds and the large vertical stabilizers produce significant weathervaning yaw torque that directs the vehicle into the wind. Although we can use asymmetric thrust, weathervaning considerations dictate that the most stable takeoffs are into the wind, a convention we have consistently adopted for Flying Fish. Difficulties arise, however, in the longitudinal dynamics as hydroplaning to takeoff speeds subjects the aircraft to fast impulsive pitch disturbances that are difficult to characterize and compensate.

Longitudinal control on takeoff requires that the vehicle handle hydroplaning dynamics; these fast impulsive pitch disturbances are difficult to accurately manage with feedback control, especially given that our effectors are aircraft control surfaces and throttles. To enable autonomous takeoff despite this challenge, the Flying Fish team first captured and characterized the inputs of a human test pilot over a variety of takeoff conditions. Of particular interest was that our primary test pilot in Phase I, Daniel Macy, was able to takeoff under most conditions with a series of near-steady control inputs to elevator and throttle. Given our decoupled controller design, we adopted a takeoff control law in which roll and yaw control laws were enacted as described above and the pitch control law was replaced by a scripted elevator command

from flight initiation through lift-off (Fig. 4.11). First, takeoff throttle is set to 100% on all motors. The open-loop pitch control initially commands full up-elevator (Fig. 4.11, 20.3s) inducing a nose-up moment to help the vehicle climb atop the water and enter a stable hydroplane. After a fixed three second interval the elevator is relaxed to lower the nose and reduce hydroplane resistance and held while the vehicle continues to accelerate to takeoff speed. After takeoff speed is achieved for at least one full second a brief hard application of full up-elevator is used to break the vehicle from the water, achieving rotation to enter the climb stage (Fig. 4.11, 26.8s). Once the aircraft has rotated, the controller immediately relaxes the elevator to prevent stall (Fig. 4.11, 27.7s) then transitions to climb mode and the longitudinal PD control law is activated. In the example, a particularly high rotation moment entering climb results in a strong down-elevator application by the close-loop longitudinal control to avert stall (Fig. 4.11, 27.8s).

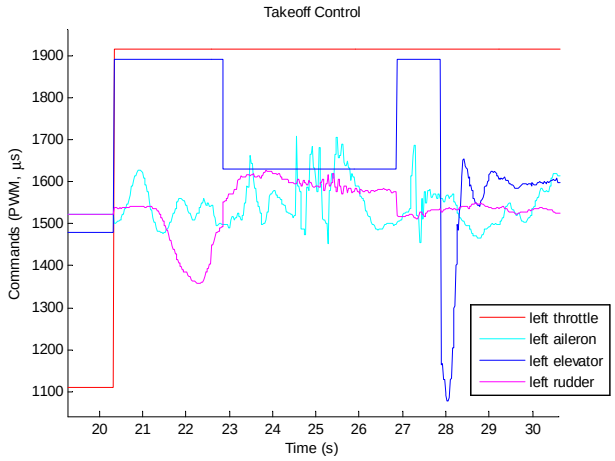



Figure 4.11: Mixed Open/Closed-Loop Takeoff Commands

## 4.3 Guidance

The Flying Fish guidance module decomposed operations into a mode-based "watch circle crossing" flight plan or script. Flying Fish satisfies its persistent surveillance

mission through alternating periods of passive drift and repositioning flight. Despite uncertainty, especially in drift behavior, the mission remains inherently cyclic (Fig. 4.12) and the robust satisfaction of a single cycle is extensible to continuous operation. A single cycle can be decomposed into the following fixed sequence of flight segments: drift, acceleration, takeoff, climb, cruise, descent, and landing. Each mode commands a specific trajectory type, and certain mode transitions require switching guidance to ensure robust system response. From a guidance perspective each segment of the flight-drift operation can be implemented using straightforward techniques whereas the complex behavior resulting from the sequencing of these steps might otherwise be very difficult to automate robustly.

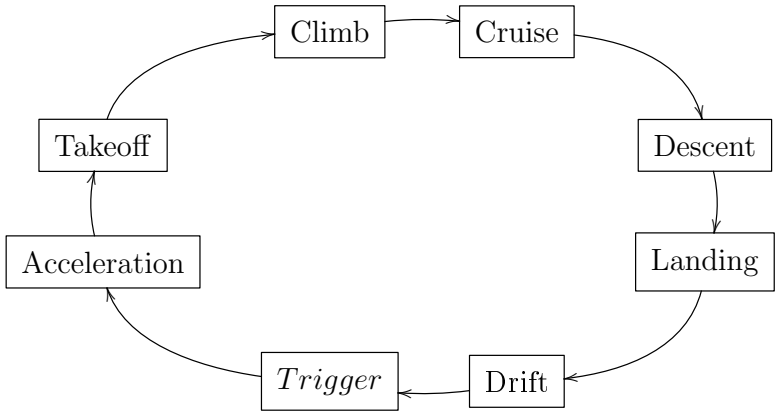



Figure 4.12: Cyclic Mission Profile

The guidance logic design was purposely straightforward to minimize computational overhead and maximize our ability to validate our software through simple testing. As described above, we define a schedule of gains and trim states for each potential mode that guidance will exercise during flight. A guidance and control database $S$ defines control law gains $K$, guidance law gains $G$, vehicle trim states $\check{X}$, and trimmed control inputs $\check{U}$ as follows:

$$\mathcal{S} = \left\langle K, G, \check{X}, \check{U} \right\rangle \tag{4.15}$$

$$K = \{K_{ij,k} | \, i \in \{P, D\}, j \in \{\phi, \theta, \psi\}, k \in \{\text{climb, cruise, descent}\}\} \tag{4.16}$$

$$G = \{G_{i,j} | \, i \in \{P, D\}, j \in \{\text{climb, cruise, descent}\}\} \tag{4.17}$$

$$\check{X} = \left\{\check{X}_i = \left[\check{V}_i, \check{h}_i, \check{\phi}_i, \check{\theta}_i, \check{\psi}_i\right] | \, i \in \{\text{climb, cruise, descent}\}\right\} \tag{4.18}$$

$$\check{U} = \left\{\check{\delta}_{i,j} | \, i \in \{a, e, r, t\}, j \in \{\text{drift, accel., takeoff, climb, cruise, descent, land}\}\right\} \tag{4.19}$$

Each mode is also subject to a set of state constraints that are used within the guidance logic to: 1) trigger mode-transitions, 2) diagnose off-nominal/out-of-range behavior, and 3) set mode-appropriate constraints on reference variables. Finally, each guidance mode has a minimum operation time that prevents disturbances from triggering premature transitions.

At the beginning of a vehicle deployment a watch region, or operational boundary, is defined for the current mission. The project sponsor defined the watch region as a circle specified by center GPS coordinates and a radius. The watch circle can be set manually; we also implemented an automated watch-setting routine that averages sensor data over a period of time (e.g., 30 seconds) to define a watch region center and baseline wind estimate. Assuming the pitot tube measures wind speed during drift, and that the vehicle weathervanes into the wind, initial environmental data collected includes minimum, maximum, and average wind speed and heading as well as initial wave statistics from the INS. Once the watch circle is set the vehicle is considered "on station" and begins its sequential execution of guidance modes as specified below.

### 4.3.1 Sequential Guidance Modes

1. **Drift** ("on station")

   This is the only mode in which no control action is exerted. The system commands the control surfaces to predefined neutral trims for the duration of the drift. The system can also shut down servos in this mode to conserve power. In drift mode the guidance system monitors range and direction to the center

of the watch circle. When the range exceeds the watch circle radius the guidance system initiates flight by transitioning to acceleration mode. Effectively we have:

$$K_{P\theta,\text{drift}} = K_{D\theta,\text{drift}} = K_{P\phi,\text{drift}} = K_{D\phi,\text{drift}} = K_{P\psi,\text{drift}} = K_{D\psi,\text{drift}} = 0 \quad (4.20)$$

$$\check{\delta}_{e,\text{drift}} = \check{\delta}_{r,\text{drift}} = \check{\delta}_{a,\text{drift}} = \check{\delta}_{t,\text{drift}} = 0 \quad (4.21)$$

2. **Acceleration**

The acceleration to takeoff is dominated by high-speed highly-variable hydrodynamic effects. Lateral closed-loop control is commanded to a wings-level roll attitude and wind-heading ($\angle\vec{w}_I$) yaw reference with gains $\{K_{P\phi,\text{accel}}, K_{D\phi,\text{accel}}, K_{P\psi,\text{accel}}, K_{D\psi,\text{accel}}\}$, and trim control surface inputs $\{\check{\delta}_{r,\text{accel}}, \check{\delta}_{a,\text{accel}}\}$:

$$\phi_d = 0 \quad (4.22)$$

$$\psi_d = \angle\vec{w}_I \quad (4.23)$$

The longitudinal pitch dynamics remain open-loop as the vehicle drives into a stable hydroplane (accelerates onto the pontoon step) to achieve takeoff speed. The open loop longitudinal elevator trim for acceleration $\check{\delta}_{e,\text{accel}}$ is constant moderate-up elevator to counter nose-down moment from hydrodynamic drag as the vehicle accelerates under full throttle:

$$\check{\delta}_{t,\text{accel}} = \max(\delta_t) \quad (4.24)$$

The guidance system transitions to takeoff mode when both a minimum operating time has elapsed and a minimum airspeed is achieved. The minimum time is particularly critical to reduce the likelihood of a wind gust or high steady

wind speed triggering premature transition to takeoff mode.

3. **Takeoff** (transition)

   At liftoff speed the autopilot initiates a scripted two-part open-loop elevator command. An initial full-up elevator pulse is required to pitch up, breaking the aircraft free from the water. The elevator must then be quickly relaxed as the required break-away elevator will rapidly rotate the aircraft to stall. With the elevator near neutral the aircraft gains speed at a slow climb entering the climb mode.

$$\check{\delta}_{e,\text{takeoff}} = \max(\delta_e) \tag{4.25}$$

$$\check{\delta}_{t,\text{takeoff}} = \max(\delta_t) \tag{4.26}$$

4. **Climb**

   If the aircraft rotates into a climb without stalling or violating the pitch envelope, the climb guidance mode becomes active. Otherwise (an infrequent but observed occurrence in adverse chop conditions), the aircraft settles back on the water and the takeoff is automatically aborted until the flight is indicated again (at the boundary of the watch circle). Full lateral longitudinal closed-loop control is in effect for climb with climb-specific gains $\{K_{P\phi,\text{climb}}, K_{D\phi,\text{climb}}, K_{P\psi,\text{climb}}, K_{D\psi,\text{climb}}, K_{P\theta,\text{climb}}, K_{D\theta,\text{climb}}\}$ with trim control surface inputs $\{\check{\delta}_{r,\text{climb}}, \check{\delta}_{a,\text{climb}}, \check{\delta}_{e,\text{climb}}\}$. The guidance law adopted for climb generates a desired pitch angle $\theta_d$ for the longitudinal controller using a trim reference pitch for climb $\check{\theta}_{\text{climb}}$ and climb guidance gain $G_{P,\text{climb}}$ to hold a prescribed trim climb airspeed ($\check{V}_{\text{climb}}$) by:

$$\theta_d = \check{\theta}_{\text{climb}} - G_{P,\text{climb}}(\check{V}_{\text{climb}} - V) \tag{4.27}$$

Lateral guidance commands wings-level wind-heading flight at full throttle:

$$\phi_d = 0 \tag{4.28}$$

$$\psi_d = \angle \vec{w}_I \tag{4.29}$$

$$\check{\delta}_{t,\text{climb}} = \max(\delta_t) \tag{4.30}$$

Climb continues for a minimum preset time and the vehicle must pass a minimum altitude before transition to cruise.

5. **Cruise**

   At the termination of climb the guidance logic switches into the cruise mode. Lateral and control remain active but switch to cruise-specific gains and trim inputs as in previous modes. The cruise altitude-intercept guidance law reduces the throttle and gives a desired pitch angle guidance-law, based on the cruise trim pitch reference $\check{\theta}_{\text{cruise}}$ and cruise reference altitude ($\check{h}_{\text{cruise}}$), as:

$$\theta_d = \check{\theta}_{\text{cruise}} - G_{P,\text{cruise}}(\check{h}_{\text{cruise}} - h) + G_{D,\text{cruise}}(\dot{h}) \tag{4.31}$$

$$\check{\delta}_{t,\text{cruise}} \approx 0.7 \max(\delta_t) \tag{4.32}$$

   The goal of watch circle crossing is to maximize the time spent in the watch circle and minimize energy use. This is accomplished by flying upwind across the watch circle, which in turn leads to maximizing the drift period before the vehicle will again exit the watch circle on the downwind side (note that wind drift has been shown to dominate current-based drift for both our seaplanes, a condition expected except in rare high-current, low-wind situations). Maximizing drift time is particularly important to recover expended energy via solar recharge. In Phase I a baseline cruise guidance strategy attempted to fly the

vehicle over the watch center before transition to an upwind descent. In the event that combined environmental and turning constraints prevented a center crossing the vehicle entered the descent mode at the point that half of the circle had been crossed from the point of liftoff (to prevent back-tracking to cross the center). As we show below, this simple strategy did not necessarily take the shortest path to the upwind landing site. In Phase II the guidance strategy was updated to target the opposite side of the circle from the point of watch-region departure and fly directly to that point during the entire cruise phase, as described below. The transition to descent is initiated at the point that the vehicle reaches trimmed glide-slope range of the landing target. To cross the watch area (Phase II strategy), the cruise guidance law attempts to intercept a target point at which the transition to descent would result in an appropriate landing site. For this we create a sub-mode "turn" of the cruise mode and write a bank-to-turn target-intercept guidance law that generates a desired bank angle $\phi_d$ from the required turn heading to the target $\psi_{\text{turn}}$ as:

$$\phi_d = \text{sign}(\psi_{\text{turn}} - \psi) \cdot \min(20\,\text{deg}, \|0.5(\psi_{\text{turn}} - \psi)\|) \tag{4.33}$$

This guidance law gives 1 degree of bank for every 2 degrees of heading error up to the saturation bound of 20 degrees bank. A 2.0 degree deadband is implemented in the system around the desired heading to minimize oscillation. We specify the yaw reference instantaneously equal to the current heading:

$$\check{\psi}_d = \psi \tag{4.34}$$

Finally, a discrete rate-limiting roll-in/roll-out update provides smooth entry

and exit from the banked turn:

$$\phi_d\Big|_0 = 0 \tag{4.35}$$

$$\phi_d = \phi_d\Big|_i = \phi_d\Big|_{i-1} - (0.5)\text{sign}(\phi_d\Big|_{i-1} - \phi_d), \quad i = 1,\, 2,\, 3,\, ... \tag{4.36}$$

We show below that this was a more effective watch crossing mechanism than strict center-crossing. However it is important to note that this guidance method can degrade for highly-variable environmental conditions as the implicit assumption that the point of watch-circle departure is near the most-downwind watch-circle radius may not hold, suggesting future adaptation based on observed drift profiles from previous cycles.

6. **Descent**

Once the vehicle either crosses into the upwind side of the watch circle (Phase I) or the aircraft reaches the appropriate distance from the landing target (Phase II) the guidance law transitions to a trimmed descent to the far upwind edge of the watch region. The longitudinal and lateral control laws remain active after changing to the gains and trim control settings for descent. The descent mode utilizes the same outer-loop guidance law as the climb guidance, above (Eq. (4.27)), with a small negative pitch trim $\check{\theta}_{\text{descent}}$ and an airspeed 20% above stall:

$$\theta_d = \check{\theta}_{\text{descent}} - G_{P,\text{descent}}(\check{V}_{\text{descent}} - V) \tag{4.37}$$

$$\check{V}_{\text{descent}} = 1.2\min(V) \tag{4.38}$$

$$\phi_d = 0 \tag{4.39}$$

$$\psi_d = \angle\vec{w}_I \tag{4.40}$$

The vehicle performs powered descent into the wind with ~30% throttle to

provide a safe gentle approach to landing:

$$\check{\delta}_{t,\text{descent}} = 0.3 \max(\delta_t) \tag{4.41}$$

7. **Landing** (transition)

   When the vehicle is within 7-10 meters of the surface, the ultrasonic altimeter provides surface ranging that allows the autonomous aircraft to more reliably predict approach and contact with the water. We maintain the gentle descent until 0.5-1.0m above the water surface, depending on swell height, at which time the aircraft is flared and throttle reduced to zero. Upon landing, the guidance law deactivates the flight control system, sets the system in drift mode, and begins monitoring range and heading to the watch-center for another drift-fly cycle.

$$\check{\delta}_{e,\text{landing}} = \max(\delta_e) \tag{4.42}$$

$$\check{\delta}_{t,\text{landing}} = 0 \tag{4.43}$$

### 4.3.2 Guidance Mode Transitions

Step transitions between modes with different intercept-guidance-laws can excite oscillatory vehicle modes that can increase tracking error, control action, and the risk of flight destabilization. In the preceding sequential mode analysis we described lateral trajectories that matched, or nearly matched, at the mode transitions but the longitudinal trajectory structure provided no guarantee of alignment. The climb/cruise and cruise/descent transitions will routinely encounter jump discontinuities in the pitch reference fed to the longitudinal controller because the governing guidance equations (Eqs. (4.27), (4.31), (4.37)) do not utilize the same input variables. Specifically, the pitch guidance reference changes from airspeed to altitude during the climb/cruise transition and back from altitude to airspeed during the cruise/descent transition.

Since the guidance input trajectory is neither of consistent type nor consistent units the autopilot input trajectory cannot be directly interpolated. Instead the output of the guidance laws (the variables between the inner and outer loops) are initially matched to mitigate the impact of the inevitable transition. This is accomplished for Flying Fish by pre-computing the initial guidance output of the goal mode prior to transitioning and then linearly interpolating from the current state to the expected initial state of the next mode. The critical smoothing algorithm variables: transition time $T_s$ and transition step size $\Delta\theta_s$, distinguished by the subscript $s$, are used in conjunction with the fixed digital execution frequency $f$ to specify the transition algorithm following a mode change, as follows:

1. Select the correct last mode and compute the final pitch guidance output:

   (a) Last Mode: Climb (Eq. (4.27)):

   $$\theta_1 = \theta_d(t) = \check{\theta}_{\text{climb}} - G_{P,\text{climb}}(\check{V}_{\text{climb}} - V(t))$$

   (b) Last Mode: Cruise (Eq. (4.31)):

   $$\theta_1 = \theta_d(t) = \check{\theta}_{\text{cruise}} - G_{P,\text{cruise}}(\check{h}_{\text{cruise}} - h(t)) + G_{D,\text{cruise}}(\dot{h(t)})$$

2. Select the correct next more and compute the initial pitch guidance output:

   (a) Next Mode: Cruise (Eq. (4.31)):

   $$\theta_2 = \theta_d(t) = \check{\theta}_{\text{cruise}} - G_{P,\text{cruise}}(\check{h}_{\text{cruise}} - h(t)) + G_{D,\text{cruise}}(\dot{h(t)})$$

   (b) Next Mode: Descent (Eq. (4.37)):

   $$\theta_2 = \theta_d(t) = \check{\theta}_{\text{descent}} - G_{P,\text{descent}}(\check{V}_{\text{decent}} - V(t))$$

3. Compute interpolation step $\Delta\theta_s$ from execution frequency $f$ and interpolation time $T_s$:

   $$\Delta\theta_s = {}^{(\theta_2-\theta_1)}/_{(T_s \cdot f)} \tag{4.44}$$

4. Replace guidance law for $t = T_s$ seconds with a discrete update of the desired pitch variable $\theta_d$:

$$\theta_d\big|_0 = \theta_1 \tag{4.45}$$

$$\theta_d = \theta_d\big|_i = \theta_d\big|_{i-1} + \Delta\theta_s,\ i = 1, 2, 3, ..., \lfloor T_s \cdot f \rfloor \tag{4.46}$$

The slope of the interpolation is variable based on the magnitude of misalignment between the pitch trajectories and is secondarily governed by the selection of the interpolation time $T_s$. In general a fixed period of time of around 3s proved to be appropriate for the majority of flight test conditions. While the other gains and trims are not smoothed by the scheduling algorithm, no fast switching commands were required for the lateral reference, and the natural stability of the aircraft in its trim-states along with moderate gain minimized negative oscillatory behaviors. Because of its large magnitude change, the effects of throttle change on the Phase II platform were more pronounced than in the Phase I vehicle, thus requiring interpolation between the trim throttles for different operational modes since throttle was not regulated explicitly through PD control. No other trajectories, states, internal variables, or settings required smoothing between modes.

## 4.4 Results

From late 2007 through 2010, the Flying Fish team successfully field-tested and demonstrated autonomous flight with the Phase I and Phase II vehicles in freshwater lakes and in the open ocean. The Phase I vehicle performed initial airworthiness tests in summer 2007 before flying in a successful multi-day sea-trial in the Monterey Bay region of the Pacific Ocean. The Phase I vehicle returned to Monterey the following year for additional tests. The Phase II vehicle has flown only in Northern Michigan at the University of Michigan's Bio-Station on the shore of Douglas Lake with an FAA

Certificate of Authorization (COA). Guidance and control results from these flight tests are summarized in this section.

Figure 4.13: Flying Fish w/Dolphins

### 4.4.1 Gain Tuning Strategy

Once sufficient trim data had been collected, gains were estimated following the procedures outlined in Sec. 4.2.1, and the aircraft was flown to a safe recovery altitude by a human pilot to begin a series of controller tuning tests. Once at altitude a controller was activated and the pilot surrendered command to the computer system for the test team to observe. In this manner a series of high altitude tests progressed from controlled level flight through climbing, descending, and turning flight capabilities. Eventually brief "watch circle hop" flight sequences (climb/cruise/descent) were accomplished, at altitude, by the mode switching guidance system. Once tuned controllers for each flight mode were developed the vehicle was moved to the water for automated takeoff testing. The trim states were subsequently augmented with control trims for each of the takeoff modes and for the landing flare (Table 4.2). For the Phase II vehicle this process was repeated but in a more abbreviated time scale to support flight testing deadlines.

Table 4.2: Control Trim States by Mode

| Control \ Mode | Drift | Accel. | Takeoff | Climb | Cruise | Descent | Land |
|---|---|---|---|---|---|---|---|
| $\check{\delta}_e$ (deg) | 0.0 | 3.0 | 20.0 | 6.02 | 3.54 | 4.77 | 20.0 |
| $\check{\delta}_t$ (%) | 0.0 | 100.0 | 100.0 | 100.0 | 70.0 | 30.0 | 0.0 |

Stability and robustness were prioritized so controllers were tuned conservatively, weighting gentle responses over tracking accuracy. The decision to tune for moderate response rates and stability over tracking accuracy is well supported for the given test environment; for an ocean landing the potential runway area is immense and a small heading or lateral position error can often be neglected whereas a strong roll correction, or any other strong control response near the water could be catastrophic. Similarly, given the inherent uncertainty/variability of the ocean height if the vehicle operates from sufficient altitude and has a sufficiently slow approach to the water even longitudinal tracking error is tolerable and preferable to a pitch overshoot near the water. Control and guidance mode gains for each mode of the Phase II vehicle are presented in Table 4.3.

Table 4.3: Phase II Guidance and Control System Gains

(a) Control Gains

| Gain | Climb | Cruise | Descent |
|---|---|---|---|
| $K_{P\theta,mode}$ | 0.25 | 0.25 | 0.25 |
| $K_{D\theta,mode}$ | 0.12 | 0.15 | 0.12 |
| $K_{P\phi,mode}$ | 0.3 | 0.33 | 0.33 |
| $K_{D\phi,mode}$ | 0.04 | 0.08 | 0.08 |
| $K_{P\psi,mode}$ | 0.15 | 0.33 | 0.2 |
| $K_{D\psi,mode}$ | 0.05 | 0.1 | 0.1 |

(b) Guidance Gains

| Gain | Climb | Cruise | Descent |
|---|---|---|---|
| $G_{P,mode}$ | 0.1 | 0.25 | 0.3 |
| $G_{D,mode}$ | 0.0 | 0.15 | 0.0 |

### 4.4.2 Longitudinal Control Performance

After the initial controller tuning the Phase II vehicle was deployed for flight tests over Douglas Lake in Northern Michigan. Preliminary observations suggested that the control system was effective and reasonably well-tuned. The vehicle first achieved autonomous flight on the second test day and was able to "hop" across the watch circle shortly thereafter. Figures 4.14-4.16 present the longitudinal system response, with rate and control signal overlays, for an autonomous flight. Upon closer inspection it becomes clear that while the system has been largely effective at achieving its basic mission the individual inner and outer loop controllers are not well tuned. The pitch response (Fig. 4.14) of the Phase II vehicle, for example, while stable and capable of a watch-crossing flight has both poor pitch tracking and poor pitch reference generation. Note that the INS pitch reference is being used for control purposes as a surrogate for AOA and flight path angle in the absence of accurate sensors for those states. In the example it is clear that the outer-loop pitch reference is too high and that the inner-loop pitch control achieves only moderate command shape following with a large steady offset error. An adjusted pitch reference line (dotted magenta) that was shifted down by 5 degrees has been added to Fig. 4.14 to illustrate that the command shape following is actually good outside of the steady error. We see that the outer-loop guidance law is increasing the pitch reference to the point that the error signal of the inner loop controller generates sufficient elevator response to achieve the flight goals (airspeed/altitude tracking). The altitude response (Fig. 4.15) is reasonable considering the small period of time spent in cruise in the example flight and given that no specific altitude was targeted (the system simply adopted the cruise altitude achieved after a fix duration climb). The airspeed tracking is also reasonable but it is clear the airspeed is impacted by poor pitch control on descent as the strong nose down tendency dramatically increases airspeed entering the descent phase (Fig. 4.16). Note that, for clarity, the data is segmented into the separate stages of flight: Takeoff

Initiation (TI), On Step/hydroplane (OS), Liftoff (LO), Climb (CL), Cruise (CR), Descent (DS), and Landing (LD). In this data we see that the pitch controller displays an induced oscillation and undershoot during descent. The oscillation is attributed to the large dynamic change, especially from thrust reduction, between the cruise and descent phase (despite reference and control smoothing) and imperfect control tuning. Undershoot is attributed to insufficient trim elevator in conjunction with the lack of an integrator component to increase command action in response to steady error. While longitudinal control performance is less than ideal, it is nevertheless sufficient for autonomous aerial hops. Specifically, even though the errors are clearly discerned from the data the performance of the closed-loop system is sufficient for safe operations and the continuation of the flight testing program.

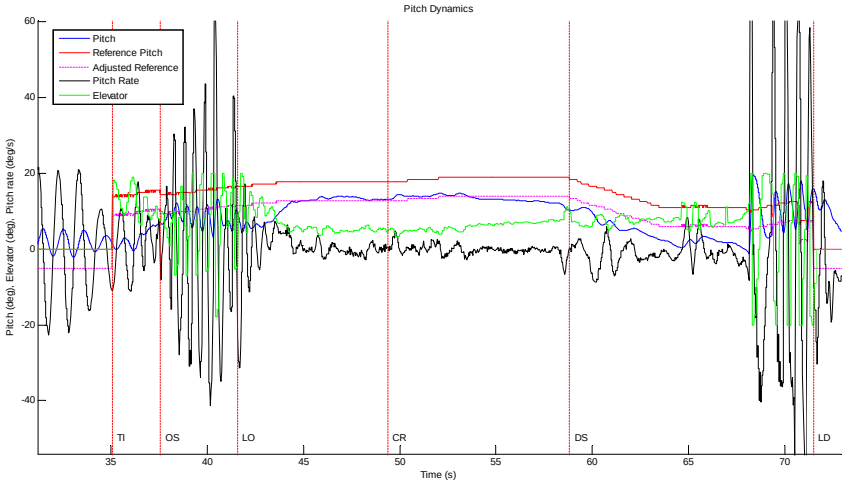



Figure 4.14: Longitudinal Control - Phase II, Pitch

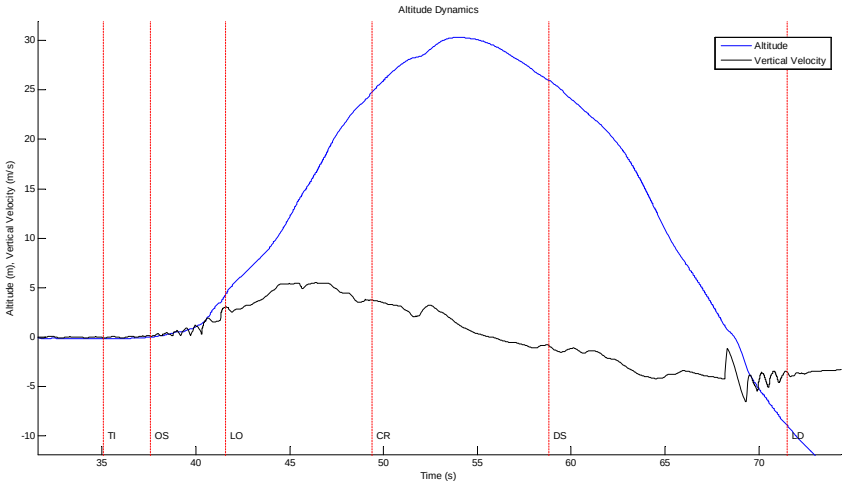



Figure 4.15: Longitudinal Control - Phase II, Altitude

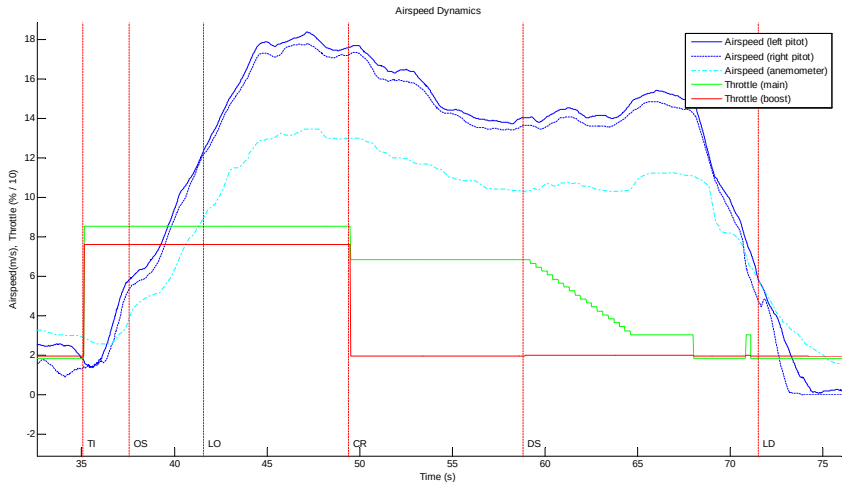



Figure 4.16: Longitudinal Control - Phase II, Airspeed

### 4.4.3 Lateral Control Performance

The lateral controller performance of the Phase II vehicle, much like Phase II longitudinal controller performance, proved effective at meeting mission goals but had limited controller accuracy. The profiles and rates of the yaw and roll states are presented, with control signal overlays, in Figs. 4.17-4.18. The INS yaw and pitch

measurements are reasonable representations of the heading and bank angle of a non-slipping aircraft. The yaw/heading response shows accurately but slowly tracks as only the cruise mode applies significant heading control; every flight mode, outside of cruise, adopts the current heading at the inception of that mode and exerts only moderate rudder action to try to maintain that heading. We desire non-slipping upwind flight and hence want a minimum of rudder deflection except as necessary to maintain tracking through a gust. Note that, again, the plots are segmented by flight state for clarity. Looking at the heading track we see that vehicle correctly tracks into the prevailing wind during the approach to cruise and, at the start of the cruise, the heading is changed to target the landing point. The yaw/heading state tracks slowly onto the new heading subject to the roll-in delay of the banked-turn controller (Fig. 4.18) and after a small overshoot follows the command as it follows the changing heading to the target. The roll/bank controller is a little less accurate, though no less effective in field testing, as the controller attempts to follow the desired bank trajectory but lags the reference command significantly and overshoots with some oscillation during the banked turn. We also see a low/moderate oscillation ($\pm$2deg) over the entire flight. This oscillation could be a natural response to wind disturbance but the periodicity may indicate suboptimal controller tuning. Lateral control performance is sufficient for safe autonomous hops and for the continuation of other flight test operations.

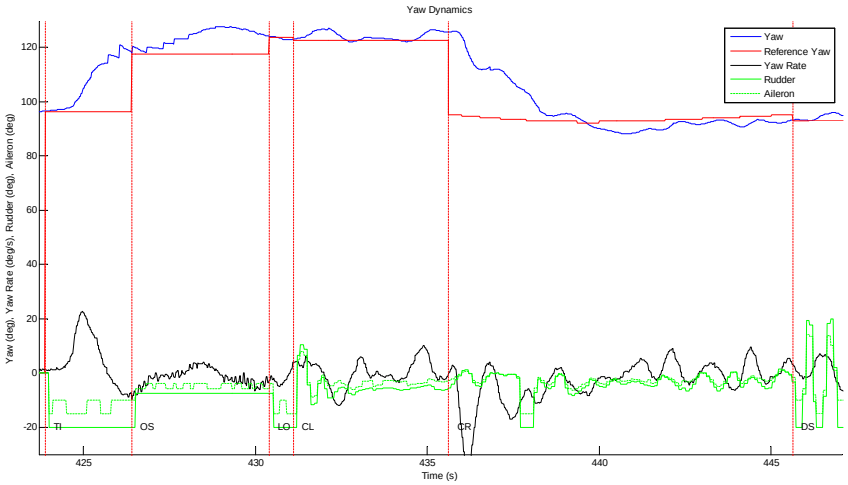



Figure 4.17: Lateral Control - Phase, Yaw/Heading

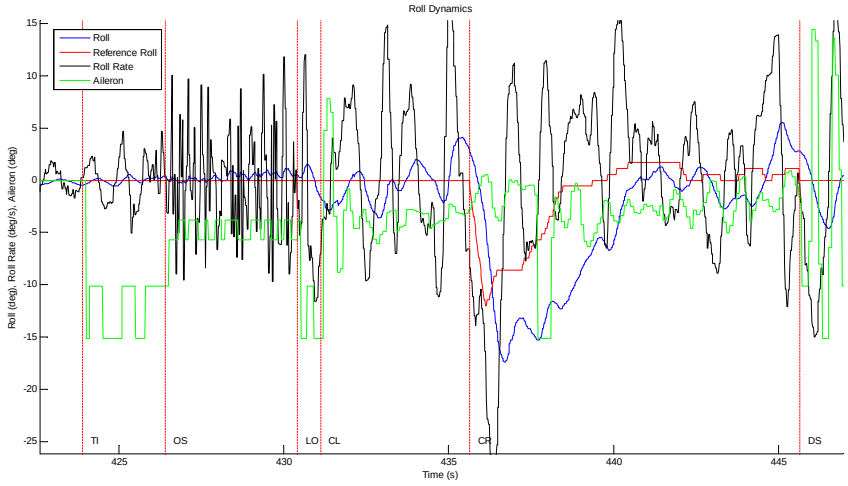



Figure 4.18: Lateral Control - Phase, Roll/Bank

### 4.4.4 Guidance Sequence Testing

Once control laws were tuned for all flight modes the system began fully-autonomous flight testing. Here we present Phase I flight profiles from the earliest guidance profiles. Much like the piloted flight profile presented Sec. 4.1 (Fig. 4.2), the successful autonomous takeoff control inputs (Fig. 4.19) are steady from the activation of full throttle until the takeoff elevator pulse (Fig. 4.19), and elevator is subsequently

relaxed when entering the climb mode. Note that in the plots control inputs are reported in microseconds of PWM signal to provide a common scaling for the different control ranges and to provide intuition for the reader acquainted with RC servo operation. The airspeed and altitude profiles of an early autonomous flight are presented in Fig. 4.20 and flight attitude data is presented in Fig. 4.21. We observe that the vehicle climbs quickly until initiating a short cruise segment at 136s with some overshoot after the mode change. Finally the guidance system switches to descent at 139s, visible as a rapidly initial climb to bleed airspeed, and descends to the water surface. Just as they were observable in the piloted flight profiles presented in Sec. 4.1 the pitch disturbances during takeoff acceleration and the right-roll tendency immediately after liftoff can be clearly seen in the automated flight response (Fig. 4.21). Also visible in the Phase I flight data (piloted and autonomous) is a right-roll tendency just after takeoff (Fig. 4.21) caused by an early stall tendency in the right wing due to manufacturing inconsistencies. This defect necessitates tight closed-loop lateral control for takeoff and also highlights the need for a quick release of the takeoff elevator deflection after separation from the water is achieved. Note that the Phase II vehicle benefited from manufacturing improvements and never exhibited a strong asymmetric stall tendency.

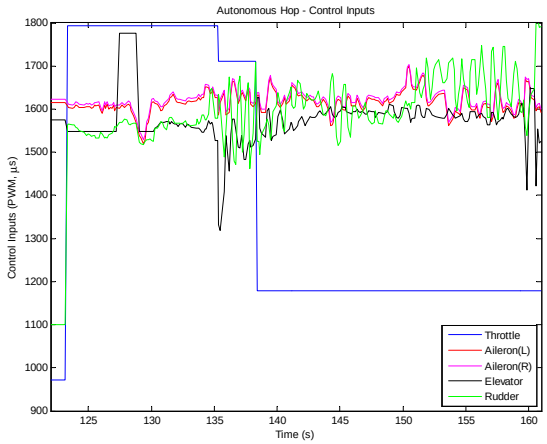
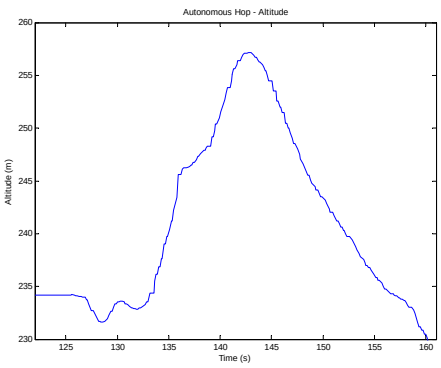
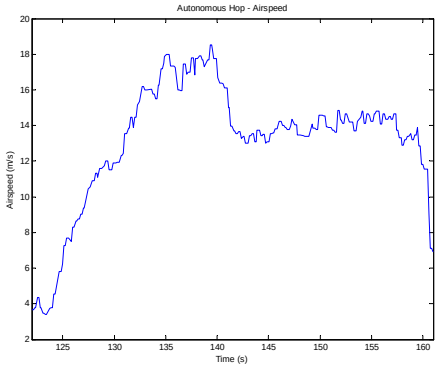



Figure 4.19: Autonomous Flight Control, Phase I

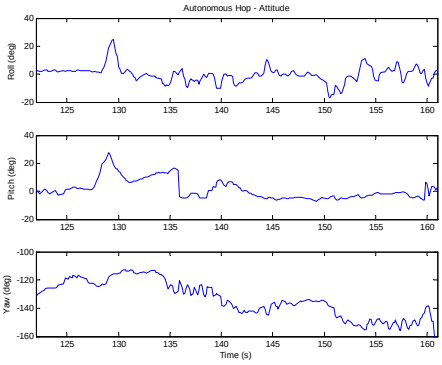



(a) Altitude

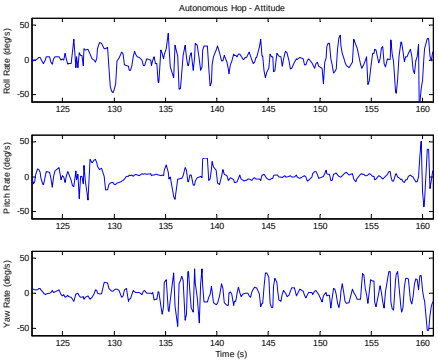



(b) Airspeed

Figure 4.20: Autonomous Flight Profile, Phase I



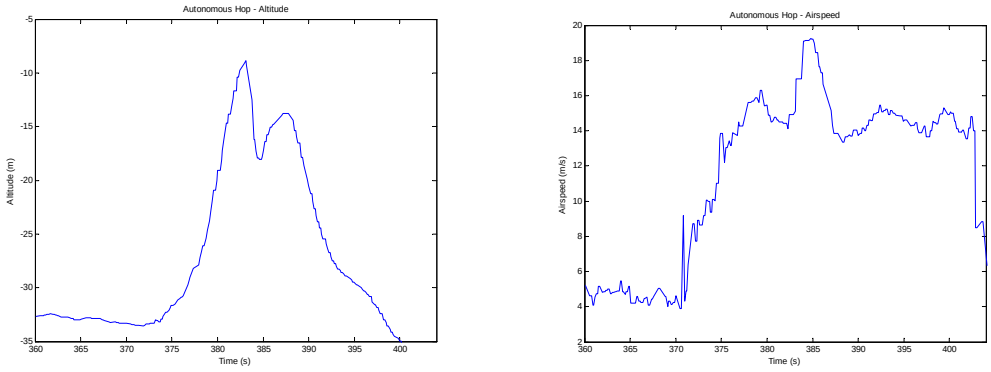

(a) Attitude




(b) Attitude Rates

Figure 4.21: Autonomous Flight Attitude, Phase I

### 4.4.5 Mode Transition Performance




(a) Altitude (b) Airspeed

Figure 4.22: Autonomous Flight, Phase I - Monterey 2007

Oscillatory excitation of the phugoid mode from impulsive mode-switching can be observed in the large altitude and speed variations (~383s, Fig. 4.22b) in the Phase I flight data during the first Monterey Bay sea trial in 2007. These flights predated the inclusion of mode-switch trajectory smoothing and switched instantaneously from climb to cruise and cruise to descent. After the inclusion of trajectory smoothing to the transitions from climb to cruise and cruise to descent the flight profiles were significantly improved. Figure 4.23 shows the response of the attitude variables of the Phase I vehicle before and after the implementation of the mode-switch interpolation strategy for the same vehicle with the same control gains.

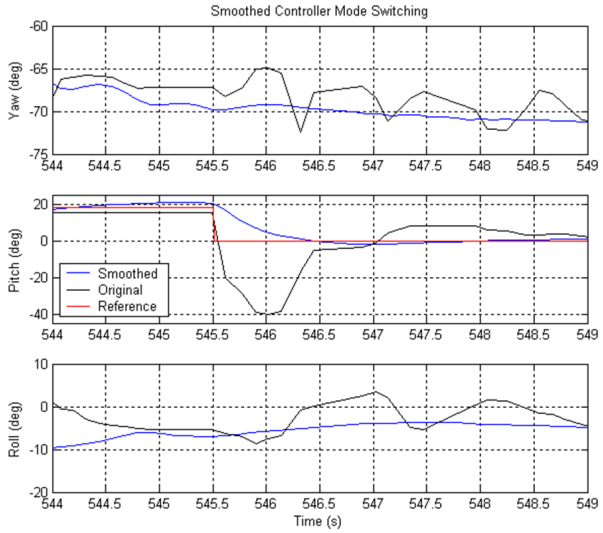



Figure 4.23: Smoothed Pitch Response, Phase I

### 4.4.6 Watch Circle Crossing Performance

Following the first autonomous flight tests in Monterey bay the Phase I Flying Fish was deployed for a watch-circle maintenance test (Fig. 4.24). The first watch-crossing guidance strategy that the Flying Fish team tried was, as previously discussed, to target the circle center and to descend slowly from the center to the edge of the circle on an upwind heading. Unfortunately the early turning guidance suffered several deficiencies, primarily the use of insensitive wings-level slipping (thrust-based) for turning. This deficiency combined with strong cross winds can be clearly seen in the steadily accumulated heading error of the GPS ground-track relative to the watch-circle boundary.

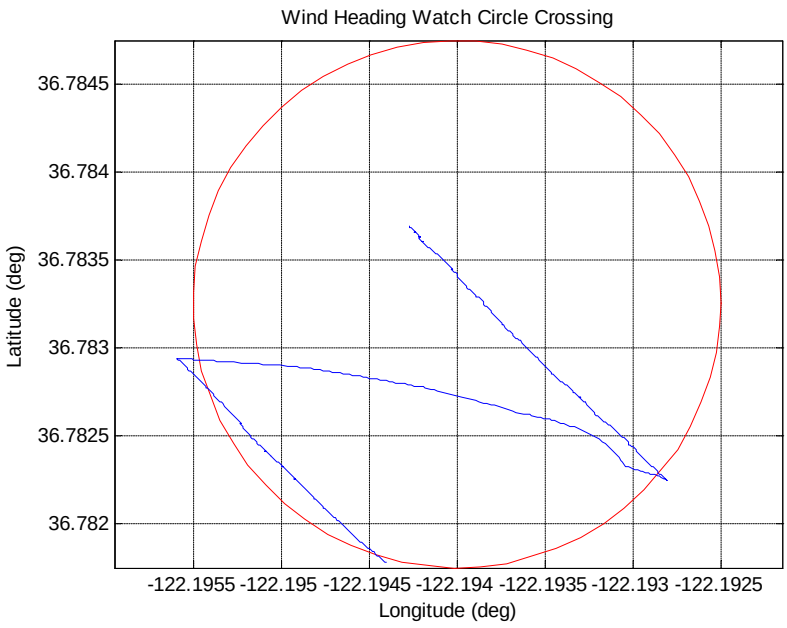



Figure 4.24: Watch Crossing, Phase I

After the Monterey Bay testing was the banked turning intercept-guidance law outlined in Sec. 4.3.1 was implemented in the FMS and the Phase I system was deployed for another round of field testing. After field tuning the guidance strategy over several flights the updated guidance system flew directly over the watch-center in our last field test deployment with the Phase I vehicle (Fig. 4.25).

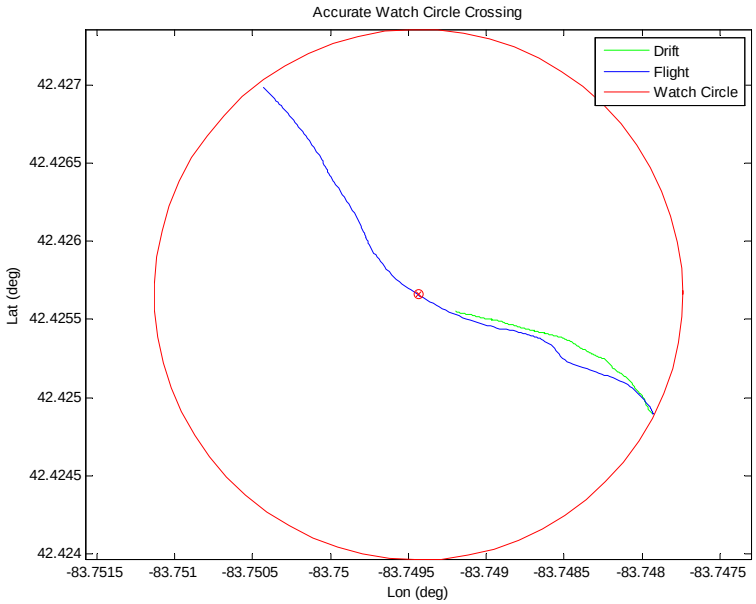



Figure 4.25: Accurate Watch Crossing, Phase I

In the second phase the vehicle the watch-crossing strategy was updated, as discussed in the guidance section above, to target the point on the circle opposite the point that the vehicle encounters the circle boundary. After the upwind takeoff and upon reaching cruise altitude the vehicle turns directly to this target and descends once it is in range. The strategy is meant to reduce flight time by flying direct to the destination and to maximize drift time (under the assumption that drift starts from the center of the circle). During flight tests the vehicle accurately selected the opposing landing site, initiated upwind flight at the watch-circle boundary, and executed banked turns to cross the circle within only a few days of its maiden autonomous flight (Fig. 4.26). Note that while the landings were held short of the watch boundary to meet sponsor demonstration requests during this early controller development the crossing and target tracking accuracy was still good.

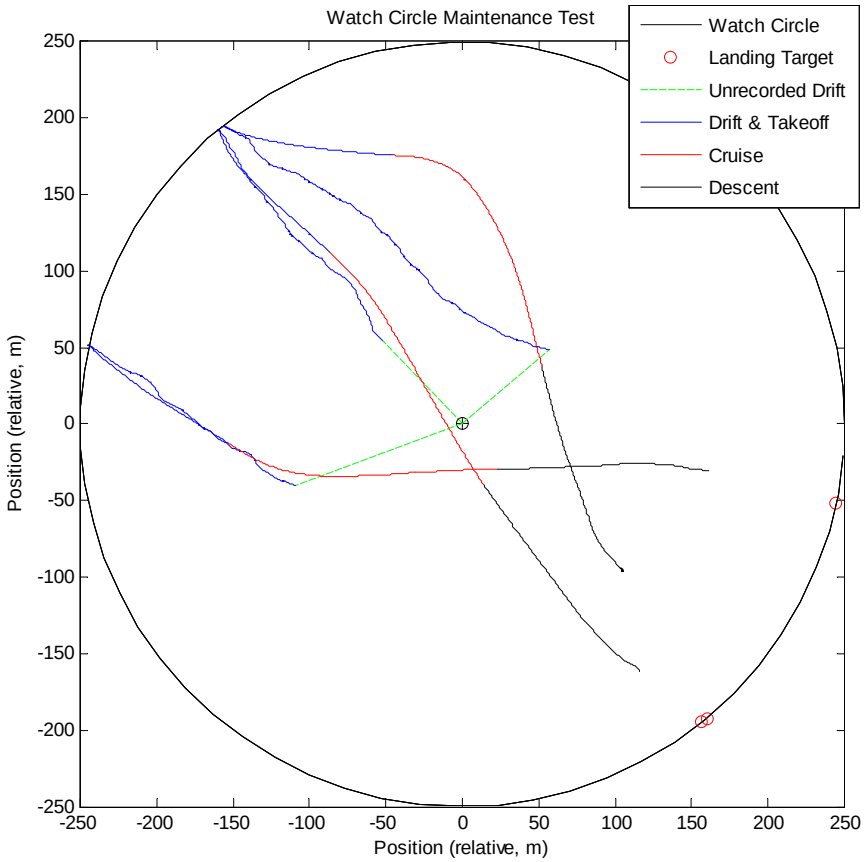



Figure 4.26: Improved Circle Crossing, Phase II

### 4.4.7 Unattended Deployment Requirements

While the control performance necessary to conduct experimentation and basic autonomous relocation was achieved, a higher level of control performance will be required for long-term unattended deployment. The tolerance for control error is necessarily lower for unattended deployment as the unmonitored system will be required to execute numerous flight sequences without intervention or recovery. Furthermore, the high level challenges facing long-term deployment, including energy-based mission planning, will require robust and reliable underlying control functionality to be successful. Additional flight testing would provide an opportunity to improve control tuning and better understand transition dynamics. Further testing would also help to characterize the stability margins of the trim conditions and better delineate the

bounds of the stable closed-loop flight envelope.

A future flight test program in preparation for unattended deployment would likely include several components: (1) system identification, (2) precise control tuning, (2) trim state experimentation, (3) dynamic mode excitation, (4) envelope exploration, and (5) control analysis and refinement. A test pilot or autopilot scripting mechanism could be employed to direct specific maneuvers and control actions and measure system response from which system identification tools could extract dynamic response characteristics.[26] System response information can be used to improve gains through either first-principle methods such as Ziegler-Nichols gain relationships (Eqs. (4.4)-(4.6)) or through the development of a dynamic models for control simulation. Further piloted tests could be used to induce oscillation or to force the autopilot to seek a stable trim from a variety of flight conditions for flight envelope and stability analysis. The test-pilot or autopilot can also conduct a program of trim state expansion to characterize variations in trim conditions over a range of flight conditions; this work would target the development of automated trim state determination for application following any future vehicle/payload reconfiguration. The combination of these flight activities should provide a comprehensive characterization of vehicle and controller performance as well as a set of well-tuned control gains for reliable accurate command following.

# CHAPTER V

# Fault-Tolerant Air Data System

Conventional air data systems (ADS) probes provide direct measurement of aircraft-relative wind as a combination of airspeed ($V$), angle-of-attack (AOA, $\alpha$), and sideslip (SS, $\beta$) as well as pressure-based altitude ($h$) estimates. As stability margins are traded for efficiency and performance and the level of automation increases the accuracy and reliability of wind-relative velocity measurements becomes increasingly important. ADS are therefore among the most critical sensor packages onboard an aircraft and are generally comprised of similar, though not strictly homogeneous, environmental sensors. The nature of ADS measurements requires that probes be directly exposed to the flight vehicle's operating environment and, as such, they are vulnerable to a wide range of external/environmental factors. Outside of direct human error and mechanical damage, foreign material infiltration and atmospheric anomalies (e.g., super-saturated air) are two of the most common causes of ADS probe failures.[27, 28, 29, 30] As these failures will tend to affect similar sensors identically, the majority of nearly homogeneous ADS are potentially subject to complete sensing loss despite redundancy. The immediate consequences of ADS failure are incorrect airspeed/direction and altitude readings. A human pilot will likely recognize the failure if the reading is clearly incorrect, but will be less likely to immediately notice if the reported values are reasonable even if they are based on incorrect measure-

ments. Despite recognition, the human pilot might still have trouble maintaining stable flight without airspeed, particularly in turbulent atmospheric conditions. An autopilot, whether part of a manned or unmanned aircraft system, nominally incorporates airspeed into its flight control laws. Upon failure, if the erroneous data is not detected, control excursions can be substantial and induce unsafe flight conditions (e.g., pitching to a dive given an airspeed approaching stall). Deployed commercial and UAS autopilots are generally not adaptive to incorrect ADS information, so the pilot of a manned aircraft will typically initiate manual flight control without direct knowledge of airspeed, while a UAS will either execute a flight termination sequence or be controlled remotely, again without airspeed data.

Subsequent to the observation of ADS failures first on the Phase I vehicle and later on the Phase II Flying Fish platform a broader investigation into the rates and impact of ADS failure on commercial, military, general, and unmanned aviation was conducted. This study suggested that both the prevalence, rates, and impact of such failures are nontrivial across all classes of aviation, affecting commercial,[28, 30] military,[31, 32] and general aviation aircraft.[27, 29, 30]. Reports on the subject suggest that the problem may be worsening with growing air travel volume, and while research is being conducted on related topics no uniform solution yet exists. Aviation safety databases provide evidence of significant commercial aviation losses due to ADS failure. The potential for an autopilot or pilot to react improperly to erroneous wind data introduces appreciable risk, as evidenced by accidents such as Aero Peru Flight 603, in which ground crews failed to remove tape from the pitot-static system after cleaning the aircraft, or the X-31A at NASA Dryden, in which pitot icing compromised system readings. Note that although redundant ADS probes are present on most high-cost aircraft, common failure modes or incorrect failure diagnosis has also resulted in catastrophic accidents, such as Austral Lineas Aeroeas Flight 2553 in which the flight crew incorrectly decided to reference the pilot's failed airspeed indi-

cation and induced structural failure by exceeding safe airspeed limits. The Aviation Safety Network database has records of at least eleven ADS (pitot probe) failures over the past three decades that have resulted in significant damage and/or loss of life.[30] These examples alone represent a nontrivial financial loss and 342 documented fatalities (339 in the past 15 years). More recently, interim accident reports for Air France Flight 447 indicate air data system anomalies were experienced[33] and that Airbus platforms alone have had 35 documented incidents of multiple ADS failures since 2003.[34] This is a nontrivial result even over the large number of total flight operations conducted by Airbus airliners given the likelihood that many transient ADS failures were not documented. The effect of ADS failures can also be observed in general aviation (GA) aircraft incident records,[27, 28, 29] but concise statistics have proven more difficult to collect. For example, the Aviation Safety Reporting System database[30] contains numerous instances of general aviation ADS-related failures but aggregate results for this specific contributing factor are not readily available. Further complicating analysis, the varied causes, effects, and results of ADS faults can lead to failure statistics being associated with a number of different classifications (e.g. inclement weather, instrument fault, and flight control failure). Furthermore, though no comprehensive failure statistics are available, it follows that failures would also be experienced in UAS ADS probes subject to similar failure-inducing factors.

The avionics of commercial aircraft now feature multiple redundant air data sensor units with some but not all designs utilizing purposeful dissimilarities between related/redundant software and hardware to increase robustness to a particular flaw; these systems require relatively complex redundancy negotiation and consensus voting strategies to operate.[10, 11] It has been noted that this complexity may induce unexpected and counterintuitive results, even to the point of introducing new failure modes,[35] it is still the case that these systems have demonstrated high reliability. Nevertheless, these systems are still fundamentally vulnerable to failures in their

external sensing apparatus. Further, while these complex redundancies and failure mitigation strategies enable the negotiation of failing redundant sensors these systems are still largely unable to handle common failure mechanisms simultaneously disabling entire classes of like sensors. These shortcomings are the fundamental reasons to research novel mechanisms for ADS failure mitigation.

This chapter will present an ADS failure mitigation algorithm that fuses data from multiple wind and inertial sensors to diagnose and react to air-data sensor failures. The methods can be applied to a range of systems and sensor types but, for the purposes of this research, the specific sensor measurements are defined in the context of instrumentation affixed to the first and Phase II Flying Fish systems[24, 36] with emphasis placed on the latter platform's more sophisticated ADS instrumentation. The Phase II Flying Fish ADS incorporates two five-hole ADS probes and a propeller anemometer. The 5-hole probes combine pitot/static airspeed measurement and barometric altitude with lateral/vertical differential pressures for the determination of angle of attack and side-slip angle. A heating element on each 5-hole-probe allows for cold weather operation and has sufficient heating capacity to rapidly evaporate freshwater blockages. Heat-based pitot clearing has not been evaluated in a marine environment where the mineral content of the water may contraindicate the application of evaporative clearing. The propeller-anemometer uses hall-effect sensors to measure the rotation rate of a small high-pitch propeller in order to determine airspeed. Dual hall-effect sensors within the anemometer head provide redundant measurement of propeller rotation. Following the background materials our sensor fault detection and data-fusion algorithms are presented. ADS fault management results are provided with flight data from both generations of unmanned seaplanes, demonstrating the efficacy of this solution.

## 5.1 Background

Most research into ADS fault tolerance and recovery tends to fall into one of three categories: (1) signal-based diagnostics, (2) alternative sensing mechanisms, and (3) strategies for operating without traditional ADS sensors.

Looking first to ADS-specific failure research we find the seminal work of Houck and Atlas which provides insight into fundamental mechanisms for ADS failure diagnosis. Houck and Atlas analyzed failed ADS sensor signals and were amongst the first to propose that probe blockage reduced signal energy levels, that large signal variations were generally sufficient (but not necessary) to demonstrate sensor functionality, and that such signal characteristics alone might be used to indicate air-data probe health.[37] Very few examples of this type of analysis exist for ADS-specific applications. Houck and Atlas ultimately proposed that even at a fixed altitude the nominally-constant static pressure varied slightly as a function of acceleration and that the derivative of the static-port pressure signal would be a good indicator of probe health. Unfortunately, independent static pressure measurements are not always available in UAS applications as the desire for volume, weight, and cost savings make the implementation of a single pressure transducer for pitot-static measurements more likely. Regardless, Houck and Atlas' methods utilize or suggest several of the tools that will be employed in this dissertation including individual signal characterization and comparison with predetermined signal statistics and operating thresholds.

A widely applied strategy for avoiding common failures modes within a given class of sensors is to employ alternative instrumentation for the redundant measurements. Variations on the common pressure-based ADS pitot-probe include body-distributed flush air-data sensing (FADS) systems[38] and self-aligning multi-hole conical probes.[39] FADS systems employ pressure ports with openings flush to, and distributed over, a vehicle's aerodynamics surfaces while self-aligning conical probes are driven by

pressure forces into alignment with local the airflow. In both cases the geometrically-related measurements collected at distributed sensing locations are used with pressure and/or flow models to estimate distributed airflow characteristics from which the ADS states can be resolved. These systems can provide both fault-tolerance and error reduction provided they are designed such that ADS states are observable from different subsets of probes. Further, the novel design of the sensors changes the potential failure modes thereby reducing the likelihood of simultaneous failures exhibited by the more common traditional ADS probes. However these sensors are still pressure-based thus susceptible to the same environmental factors as the more common pitot-probe ADS sensors. Other alternatives that would not be as susceptible to blockage situations include surface application of piezoelectric pressure sensors, ultrasonic sensing, or mechanical systems such as propellers, turbines, and directional vanes. The system presented in this research will, as previously indicated, include a preliminary prototype for alternative low-cost ADS sensing in the form of a propeller-anemometer for our relatively slow-speed seaplane UAS flight operations. In this manner, the experimental system avoids having failure modes common across all of the vehicle's air-data sensors.

In addition to physical sensor redundancy and non-homogeneity, researchers have also proposed schemes for flight operation in the absence of ADS measurements. Continued improvements to sensors and filtering in high-accuracy inertial navigation systems (INS), such as those found onboard military and commercial aircraft, have given rise to model-based mechanisms for deriving indirect estimates of ADS states.[32, 40, 41]In the UAS segment, the development and proliferation of lower-cost micro electromechanical (MEMs) GPS and GPS-integrated INS, has given rise to a number of UAS projects utilizing flight control laws that omit ADS variables completely.[42, 43] Implementations of the latter generally impose performance limitations (e.g., reduced tolerance for elevated wind speed or high wind variability), require particularly wide

stability margins, and/or must employ some alternate motion sensing mechanism such as machine vision and optical flow.[44] The concept of ADS state estimation has evolved in two basic formulations differing by a time derivative. In both cases, starting from known initial conditions (nominally conditions at the time of ADS failure), the estimation algorithms infer wind from either: a) the difference between a wind-unaware dynamic estimate of inertial velocity and the measure of inertial velocity[32] or, b) the integration of lateral, longitudinal, and vertical accelerations by a dynamic model that includes wind.[40, 41] Regardless of the method, trigonometry is applied to the resulting velocity triangles to estimate angle of attack, sideslip, and airspeed. The difficulty with these methods is their reliance upon high-accuracy high-rate inertial sensors and upon high-fidelity dynamic models. While these assumptions are reasonable for the military vehicles for which much of the referenced research was intended, they are not necessary appropriate for a slow-flying, low-cost UAS with less accurate MEMs-based sensors. Targeting a low-cost UAS platform, the algorithm presented in this chapter will develop an average wind estimate based on the difference between measured airspeed and estimated sensor motion. Application of the average wind field to INS velocities is shown to provide reasonable airspeed estimates in the absence of trusted ADS measurements over the average mission profile of the Flying Fish systems.

## 5.2 Implementation

Fault tolerance for our ADS is based on a confidence filter (Fig. 5.1) composed of three algorithms: (1) a signal-fault detection scheme, (2) a confidence-discriminate data-fusion procedure, and (3) an IMU driven wind-estimation calculation. Signal fault detection extracts and tests signal characteristics to estimate the likelihood of sensor failure. The confidence-discriminate data-fusion module combines the signal fault

results into sensor confidence values. Confidence values are used to judge sensor fitness and eliminate signals from failed sensors before being used to build a weighted-average data-fusion airspeed estimate. The wind-estimation module utilizes the resulting composite air-data vector to refine the local wind model which is subsequently used, in conjunction with inertial navigation measurements, to estimate the air-data vector in the event of total ADS sensor feedback failure. This wind-estimate also provides a baseline wind vector for judging individual sensor measurements. The three algorithm stages are executed sequentially. Wind estimates are fed back to the signal-fault detection scheme to become one of the confidence-rated signals combined in the data-fusion cycle.

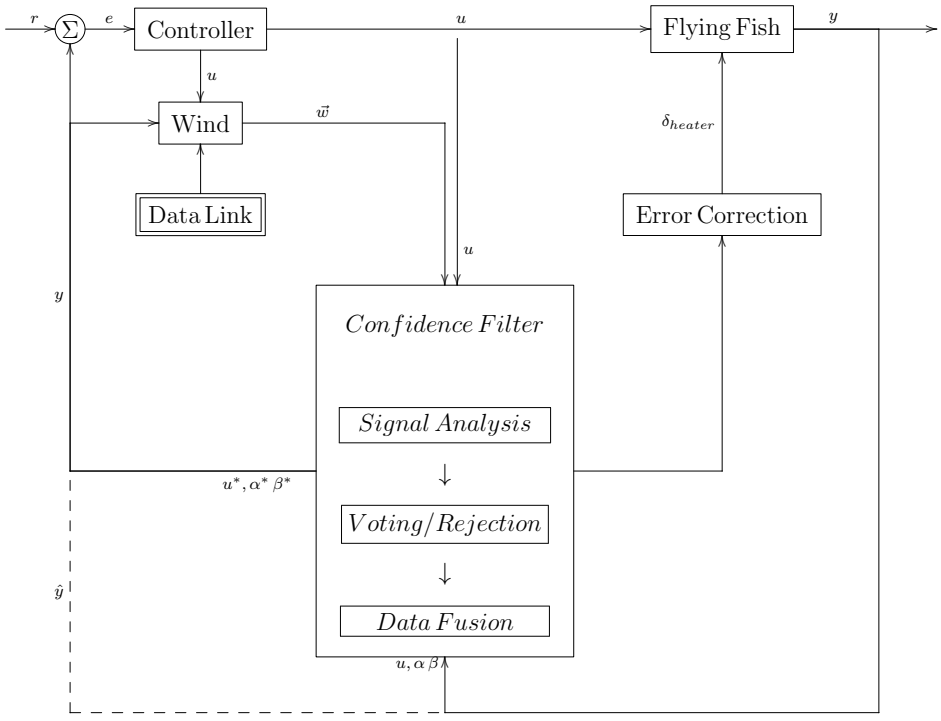



Figure 5.1: System Configuration with ADS Confidence Filter

### 5.2.1 Signal Fault Detection

The purpose of the signal-fault detection scheme is to discern anomalous operating conditions that may indicate a sensor failure. The number, type, and redundancy of

the sensors are not considered at this stage, rather each signal is judged based on its individual parameter-based model. Signal models are captured from the sensor specifications and the offline analysis of both failed and operational sensor signals. This stage of the algorithm is composed of three major procedures: signal characteristic extraction, signal model determination, and signal fault testing.

Common methods for signal characterization in fault detection schemes include statistical metrics, such as arithmetic average and variance, and spectral analysis methods, such as the wavelet[45] or Fourier[46] decompositions. Spectral decompositions are typically applied for fault detection in systems with cyclic behavior[45] or harmonic content/excitation.[47] The efficacy of frequency decomposition methods for common classes of air-data sensors are, as of yet, unknown, but pressure-based sensors do not generally have a strong frequency component. Frequency-based diagnostics may prove applicable to rotary propeller anemometers, but preliminary analysis for this application yielded inconclusive results subject to sparse failure data, uncertain system degradation trends, and relatively slow measurement frequency. For the remaining pressure-based ADS signals we employ an arithmetic average and variance algorithm for signal characteristic extraction. Unlike decomposition strategies, which are focused on frequency-keyed information, the mean and variance are utilized to obtain smoothed, low-pass filtered, characteristics of a signal.[45] In this case we will be using a sliding average and variance formulation. The $k^{\text{th}}$ arithmetic-average and variance over $m$ samples of the $n^{\text{th}}$ signal ($s_n(k)$) are given, respectively, for the set of non-negative/not-all-zero weighting coefficients ($\tau_i$), by the formulae:

$$\bar{s}_n(k) = \frac{\sum_{i=1}^{m} \tau_i s_n(k+1-i)}{\sum_{i=1}^{m} \tau_i} \tag{5.1}$$

$$\nu(s_n(k), k) = \frac{\sum_{i=1}^{m} \tau_i (s_n(k+1-i) - \bar{s}_n(k))^2}{\sum_{i=1}^{m} \tau_i} \tag{5.2}$$

To reduce the number of computations per iteration, the sums of time-invariant weighting factors are normalized producing the following simplified formulation:

$$\bar{s}_n(k) = \sum_{i=1}^{m} \tau_i s_n(k+1-i), \quad \sum_{i=1}^{m} \tau_i = 1 \tag{5.3}$$

$$\nu(s_n(k), k) = \sum_{i=1}^{m} \tau_i (s_n(k+1-i) - \bar{s}_n(k))^2 \tag{5.4}$$

Exponential weighting is utilized to favor the most recent data, reducing phase delay between the raw and filtered signals. Our parametrized exponential weighting formula is given by:

$$\vec{\tau}(\xi) = [\tau_1(\xi), ..., \tau_m(\xi)]^T = \begin{cases} \tau_i(\xi) = \frac{\xi(1-\xi)^{i-1}}{\sum_{k=1}^{m} \xi(1-\xi)^{k-1}}, \quad i = 1, ..., m & 0 < \xi \leq 1 \\ \tau_1 = \tau_2 = ... = \tau_m = 1/m & \xi = 0 \end{cases} \tag{5.5}$$

The decay parameter $\xi$ determines the relative influence of aging data points, defining a continuum between preserving only the most recent sample at one extreme ($\xi = 1$) and approaching equal weighting of all points at the other extreme ($\xi$ arbitrarily close to zero). We explicitly define equal weighting for the special case $\xi = 0$. In this case, the unweighted arithmetic mean and variance equations are recovered (Eqs. (5.1)-(5.2)).

To apply Eqs. (5.1)-(5.5) the sliding sample-window size, $m$, and the weight decay parameter, $\xi$, must be selected. These values were tuned empirically to balance signal tracking against delay and low-pass filter performance for each signal. It may be possible to formulate an optimal tuning of these parameters if a cost function can be formulated based on the signal following characteristics and low-pass filtering requirements. This approach did not prove necessary for the Flying Fish sensor systems but may be appropriate for any larger or more complex ADS. The resulting

tuned sliding average and variance formulations were then applied to the signals of functional, failed, and failing sensors to extract sensor model parameters for each flight mode (taxi, takeoff, climb, cruise, turning, descent, and landing). Average variance ($\bar{\nu}(s_n, mode)$), peak signal change rate ($\dot{s}_{max}(s_n, mode)$), and peak rate of variance change ($\dot{\nu}_{max}(s_n, mode)$) were extracted from functional sensor data for each signal and each flight mode. Performing the same calculations on operational, failing, and failed sensor data enabled the determination of tolerances for deviation in the model parameters. During this process it was discovered that the peak rate of change during a failure was generally within the normal dynamic range of the UAS sensors. That is to say, the initial dynamics of probe failures are almost indistinguishable from the dynamic response rates of the functional system based on rates alone. Subsequently, peak rate of change was discounted as a fault detection metric but tolerances for deviation from average variance ($\mathbb{T}_{\bar{\nu}}(mode)$) and peak rate of variance change ($\mathbb{T}_{\dot{\nu}_{max}}(mode)$) were recorded for each sensor. Drawing from the manufacturer's specifications and failed sensor data the saturation limits of the sensor ($s_{n,sat} = \{s_{n,min}, s_{n,max}\}$) were quantified to give a total of four fault-detection parameters per sensor, per mode: $T_{\bar{\nu}}, T_{\Delta\nu_{max}}, s_{n,min}, s_{n,max}$. The collected set of all fault-detection parameters, averaging window, and weighting decay parameter for each sensor comprise the parametrized sensor signal model.

The final step in the signal-fault detection process is the evaluation of extracted characteristics using the metrics and tolerance stored for each sensor, for each flight regime. The combination of model parameters produces three distinct sensor-fault tests. The first test determines if the signal variance exceeds the variance deviation threshold for the current flight mode:

$$\|\nu(s_n(k), k) - \bar{\nu}(s_n, mode)\| > \mathbb{T}_{\bar{\nu}}(mode) \tag{5.6}$$

The result of the variance test is recorded as a binary pass ("1") or fail ("0") vote ($\Lambda_{var}(s_n, k)$) for each sensor at each time step, $k$. The second test determines if the signal variance increases or decreases too quickly and violates the peak-variation-rate tolerance. In order to make this determination we require a smooth baseline measurement of the signal variance for which we compute a sliding average of the variance results ($\bar{\nu}(s_n(k), k)$). Again the window size, $m$, and the weight decay parameter, $\xi$, are selected empirically to find an acceptable trade-off between signal following, smoothing, and delay. The signal test is formulated as:

$$\|\nu(s_n(k), k) - \bar{\nu}(s_n(k),\, k)\| > \mathbb{T}_{\dot{\nu}_{max}}(mode) \tag{5.7}$$

The result is a binary pass-fail vote for the variance rate test ($\Lambda_{rate}(s_n, k)$) of the $n^{\text{th}}$ sensor at the $k^{\text{th}}$ time step. The final test considers if the sensor has entered a saturation region. Data analysis indicated a threshold of 3% of the saturation limits provided an appropriate balance between missed and false fault detection. The set of all flight data collected for the original Flying Fish demonstrated that the output of an operational sensor remained outside the 3% saturation threshold for >99% of all measurements. Conversely, failures that produced saturated signals approached within 3% of the saturation limit for 99% of subsequent incorrect measurements. The only observed occurrence of saturation aside from failure was for low-pressure saturation subject to extremely-low/zero air speed (or a slight tailwind) while the vehicle was drifting. The tests for saturation take the form:

$$s_n(k) < s_{n,min} + (0.03)(s_{n,max} - s_{n,min}) \tag{5.8}$$

$$s_n(k) > s_{n,max} - (0.03)(s_{n,max} - s_{n,min}) \tag{5.9}$$

This test produces, as with the previous tests, a binary indication of a saturation test

failure of the $n^{\text{th}}$ sensor at the $k^{\text{th}}$ time step ($\Lambda_{sat}(s_n, k)$). Finally, the time history of sensor fault votes is output to the confidence-discriminate data-fusion algorithm to develop sensor confidence values.

### 5.2.2 Confidence-Discriminate Data Fusion

The goal of confidence-discriminate data-fusion is to leverage signal confidence and redundant-data comparisons to combine like measurements while excluding failed sensors. This sensor discrimination and data fusion process is comprised of three steps: signal confidence determination, failed sensor rejection, and final data fusion.

Confidence of the $n^{\text{th}}$ signal ($C(s_n, k)$) is developed from the time history of signal-fault votes for that sensor in a two step process. First the signal-fault votes are accumulated into a probability that the given sensor has passed the specific fault test. This probability is created by using a large-window moving average which allows failure votes to have a long influence period while also mitigating spurious intermittent false-negative/positive votes. For this we define probabilities for each of the three primary fault types: signal variance exceeds threshold ($P_{var}$), rate of variance change exceeds threshold ($P_{rate}$), and signal exceeds saturation tolerance ($P_{sat}$). The probabilities are created by a moving average of the binary voting history for each fault:

$$P_{var}(s_n, k) = \bar{\Lambda}_{var}(s_n, k) \tag{5.10}$$

$$P_{rate}(s_n, k) = \bar{\Lambda}_{rate}(s_n, k) \tag{5.11}$$

$$P_{sat}(s_n, k) = \bar{\Lambda}_{sat}(s_n, k) \tag{5.12}$$

The second step is to combine the three failure-voting probabilities into the final signal confidence value by a weighted average. The two variance probabilities are given even weighting (0.3) while the saturation probability is given a slightly higher

weighting (0.4).

$$C(s_n, k) = (0.3)P_{var} + (0.3)P_{rate} + (0.4)P_{sat} \tag{5.13}$$

This distribution is best explained by examining the features of our fault detection process. Specifically, a complete saturation fault indicates a definitive failure condition whereas the variance-based faults indicate only some [non-zero] likelihood of sensor failure. By giving a slightly greater weight to the saturation test we can select a required sensor acceptance threshold (0.7) that is always exceeded in the event of a complete saturation failure but that cannot be surpassed by any single variance failure. Armed with confidence values for each signal the algorithm can execute both sensor-recovery actions and the algorithms responsible for error-rejection and data fusion.

The heating elements installed to the 5-hole probes of the Phase II vehicle afford the failure-mitigation algorithm a mechanism for recovering from water blockage failures. When confidence in a sensor drops below the acceptance threshold ($< 0.7$) the signal is automatically eliminated from the fusion algorithm. The control system can subsequently activate a digital control line to a MOSFET power amplifier that delivers battery-level voltage to a Nichrome heating element integrated to the 5-hole probe. The heating system was designed for high-energy short-term heating to rapidly clear a blocked sensor and not for sustained heating in freezing temperatures; the heater is never applied for more than 30s and requires at least 30s between applications. If the probe is successfully cleared the sensor will automatically be reintegrated to the data fusion algorithm when the sensor confidence reaches the acceptance threshold. During heating false readings can be induced by increased pressure behind the blockage; these events are unlikely to increase sensor confidence and should not impact the data fusion algorithm. In the event of multiple simultaneous blockages the system would attempt to clear only one probe at a time to reduce the risk of having two

sensors producing similar erroneous recovery signals. The heating system can also be activated by a command from the ground station if necessary.

One of the most important requirements for a fault-tolerance system is an accurate mechanism for judging and rejecting questionable signals from the set of all available sensors. Willsky's survey of design methods for failure detection provides a good summary of this field.[48] Common methods include neural networks, voting or outlier rejection, model-based analysis, and filter-based techniques, including recursive least-squares and the Kalman filter.[49, 50, 51, 52, 53] The algorithm defined for this work adopts an outlier rejection (OR) and voting sequence. These methods are closely related but typically utilize different operating principles. Voting schema are comprised of rule-based judgments. Outlier rejection, on the other hand, generally relies on statistical analysis and, while subjectivity remains in the selection of metrics and thresholds, OR methods are usually governed by commonly accepted statistical practices/measures (e.g. using a fixed multiple of the standard deviation to define an outlier). A comprehensive treatment of outlier rejection in statistical data can be found in Barnett and Lewis.[54] For this dissertation we will utilize sensor confidence as a per-signal voting mechanism and employ a simple outlier rejection scheme whenever three or more redundant signals are available. As previously indicated a confidence threshold of 0.7 was empirically selected; whenever a signal's confidence drops below this threshold it is rejected from data fusion. If three or more redundant measurements remain after confidence-based elimination they are subject to a consensus-seeking outlier rejection scheme that eliminates signals too dissimilar (subject to a threshold) from any majority amongst all signals. Utilizing Eqs. (5.1) and (5.2) confidence-weighted average and variance are computed over the full set of redundant sensors. The square root of the variance gives the standard deviation of the set of signals. Any signal that is more than one standard deviation from the average

is eliminated.

$$\|s_n(k) - \bar{s}_n(k)\| > \sqrt{\nu(s_n(k))} \tag{5.14}$$

The remaining step is data fusion. Hall and Llinas provide a comprehensive introduction to data fusion;[55] a comparison and classification of data fusion operators can be found in Bloch's 1996 manuscript.[56] It can be shown that a great many of the filter-based data fusion algorithms are based on least-square error concepts.[57] A noisy measurement ($z_i$) of some value ($x_i$) subject to zero-mean uncorrelated white noise ($v_i$) with variance $\nu_i$, can be written as:

$$z_i(k) = x(k) + v_i(k), \quad v_i \sim N(0, \nu_i) \tag{5.15}$$

$$E(v_i v_j) = 0, \quad i \neq j \tag{5.16}$$

We can formulate an unbiased estimator ($x^*$) as the weighted summation of noisy measurements ($z_i$):

$$x^*(k) = \sum_{i=1}^{n} \tau_i z_i(k) \tag{5.17}$$

$$E(x^*(k) - x(k)) = 0 \tag{5.18}$$

Minimization of the expected error between the estimator and signal recovers the weighted arithmetic average. Subsequent minimization of the mean square error between the estimator and signal produces an ideal weighting based on variance:[57]

$$\tau_j = \frac{\prod_{i \neq j}^{n} \nu_i}{\sum_{h=1}^{n} \prod_{k \neq h}^{n} \nu_k} \tag{5.19}$$

However, since the signal confidence calculations already indirectly consider signal variance it is more useful at this juncture to substitute confidence-based weights as they capture a greater amount of data than variance-based weighting alone. A linear

confidence weighting, which reduces to zero at the confidence threshold, can be given for $n$ redundant sensors by:

$$\tau_{i,conf} = \frac{c_i(k) - (0.7)}{(\sum_{j=1}^{n}(c_j(k) - 0.7))} \tag{5.20}$$

Recall that the low-confidence signals have already been eliminated, so the weighting strategy above gives a normalized positive weighting that satisfies the requirements for the weighted average. The resulting combined air-data measurements are output. If these results are determined valid they will be used for the next guidance, navigation, and control cycle; if the results are deemed invalid based on a comparison with independent wind estimates along with sensor confidence levels, an inertial-only wind estimate is used as a backup, a procedure described in the following section.

### 5.2.3 Wind Estimation

The goal of the wind estimator is the generation of an air-data vector that can serve as both a reference for ADS failure detection and a failsafe reading to promote safe pilot/autopilot operation of ADS-dependent flight controls in the event of partial/complete air-data sensor failure. The wind estimation scheme has two effective modes: nominal operation, wherein some number of ADS sensors are functional and wind estimation is dominated by direct measurements, and failsafe operation where, in the absence of ADS inputs, winds are estimated from previously collected wind statics. Refinements to this model would require use of dynamics models for both the aircraft and environmental wind processes, neither of which were readily available for the Flying Fish platforms given time and budget constraints. Such models are typically more readily available for commercial aircraft than for small UAS. In the future, as Flying Fish model development continues, it is anticipated that the addition of a Kalman filter would improved ADS estimates and increased system

tolerance to long flight intervals subject to extended ADS failure.[58, 59, 60] The primary mechanism for wind estimation and, subsequently, air-data vector estimation is a three step process of extracting inertial wind measurements from body-relative sensors, updating the wind estimate, and recovering air-data measures from the wind model and current inertial measurements.

Wind model relationships to body and inertial measurements are expressed by rotation matrices for pitch ($\theta$), roll ($\phi$), and yaw ($\psi$) Euler angles about the $x$, $y$, and $z$ axes, respectively:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{5.21}$$

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{5.22}$$

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.23}$$

The first step in the wind estimation procedure is to resolve an inertial-referenced environmental wind measurement from high-confidence air-data vector and vehicle motions. First, measured airspeed ($V(k)$) for the $k^{\text{th}}$ iteration is resolved into a vector in the aircraft body frame ($B$) using angle-of-attack ($\alpha(k)$) and sideslip ($\beta(k)$):

$$\vec{V}_B(k) = R_y(\alpha(k))R_z(-\beta(k)) \begin{bmatrix} V(k) & 0 & 0 \end{bmatrix}^T \tag{5.24}$$

This vector is subsequently rotated into the inertial frame ($I$) using the aircraft's roll ($\phi(k)$), pitch ($\theta(k)$), and yaw ($\psi(k)$) Euler angles:

$$\vec{V}_I(k) = R_x^T(\phi(k))R_y^T(\theta(k))R_z^T(\psi(k))\vec{V}_B(k) \tag{5.25}$$

To develop an inertial wind measurement ($\vec{w}_I(k)$) we must add the vehicle's inertial frame velocity ($\vec{v}(k)$) as measured by the inertial measurement unit (IMU) to the body relative airspeed vector:

$$\vec{w}_I(k) = \vec{V}_I(k) + \vec{v}(k) \tag{5.26}$$

The next step is to update the actual wind model. For the small unmanned seaplane we employ a simple spatially-uniform average-based wind model. The locally measured wind (primarily during drift) is accepted as the global wind estimate with a weighted time average of the wind measurements used as a reasonable estimate of the current steady wind. Further the wind is assumed to only have velocity components in a local horizontal plane, that is, there is no vertical component of wind. This set of assumptions are reasonable for the Flying Fish mission as the vehicle will transit over only fairly short distances ($<$1000m) and a narrow/low altitude range ($<$100m) during each flight. With an updated wind estimate we can construct the inertial-measure-based ADS estimates. First we recover the estimated inertial-frame airspeed vector ($\vec{V}_I^*(k)$) by differencing the wind estimate with the vehicle velocity:

$$\vec{V}_I^*(k) = \vec{w}_I^*(k-1) - \vec{v}(k) \tag{5.27}$$

Rotating the inertial airspeed vector estimate into the body frame produces a body-frame relative airspeed vector estimate ($\vec{V}_B^*(k)$):

$$\vec{V}_B^*(k) = R_x(\phi(k))R_y(\theta(k))R_z(\psi(k))\vec{V}_I^*(k) \tag{5.28}$$

Trigonometry can then be applied to recover the angle of attack and sideslip values:

$$\alpha(k) = \tan^{-1}\left(\frac{V^*_{B,z}(k)}{((V^*_{B,x}(k))^2 + (V^*_{B,y}(k))^2)^{1/2}}\right) \tag{5.29}$$

$$\beta(k) = \tan^{-1}\left(\frac{V^*_{B,y}(k)}{((V^*_{B,z}(k))^2 + (V^*_{B,x}(k))^2)^{1/2}}\right) \tag{5.30}$$

Airspeed is recovered from the magnitude of the body-frame airspeed vector estimate:

$$V(k) = \left\|\vec{V}^*_B(k)\right\| \tag{5.31}$$

### 5.2.4 Signal Fault Detection Algorithm

The fault-mitigation algorithm was implemented and tested in Matlab and translated into C for the embedded flight system. Development in both Matlab and C focused on efficient implementation of the heavily-used sliding average and variance calculations and the associated data buffering. The computational complexity of the algorithm is fairly low with mathematical operations and memory requirements that scale linearly for each signal as a function of averaging window size. Furthermore, computation requirements are fairly uniform across different sensors in a single system and the complexity of the full algorithm, from detection through fusion, is approximately linear over the number of filtered sensors, $n$ ($\sim O(n)$). This section summarizes the sequential stages of the fault-mitigation algorithm before presenting filter validation and tuning details.

1. Signal Fault Detection Block:

   *Input:* Sensor signals: $s_n(k)$

   *Output:* Fault detection votes: $\Lambda_{var}(s_n, k)$, $\Lambda_{rate}(s_n, k)$, $\Lambda_{sat}(s_n, k)$

(a) Extract signal characteristics (Eqn. 5.3-5.4)

(b) Select test parameters for current flight mode

(c) Perform fault detection tests:

i. Variance within expected thresholds, vote Pass/Fail (Eqn. 5.6):

$\Lambda_{var}(s_n, k) = \{\mathrm{v} | \, \mathrm{v} \in \{1, 0\}\}$

ii. Rate of variance change within expected thresholds, vote Pass/Fail (Eqn. 5.7):

$\Lambda_{rate}(s_n, k) = \{\mathrm{v} | \, \mathrm{v} \in \{1, 0\}\}$

iii. Signal response sufficiently far from saturation, vote Pass/Fail (Eqn. 5.8-5.9):

$\Lambda_{sat}(s_n, k) = \{\mathrm{v} | \, \mathrm{v} \in \{1, 0\}\}$

2. Confidence-Discriminate Data-Fusion Block:

*Input:* Fault votes: $\Lambda_{var}(s_n, k)$, $\Lambda_{rate}(s_n, k)$, $\Lambda_{sat}(s_n, k)$

*Output:* High-confidence ADS values: $\alpha(k)$, $\beta(k)$, $V(k)$

(a) Sensor confidence assessment:

i. Compute probabilities for each failure type (Eqn. 5.10-5.12):

$P_{var}$, $P_{rate}$, $P_{sat}$

ii. Compute confidence for each sensor (Eqn. 5.13):

$C(s_n, k)$

(b) Sensor voting / outlier rejection:

i. Reject low confidence signals ($C(s_n, k) < 0.7$)

ii. Reject outliers (Eqn. 5.14)

(c) Confidence-weighted sensor fusion (Eqn. 5.3, 5.20)

3. Wind Estimation Block:

   *Input:* High confidence ADS values: $\alpha(k)$, $\beta(k)$, $V(k)$

   *Output:* Estimated ADS values: $\alpha^*(k)$, $\beta^*(k)$, $V^*(k)$

   (a) Extrapolate $k^{\text{th}}$ Wind Measurement: $\vec{w}_I(k)$

      i. Resolve airspeed ($V$) as body-frame vector (Eqn. 5.24)

      ii. Rotate air-data vector into inertial frame (Eqn. 5.25)

      iii. Compute wind from inertial airspeed estimate and vehicle motion (Eqn. 5.26)

   (b) Update wind estimate: $\vec{w}_I^*(k)$ (Eqn. 5.3)

   (c) Construct airspeed, AOA, and sideslip estimates from wind estimate:

      i. Compute inertial-frame air-vector estimate (Eqn. 5.27)

      ii. Rotate estimated air-vector into body-frame (Eqn. 5.28)

      iii. Determine estimated ADS values from body-frame air-vector (Eqn. 5.29-5.31)

Algorithm tuning was conducted for a subset of all recorded failures which were analyzed for nominal and failed signal characteristics. The confidence filter was first applied to the Phase-I vehicle where the windowing and decay parameters were tuned by trial and error. The windowing parameters were subsequently scaled, by the ratio of sensor sampling rates, for application to the Phase-II vehicle and adjusted slightly to improve performance for the faster-sampling system. The moving average window sizes and decay parameters used by the confidence algorithm for each ADS signal are presented in Table 5.1. The tuned filters were validated against unprocessed flight test data in the same way they would be used during a flight test. The flight testing schedule preceded full validation of the C code so no live flight test data could be collected before the end of the project.

Table 5.1: Moving Weighted Average Parameters

(a) Phase-I Flying Fish

| Parameter | $m$ | $\xi$ | $t$ (sec) |
|---|---|---|---|
| $\bar{s}_n$ | 40 | 0.4 | 4.0 |
| $\bar{\nu}(s_n)$ | 50 | 0.0 | 5.0 |
| $\bar{\Lambda}_{var}(s_n)$ | 12 | 0.07 | 1.2 |
| $\bar{\Lambda}_{rate}(s_n)$ | 8 | 0.1 | 0.8 |
| $\bar{\Lambda}_{sat}(s_n)$ | 20 | 0.3 | 2.0 |
| $\angle\vec{w}_I$ | 300 | 0.005 | 30.0 |
| $\|\vec{w}_I\|$ | 300 | 0.005 | 30.0 |

(b) Phase-II Flying Fish

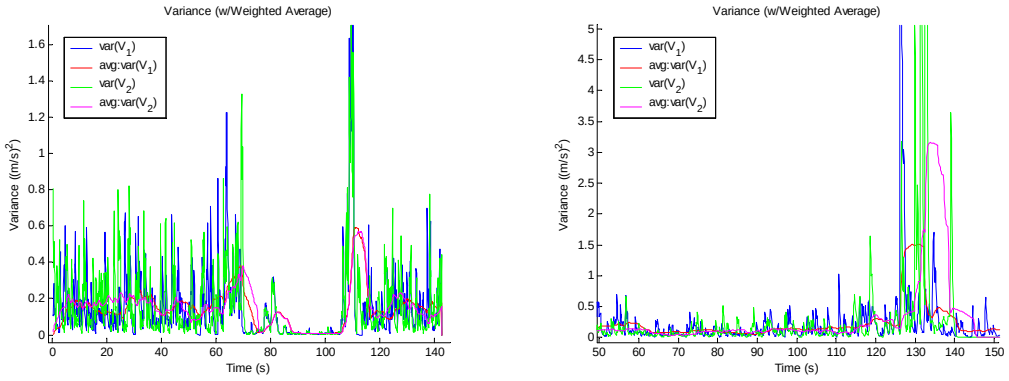
| Parameter | $m$ | $\xi$ | $t$ (sec) |
|---|---|---|---|
| $\bar{s}_n$ | 200 | 0.4 | 4.0 |
| $\bar{\nu}(s_n)$ | 200 | 0.006 | 4.0 |
| $\bar{\Lambda}_{var}(s_n)$ | 60 | 0.07 | 1.2 |
| $\bar{\Lambda}_{rate}(s_n)$ | 40 | 0.1 | 0.8 |
| $\bar{\Lambda}_{sat}(s_n)$ | 100 | 0.3 | 2.0 |
| $\angle\vec{w}_I$ | 1200 | 0.005 | 24.0 |
| $\|\vec{w}_I\|$ | 1200 | 0.005 | 24.0 |

Table 5.1 provides the effective time over which the signals were averaged subject to the 10Hz sampling rate of the Phase-I vehicle and the 50Hz sampling rate of the Phase-II vehicle. The primary signal filter on both vehicles computes the average and variance of every ADS signal over a 4s sliding window with moderate sample decay to bias estimates towards the most recent data. It was experimentally determined that a comparable window size was appropriate for averaging the resulting signals in order to detect changes in variance characteristics. This secondary average does not generally benefit from decay weighting as older values are equally important when attempting to discern if a new measurements is anomalous. Averaging of variance-based error detection votes for the probability of active failures was determined to be most accurate for window lengths of $\sim 1/4$ the length of the primary filter (~1s average of vote results). Averaging votes for the probability of saturation detection yielded better results at half the window length of the primary filter (~2s average of vote results). The decay parameters for these signals were tuned based on the prevalence of false-positives in operational sensor data, per unit time; higher concentrations of

false-positives and longer averaging windows both require increased decay values to allow the filter to pass distributed false-positives without producing a false failure detection. The averaging window for the failsafe wind estimate was selected to be several times longer than the primary filter (~20-30s) to provide a smooth average of variable wind data with a low decay weighting applied to provide a slight bias towards more recent data.

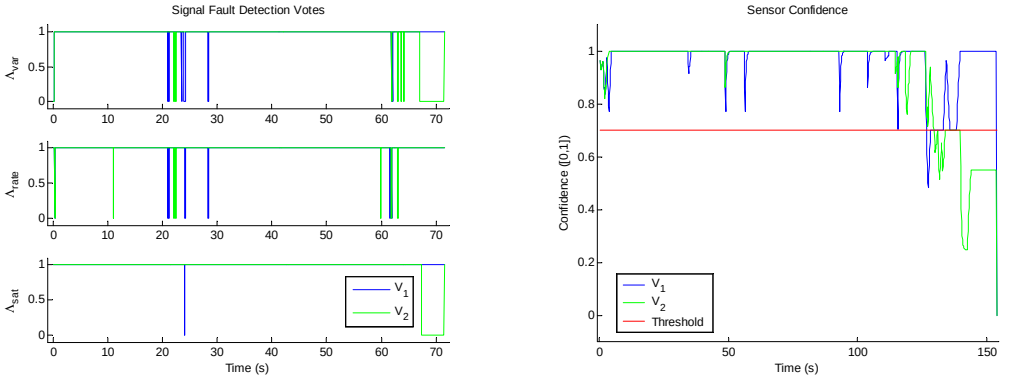
## 5.3 Results

The ADS filtering algorithm was initially tuned and tested with pre-recorded flight data from the Phase I ADS. The available data sets provided a good basis for testing and development as they contained a variety of ADS failures; careful analysis of the data sets yielded the requisite model parameters and filter thresholds. After the tuned filter was successfully tested against the entire range of Phase I ADS failures the windowing parameters were adjusted for the higher measurement rate of the Phase II vehicle. The new filter tuning was subsequently validated through testing and analysis of a small number of pre-recorded Phase II ADS failures. The ongoing Phase II flight testing program did not afford an opportunity to fly the full algorithm but did provide several additional recorded ADS failures. The algorithm was subsequently subject to synthetic testing wherein raw sensor records were delivered on schedule to the ADS filter to validate filter operation in advance of future flight testing opportunities. These results are summarized below.




(a) Functional Sensor Variance (b) Failed Sensor Variance

Figure 5.2: Variance Analysis




(a) Fault Detection Votes (b) Sensor Confidence

Figure 5.3: Failure Detection Results

Phase I flight test data revealed a number of ADS failure types including: single and double in-flight pitot/static failures, intermittent failures, and flight datasets that begin with failed sensors. As discussed above signal characteristics are extracted from the analysis and comparison of functional (Fig. 5.2a) and failed (Fig. 5.2b) ADS sensor data. After tuning the characteristic extraction and signal conditioning parameters the algorithm successfully rendered valid binary votes for the three classes of signal-faults (Fig. 5.3a) on test data sets not used to tune parameters. The results of the binary fault detection decisions are then combined to determine composite sensor

confidence (Fig. 5.3b). Further initial test of the wind-model air-data estimates shows good correlation with functional ADS sensors (Fig. 5.4). The complete algorithm, combining the wind model, signal confidence, voting, and data fusion to produce a single high-confidence airspeed has been validated for a wide range of cases including single (Fig. 5.5) and double (Fig. 5.6) airspeed sensor failure cases.

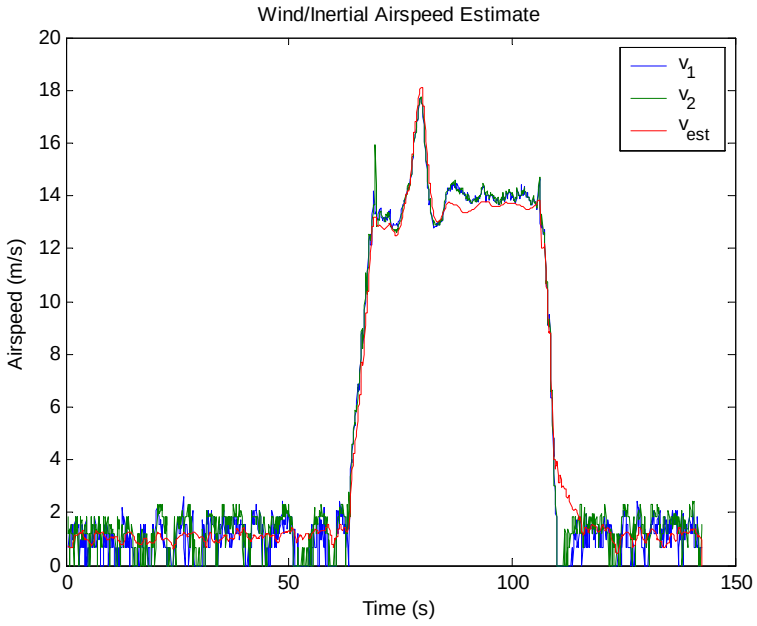



Figure 5.4: Nominal Operation Case with Wind Model Estimate

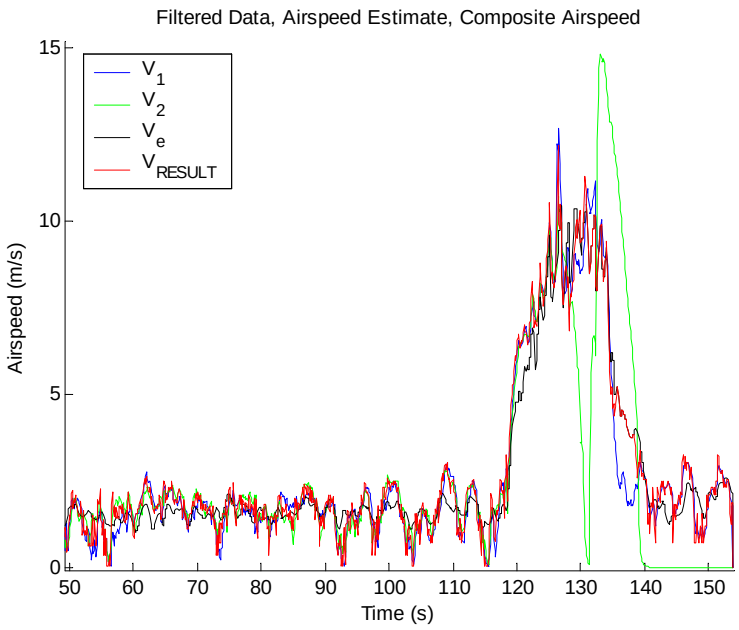



Figure 5.5: Single Probe Failure Case

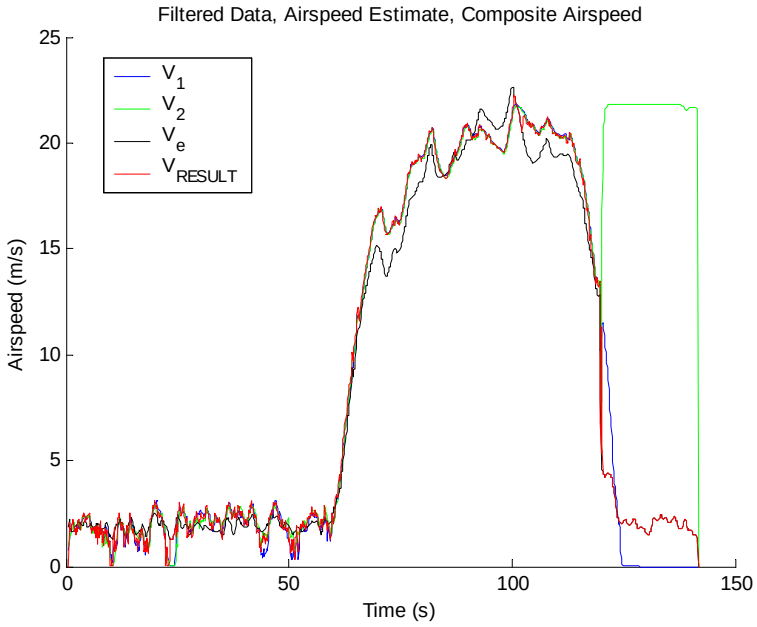



Figure 5.6: Double Probe Failure Case

The failure mitigation system performed well during preliminary testing with Phase II vehicle data. The algorithm accurately handled errors both during high-speed taxi tests (Fig. 5.7) and during simple flight tests (Fig. 5.8). During the high-speed taxi

test (Fig. 5.7) the algorithm correctly eliminates an erroneous sensor excursion during the approach to hydroplaning speeds (75s) and also correctly rejects a high-pressure blockage that biases one of the airspeeds high after 85s. Flight test results (Fig. 5.8) show the algorithm correctly rejecting low-speed saturation in (0-100s) and, similar to the taxi test, rejecting a high-pressure biased signal during descent/landing (125s). These tests also demonstrate some interesting dynamics and issues associated with the propeller anemometer. At low airspeed the counting limit of the digital timer and the rotating friction of the propeller produce saturation effects. Conversely at high speeds the sensor response is increasingly non-linear as the small 3cm propeller is driven to rotation speeds in excess of 11000rpm. Furthermore the installed prototype anemometer was subject to greater wear than was originally anticipated. Continuous high-speed rotation during flight began to erode the propeller's waterproof bushings resulting in misalignment of the propeller and hall-effect sensors. The Flying Fish team found that while the anemometer could be realigned on shore each morning, giving good results for early flight tests (Fig. 5.7), the progressive wear and stresses of flight testing resulted in non-negligible signal degradation over the course of a day (Fig. 5.8). This design issue can be addressed with an update to the propeller bearing of the miniature anemometer prototype.

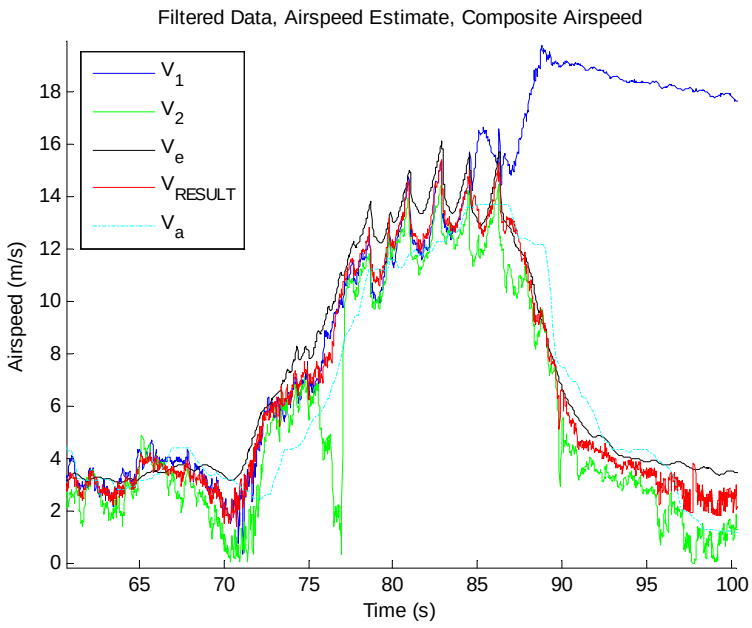



Figure 5.7: High Speed Taxi, Double Failure Case

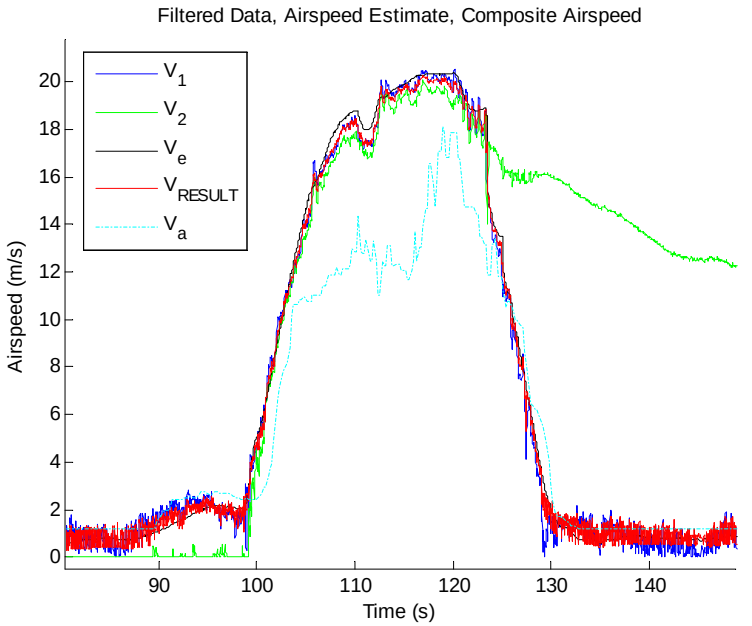



Figure 5.8: Flight Test, Single Failure Case

Continued testing allowed the algorithm to demonstrate its ability to handle increasingly difficult failures. One of the first major trials for the failure mitigation system was a flight test that began with a blocked pitot-static probe. Consensus voting was

able to distinguish the correct signals after sensor confidence was established (Fig. 5.9) and the signal was successfully reintegrated to the confidence voting procedure when the probes blockage cleared (85s). More impressive however are the results obtained during flight testing in a rainstorm (Fig. 5.10). The ADS algorithm successfully rejects several erroneous ADS sensor excursions and negotiates a complete sensor failure and two subsequent probe recoveries.

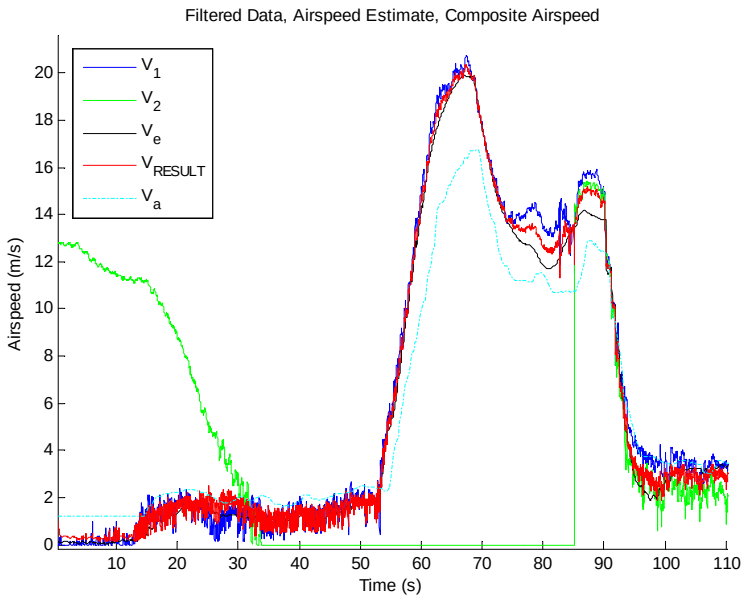



Figure 5.9: Initially Failed Probe Case

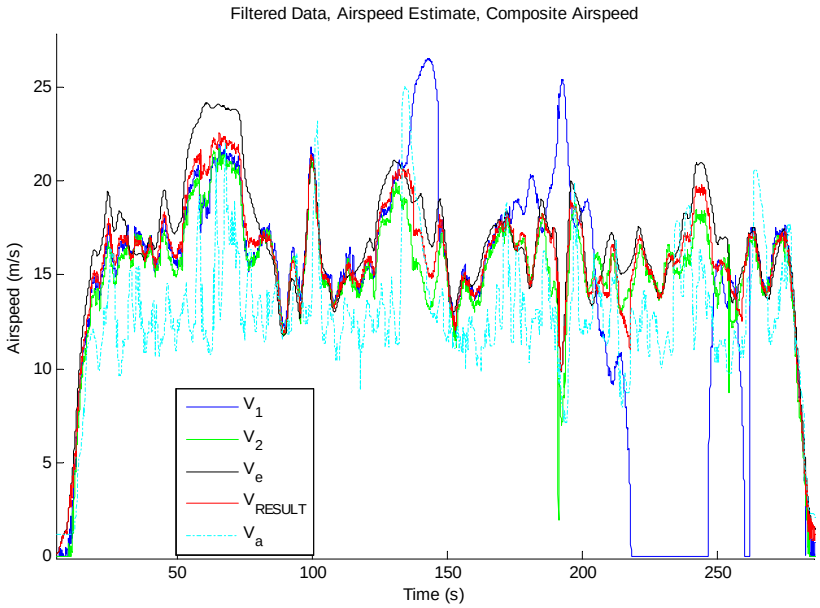



Figure 5.10: Flight Through Rain

# CHAPTER VI

# Energy-Aware Flight Management

The ultimate goal of the Flying Fish project is long-endurance unattended autonomous mission execution on the open ocean. To survive without recovery and potentially without a continuous ground station communication link, the onboard flight management system must be able to: (1) plan the pace and order of goal satisfaction subject to dynamic system and environmental constraints, (2) possess the navigation, guidance, and control faculties to execute a given plan, and (3) monitor for, diagnose, and recover from vehicle subsystem failures. In Chapter 2 we detailed the vehicle avionics including the requisite sensing, navigation, and control systems, then in Chapters 3-4 we developed the dynamics of flight, system models, and control laws necessary to execute mission plans. Chapter 5 introduced a system for fault mitigation of the most failure-prone Flying Fish subsystem for future integration to the flight planner. The primary remaining requirement to support unattended persistent autonomous operation is the development of sufficient sensing, modeling, planning, and decision-making utilities to safely and efficiently order and manage a series of flight operations subject to variable environmental conditions, changing mission parameters, and the impact of known and pop-up constraints/conflicts. This chapter focuses on the optimization of task-level plans given dynamics, energy, and obstacle constraints; task-level management of subsystem failures is left for future work.

A UAS mission planner must create flight trajectories that remain within the vehicle's operational envelope, that are potentially subject to constraints and one or more optimality conditions, and that satisfy the specified mission requirements, nominally reaching all mission goal waypoints. The planner must be capable of devising obstacle avoidance maneuvers given a priori obstacle data (constrained planning) and observational data (sense and avoid) with the potential to re-plan if pop-up obstacles or unanticipated environmental conditions invalidate the existing plan. These attributes are generally sufficient to define mission planning goals for UAS, however in the context of the solar-regenerative seaplane-UAS (S-UAS), flexible long-term planning depends critically upon system energy management, sea state, and long-term and short-term solar and wind conditions. The ideal solar-regenerative S-UAS planner would be able to forecast energy collection and expenditure to plan a sequence of flights that achieve primary, auxiliary, and opportunistic mission goals without exceeding the total energy budget. A well-informed planner for such a system should also account for long-term/overnight survivability constraints and possess the ability to develop reasonable estimates of, and responses to, the effects of inclement weather on energy and position. Critically, given the remote nature of the deployment environment many/all of these capabilities must also be executed with the limited resources of an embedded computer that is simultaneously tasked with the management of all vehicle subsystems.

For planning, the greatest assets of the seaplane-UAS—energy harvesting, flexible flight/drift profiles, extensive deployment footprint, and long-term mission timescales—also stand as the greatest challenges to effective management and planning. The inclusion of broad operational flexibility and environmental effects increases the search dimensionality but the omission of these elements would, at best, produce ineffective/inefficient plans. Flying Fish planning problems must be solved on multiple temporal scales, ranging from the duration of a single flight to day/night and

seasonal scales; on multiple physical scales, from the single-flight range of the vehicle to the scale of the a lake, sea, or ocean; and over multiple domains, including energy and the aforementioned time and space. Further, as discussed in this chapter, the multidimensional optimization problem under consideration possess many traits, arising from the very phenomena that allow sustainable deployment, which render most solution methods ineffective or inapplicable.

In this chapter energy and environmental models for the planner are developed to supplement vehicle performance models presented in earlier chapters.The mission planner is subsequently described in the context of AI search methods applied to an extended Travelling Salesman Problem (TSP). The planner is then used to explore the space of possible Flying Fish mission plans subject to energy and boundary constraints and a multi-objective cost function over which solutions are optimized. Within this framework the unique characteristics and specific challenges of the solar-regenerative S-UAS optimization problem are discussed and examined in the context of multi-day, overnight, and mid-day mission scenarios. Applicable assumptions and simplifications are presented to yield a tractable problem space despite the presence of cycles arising from energy dynamics over cyclic flight-drift operations. Example scenarios or each mission type are presented and analyzed.

## 6.1 Energy and Environment Models

Energy-aware flight planning requires models of energy use and harvesting capabilities of the vehicle as well as environmental conditions focused on the critical phenomena effecting the solar-regenerative UAS: wind, water-motion, and solar insolation. Models to estimate energy collection, storage, and expenditure are presented below.

### 6.1.1 Environment Models

Modeling energy collection requires accurate determination of the locally incident solar power density ($\mathcal{P}_{\text{spec}}(t, \cdot)$) and the array-relative incidence angle of the arriving solar rays. The incidence of the solar rays will be resolved in the inertial frame from the solar azimuth ($a_{sun}$) and elevation ($e_{sun}$) and rotated into vehicle coordinates using vehicle models presented later. The solar position and irradiance models used in this research are derived from calculation utilities published by the National Renewable Energy Laboratory (NREL).[61, 62, 63]

For brevity and clarity we introduce a dot-notation (e.g. $\mathcal{P}_{\text{spec}}(t, \cdot)$) which represents the full set of variables required to characterize the configuration of a solar array relative to the Sun. These variables include: the Earth-relative position and attitude of the solar array, the Earth's rotation angles and orbital relationship to the Sun, and the characteristics of the atmosphere between the array and Sun. In practice a number of realizations of these variables may be used. The NREL model utilizes a (North-referenced) azimuth (or aspect) and a tilt (referenced to the Earth's rotational axis) to represent solar panel attitude; solar panel position is characterized using latitude, longitude, and mean-sea-level (MSL) altitude. The NREL calculators base their atmospheric parameters on local dry-bulb temperature and surface pressure. Finally, the NREL model extrapolates the Earth-Sun relationship from the Gregorian calendar date and the current Coordinated Universal Time (UTC). Other representations might, for example, use an Earth-Centered Earth-Fixed (ECEF) coordinate system and Euler angles to characterize the solar panel configuration and orbital parameters and time of day to characterize the Earth-Sun configuration. To make clear the explicit dependence on time, the current UTC time of day is included separately from the dot-notation.

The NREL solar position calculator uses a Fourier Series model to resolve a vec-

tor, in an ECEF coordinate system, from the Earth to the Sun.[64] Subsequently the system calculates the declination, right-ascension, and local mean sidereal time from local date, time, and vehicle longitude to characterize the local hour-angle of the sun relative to solar noon.[61] The latitude of the vehicle can then be applied to determine solar zenith, unrefracted solar elevation, and unrefracted azimuth[65] which can be corrected for the expected atmospheric refraction based on the sun's proximity to the horizon, local temperature, and local pressure.[66] The resulting refraction-corrected azimuth ($a_{sun}$) and elevation ($e_{sun}$) angles can be applied to the extrapolated atmosphere-corrected solar power density ($\mathcal{P}_{\text{spec}}(t,\cdot)$) for the given date and geographic location to uniquely identify the solar insolation and incidence vector. The inertial-frame solar incidence vector is given, for solar azimuth and elevation angles, by:

$$\vec{s}_I(t,\cdot) = \begin{bmatrix} \cos(a_{sun}(t,\cdot))\cos(e_{sun}(t,\cdot)) \\ \sin(a_{sun}(t,\cdot))\cos(e_{sun}(t,\cdot)) \\ \sin(e_{sun}(t,\cdot)) \end{bmatrix} \tag{6.1}$$

Given the solar insolation and incidence model we can apply the vehicle solar-power model in Eq. 6.2 to determine the available power (integrand) and energy (integral) harvested from the sun. An example of simulated daily solar energy is presented in Fig. 6.1. This figure shows solar incidence angle (elevation only for a horizontal solar array), total incident solar power, and efficiency-scaled harvestable solar power collected by the Flying-Fish-scale 1.34m$^2$ solar panel.

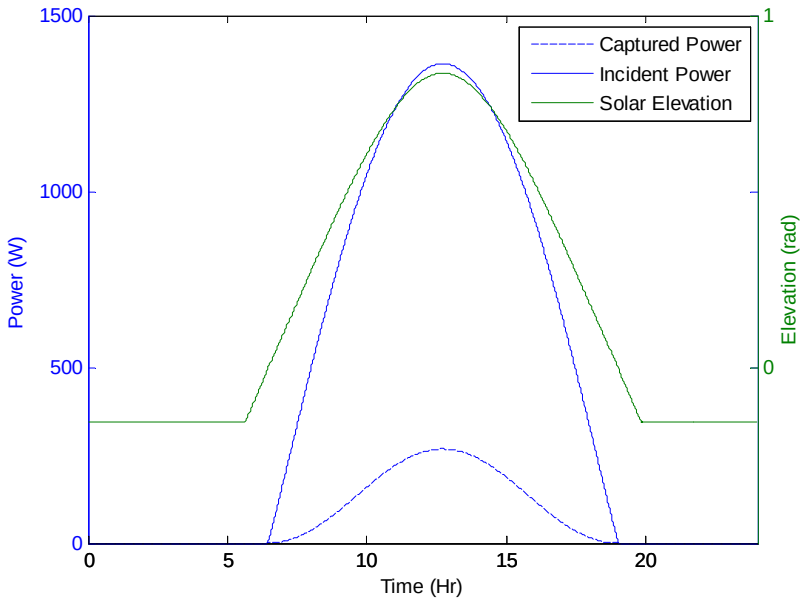



Figure 6.1: Daily Solar Energy Model

The greatest sources of uncertainty in the solar model are global and local atmospheric conditions, with cloud cover providing the most significant potential attenuation of available solar energy. Atmospheric weather processes including wind, cloud formation, and precipitation cannot be reduced to simple deterministic models and are typically forecast using large-scale grid-based numeric simulations leveraging distributed networks of sensors and statistical/probabilistic data [67, 68] to represent nominal/seasonal weather [69] or severe weather conditions [70]. The estimation of weather patterns and cloud cover is beyond the scope of this work. However in practice a system such as Flying Fish can measure and respond to the instantaneous solar conditions provided to the flight computer via the maximum power point tracker subsystem. If it is assumed that there is no array degradation, the solar position and irradiance models can be used to estimate the expected solar input for comparison to measured input. The resulting on-line estimate of average atmospheric attenuation could subsequently be applied within the planner of the deployed system to dynamically account for cloud cover. Other, more robust alternatives, might include

utilizing a gimballed or sky-looking camera or miniature pyrometer to make more precise cloud-cover assessments and extrapolate these effects within the planning architecture. Long term weather uncertainty would be best addressed by the addition of a satellite-link for weather data, especially in well-monitored coastal regions. Our planner currently incorporates ideal solar insolation conditions, leaving estimation of cloud cover, etc. as future work.

Long-term wind estimation is a challenge for a mobile platform in variable atmospheric conditions. Wind estimates and forecasts may ultimately be available via satellite link but were not available to the system used over the course of this research. The on-board planner currently uses a simple data-driven steady wind estimation algorithm based on a weighted running average of ADS data, with newer data weighted more heavily (as detailed in Chapter 5). We therefore assume it is reasonable to estimate the immediate future winds (an hour or more) from the characteristics of recent wind data. Under this assumption the wind is modeled to have the same average direction and prevailing speed as recently sampled wind data. Rather than using a statistical model, which adds significant complexity but no guarantee of matching real world wind performance, it was decided that the planner would be best served by the application of a constant reasonable, but pessimistic (high), estimate of wind and planning for a worst-expected-case scenario. Nevertheless, the planning infrastructure was developed such that the wind input is not required to be constant; a statistical wind model or field-measured data can be written directly to the (protected) global wind variables within the C code and will be appropriately integrated through the estimation process. Planning results presented in this chapter will utilize constant winds for each simulation.

### 6.1.2 Vehicle Energy Dynamics

Vehicle energy models must be incorporated within the autonomous flight planner to reasonably estimate the collection, storage, and expenditure of system energy during flight and drift operations. Energy collection is a function of many interdependent variables but the solar conditions, solar-collection efficiency, and battery-charge status dominate system response. Power collection is determined from the angle at which solar rays are incident to the array ($\angle\vec{s}_A$), the spectral power density ($\mathcal{P}_{\text{spec}}$) of this incident light, the solar array area ($\mathcal{A}_{\text{sol}}$), and solar collection efficiency ($\eta_{\text{sol}}$). Energy collected by the solar array ($\mathcal{E}_{\text{sol}}$) can be written as the time integral of the instantaneous power available from the array ($\mathcal{P}_{\text{sol}}$):

$$\mathcal{E}_{\text{sol}}(t,\cdot) = \int_{t_0}^{t_f} \eta_{\text{sol}}\mathcal{A}_{\text{sol}}\mathcal{P}_{\text{spec}}(t,\cdot)\,\cos(\angle\vec{s}_A(t,\cdot))dt \tag{6.2}$$

In Eq. 6.2 the incidence of the solar light is denoted as the angle of the solar-incidence vector in the coordinate frame of the solar array ($\vec{s}_A$). To compute the array-frame incidence vector ($\vec{s}_A$) the inertial-frame incidence vector ($\vec{s}_I$) is computed using solar motion models discussed below and rotated by the vehicle's Euler angles plus the positive pitch-axis offset angle of the solar array in the vehicle frame, $\theta_{array}$ using elementary rotations $R_x(\phi)$, $R_y(\theta)$, and $R_z(\psi)$ (Eqs. 5.21-5.23):

$$\vec{s}_A(t,\cdot) = R_x(\phi)R_y(\theta+\theta_{array})R_z(\psi)\vec{s}_I(t,\cdot) \tag{6.3}$$

The vector inner product may subsequently be applied to determine the angle of solar incidence to the array; the inner product provides a relationship between the length of two vectors $\{\vec{v}_1,\vec{v}_2\}$ and the angle between them $\varphi$:

$$\vec{v}_1\cdot\vec{v}_2 = \|\vec{v}_1\|\,\|\vec{v}_2\|\cos(\varphi) \tag{6.4}$$

From this relationship the angle of solar incidence to the array ($\angle\vec{s}_A$) can be determined from the solar incidence vector in the array frame ($\vec{s}_A$) and the z-axis of the array-fixed frame ($\hat{k} = [0, 0, 1]$):

$$\angle\vec{s}_A = \arccos\left(\frac{\vec{s}_A \cdot \hat{k}}{\|\vec{s}_A\| \left\|\hat{k}\right\|}\right) = \arccos\left(\frac{\vec{s}_{A,z}}{\|\vec{s}_A\|}\right) \tag{6.5}$$

Only the cosine of the incidence is required to compute solar power and energy (Eq. 6.2) and Eq.6.5 immediately simplifies to an expression for the cosine of the incidence angle as a function of the magnitude ($\|\vec{s}_A\|$) and z-component ($\vec{s}_{A,z}$) of the incidence vector:

$$\cos(\angle\vec{s}_A) = \frac{\vec{s}_{A,z}}{\|\vec{s}_A\|} \tag{6.6}$$

A comparable derivation for an un-tilted panel can be found in related literature.[18] The array area, array offset angles, and solar collection efficiency are know a priori and the vehicle attitude and position are assumed to be known from sensor measurements. The remaining variables, the inertial-frame incidence vector and spectral power density of the sun, are external environmental processes and are accordingly discussed in the environmental modeling section below.

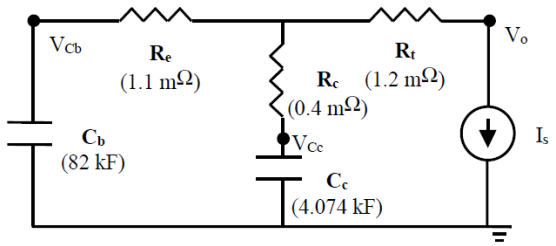



Figure 6.2: NREL-Saft Battery Model[71]

Energy input to the system is further subject to the system's ability to accept power; a fully charged battery, for example, cannot accept additional energy regardless of the quantity of available power. To implement the battery model we initially em-

ployed polynomial curve-fits to laboratory-measured charge/discharge curves from actual Flying Fish flight batteries.[72] While this approach was accurate for the measured data for moderate loading conditions the resulting battery model was unable to accurately represent the dynamic response of lithium batteries during the heavy loading such as those seen during takeoff. A number of alternative lithium battery models have been developed in recent years[73, 74, 75] with much of the research driven by growing interests in electric-powered transportation.[76, 71] The current Flying Fish battery model is adapted from a model presented in an NREL lithium-battery technical report.[71] The model was attributed to lithium battery manufacturer Saft, but appears to be a variation on previously known models.[73, 74] The NREL-Saft model represents the battery as a parallel resister-capacitor network with input/output impedance (Fig. 6.2). The linear model of this system is given by:

$$\begin{bmatrix} \dot{V}_{Cb} \\ \dot{V}_{Cc} \end{bmatrix} = \begin{bmatrix} \frac{-1}{C_b(R_e+R_c)} & \frac{1}{C_b(R_e+R_c)} \\ \frac{1}{C_c(R_e+R_c)} & \frac{-1}{C_c(R_e+R_c)} \end{bmatrix} \begin{bmatrix} V_{Cb} \\ V_{Cc} \end{bmatrix} + \begin{bmatrix} \frac{-R_c}{C_b(R_e+R_c)} \\ \frac{-1}{C_C} + \frac{R_c}{C_c(R_e+R_c)} \end{bmatrix} [I_s] \tag{6.7}$$

$$\begin{bmatrix} V_o \end{bmatrix} = \begin{bmatrix} \frac{R_C}{(R_e+R_c)} & \frac{R_e}{(R_e+R_c)} \end{bmatrix} \begin{bmatrix} V_{Cb} \\ V_{Cc} \end{bmatrix} - \begin{bmatrix} R_t + \frac{R_c R_e}{(R_e+R_c)} \end{bmatrix} [I_s] \tag{6.8}$$

This model simulates battery voltage potential over a large charged capacitor. The charge/discharge response is governed by the output impedance and a smaller capacitor/resistor combination reproduces the nonlinear depletion region of a lithium battery under load. Effective battery capacity is represented by the charge stored by the capacitor at a given voltage potential; the energy stored by a capacitor, charged to a voltage potential $V$ with capacitance $C$ can be expressed as:

$$\mathcal{E}_{\text{charge}} = \frac{1}{2} C \cdot V^2 \tag{6.9}$$

Subject to the usable voltage range of a lithium battery $(V_{min}, V_{max})$ the maximum available energy and the energy remaining at an intermediate voltage $V(t)$ can be written:

$$\mathcal{E}_{\text{batt,max}} = \frac{1}{2}C \cdot V_{max}^2 - \frac{1}{2}C \cdot V_{min}^2 \tag{6.10}$$

$$\mathcal{E}_{\text{batt}}(t) = \frac{1}{2}C \cdot V_{max}^2 - \frac{1}{2}C \cdot V(t)^2 \tag{6.11}$$

Each Flying Fish battery pack is composed of 5x lithium-polymer cells in series, each with a nominal voltage of 3.7V and an operating range of $(V_{min}, V_{max}) = (3.1V, 4.2V)$, to provide an energy capacity of approximately 72kJ. The parameters of the battery model (Eq. 6.8) have been tuned to deliver the voltage and capacity of a single cell by the appropriate selection of impedance and capacitance ($R_e = 1.1\text{m}\Omega, R_c = 0.4\text{m}\Omega, R_t = 2.2\text{m}\Omega, C_b = 18.45\text{kF}, C_c = 4.0\text{kF}$). This model also changes the sign of the current input $I_s$ block of Eq. (6.8) which is a suspected error in the original NREL report as the published model gives increasing voltage under heavy loads. Series battery voltage is simulated by scaling the linear model to the cell-count of each Flying Fish battery pack. Battery bank capacity is simulated by dividing system loads by the number of batteries in each bank. While the division of loads omits the effect of variation between battery packs these variations are reasonably represented in the average so long as we assume cells do not fail or electrically short. The battery dynamics make the equal division of loads amongst batteries in a single bank a reasonable assumption for the application of this model; any load inequity on battery packs in the bank would yield a proportional inequity in pack potential and result in increased load to the other packs in the bank achieving equilibrium. Flying Fish has two large main battery banks charged by the solar array (and loaded by the avionics and primary propulsion) and a smaller central bank that is charged by the primary banks (and loaded by the avionics and auxiliary propulsion system). A simulation of the Flying Fish Phase II power system with the NREL-Saft battery model over two

sequential 1.5-minute flight cycles with solar charging is shown in Fig. 6.3. Note that in this simulation drift time is quite short to illustrate recharge; in practice a significantly extended drift time would allow all batteries to more fully recharge between flights.

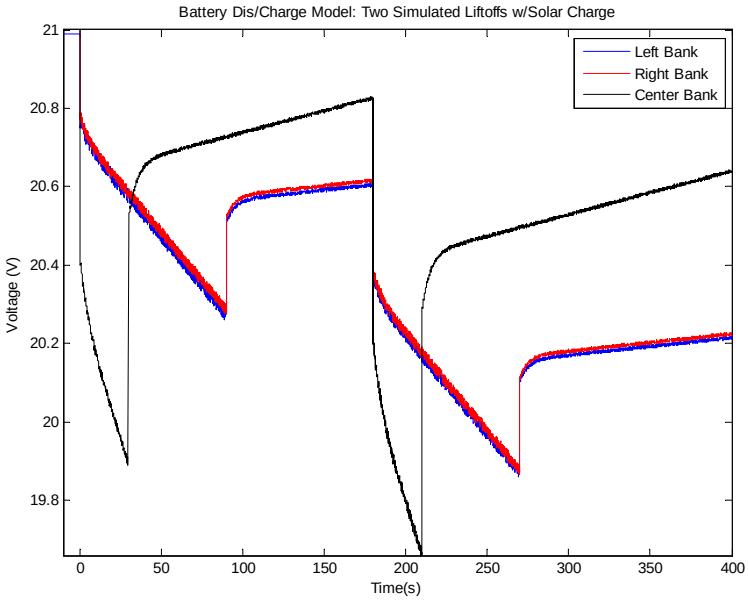



Figure 6.3: Battery Simulation: Two Flights

Energy expenditures are characterized by two distinct loading processes: maintenance and flight. All avionics loads are assumed, with the exception of surface-deployed payloads, to be active during flight but the system is subject to a number of loads that are "sheddable" when the vehicles is on the water. Assuming that the vehicle must maintain situational awareness on the water the fixed "hotel" loads include: avionics computer (ACPU), inertial navigation system (INS), wireless communications (can be idled, but must monitor command channels), and the miscellaneous regulator/interface overhead. The routinely sheddable loads on the water include the control actuation mechanisms (regulators, servos, & controllers) and the ultrasonic altimeter. Auxiliary payloads are assumed to be "anytime" sheddable for the purposes of survival, although none were modeled in this work. Table 6.1 provides an estimate

of fixed and sheddable loads onboard a seaplane UAS such as the Flying Fish Phase II platform.

Table 6.1: Flying Fish Fixed and Sheddable Loads

| | $P_{on}$ (W) | $P_{idle}$ (W) | $P_{sleep}$ (W) |
|---|---|---|---|
| ACPU | 1.0 | - | - |
| Modem | 4.83 | 0.7 | 0.133 |
| INS | 1.2 | - | - |
| Misc | 0.9 | - | - |

(a) Fixed Hotel Loads

| | $P_{on}$ (W) | $P_{idle}$ (W) | $P_{off}$ (W) |
|---|---|---|---|
| Ultrasonic | 1.47 | - | 0.0 |
| Servo Controller | 0.5 | - | 0.0 |
| Control Servos (ea) | 3.15 | 0.264 | 0.0 |
| Servo Receiver | 0.075 | - | 0.0 |

(b) Sheddable Loads

Note that for simplicity this analysis ignores the complexity that load-shedding imparts to electrical systems and general vehicle design. Whenever a load is made to be sheddable additional power switching must be added which increases the complexity of both the wiring and control systems and creates additional potential failure points. There are also safety issues to be considered if a critical load might be accidentally "shed" during flight. For example, while the Flying Fish has the infrastructure to switch on/off the power applied to servo mechanisms the servos have remained continuously powered on throughout development for the sake of safety and reliability in the development environment. Operating in this fashion, the best-case for field-test load-shedding is servo idling and ultrasonic-altimeter deactivation. The actual

maintenance energy cost, given these conditions, has therefore been extrapolated from battery usage statistics over long-duration avionics tests; specifically, the average avionics power draw is about 6W with idled but powered servos. Applying this fixed power requirement over the duration of the mission allows the system to compute maintenance energy expenditures, which are especially critical for forecasting overnight survival.

Energy expenditure due to flight (propulsive) loads are modeled from flight and laboratory-derived calibration curves (Fig. 6.4). Second-order polynomial fits ($0.75 < R^2 < 0.8$) provide a basic relationship between the throttle settings and power requirements for both the main and boost propulsion systems. The curves are applied to the throttle vector over a flight profile to determine system loads and then, by applying the battery model, to compute cumulative energy expenditures. Throttle settings for each stage of flight, takeoff, climb, cruise, and descent, have been extracted from flight data and are used to estimate the power required for each segment. The first generation model assumed that the straight-line cruise segment of each Dubins path trajectory serves as a reasonable average flight direction for that segment.[72] A solution of the wind-heading velocity triangle was used to determine the slipping-flight speed along the flight path which divides the entire turn-fly-turn segment length to produce segment flight time. In the current model we apply the wind-aware bank-to-turn unicycle model, presented in Ch. 3, throughout the flight path, numerically integrating the implicit result of the velocity triangle over the entire path. Once an estimate of flight time in each segment is determined the required energy, per segment, can be determined and the total flight expenditure can be computed from the sum of the energies required for each segment. The difference in Dubins trajectory length between turning *at* a waypoint and turning *before* a waypoint, to intercept it on an arc, is assumed to be negligible. We also assume that it is reasonable to meet the Dubins path requirement for instantaneous turn rate change between segments

by allowing instantaneous changes in bank-angle. This is not an entirely unreasonable assumption as Flying Fish has high roll authority and actual turn performance is expected to be able to quickly catch up to the model trajectory despite initial inaccuracies between the flown and project path. If planning and simulation fidelity are subsequently judged insufficient for long-term planning it would be possible to characterize and implement more accurate roll dynamics in future work.

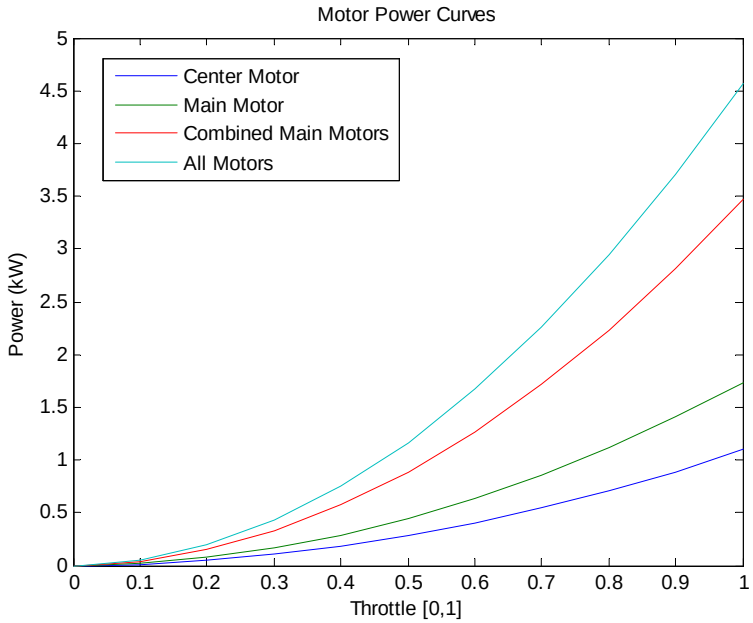



Figure 6.4: Empirically Determined Curve-Fit Motor Power

## 6.2 Mission Planner

The mission planner constructs sequences of flight and drift segments that achieve specified observation waypoint goals in the air and on the ground, and that satisfy energy and performance constraints given current and expected environmental conditions. Below, we first define the domain and structure of the planning system and then define mission goals, constraints, and planning parameters. We define a compact plan representation applicable to both algorithm and software development. We subsequently describe the types and implementation of constraints and goals. We

formally define a mission and discuss the implementation of the collected vehicle and environmental models into a cohesive planning and simulation utility. This framework is subsequently used to mathematically define utility, benefit, value, and cost as they pertain to the planning problem. An example plan-environment is referenced and elaborated over the course of the section and will serve as a standard test for subsequent planning strategies (Fig. 6.5). For simplicity the planning environment uses a relative Cartesian coordinate system with its origin at the latitude and longitude of the center of the desired operating region. The axes are oriented such that the positive y-axis is directed North, the positive x-axis is directed East, and the positive z-axis is directed towards the center of the Earth. Distance in the planning frame is measured in meters. The planning environments (goals and constraints) are generated randomly; the same randomly-generated environment is used for all of the presented result for consistency.

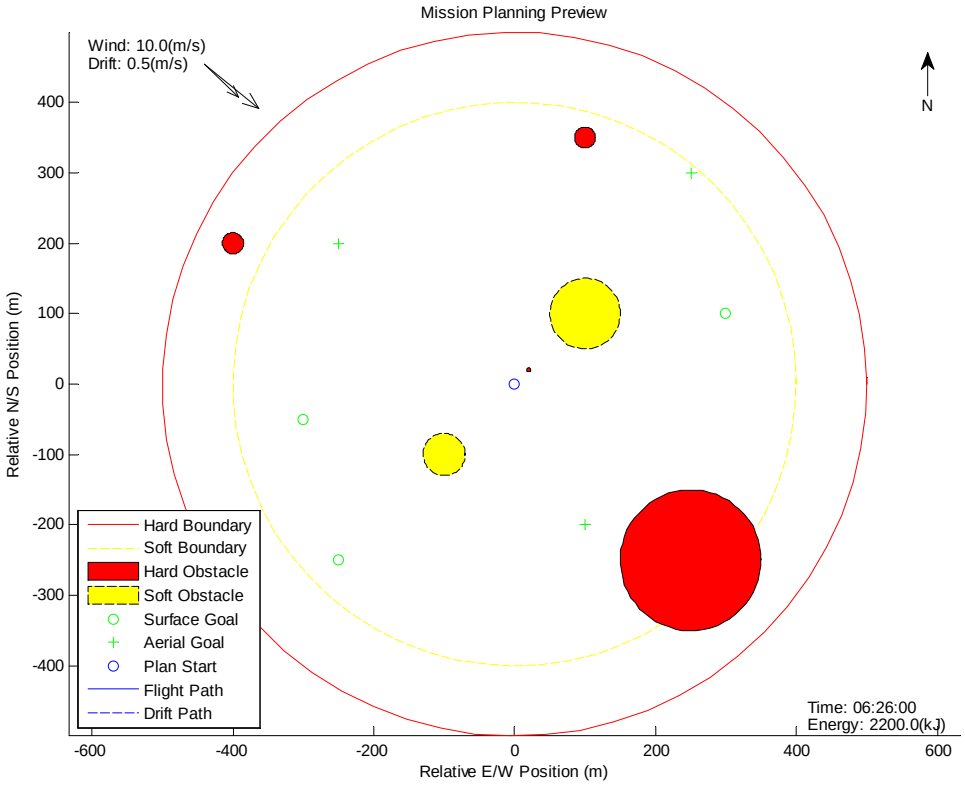



Figure 6.5: Planning Environment

Within the Flying Fish planner every entry, be it a goal or constraint, takes the form of an augmented waypoint object that is endowed with a uniform set of attributes: time, position, velocity, attitude, spatial dimensions, activation value, execution priority, description, and type. Not all plan objects use all available attributes but almost any type of plan object can be represented without additional attributes, making bookkeeping and development relatively straightforward. Every planner object takes the form:

$$\begin{aligned} \mathbb{P}_i &= \{\text{type}, \text{time}, \text{position}, \text{velocity}, \text{attitude}, \\ &\quad \text{dimensions}, \text{priority}, \text{value}, \text{description}\} \\ &= \{\text{TYP}, t, (x, y, z), (\dot{x}, \dot{y}, \dot{z}), (\phi, \theta, \psi), (r, h), \bar{p}, \mathrm{v}_0, \text{DES}\} \end{aligned} \tag{6.12}$$

Four types of environmental constraints are defined for this planning problem: *hard obstacles* ($\mathcal{C}_{OH}$), *soft obstacles* ($\mathcal{C}_{OS}$), *hard operational boundaries* ($\mathcal{C}_{BH}$), and *soft operational boundaries* ($\mathcal{C}_{BS}$). Boundary constraints are define as containment regions in which the vehicle must remain and obstacles are defined as regions into which the vehicle must not enter. Hard and soft constraints are differentiated, respectively, as either inviolable barriers or advisory barriers, for which infringement is permitted but costly. This specification is included for completeness but, for the purpose of this research, all boundary constraints are considered inviolable. Constraint attributes allow velocities to be defined, yielding moving constraints (mobile water/aircraft). Mobile constraints are considered in every planning problem presented in this research. Constraint attributes also allow for the definition of constraint geometry; vertical cylinders are used for all of the planning problems presented in this dissertaion. The altitude associated with a constraint is important as aircraft can utilize altitude as well as lateral separation for deconfliction. For example, ocean surface constraints (currently, those with altitude constraints up to 10m) may be overflown if an additional 10m of altitude separation is planned and aerial constraints (aircraft or airspace restrictions) are not considered for a drifting vehicle. An example constraint environment is presented in Table:6.2 and is presented graphically in Fig. 6.5.

Table 6.2: Planner Environment Description

| | Type | Position (m) | Vel. (m/s) | Size (m) | Description |
|---|---|---|---|---|---|
| $\mathcal{C}_{BH,1}$ | Hard Bnd. | (0,0,0) | (0,0,0) | (500,100) | "Shore" |
| $\mathcal{C}_{BS,1}$ | Soft Bnd. | (0,0,0) | (0,0,0) | (400,500) | "Airspace" |
| $\mathcal{C}_{OH,1}$ | Hard Obs. | (-100,200,0) | (-10,20,0) | (15,15) | "Boat" |
| $\mathcal{C}_{OH,2}$ | Hard Obs. | (20,20,0) | (0,0,0) | (3,4) | "Buoy" |
| $\mathcal{C}_{OH,3}$ | Hard Obs. | (250,-300,0) | (0,0,0) | (150,5) | "Reef" |
| $\mathcal{C}_{OS,1}$ | Soft Obs. | (100,-200,0) | (1,-1,0) | (50,0) | "Oil Slick" |
| $\mathcal{C}_{OS,2}$ | Soft Obs. | (-100,-100,0) | (0,-1,0) | (30,0) | "Algae Bloom" |

In the constrained environment, we define two types of mission goals: *surface* ($\omega_S$), and *aerial* ($\omega_A$). Goals define either aerial or surface waypoints and are considered satisfied when the vehicle passes within an acceptable proximity to the specified four-dimensional waypoint position and time. Plan object attributes can be used to specify approach tolerances, through the dimension attributes, and moving goals, through the velocity attribute. This detail is included for completeness, as a deployed planner may reasonably need to direct a trajectory to a non-stationary target but, for this research we hold all goals stationary for consistency. Beyond these attributes, goals are also endowed with auxiliary value and priority. Generally, total value ($\mathrm{v}_i$) of the $i^{\text{th}}$ goal is computed as the sum of the initial value of that goal ($\mathrm{v}_{0,i}$) and a weighted time-since-last-visit incentive to encourage re-exploration if appropriate: ($\mathrm{v}_{t,i}$):

$$\mathrm{v}_i = \mathrm{v}_{0,i} + \mathrm{v}_{t,i} \cdot \Delta t \tag{6.13}$$

Initial goal values ($\mathrm{v}_{0,i}$) are set to zero at the moment of goal satisfaction with future value derived only from the time-since-last-visit incentive ($\mathrm{v}_{t,i}$). Goal values are not required for planning, and may be set to zero or unity, for example, in the case

that all goals must be visited exactly once. User-defined priority is a secondary valuation criterion that can be applied to either goals or constraints. In the event that differentiation between soft constraints is required, or in the case that goal values are insufficient to discriminate between goals, the priority can be used as an auxiliary ranking system. For example, given two soft constraints of which one must be violated for vehicle survival the secondary priority value may be used to deconflict the selection. Similarly, if initial goal incentive goal valuations are insufficient to encode preference for the completion of a critical goal, then a goal priority may be applied. Priorities are assigned values between 0 and 1, inclusive, with higher values indicating higher priority. An example set of goals are given in Table 6.3 and are also graphically represented in Fig. 6.5.

Table 6.3: Planner Mission Description

| | Type | Position (m) | Priority | Value | Description |
|---|---|---|---|---|---|
| $\omega_{S,1}$ | Surf. | (-250,-250,0) | 8 | 20 | "Water Sample" |
| $\omega_{S,2}$ | Sur. | (-300,-50,0) | 7 | 10 | "Water Sample" |
| $\omega_{S,3}$ | Surf. | (300,100,0) | 8 | 20 | "ROV Comm" |
| $\omega_{A,1}$ | Air | (250,300,20) | 8 | 10 | "Algae Image" |
| $\omega_{A,2}$ | Air | (-250,200,20) | 7 | 10 | "Unknown Object" |
| $\omega_{A,3}$ | Air | (100,-200,20) | 8 | 10 | "Oil Slick Image" |

The system is also subject to intrinsic constraints that must be respected during plan execution. The primary constraint is that the flight vehicle is subject to real energy limitations; batteries can store only a finite amount of energy and, if ever the entirety of the system energy is depleted, the system become derelict. Another major consideration for the seaplane-UAS mission is that while operation over the ocean provides several clear operational benefits it also imparts several added concerns for mission planning and constraint definition. Considering beneficial attributes

first, the ocean provides an effectively infinite landing surface, with little to no [low-altitude] air traffic, and any surface obstacles in open water conditions will generally be sparse, slow-speed boats or buoys that are easily avoided once detected. Unfortunately this near-infinite operational surface is continuously changing shape, subject to harsh environmental conditions, and devoid of stationary loiter conditions without expending energy to counter wind and current. From this last condition alone we find that significant planning activity may arise just from the specification of a hard or soft boundary. Environmental disturbances will naturally result in the traversal the operating region and subsequently require routine flight to avoid constraint violation

Aircraft-to-aircraft collision avoidance, cast as sense-and-avoid within the UAS industry [77], is a widely-studied problem for both manned and unmanned aircraft.[78, 79] A comprehensive survey and analysis of collision avoidance can be found in Campo's 2010 thesis on the subject.[80] Today, a growing infrastructure of collision mitigating technologies such as the Traffic Collision Avoidance System (TCAS)[81] and Automatic Dependent Surveillance-Broadcast (ADS-B)[82, 83] are becoming available to a broad range of aircraft. The mission considered in this manuscript involves relatively short low-altitude "hops" in an open-water environment, potentially much further than best-glide distance from land. With the focus primarily on energy and the fundamental properties of the seaplane-UAS drift-fly missions, sense-and-avoid collision avoidance is not considered in the mission planner. We anticipate the need for this in future work, although avoidance of unmapped surface ships/obstacles is perhaps even more critical for a low-altitude seaplane than would be air-to-air collision avoidance.

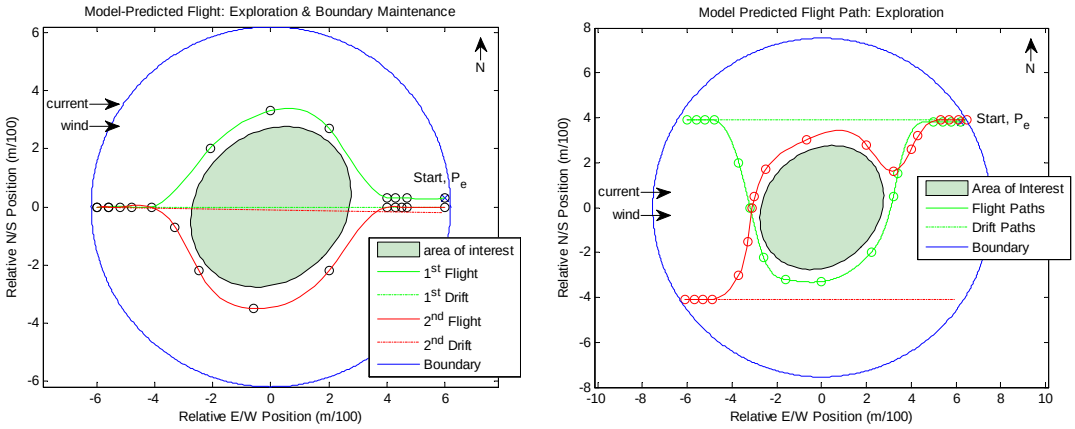
As previously discussed a seaplane-UAS is always subject to drift and, except in the rare circumstance of zero/negligible current and wind, is never at rest while on deployment. As a result mission goals, the collection of waypoints that the vehicle must reach, are not necessarily sufficient to characterize a mission. In the limiting case the system will still be required to satisfy the watch circle/operating-region

constraints, even in the absence of mission goals. Therefore we define the Flying Fish mission as the satisfaction of zero or more surface goals $\omega_{S,i}$ and zero or more airborne goals $\omega_{A,i}$ subject to the set of all vehicle and environmental constraints and we require either an operational region containment constraint (e.g., watch circle definition) or a goal to prevent the mission plan from devolving into unending drift. The planning problem for a single mission is then given by the non-empty set defined by Eq.6.14 meeting the requirements specified by Eq.6.15:

$$\mathbb{P} = \{\mathbb{P}_1, \ldots, \mathbb{P}_n\} = \{\vec{\mathcal{C}}_{OH}, \vec{\mathcal{C}}_{OS}, \vec{\mathcal{C}}_{BH}, \vec{\mathcal{C}}_{BS}, \vec{\omega}_A, \vec{\omega}_S\} \in Missions \tag{6.14}$$

$$\{\vec{\mathcal{C}}_{BS} \bigcup \vec{\mathcal{C}}_{BH} \bigcup \vec{\omega}_A \bigcup \vec{\omega}_S\} \neq \emptyset \tag{6.15}$$

Figure 6.6 presents two variations on an example mission wherein the vehicle is directed to explore the boundary of some surface phenomena (e.g. an algae bloom). In Fig.6.6a the system is directed to perimeter goals around the surface phenomena on successive flights, to conduct a sensor survey, before landing upwind and drifting back through the surface region of interest (ROI). In the second case (Fig. 6.6b) the ROI is a hard obstacle that requires similar sequential edge exploration but does not allow return drift through the center of the containment region; this ROI might be, for example, an oil spill, for which drifting in the constrained region could negatively impact the system. A real mission example currently under consideration involves the application of satellite imagery to provide estimated positions for several ROIs in an inland lake. The first sub-mission specifies overflight (aerial goals) of these ROIs with a hyperspectral imager to determine if additional surface measurements are warranted. After the successful completion of the first sub-mission, the results of the overflight data indicate some number of surface goals (a subset of the previous aerial goals) that should be reached in order to take water samples.



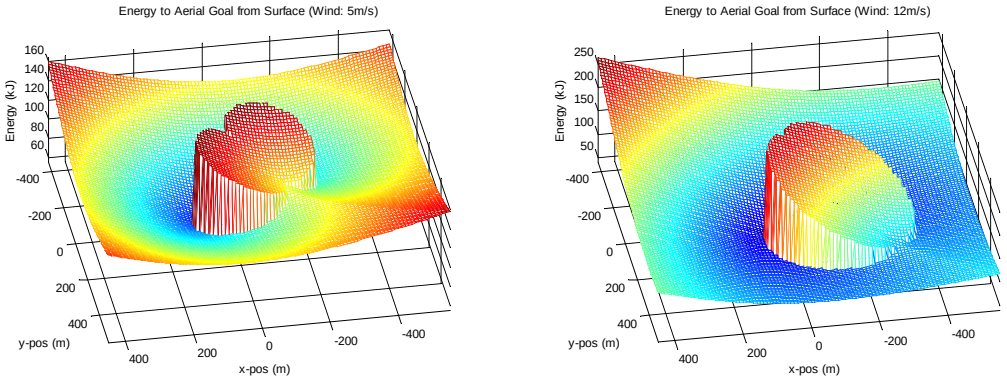

(a) Unobstructed Exploration

(b) Obstructed Exploration

Figure 6.6: Example Missions

Given a constrained environment and mission description we can now apply the previously-presented models to develop simulations for the critical dynamic processes required for every stage of mission execution. We utilize the solar, battery, motor, and wind models presented in this chapter in concert with the aerodynamic and drift models presented in Chapter 3 to develop a takeoff/climb simulation. Adding the kinematic flight model and Dubins path constraints from Chapter 3 yields a guided cruise simulation between any two airborne locations with specified headings. At this stage we have the tools necessary to simulate energy output and collection for flight from the water to any aerial goal. Figures 6.7a and 6.7b present the simulated cost from takeoff at the origin to reach an aerial goal at the given x-y coordinates (and a 25.0m cruise altitude) subject to 5m/s and 12m/s winds, respectively. The Northwesterly wind is directed across the plot in the x-y plane at -45.0 degrees from the x-axis. The structure of the plots clearly shows the characteristics of the upwind-constrained takeoff and limited turn rate, creating a high-cost region around the liftoff location wherein more extensive maneuvering is required to reach the goal despite proximity to the starting location. In the low-to-moderate 5m/s wind case the energy benefit of flying with a tailwind is, at least with a range of 500m, overshadowed by the cost of

then having to turn downwind from the takeoff vector. However, in the high 12m/s wind case the benefits of a tailwind are far more pronounced and the cost of reaching liftoff-proximal goals is also more pronounced given the higher cost of maneuvering back into a headwind.



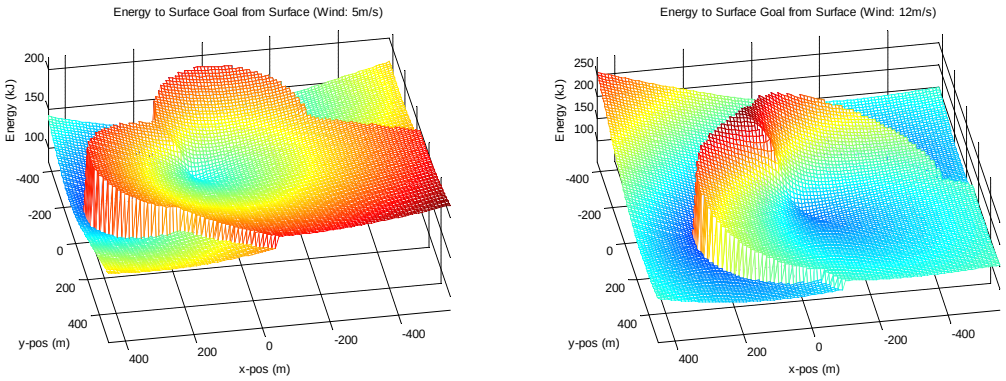

(a) Wind: 5.0m/s

(b) Wind: 12.0m/s

Figure 6.7: Energy Required to Takeoff and Fly to an Aerial Goal (Altitude = 25.0m)

The Dubins path solver generates the transition between an arbitrary position/heading to a new position with an upwind heading allowing the cruise simulation to be extended to reach the constrained final approach vector. Note that final approach must be pre-simulated for each landing to propagate the trajectory backwards from the desired landing point to the start of descent. Final approach and touchdown are simulated using the same models as takeoff. Throttle and trajectory smoothing are applied where necessary. Together these simulations estimate the energy output and collection for flight between any two points in the plan environment. Figures 6.8a and 6.8b present the simulated cost to takeoff from the origin and fly to a surface goal at the given x-y coordinates subject to 5m/s and 12m/s wind, respectively, given a Northwesterly wind directed across the plot in the x-y plane at -45.0 degrees from the x-axis. The structure of the plots now reveals the limits of turn-constraint maneuvering between the mutually heading-constrained takeoff and landing. The lobes of the plateau are clearly larger due to the spatial requirements of two sequential

constrained-rate turns separated by the required return landing distance. Again we observe that transitions to liftoff-proximal locations are more expensive due to increased maneuvering requirements. We observe again that in the high-wind case the plateau is pushed downwind and yields more pronounced costs for upwind and liftoff-proximal maneuvers.




(a) Wind: 5.0m/s

(b) Wind: 12.0m/s

Figure 6.8: Energy Required to Takeoff and Fly to an Aerial Goal (Altitude = 25.0m)

Drift is simulated using the solar, battery, and wind models discussed in this chapter along with the empirically-designed drift model presented in Chapter 3. An example of the drift model simulation was presented in Chapter 3. The drift simulations indicate relatively low energy expenditure compared with flight, with positive energy input during periods of modest to high solar insolation(Fig. 6.1). The greatest impact of drift on solar energy collection is that higher drift speeds, due to the combination of ocean currents and wind, will reduce total energy-harvesting time. It is also the case that, since the panel is oriented slightly to the rear of the vehicle, drifting away from the sun near dawn or dusk yields a high off-axis solar incidence angle to the panel which attenuates solar energy collection proportional to the cosine of total incidence angle (Eq. 6.2).

We now define the set of admissible transitions and specify the evaluation metrics that

will be used to judge the utility and cost of both transitions and candidate mission sequences (plans). From a given surface position there are a finite set of admissible transitions that have already been discussed, including: flight to an airborne goal, flight to a surface goal, and drift. The latter action is defined as a Non-Operation or No-Operation (no-op) transition. At every surface point not currently in violation of a constraint, the admissible set of transitions includes at least the no-op action wherein the planner may elect to drift until a constraint is violated or it is otherwise appropriate to take operational action (e.g. when the system energy becomes full). No comparable no-op exists for the airborne case, as all airborne actions expend more energy than they collect, but a reasonable and useful analog can be defined by specifying an auxiliary action that states that the system may fly from any aerial position and land at the upwind/updrift boundary of the operational area from a drift-direction perspective. Implicit in this active-no-op is the maximization of the next available true-no-op by positioning the vehicle at the maximum updrift location. Notably the up-drift-reposition no-op is reasonable for inclusion to the admissible transitions of a surface goal as well. The set of no-op actions is outside the specification of a mission and has no associated goal-seeking value. Given these definitions the admissible set of transitions from any aerial goal include transitioning to: (1) another aerial goal, (2) a surface goal, or (3) the updrift no-op location. Similarly, the admissible set of transitions from any surface goal include transitioning to: (1) an aerial goal, (2) another surface goal, (3) the up-drift no-op location, (4) a drift no-op. An example of the collected admissible transitions between a set of goals (i.e., neglecting no-ops) is presented in Fig.6.9.

Transitions and missions are evaluated based on a number of parameters. The first is Cost ($\mathbb{C}$), which will be defined as the amount of any limited resource that is expended during transitions. Cost may be used, for example, to represent energy expenditure, transition time, or some equivalent to tachometer time (an indicator

of motor usage common on GA aircraft). The selection of the physical metric will depend on the planning strategy. Using, for example, the energy expenditure over the $i^{\text{th}}$ transition ($\mathcal{E}_{\text{cost,i}}$) the costs of the $i^{\text{th}}$ transition and the $k^{\text{th}}$ flight plan are denoted, respectively,as:

$$\mathbb{C}_{k,i} = \mathcal{E}_{\text{cost},k,i} \tag{6.16}$$

$$\mathbb{C}_{k} = \Sigma_i(\mathcal{E}_{\text{cost},k,i}) = \Sigma_i \mathbb{C}_{k,i} \tag{6.17}$$

The second critical evaluation parameter for the planner is potential transition Benefit ($\mathbb{B}$). Potential benefit is loosely defined as the *potential for reward*, in the form of some quantity of limited resource that may be recaptured, following a particular set of transitions. For example the planner might estimate the amount of energy that could be recovered ($\mathcal{E}_{\text{rcvr,i}}$) over the next drift transition or some alternate formulation of potential energy based on the distance traveled upwind. The benefit function for the $i^{\text{th}}$ transition and the $k^{\text{th}}$ flight plan are, respectively:

$$\mathbb{B}_{k,i} = \min(\mathcal{E}_{\text{rcvr},k,i+1}, (\mathcal{E}_{\max} - \mathcal{E}_{\text{sys},k,i})) \tag{6.18}$$

$$\mathbb{B}_{k} = \min(\Sigma_i \mathcal{E}_{\text{rcvr},k,i}, (\mathcal{E}_{\max} - \mathcal{E}_{\text{sys},k,0})) \tag{6.19}$$

where $\mathcal{E}_{\max}$ is the system maximum energy capacity and $\mathcal{E}_{\text{sys},k,i}$ and $\mathcal{E}_{\text{sys},k,0}$ are the current energy storage of the system at the end of the $i^{\text{th}}$ transition and at the beginning of the $k^{\text{th}}$ mission planning cycle, respectively. In this formulation the minimum of the difference between energy storage capacity and actual storage must be used as the planner should not consider potential energy-harvesting benefits that are not realizable due to batteries capacity saturation limits. The third evaluation parameter is value ($\mathbb{V}$), which has already been discussed in the context of mission goals. Value is defined as the reward, *not* in the form of limited resources, associated

with some set of transitions. Value may be, as in the case of goal value, an intangible reward meant to entice the planner to trade some limited resources for goal and mission completion. The value of the $i^{\text{th}}$ transition and the $k^{\text{th}}$ flight plan can be written as:

$$\mathbb{V}_{k,i} = \mathtt{v}_{0,k,i} + \mathtt{v}_{t,k,i} \cdot \Delta t \tag{6.20}$$

$$\mathbb{V}_k = \Sigma_i(\mathbb{V}_{k,i}) \tag{6.21}$$

Note that the initial value ($\mathtt{v}_{0,k,i}$) and time-incentive value gain ($\mathtt{v}_{t,k,i}$) associated with a given transition may be identically zero, or possibly negative if a transition is particularly undesirable. Given the cost, benefit, and value definition a fourth aggregate parameter, Utility ($\mathbb{U}$), is defined which combines all of the other parameters in a weight sum. The utility, given the a set of weights $\{\tau_{\mathbb{V}}, \tau_{\mathbb{B}}, \tau_{\mathbb{C}}\}$, is given for the $i^{\text{th}}$ transition and the $k^{\text{th}}$ flight plan as:

$$\mathbb{U}_{k,i} = \tau_{\mathbb{B}} \cdot \mathbb{B}_{k,i} + \tau_{\mathbb{V}} \cdot \mathbb{V}_{k,i} + \tau_{\mathbb{C}} \cdot \mathbb{C}_{k,i} \tag{6.22}$$

$$\mathbb{U}_k = \tau_{\mathbb{B}} \cdot \mathbb{B}_k + \tau_{\mathbb{V}} \cdot \mathbb{V}_k + \tau_{\mathbb{C}} \cdot \mathbb{C}_k \tag{6.23}$$

To summarize, the planner contains a full simulation suite that estimates energy expenditure and collection, subject to wind, ocean currents, and solar dynamics over any mission action. This section developed careful definitions of missions, goals, and constraints, a mechanism for recording plan attributes, and a mission evaluation framework. Below we define strategies to search the set of possible action sequences, determine the best such sequence, and ultimately build the sequence of actions to take over the course of a full day.

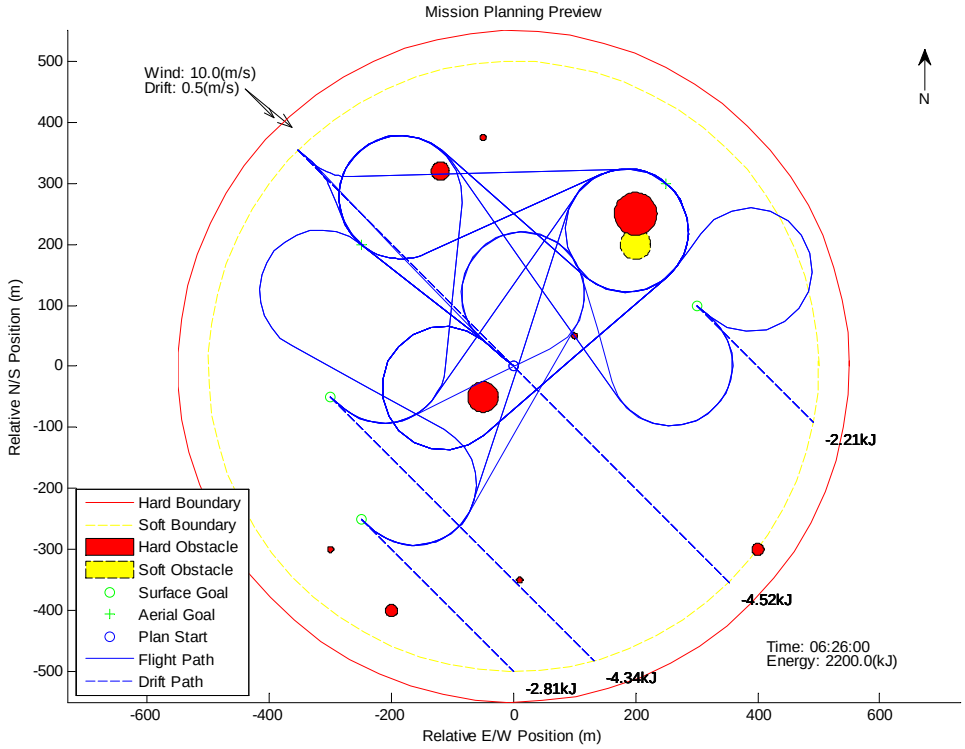



Figure 6.9: Expansion of All Possible Mission Trajectories Between Goal Points

## 6.3 Search Strategies

The mission planning problem is an example of combinatorial optimization for which the planner must build, given an initial state, a finite sequence of actions from a finite set of admissible behaviors such that the resulting sequence satisfies required mission goals and respects plan constraints subject to an optimality criterion such as that represented by Equation 6.23. A mission is satisfied by any sequence of transitions that visits every goal at least once without violating any environmental or system constraint. If this definition is restricted to require that each goal be visited exactly one time, then the planning problem becomes a form of the Traveling Salesman Problem (TSP), a well known NP-hard problem.[84] The much-studied TSP is concerned with finding the optimal (shortest distance) "tour" for a traveling

salesman to visit every city in the set of all cities exactly once (Fig. 6.10).[84, 85, 86] The Flying Fish problem is more specifically concerned with visiting every waypoint goal from a given starting position, a generalization of the TSP that is often denoted as the Traveling Salesman Path Problem. Within this thesis the term TSP encompasses the Traveling Salesman Path Problem as well as the traditional TSP. An exact solution to TSP can most obviously be found by exhaustive or "brute force" exploration of all permutations of available paths in a set of goals. However this mechanism is generally impractical for even moderately sized sets of goals. The complexity for brute-force search of a TSP with $n$ mutually connected goals (from a single source goal) is $O((n-1)!)$ and the problem of finding the minimum tour from any source node worsens to a polynomial factor of $O(n!)$. As such, even a problem with only ten goals has greater than 360,000 possible paths to explore from any single source goal and over 3,000,000 paths when searching every permutation of starting goals. Some of the greatest advances in TSP solutions have been based on applications of dynamic programming, realizing solutions in time $O(n^2 2^n)$[87] or better.[88] There also exists a wide variety of solution mechanisms that utilize some form of heuristic to guide exploration which can be shown to converge relatively fast even for large problems but which often do not guarantee optimality in the general case. One of the more well-known heuristics is nearest-neighbor, in which the nearest unvisited city is always selected at each junction. It has been shown that the nearest-neighbor heuristic, which may produce nearly-optimal paths in some cases, can also produce the most expensive possible path under certain pathological conditions.

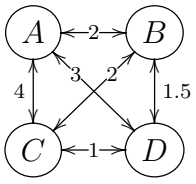



Figure 6.10: Symmetric Graph Problem with Four Nodes

The TSP can be represented, using graph theory, as set of nodes (cities) and a set of edges (roads) which connect pairs of nodes (Fig. 6.10). Individual formulations of the TSP and many of the available solution mechanism are subject to a few common assumptions or rules about the structure of the TSP. The first assumption is that the search transition graph is symmetric, i.e., the cost to travel in one direction between two goals is the same cost incurred traveling in the opposite direction between the same two goals. In physical terms this would mean that the road between any two cities is not expected to be longer in one direction than it is in the opposite direction. The second common TSP assumption is that all transitions have a positive cost. The seemingly obvious physical assertion is that the salesman cannot travel less than zero miles between two cities, given a distance cost metric. Finally, TSP solutions often rely on the application of the triangle inequality which states that the cost over two successive transitions to a specific terminal node must be equal to or greater than the cost that would be incurred traveling directly to the terminal node. This assumption is most often applied over graphs with straight-line edges, or else by heuristics that evaluate the triangle inequality based on optimal straight-line estimates. TSPs that obey the triangle-inequality are collectively referred to as metric-TSPs. In physical terms this assumption specifies that the salesman cannot reach a goal with less cost by first going to another goal; at best the intermediate goal will be on the optimal path to the terminal goal. The four-node graph TSP presented in Fig. 6.10 meets these criteria. As discussed later, the Flying Fish energy-based mission planner does not guarantee any of these basic TSP assumptions.

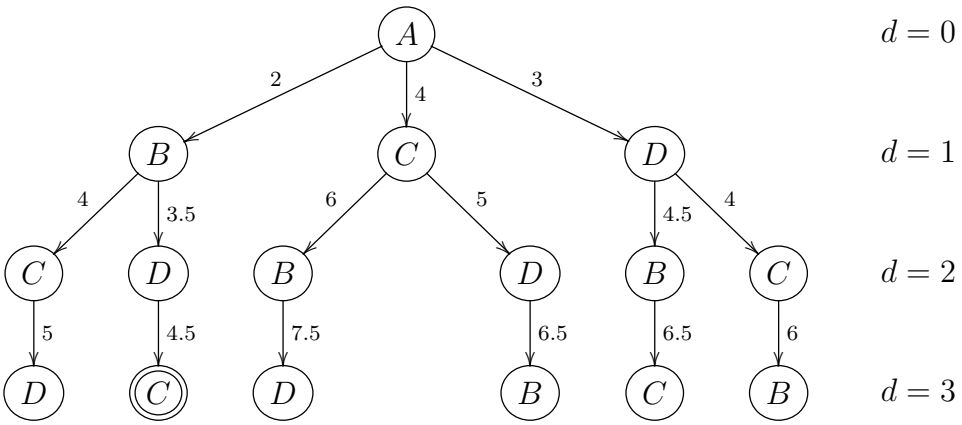



Figure 6.11: Example Search Tree for Four Node Graph

The solution mechanisms that are applied to the TSP, or more generally any problem with a graph structure, can be loosely classified as search algorithms. The application of a search algorithm to a graph yields a tree structure in which the available paths branch out from the root node for increasing exploration distance in the graph (Fig. 6.11). Search depth ($d$) is defined as the number of steps or transitions taken from the root node. Figure 6.11 presents the the complete search tree for the example TSP graph presented in Fig. 6.10. In this simple case the optimal solution to the Traveling Salesman Path Problem starting from node $A$ is easily determined; the sequence $\{A, B, D, C\}$ traverses every node of the graph with the lowest possible path cost ($\mathbb{C} = 4.5$). Note that this is not a unique optimal tour amongst all tours in the graph as the reverse permutation $\{C, D, B, A\}$ has the same cost. The complexity of a search tree is often characterized by a branching factor which is defined as the number of branches, or child nodes, that extend from a parent node at a given search depth. In mutually-connected graphs, like the example case, the exact branching factor ($b_f$) can be written for a graph with $n$ nodes as:

$$b_f = (n - 1) - d \tag{6.24}$$

From this equation the branching factor of the example problem can be determined

as $(3-d)$. Whenever the underlying graph geometry becomes more complicated, for example if there are nodes that are not mutually connected or nodes that can be re-visited, specification of the exact branching factor becomes more involved. Under these circumstances it is typical to compute an average or worst-case branching factor.

Applying search to the Flying Fish problem poses a series of specific challenges. First, the search problem is distinctly asymmetric. The impact of wind on the cost of each flight segment and the upwind-constrained landing and takeoff headings give rise to transitions that can have radically different costs for traversal in one direction versus the other. For example, it would clearly require more energy to fly from a downwind goal to an upwind goal that it would to fly the reverse path. While there are mechanisms to convert an asymmetric graph problem into a symmetric graph problem, they generally require more resources to solve and effectively search dual/complementary symmetric graphs.[89] Additionally, none of the transition costs are known in advance and each transition cost is potentially dependent on every previous transition, as the departure heading of each node affects the arrival heading of subsequent nodes. Furthermore, the necessary inclusion of the non-goal-seeking no-op actions (drift or upwind flight) has a critical impact on the size of the search tree. Consider the tree for the Flying Fish planning problem for $n$ total goals and one no-op action, which is admissible from every node (Fig. 6.12).

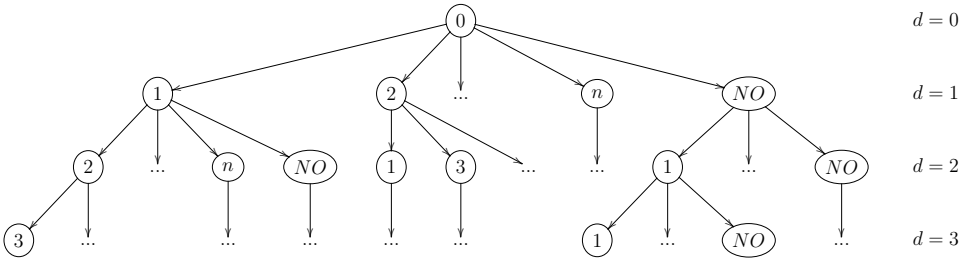



Figure 6.12: Example Search Tree for Four Node Graph

In this problem the no-op action cannot be represented by staying at a constant search depth as the time of day and vehicle state change nontrivially over a no-op

transition. As a result there exists a no-op transition out of every tree node at every tree depth. At every depth, every path that includes $m$ goals has an admissible no-op node below which the entire tree is expanded again, as if from depth $m$ with different initial conditions. This attribute results in potential branching to infinite depth. The average branching factor at a given depth is a complex function of the preceding tree structure making it difficult to determine the average branching factor for any non-trivial depth. The average branching factor at depth $d = 1$ can be resolved as follows:

$$\begin{aligned}
\bar{b}_f &= \frac{(\text{goal parents})(\text{goal children}) + (\text{no-op children})}{(\text{all parents})} && (6.25)\\
&= \left.\frac{(n-d+1)(n-d+1)+(n-d+2)}{(n-d+2)}\right|_{d=1} && (6.26)\\
&= \left. 1 + \frac{(n-d+1)^2}{(n-d+2)}\right|_{d=1} && (6.27)\\
&= 1 + \frac{n^2}{(n+1)} && (6.28)
\end{aligned}$$

To limit search-space size, we define a *search horizon* as a limit, by some choice of measure, at which further tree exploration will be disallowed. An example search horizon is search tree depth. The resulting depth-limited search requires a depth horizon at least equivalent to the desired number of goal actions and will include comparably large numbers of no-op branching sub-trees to the search space. Another search horizon measure, which is particularly well suited to the Flying Fish TSP problem, is total plan time. Given that there is as much as an order of magnitude difference between the duration of a watch-circle traversal by flight (fast) and drift (slow) a plan-time search horizon tends to restrict no-op branching more than the depth limit horizon.

Perhaps the greatest complication to the Flying Fish planning problem is the inclusion of solar energy harvesting. To begin, the triangle inequality cannot be guaran-

teed to hold if an external (variable) cost reduction is included. Subject to energy harvesting, the cost over any set of transitions to a single goal may be dominated by external factors (solar availability) and not the physical relationship between the search nodes. Given this condition, the Flying Fish problem is a non-metric TSP. More critical however is the fact that the most logical cost for Flying Fish, energy, results in graphs with negative transition costs whenever a positive amount of energy is harvested. Negative edge costs can give rise to negative-cycles within the search space and yield non-monotonic utility. The negative cycle is desirable from an energy recovery perspective but can yield sub-optimal search results or prevent a search algorithms from converging to any solution. Of course, the omission of energy harvesting would miss the goal of planning for an energy-regenerative system, so it is not possible to eliminate this issue.

Under the planning requirements of the solar-regenerative seaplane-UAS the basic TSP must be nontrivially extended. Let the seaplane-UAS TSP be called the Frequent Flier Salesman Problem (FFSP). The FFSP is defined to have the same goal as the traditional TSP, the determination of the minimum cost tour through a set of goals/cities. However, the FFSP assumes that the Salesmen (a UAS sensor payload in this case) will travel by aircraft, purchasing all tickets using frequent flier miles (renewable energy in this case). The salesman's frequent flier miles are assumed to be bounded above by program limits ($m_{max}$) and from below by the salesman's desire not to be stranded ($m_{min} = 0$) (analogous to battery capacity). Over a given transition the salesman will be required to expend a relatively large number of miles to reach a goal while simultaneously accruing a fractional number of those miles for the distance traveled to the goal. Furthermore, there are assumed to be "reward scenarios" in the frequent flier program wherein certain flight offers more frequent flier miles than they cost, but these flights are always directed to non-goal destinations (drift). If, at the beginning of every plan, the salesman starts with with $m_{max} - n$ frequent flier miles

$(0 < n < m_{max})$ then the cost ($\mathbb{C}$), in miles, of any complete or partial admissible path through any number of cities is bounded by: $-n < \mathbb{C} < m_{max}$. The FFSP is effectively an asymmetric non-metric TSP with bounded, but often negative, edge costs and a path cost everywhere bounded by initial plan conditions. The largest single challenge that faces the FFSP problem is the definition and identification of provable optimality conditions. It is difficult to define an optimal path when subsequent transitions can be selected to drive plan cost to zero (or below, if the initial condition permit). Moreover, when a search is underway numerous branches can appear equally promising from a cost perspective thus the search space can become formidable, particularly given the presence of drift-fly "cycles" and the no-op action choice.

## 6.4 Case Study Results

This section defines and presents solutions to a series of tractable problems over which different search strategies and optimality criteria are considered. Both short-term and long-term planning epochs are considered, including a night mission where no energy harvesting is available (and optimality can be strictly defined), a mid-day maximum solar insolation solution to the same mission, and finally an exploration of the dawn/dusk planning problem. In each case the capabilities and limitations of the proposed algorithms, cost metrics, and constraints will be evaluated.

### 6.4.1 Greedy Search

The initial planner implementation used greedy search that makes a locally-optimal action choice at each search node. Use of a locally-computed heuristic substantially reduces search time relative to strategies offering global optimality, thus supports the examination of long-duration (i.e., significant search depth) plans, possibly over the course of a full mission deployment.. Greedy search solutions can be reached quickly,

relative to other search strategies, but do not guarantee optimality. The challenge in this problem is the appropriate definition of the utility function terms to guide the short-sighted search towards transitions that remain appropriate for the distant horizon problem.

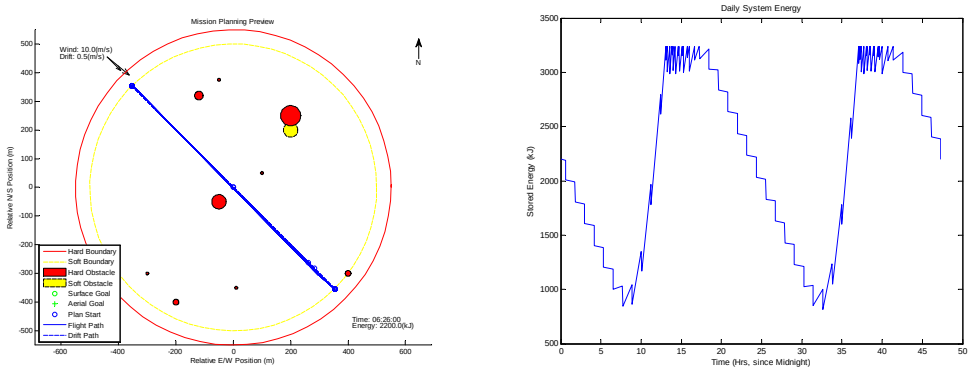
Greedy search always expands the node with the highest utility action available without regard for the past cost required to reach that particular node.[90] In a strict TSP greedy search is a nearest-neighbor heuristic. For the seaplane UAS planning problem, the TSP is broadened to allow multiple visits to each goal and to upwind/downwind watch region boundaries. In this case, the planner utilizes the utility, cost, value, and benefit expressions already presented in Eqs. 6.16-6.23. The search problem is segmented into the available permutations of all goal-to-goal paths that have exactly one takeoff and one landing. In this fashion the planner assesses, from each surface goal, the path to every other available surface goal. The Cost term in the utility function is meant to bias the planner away from expensive paths. The Value term in the utility function provides incentive for the planner to trade system energy for goal satisfaction, else the greedy planner might never choose a path that expended the extra energy required to reach a remote mission goal. Finally, the Benefit term in the utility function provides an incentive for the vehicle to reach states that have higher potential for energy recover, which generally involves landing further upwind. The potential benefit term is meant to to bias the myopic search towards energy-conscious transitions that have a higher likelihood of providing long-term benefits. Utilizing this model, the "greedy" planner expands the highest utility nodes until it reaches the plan horizon; a two day planning horizon was selected for the presented case study. The resulting greedy or best-first search algorithm is only one-step, and not globally, optimal. The greedy search is also subject significant sub-optimality if subject to pathological search topologies.


Figure 6.13: Douglas Lake Operating Region

To determine the capabilities and limitations of the greedy search algorithm it was applied to a series of planning problems of increasing complexity. The first mission was the basic boundary maintenance or "watch circle" problem; this test case was aimed at demonstrating overnight constraint-region persistence. The second mission considers the addition of a surface exploration goal to the basic persistence problem. Finally, the third mission introduces a range of surface and aerial goals to test the limitations of the planner. The test case simulations are initialized to the latitude and longitude of the Flying Fish FAA-authorized test site on Douglas Lake in North Michigan at the University of Michigan's biological research station. This choice provides vehicle-measured environmental data to serve as a sanity check for the environmental models. The search space is constrained to the boundaries of Douglas Lake's Fishtail Bay region (Fig. 6.13). Search solutions were sought for solar conditions in late March 2011.

The first critical test is to see if solar conditions are sufficient for overnight survival. The planner is initialized with no fixed exploration goals, a soft constraint providing 50m clearance from the shore, and a hard constraint that marks the shoreline. The

planner starts with fully charged batteries (~3200kJ) and plans from one minute after midnight for two consecutive days of operation. The resulting plan (Fig. 6.14a) accommodates the boundary constraints with a series of flights that either maximize energy-recovery potential or minimize energy-collection waste. The resulting balanced energy budget is shown in Fig. 6.14b.



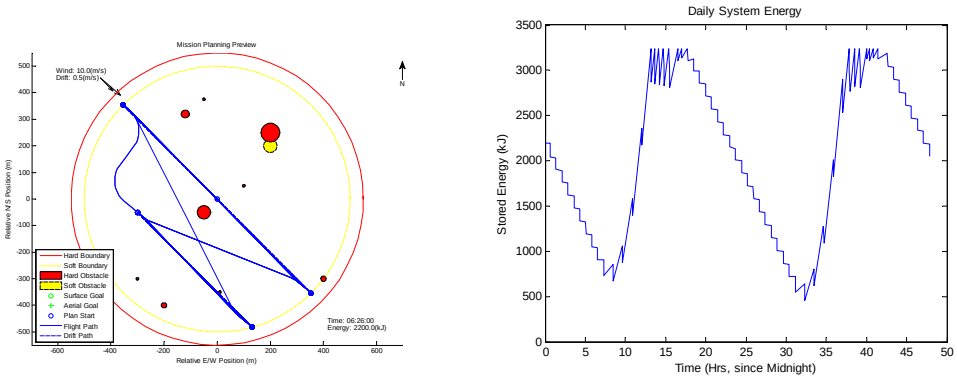

(a) Mission Plan (b) Energy Budget

Figure 6.14: Boundary Maintenance Mission

For the given location (at a high latitude) and the given time of year a sustainable plan can be found for persistent wind up to ~10m/s. Above this limit the high winds compromise ability to balance the energy budget by increasing the cost of upwind flight while also decreasing energy collection on drift, as faster drift rates reduce total downwind drift time. Under different solar conditions higher winds can be tolerated. For example, solar conditions at Douglas Lake in June provide ~20% more energy. Alternately, a deployment in March that is closer to the equator would benefit from as much as 10% improvement to solar energy intake. Additionally, in this case study, the planner would always choose to fly when the batteries were fully charged, as the energy benefit of drift drops to zero. Given that significantly fewer flights are required if every drift cycle is continued to the downwind boundary it is not clear that this strategy is either provably-optimal or necessarily a best-practice, especially if risk is

factored into the cost function in future work.

Given that a balanced energy budget was achieved with the basic mission profile a more advanced mission was attempted. The second plan was initialized with the same conditions and constraints as the base mission with the addition of a stationary surface exploration goal. The second plan (Fig. 6.15a) produced another balanced daily energy budget (Fig. 6.15b) while visiting the exploration goal as many as 100 times in a two-day plan. Here again plan viability is subject to the accuracy of the environmental measurements. A dramatic reduction in solar insolation or increase in wind speed can tip the balance of the energy budget under the sustainability constraint.




(a) Mission Plan (b) Energy Budget

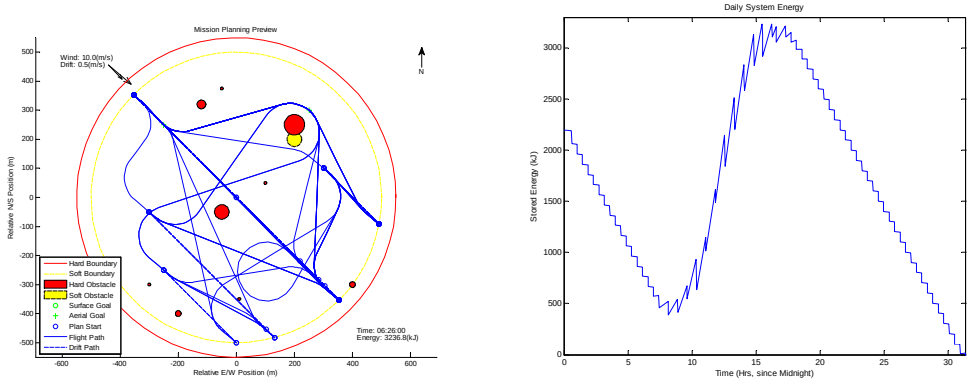
Figure 6.15: Boundary Maintenance + Surface Goal Plan

To test the system against a more challenging planning problem a set of multiple surface and airborne waypoints were combined with the base mission constraints and another two-day plan was developed (Fig. 6.16a). In this final case the energy expended due to the search incentive in combination with the myopic limitations of the greedy search resulted in system energy sacrifices and an imbalanced energy budget (Fig. 6.16b). Recall that the utility value of a transition is specified for the

Cost ($\mathbb{C}$), Benefit ($\mathbb{B}$), and Value ($\mathbb{V}$) of the transition by:

$$\mathbb{U}_{k,i} = \tau_{\mathbb{B}} \cdot \mathbb{B}_{k,i} + \tau_{\mathbb{V}} \cdot \mathbb{V}_{k,i} + \tau_{\mathbb{C}} \cdot \mathbb{C}_{k,i} \tag{6.29}$$

The objective function weighting factors are set to 1.0. Two critical factors effect performance in this case: (1) as night falls and the solar energy drops below the hotel-load the potential benefit term becomes negative, effectively adding cost to the utility function for increased drift time, and (2) as the time between goal visits grows, due to the increased number of goals to be reached, the average magnitude of the value incentive prior to visitation increases, motivating the search to sacrificing more energy for exploration even as solar-energy drops to zero. As a result of the first factor, flights that end further upwind are assessed larger "negative benefit" because they are followed by longer periods of drift subject to the hotel load without solar input. This false disincentive highlights the shortcoming of the greedy search as the algorithm elects to fly to "cheaper" nearby/downwind goals and fails to consider that the shorter subsequent drift will require many more flights to satisfy the boundary constraints. The second factor exacerbates this phenomenon by providing high incentive for exploration even while the system is losing energy. Furthermore, regardless of the choice of utility function, the factorial increase in permutations of goal waypoints reduces the likelihood that the greedy search will find the optimal path amidst all available options.




(a) Multi-Goal Mission Plan (b) Multi-Goal Energy Budget

Figure 6.16: Greedy Multi-Goal Planning

Ultimately greedy search is capable of developing overnight-survivable plans for the baseline watch-circle mission as well as the extended basic exploration mission but fails to find a sustainable plan if subject to high cumulative goal valuations after sunset. In this case the greedy search will elect to trade energy for the intangible value of ancillary goal exploration rather than conducting an upwind flight, which is subject to a false disincentive after sunset. Fundamentally, the search fails in the challenging case because the cheapest transitions (the shortest flights) are not the best transitions for long-term survival (maximum upwind flights) but greedy search will never choose a more expensive path now to get to a cheaper path later. It is merely the case that the Benefit and Value constructs were insufficient in the more challenging case to overcome the nature of the greedy search. It may be possible to develop improved policies for the utility function, for example by the inclusion of some function of available solar energy, but it is not clear that significant improvement can be had without searching further ahead, either within the utility function or through another search algorithm. This case study provides two important lessons: (1) valuing overnight exploration is dangerous from the perspective of sustainability, as the safest solution will always be to minimize cost in the absence of solar recharge, and (2) it is

difficult to develop a benefit valuation that will lead a greedy planner to make good long term decisions.

### 6.4.2 Midnight Missions: Non-Energy-Harvesting Optimal Planning

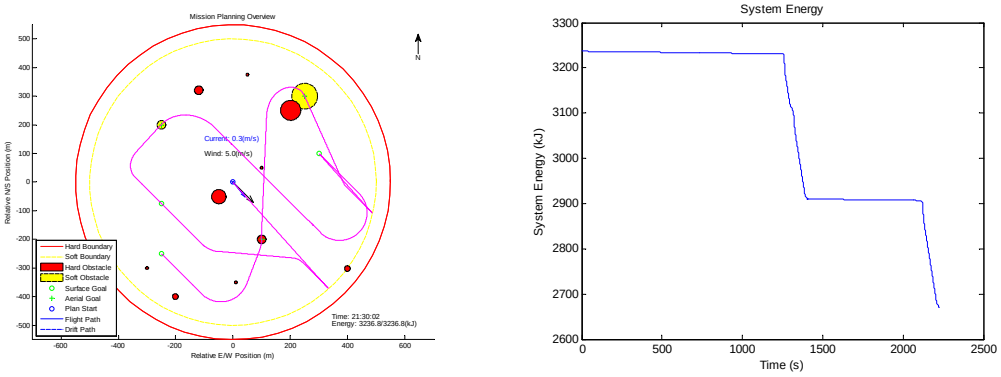
Given the nonmonotonic total energy/cost sequence exhibited by the system with energy harvesting, we first consider the non-energy-harvesting sub-problem. This subproblem is applicable in practice as the complex Flying Fish planning problem naturally devolves to the non-regenerative case over the approximately 12 hours between sunset and sunrise. When no solar energy is collected negative transition costs are eliminated from the search tree and the FFSP reduces to the slightly more tractable asymmetric, non-metric TSP problem. Notably, given that system survival is of paramount importance, it is not clear that any mission, aside from survival within the hard boundary constraints, would typically be pursued overnight. Nevertheless, plans will be developed for the example mission environment in order to characterize the globally optimal solution for that mission subject to strictly monotonically-decreasing energy thus positive cost ($\mathbb{C}$) over each transition.

For this problem a one-visit-per-goal solution is sought and goal incentive values are made identically zero ($\tau_{\mathbb{V}} = 0$). Since we are looking for the strict energy optimal plan, and energy is not actually recovered in trade for the exploration incentive, the goal valuation must be zero to avoid transition biases yielding a sub-optimal energy path. The transition benefit term is also canceled from the utility function ($\tau_{\mathbb{B}} = 0$) as the potential energy benefits of a transition are not equivalent to physically-collected energy and may also render the utility function sub-optimal for strict energy usage optimality. Under these conditions the utility function of the $i^{\text{th}}$ transition and $k^{\text{th}}$ plan of $n$ transitions are identically equal to the expended energy ($\tau_{\mathbb{C}} = 1$):

$$\mathbb{U}_{k,i} = \mathbb{C}_{k,i} = \mathcal{E}_{\text{cost},k,i}, \;\; i = 1,\, 2,\, 3,\, ... \tag{6.30}$$

$$\mathbb{U}_{k,i} = \Sigma_i^n \mathbb{C}_{k,i} \tag{6.31}$$

Subsequently a uniform-cost search algorithm is applied to the planning problem. The uniform-cost search always expands the transition with the lowest current cumulative cost from the root node and is guaranteed to converge to the globally optimal solution if every transition cost is bounded from below by some positive $\epsilon$.[91] In the non-regenerative case the Flying Fish planner is always subject to, at least, the system maintenance or "hotel" loads which guarantee globally-positive transition costs and the optimal solution from uniform-cost search. The resulting plan and energy budget from this search are presented in Fig. 6.17. The three instances for which the mission path crosses over surface constraints (obstacles) are actually goal actions wherein the system was directed to fly over the obstacles, presuming sufficient altitude separation (as described in Sec. 6.2), for exploration purposes (the soft constraints are expected to be an oil spill and an algae bloom, the hard constraint is an unidentified surface vehicle). Given positive transition costs the presented solution is guaranteed to be a globally optimal flight plan that reaches every goal. The most interesting feature of the plan is the drift segments; bearing in mind that no solar energy was available, the downwind drift segments represent non-goal-seeking transitions that capitalized on the complex internal dynamics of the search space to realize a lower-cost solution than might be found by inspection. Effectively, by drifting downwind with only a very low "hotel" cost the system is able to reach goals by a more direct route that is cheaper than the more circuitous route requiring a turn to the downwind vector and/or a turn back to an upwind vector to reach a surface goal following a downwind leg.




(a) Midnight Mission Plan

(b) Midnight Energy Budget

Figure 6.17: Midnight Uniform-Cost Planning

With a two hour search horizon the uniform-cost search converges to the optimal solution in ~315s on a dedicate desktop server (Intel Xeon X3450, Quad-Core CPU @ 2.67GHz, 8Gb RAM) which is more powerful than the current Flying Fish embedded computer (TI OMAP Single-Core CPU @ 600MHz, 256Mb RAM) but that could be envisioned in future seaplane UAS. Uniform-cost is an optimal strategy but is also computationally-complex in that it can explore a substantial part of the exhaustive search space. Given that problem spaces in practice may be longer than two hours and that the target execution architecture is an embedded system, it is appropriate to consider methods to improve search efficiency. As a first step consider A* search, which augments uniform cost with a cost-to-go heuristic used to guide exploration along paths that appear promising in the future rather than strictly the past.[92] A* search computes total utility of a node as cost to reach the node from the root plus an estimate (heuristic) of the cost to complete the search from the given node. The $k^{\text{th}}$ path over $n$ nodes has an associated heuristic cost-to-go given by $\mathbb{H}$ that gives a total A* utility of:

$$\mathbb{U}_k = \sum_{i=0}^{n} \mathbb{C}_{k,i} + \mathbb{H}_{k,i} \tag{6.32}$$

The most important feature of A* search is that an admissible heuristic guarantees

the search will converge to the optimal solution (for graphs with positive transition costs).[91] A* search admissibility requires that the heuristic be "optimistic," which is to say that it underestimates, at every step, the cost-to-go. Given that the true cost-to-go is always higher than the estimated cost-to-go the A* search can use the optimistic evaluation of each explored node to ignore any branches of the search tree for which the estimated utility exceeds the current utility without missing the optimal solution. Heuristics that produce estimates closer the true cost are called "more informed" heuristics. A poorly informed heuristic may not improve search convergence substantially beyond the performance of the uniform-cost search. A challenge for use of A* is the development of a maximally-informed admissible heuristic.

Perhaps the most common heuristic in a spatial search is the straight-line distance to the goal. However, in the TSP problem there is no single goal but rather a set of sequential goals. The first heuristic that was attempted was based on analyzing the set of unvisited goals to find the single shortest path between any two ($p_{min}$). Given $n$ unvisited goals, the power in kW required for trimmed cruise ($\mathcal{P}_{\text{cruise}}$), and the maximum airspeed ($V_{max}$) the heuristic value in kJ is given by:

$$\mathbb{H}_{k,i} = \frac{(n-1) \cdot p_{min}}{V_{max}} \mathcal{P}_{\text{cruise}} \tag{6.33}$$

Effectively the heuristic attempted to capture the absolute minimum energy required to sequentially fly between every remaining goal ignoring the need to land to reach surface goals. Unfortunately, the set of unvisited goals is similar amongst the children of a single node and as such the heuristic was not very informative. A series of refinements and related heuristics were explored but none produced notable improvements in convergence owing to the difficulty of developing a globally informed heuristic that remain admissible. Ultimately a radical notion was considered: the difficulty in formulating an informed TSP heuristic stems from the same characteristics that make

the TSP a difficult problem to solve. Specifically, the minimum-cost solution of a TSP is not always readily apparent by inspection. It is only through exploration of the search space that a solution can be assembled and by extension it is likely to require exploration of the search space to formulate a heuristic that is both well-informed and optimistic. While this conjecture may be true, requiring the solution to an embedded search problem in order to solve an outer-loop search problem is not an obvious strategy for improving overall computational efficiency. However, given no obvious alternative we explored the hypothesis that a simplified TSP could be formulated to serve as an admissible heuristic for the full TSP problem.

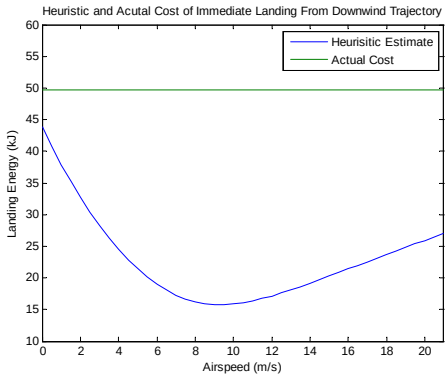
The following simplified TSP was thus formulated as a heuristic: What is the actual minimum geometric distance between all remaining unvisited goals? Unlike the approximation of actual flight path geometric distance through the remaining goals (Eq. 6.33) this new TSP sought the ideal minimum distance. This simplified TSP is symmetric, adheres to the triangle-inequality, and has fixed edge lengths that can be computed in advance. What remains is to determine which of the combinatorial orderings or transitions reaches every remain goal from the current goal in the shortest length. To solve the problem a recursive exhaustive solver was coded in C and applied to the problem of using the simplified TSP as an admissible heuristic for A* search. The TSP solver is in the class of brute-force solvers and the speed of finding a solution is largely attributed to the use of a visited-list, re-using previous calculations, and the simplicity of the problem structure itself. As previously discussed the complexity of a brute force TSP solver scales, for $n$ goals, by the number of available paths, $p$, through all goals. For a goal node the worst case brute force time complexity is $O((n-1)!)$ and for a no-op node it is $O(n!)$. The recursive search is structured to minimize the number of computation required over all paths by proceeding to maximum depth first and working backwards over permutations of each path. It may be possible to obtain additional computational efficiency, at the cost of

memory efficiency, by storing the first calculation of each graph edge to eliminate any remaining duplicate calculations. Now armed with the actual length of the shortest 3D path that connects every remaining goal ($p^*_{min}$) a new heuristic is developed using the cruise airspeed ($V_{max}$), power required for cruise ($\mathcal{P}_{\text{cruise}}$), and the magnitude of the inertial-frame environmental wind speed ($\|\vec{w}_I\|$):

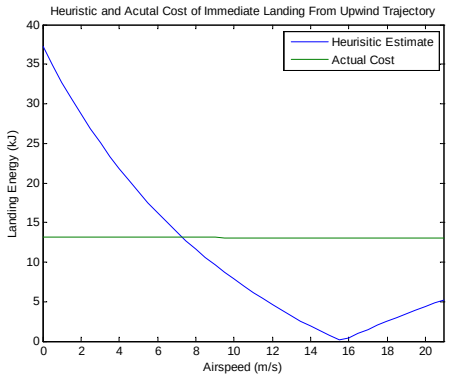
$$\mathbb{H}_{k,i} = \frac{p^*_{min}}{V_{cruise} + \|\vec{w}_I\|}\mathcal{P}_{\text{cruise}} \tag{6.34}$$

The application of this heuristic has been effective. The results of the A* search applied to the "midnight flight" case presented above exactly duplicate the optimal uniform-cost search results (Fig. 6.17) but the A* solution is found in approximately $1/3^{\text{rd}}$ the computational time. This heuristic effectively calculates the amount of energy required to fly the minimum 3D distance through every remaining goal as if the goals were in straight downwind line. Given that cross and upwind flight are less efficient the heuristic will intuitively underestimate the actual cost of flight through any arbitrary set of goals that *are not* aligned in a straight downwind line and will exactly estimate the cost of goals that *are* in the straight downwind line case. Conversely the heuristic will estimate the exact cost for flight in any straight-line sequence in the absence of wind. Furthermore, the heuristic will underestimate takeoff by a larger margin as surface acceleration, liftoff, and climb require significantly more power than cruise over any given distance. The only transition that is not intuitively guaranteed to be underestimated is landing, for which large reserves of kinetic and potential energy, rather than battery energy, are traded for glide distance. In the previously discussed limiting case, estimating the cost over a set of goals in a straight downwind line, the fact that our final-approach constraints require the vehicle to turn into the wind before descending results in higher energy requirements for landing than are estimated by the heuristic, regardless of wind speed (Fig. 6.18a). However, a new limiting case

must now be considered; while every non-landing upwind transition will certainly be underestimated the same is not necessarily true for landing transitions over every possible wind speed. Figure 6.18b shows that for landings with wind speeds less than ~7.3m/s the resulting glide distance will result in the heuristic overestimating the cost to reach the landing site. If the landing is not the last transition (e.g. a takeoff follows landing) or if it is otherwise the case that there are a comparable number of takeoffs for every direct-downwind landing (in wind less that 7.3m/s) the underestimation of takeoff is sufficient to cancel the overestimation of landing. Furthermore, if a sufficient duration of cruise precedes the direct-downwind landing (in wind less than 7.3m/s) the overestimation will again be canceled. Figure 6.19 shows the duration of cruise that must precede the landing to guarantee admissibility as a function of wind speed; for wind speeds below 2m/s the heuristic underestimation of cruise is insufficient to cancel the overestimation of landing. In summary, the proposed heuristic is admissible over all but the following rare pathological cases (all defined only for persistent winds less than ~7.3m/s): (1) Exactly one transition remains and it is a direct-downwind landing, (2) The last transition of a series is a direct-downwind landing not preceded by a sufficient combination of takeoffs and up/crosswind cruise. Furthermore, direct-downwind landings executed in the middle of a plan will also be underestimated but admissibility is ensured by the cost of the following takeoff. As a result the heuristic can be rendered less-informed by downwind landings and extend A* search time. Ultimately, in the worst-case, a sub-optimal plan resulting from an admissibility violation due to a pathological case should still be optimal to the step preceding landing. Empirical evidence suggests that the confluence of these circumstance is rare, or that the impact at the terminal transition is somehow negligible in this search space, as this heuristic has resulted in an optimal search result (in comparison to uniform-cost) for every example tested to date.




(a) Landing from Downwind Trajectory




(b) Landing from Downwind Trajectory

Figure 6.18: Comparison of Actual Cost and Heuristic Estimates of Landing

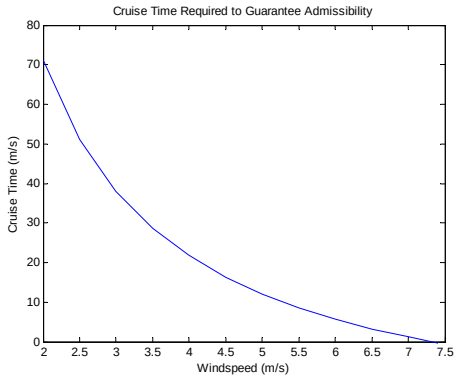
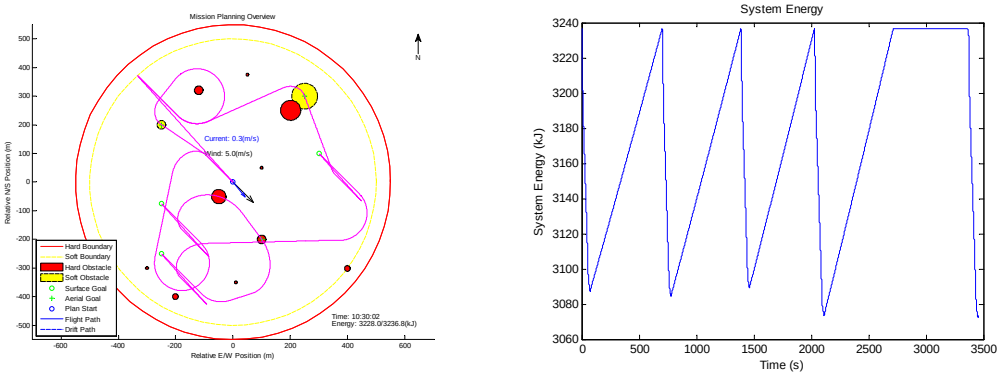



Figure 6.19: Cruise Duration Required to Guarantee Admissibility

### 6.4.3 Midday Missions: Admissible Planning with Uncertain Optimality

Above we established search strategies based on uniform cost and A* algorithms that enable optimal mission plan generation for the non-regenerative energy case. This section describes the more difficult daylight case with the potential for negative transition costs in cases where harvested energy exceeds expended energy. The same utility function as originally applied to the midnight-operation case (Eqs. 6.30-6.31) is applied in this case with benefit and value motivating the planner to select energy-consuming actions. Applying the uniform cost search for a two-hour search horizon exceeds the recursive memory allocation limits of the search code, failing to converge

due to the numerous transitions introduced by the implicit energy-collection incentive encouraging exploration of all paths with negative-cost branches. Reducing the search horizon to one hour adequately contains the exploration space, producing the plan presented in Fig. 6.20. Notably, the resulting plan comes within ~100s of the search horizon. The results in this case are more difficult to interpret than in the non-regenerative case. Again there are instances of electing to drift rather than selecting strictly goal-seeking actions. The final drift clearly extends beyond the point at which the batteries are charged suggesting some benefit was gained by allowing the system to drift downwind at 100% charge before flight. However, the saw-tooth energy budget trend suggests drift segments were nominally selected to recharge the batteries and drive the effective path cost to zero. The ability to drive an arbitrary path cost to zero at any given step is problematic for the planner. At each step where the cost is driven to zero the search can be thought to restart with fewer goals; any previously reached goal will be on the visited list and there would be no mathematically-compelling reason to backtrack beyond the threshold of zero path cost (unless all transitions in the branch following the zero-cost threshold are inadmissible or do not reach all goals). As such, it is the utility function and search-space geometry, and not a sense of global optimality, that determines the sequence of "canceled" goals that precede the final flight. Nevertheless, the search produces a viable plan provided the horizon is sufficiently close.



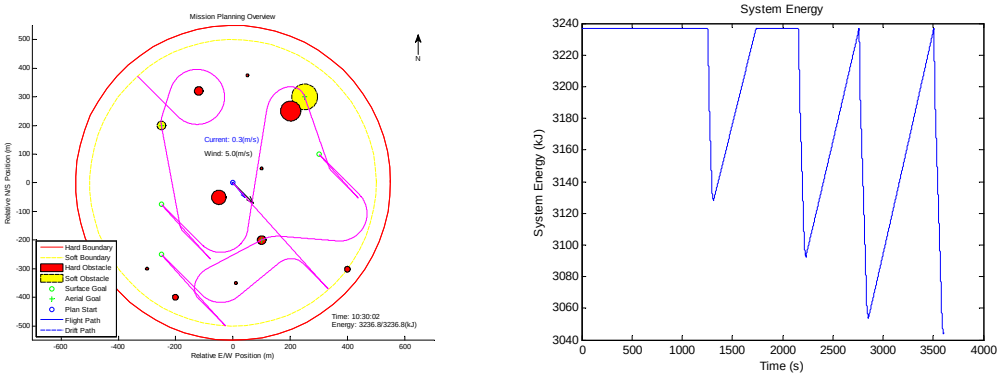

(a) Midday Mission Plan

(b) Midday Energy Budget

Figure 6.20: Midday Uniform Cost Planning

For comparison we subsequently applied the A* heuristic developed in the preceding section. Given that the heuristic estimates positive cost through all goals it is not generally admissible in the solar-regenerative case where costs can be zero or negative. Subsequently we subtracted the maximum system energy from the heuristic to make it always negative. Ultimately it was determined that the results of the search are the same regardless of whether or not the heuristic is given this negative offset. Figure 6.21 presents the plan that results from applying the simplified-TSP heuristic. Notably, sufficient separation from the boundary is available to allow drift for recovery of the terminal cost of both the uniform-cost search and A* results and the difference between the terminal cost of the two searches was only ~30kJ. However, while the uniform-cost search exceeded the memory management capabilities of the software when the search horizon was extended to two hours, the A*/TSP-heuristic search converges (nearly regardless of the horizon) in <10s, or ~30-40x the speed of the best short-horizon uniform-cost search. As expected search algorithms and heuristic metrics defined with respect to strictly positive costs no longer guarantee optimality and/or admissibility with negative transition costs. However, as shown in the above case study, the inadmissible heuristic has still produced a compelling result:

a planning cycle completed in faster-than-expected time that provides an intuitive if suboptimal plan supporting goal achievement.



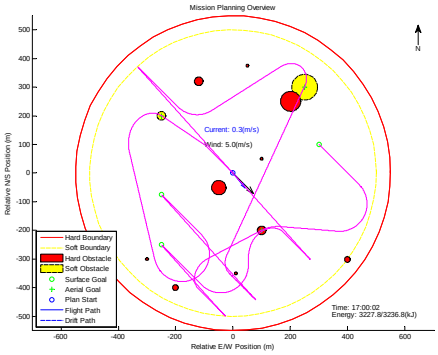

(a) Midday Mission Plan

(b) Midday Energy Budget

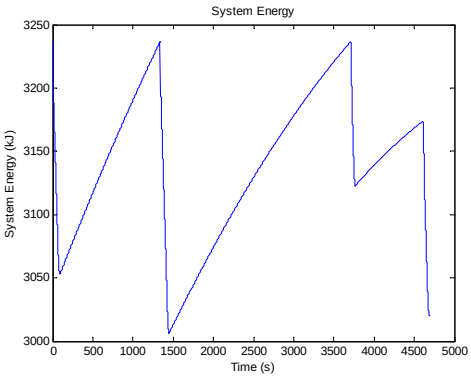
Figure 6.21: Midday A* Planning

### 6.4.4 Dawn/Dusk Missions: Planning Near Minimum Charging Conditions

At this point a set of clear planner capabilities, limitations, and attributes have been presented for the extreme cases of zero solar energy and high-intensity midday solar-regenerative planning. What remains is to consider the transition regions at dawn and dusk. Early searches looking to compare and contrast results in this region revealed that as the sun neared the horizon, but distinctly before nightfall, the A* and uniform-cost search results become indistinguishable. If searches are conducted either earlier in the afternoon or later in the morning (late afternoon in this case) the results of the uniform-cost (Fig. 6.22) and A* (Fig. 6.23) search strategies begin to diverge steadily from one another.

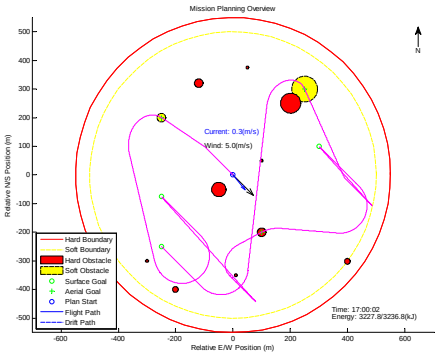



(a) Early Dusk Mission Plan

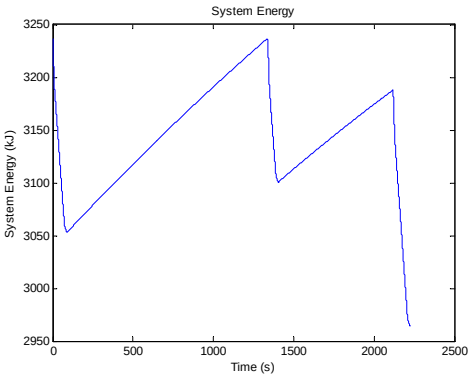



(b) Early Dusk Energy Budget

Figure 6.22: Early Dusk Uniform-Cost Planning




(a) Early Dusk Mission Plan

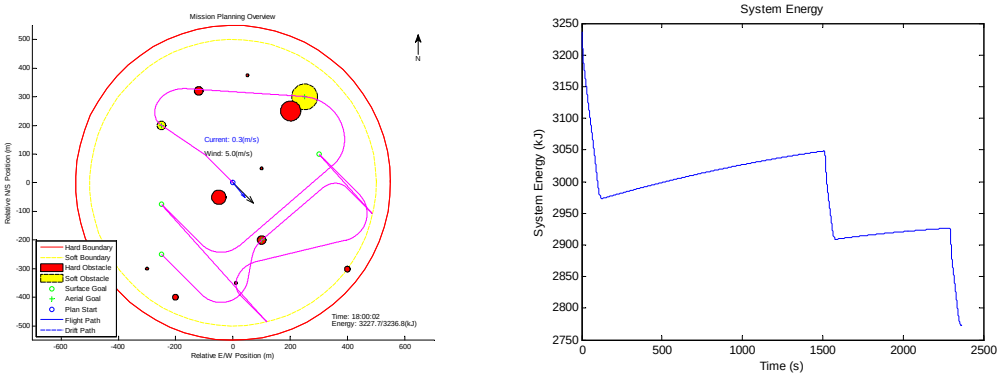



(b) Early Dusk Energy Budget

Figure 6.23: Early Dusk A* Planning

The precise threshold at which the searches begin to produce identical results has proven elusive to define. It is possible that the results do not simply synchronize at some well defined moment. Instead it is thought that there is some varying continuum of energy-density over the transitions from night-to-day/day-to-night over which the uniform-cost search and A* search results grow closer together for depleting energy-density and further apart for increasing energy-density. Ultimately the search results converge at the optimal non-regenerative solution after sunset (or prior to sunrise). Figure 6.24 presents an example of an identical late-day/early-morning results. The

convergence of the search solutions over the course of sunset (or divergence over sunrise) would seem to suggest that there may exist a range of low-solar-energy conditions for which negative path cost may be a non-issue in the determination of true optimality. That is, for reduced energy density in the span of a given plan it becomes less likely that a previously incurred cost can be driven to zero. As such, for a subset of planning conditions with negative edge costs there may still exist a clear transition ordering. Further work is required to clarify this relationship.




(a) Dusk/Dawn Mission Plan

(b) Dusk/Dawn Energy Budget

Figure 6.24: Dusk/Dawn Mission Planning

### 6.4.5 Summary

Three complementary planning strategies have been presented and evaluated. First, a greedy search algorithm was presented that is capable of generating long-term mission plans that are suboptimal but that may provide a valuable forecasting tool to determine if a planner will be able to balance energy over multiple days given the expected wind and solar insolation conditions. Second, we presented uniform cost and A* algorithms with a novel admissible heuristic for "overnight" flight planning where it is essential to optimize energy use. A similar result was observed for day-night transition periods during which the level of solar harvesting is sufficiently low that, while transition costs were not guaranteed to remain positive, the available en-

ergy was insufficient to fully recharge over arbitrary transition cycles. Under these circumstances the path cost cannot be driven arbitrarily to zero and a clearly defined optimal mission ordering can be found. Third, we examined the use of our overnight strategy for application during mid-day where transitions, especially drift segments, had negative cost due to energy harvesting exceeding energy consumption. The high negative path costs during full daylight allow the search to drive the cost of arbitrarily-ordered missions to zero after each mission. Identified solutions cannot be guaranteed optimal and the simplified-TSP heuristic is demonstrably inadmissible. However, suboptimal solutions can still be identified; recall that both search strategies assembled a series of goal-seeking transitions followed by energy recovery transitions such that the maximum terminal cost was no greater than the maximum cost of any single transition (Fig. 6.20-6.21). Further suboptimal solutions are nearly energy neutral if a drift cycle is added beyond the terminal flight, as the search stops at the moment the final goal is reached without consideration of subsequent drift.

# CHAPTER VII

# Conclusions and Future Work

This thesis has described the development and implementation of a flight management system for a first-of-its-kind solar-regenerative seaplane UAS that has been matured and field-tested over two generations of deployed flight vehicles. Research focused on developing and flight-validating a robust autonomous flight planning, guidance, and control scheme, implemented on a concurrently-developed embedded avionics system, to enable self-initiated, fully-autonomous persistent ocean surveillance via sequential drift-fly cycles over designated watch circle regions. Air data sensor failure conditions uncovered during flight testing and development were addressed with a fault-detection/recovery scheme that applied a confidence filter of weighted averages to diagnosed the fitness of signal characteristics and mean signal energy to inform a weighted data-fusion algorithm. The fault tolerance system was shown to: (1) handle multiple failures in non-homogenous sensor networks on two different vehicles, (2) correctly re-integrate recovered sensors and, (3) tolerate full ADS failure (by providing controller-safe airspeed estimates) over the timescale of the average Flying Fish drift-fly cycle. Given the objective of long-term unattended operation, which requires significant steps beyond operator-supervised flight typical of contemporary UAS, a program was undertaken to develop well defined characteristics and solution mechanisms for the Flying Fish mission planning problem. To this end the Flying

Fish mission-planning problem was expanded and a global hybrid vehicle-environment model was constructed from the collected models presented and developed over the course of the thesis. The planning problem was characterized as a particularly challenging form of the Traveling Salesman Path Problem (an already NP-hard/complete problem) which was dubbed the Frequent Flier Salesman Problem. The conditions for which an optimal solution can be guaranteed were characterized and the optimal solution was found for a series of case studies. The solution time was improved by the development of a novel heuristic, itself a TSP problem, that increases the search performance by no less than a factor of 3x in all executed test cases. The admissibility limits of this heuristic subject to a pathological case were explored. Solutions were subsequently explored for daylight conditions, under which energy-optimality becomes poorly defined and for the transitional dusk/dawn cases. In all cases the application of the previously developed TSP-heuristic was shown, despite uncertain heuristic admissibility in daylight, to converge to solutions comparable to more exhaustive searches in a fraction of the search time and to reduce to the optimal solution as the limit of solar energy approaches zero. This research is distinctive in its autonomous seaplane application, its motivation by challenges observed during flight operations, and its use of flight test data to validate implemented solutions.

## 7.1 Major Findings

The major results of this work are summarized below by chapter.

- **Chapter 2:** The development of a complex avionics and customized flight-software systems resulted in the characterization and resolution of a number of unique solar-regenerative seaplane-UAS avionics development challenges including system-integration of extensive disparate electrical subsystems with strict weight and waterproofing requirements, vehicle distributed sensing and

symmetric-redundant power and actuation systems, the development of a custom over-current protection system with both high power handling and low quiescent losses, and mode-specific software data handling for guaranteed control execution times. The resulting system has been shown, in real-world flight testing, to successfully interface and autonomously manage all vehicle subsystems including data collection, command and control, and communication

- **Chapter 3:** Approximate models have been developed for a new class of UAS based on conventional aircraft and watercraft models in conjunction with empirical results from a concurrent flight-testing program. A unique linear drift model was developed from empirical data that provides estimates of the cumulative free drift behavior of a floating seaplane arising from complex underlying interdependent nonlinear vehicle and environmental dynamics .

- **Chapter 4:** Guidance and control strategies developed for a new class of UAS have been shown to achieve fully autonomous flight, from takeoff through landing, over drift-fly cycles executed to persist in a designated watch-circle region subject to environmental disturbances. While the algorithm is itself simple, it is the first of its kind, enabling a novel form of always-on autonomy with self-initiated autonomous flight sequences.

- **Chapter 5:** A fault-tolerance confidence filter system which leverages signal-level fault-rejection/recovery schema to maintain flight operations in adverse environments despite high failure rates of critical (non-homogenous) redundant air-data sensor was validated by flight test data from two different vehicles. A set of evaluation metrics were developed and appropriately combined to accurately and robustly assess the fitness of failure-prone pressure-based sensors to support data-fusion for accurate and feedback-safe flight speed determination.

- **Chapter 6:** Physics-based trajectory planning, vehicle performance, and mod-

els of energy harvesting, usage, and storage were integrated into a discrete search tool for the determination of energy-optimal paths subject to vehicle performance and solar energy recovery dynamics. While individual models are adapted from the literature, the combination of vehicle performance, environment, and solar energy harvesting and usage models have never before been integrated into a system that optimizes energy over multiple flights with surface and airborne targets visited by the same platform. The Flying Fish planning problem was found to be of the form of an asymmetric, non-metric, negative-cost NP-hard Traveling Salesman Problem (TSP). An innovative informed heuristic was devised based on the concept of solving a simplified TSP problem as the inner-loop of the top-level TSP planner and was demonstrated, over a series of case studies, to substantially reduce search time.

## 7.2 Future Work

The possible applications of an autonomous unmanned seaplane are far-ranging but the ultimate utility of the system will not be realized until it can achieve truly unattended operation. The immediate milestones on the way to this goal include maximizing system performance and robustness and field-validating all system tools. Moving forward the system would also require additional sensing capabilities and mission payloads. Finally, to maximize the value of unattended operations the mission planner must be endowed with both greater amounts of information and the ability to utilize that information to formulate not just situation-aware plans but to also make mission-enhancing decisions.

First and foremost, the flight testing program needs to continue to aid the development of dynamic models and improve trim determinations and controller tuning. This will be important for the execution of accurate and efficient flight trajectories,

as determined by the mission planner, as well as for safety and repeatability in takeoff and landing sequences. Moreover the GNC systems must be tested over a wider range of environmental conditions for longer periods of time to ensure long-term all-weather survivability. Also in the near term, the addition of an extended Kalman filter (EKF) to the failsafe ADS fault-mitigation system would provide expanded estimation and data fusion capabilities as well as increased tolerance to full ADS failure. Furthermore long-term operational tests of the the ADS fault-mitigation system are required to validate real-time failure rejection under the full range of environmental conditions. Finally, while significant efficiency improvements have already been realized, the energy-aware planning utility needs to be further optimized for an embedded computer system and validated via live field deployments.

Amongst the future challenges for Flying Fish is the development and implementation of sensors and algorithms for obstacle detection and guaranteed collision avoidance. The planning mechanism can only reasonably estimate non-conflicting paths given perfect obstacle information. The detection, inspection, and classification of environmental obstacles and the ability to quickly respond to unexpected environmental conditions are likely to be a significant challenge. To solve this problem one might implement a machine vision systems or miniature synthetic aperture radar and work to refine object detection mechanisms for the classes of objects and environmental features that are most likely to be encountered by Flying Fish, specifically: land masses, biologicals, boats/ships, and buoys that must be detected against a non-uniform water surface. There are a number of variations on the presented search strategies as well as alternatives for cost and valuation that could be explored to optimize the mission planning strategy under a wider range of environmental conditions. It might also be beneficial to integrate a weather-satellite down-link that would provide the planning utility and ADS system with critical insight to local and distant weather patterns. The implementation of additional science/surveillance payloads would also

be an important step as this would broaden the applications of Flying Fish. Furthermore, while the interface, management, and deployment of these additional sensor systems may (or may not) be straightforward from an avionics standpoint the most interesting challenge might be to leverage the sensor's data-stream to dynamically update flight and mission plans to improve science/surveillance returns.

# APPENDIX A

# Custom Electronics Development:

## A.1 Multi-Protocol Device Interface Board with 12-bit Analog/Digital Converter

The very-low power usage of the Gumstix Overo line is achieved in part through the use of very low on-board voltage levels (1.8V, LVCMOS). While this is certainly beneficial from an efficiency standpoint it also means that the I/O voltages of the Overo are largely incompatible with industry standard logic levels making direct interface to devices with standards-based communication protocols impossible. The low voltage levels have the additional effect of limiting the input range of the onboard Analog/Digital conversion to 2.5V.

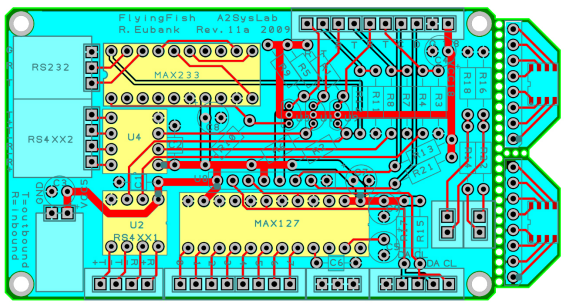



Figure A.1: Logic-Level Interface & ADC Circuit Board

Flying Fish requires the Overo to interface with the: Solar Maximum-Power-Point Tracker(s) (RS-422/485), MIDG-IIC Inertial Navigation System (RS-422), high-speed multi-drop 2-wire sensor bus (i2c), servo switch controller (RS-232), and a wireless serial modem (RS-232/USB). Flying Fish also needs to sample analog signals up to 21V. While voltage dividers may be used to reduce the analog signals to within the range of the Overo A/D the potential accuracy reduction from scaling signals by an order of magnitude in generally undesirable, necessitating an alternate A/D system.

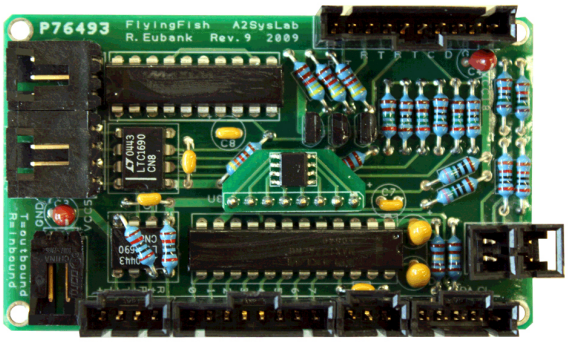
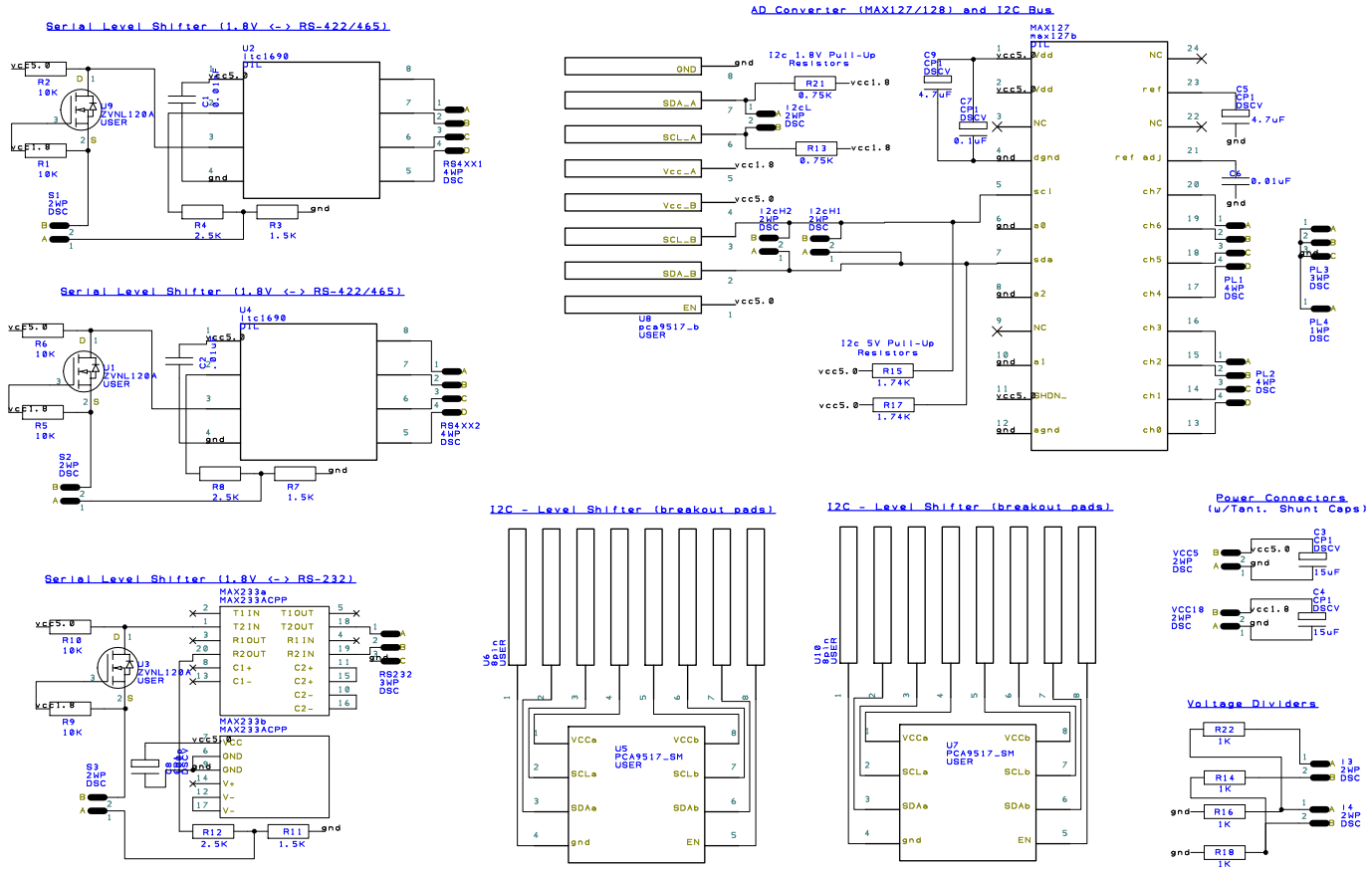



Figure A.2: Fully Assembled 3rd Generation Interface-ADC Board

In response to these requirements a custom printed circuit board (PCB) was produced that converted each of the Overo's three 2-wire serial interfaces from Overo-native voltage (0-1.8V) into TTL (0-5V), RS-232 (+/-(5-12)V), and RS-422 (+/-5V, differential) interfaces. The Overo i2c bus was given bi-directional buffered conversion on the both the clock and data lines along with the appropriate pull-up resistors (not provided onboard the Overo). Finally an 8-Channel 12-bit A/D was added to the i2c bus onboard the PCB. The circuit (Fig. A.1-A.2) was designed for robust operation with appropriate noise isolation, shunt capacitors, a shielding ground plane, and several utility connections, including: two utility voltage dividers, additional grounding, dual i2c connections, and PCB-based surface-mount to through-hole component break-out boards. The circuit diagram of this board is presented on the following page.




| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| E | D | C | B | A | Drawn<br>RDE | Check<br>RDE. CC | Projection<br>Do Not Scale | Attributes:<br>(2x) 1.8V 2-wire serial to RS-422<br>(1x) 1.8V 2-wire serial to RS-232<br>Bi-directional level shifted I2c<br>8-channel 12-bit A/D Conversion<br>(2x) Utility voltage dividers |
| Drn | Drn | Drn | Drn | Project<br>Flying Fish - Central Avionics | | | Client | |
| Chk | Chk | Chk | Chk | Title<br>Overo Logic-Level Shifting<br>I2c Analog/Digital Interface board | | | Filename<br>overo_logic11.sch | Drawing No.<br>11.a | Sheet 1<br>of 1 |

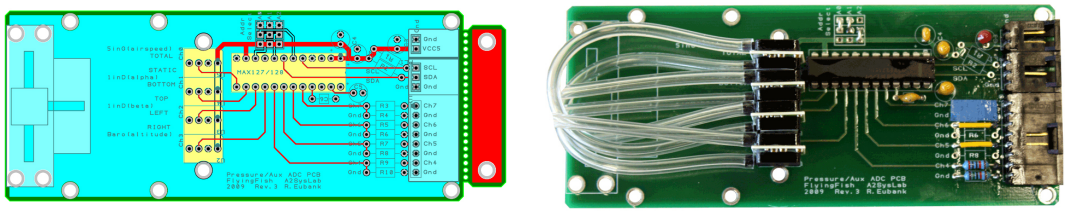
## A.2 Remote Analog/Digital Conversion Nodes

Due to the distribution of sensors and power systems in the Phase II Flying Fish it was desirable to develop remote sensing boards that could be co-located with critical analog systems. By minimizing the travel path of the wires carrying analog signals it was hoped that the induced EM noise on those wires, and subsequently sampled-signal degradation, would be minimized. This is particularly important in the vicinity of the high power, highly-variable, EM fields associated with the propulsion systems, solar optimization circuitry, and RF communication systems.

It was also desirable to place pressure transducers as close as possible to the pitot-static/5-hole probes (located on the vertical stabilizers) for two reasons: 1) to minimize the required length of heavy (relative to 24-26Ga wire) polymer tubing to plumb the pressure transducers, and 2) to minimize any sensing error that might come from extended pressure tubing runs or the need to connect/disconnect tubing regularly to service the avionics. The pressure transducers could not be easily placed on the tail (the closest possible installation) because doing so would further degrade the already tail-heavy vehicle weight-and-balance and because the low cross-section/volume of the the tail provided no ready location to house the circuit/sensors nor any mechanism to water-proofing those systems. The vertical wing stanchions however (which bridge the space from the wing to the floats) have sufficient volume/area to house the circuits and a water-proof aerodynamic faring.

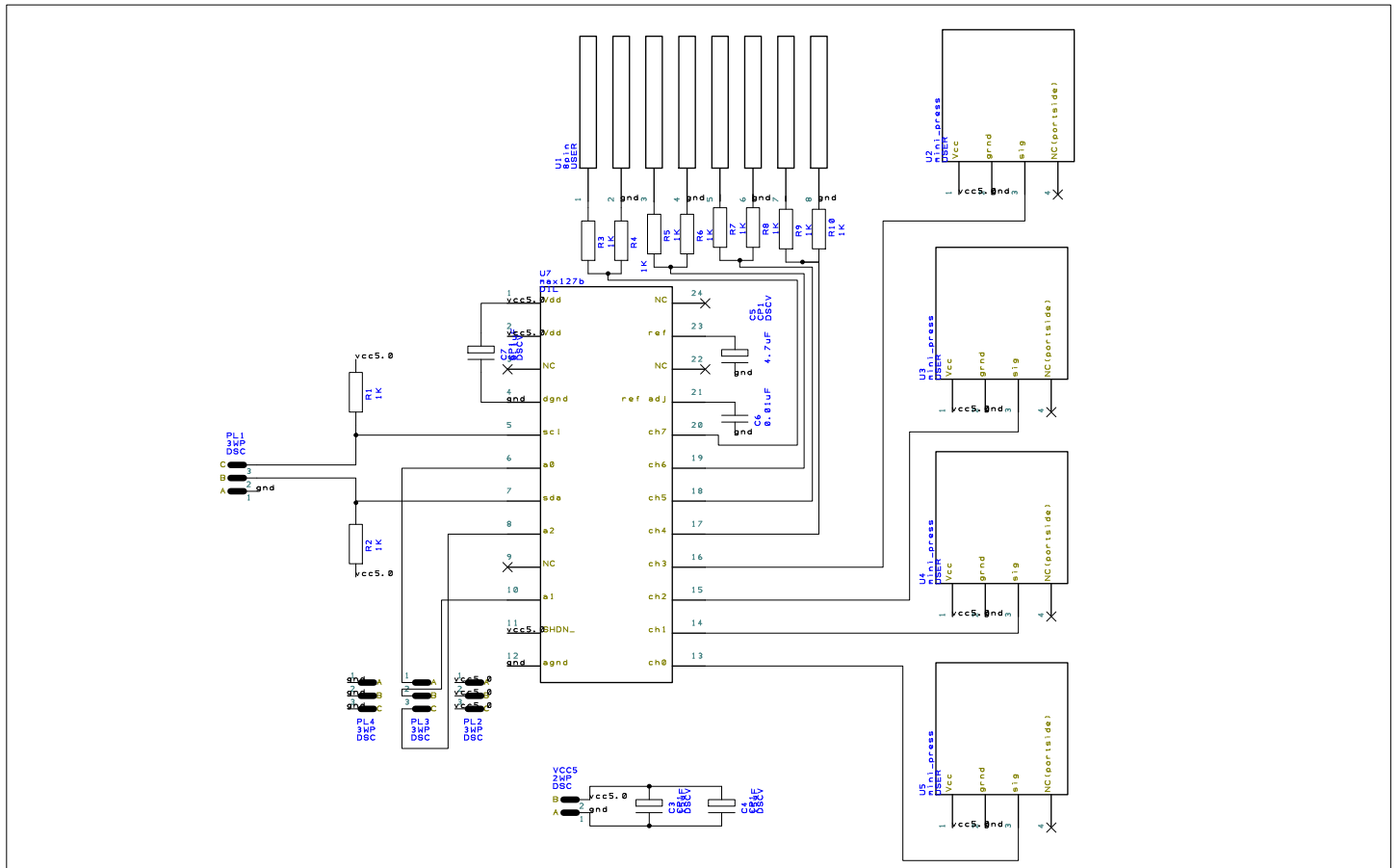
Ultimately it was decided that a PCB should be developed to provide remote ADC in each of the vertical stanchions and that the miniature pressure sensors should be integrated to this board. The PCB (Fig. ) was designed to act as its own structural support system, cantilevered off of a water-proof PVC hatch, carrying the bank of miniature pressure transducers (with PCB-integrated structural support) into the body of the vertical stanchion to the wiring interfaces within the stanchion. The

board also housed the pressure manifolds and static-pressure junction for barometric altimetry. The circuit diagram of the remote ADC node is presented on the following page.



(a) Circuit Board Design (b) Assembled Circuit Board

Figure A.3: Remote ADC With Integral Pressure Transducers




| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| E | D | C | B | A | Drawn<br>RDE | Check<br>RDE. CC. KS | Projection<br>Do Not Scale | Attributes:<br>8-channel 12-bit A/D Conversion<br>(8x) Utility voltage dividers<br>(4x) Miniature Pressure Transducers<br>w/integrated support structure<br>Structurally Integral Design |
| Drn | Drn | Drn | Drn | Project<br>Flying Fish - Central Avionics | | | Client | |
| Chk | Chk | Chk | Chk | Title<br>Remote A/D Node<br>Structurally-Integral ADC Board | | | Filename<br>remote_adc3.sch | Drawing No.<br>3 |
| | | | | | | | | Sheet 1<br>of 1 |

## A.3 High-Power Low-Loss Current Limiter

When, after the construction of the second flight vehicle, it was decided that an auxiliary propulsion system would be required for takeoff under adverse conditions the engineering team had to develop and implement an interface for the high-energy propulsion system to a power system that was not designed to handle the additional loading. As discussed in the vehicle development section II high-energy brushless motor systems have fairly strenuous electrical and physical requirements necessitating both high power delivery capabilities and physically proximal supply source to mitigate electro-magnetic field (EMF) effects. Ultimately it was decided that primary power delivery from the existing symmetric main battery banks was too problematic due to the high likelihood of EMF problems (both from supply power "ringing" and from high-power delivery lines passing near sensitive electrical signals) and the high weight penalty and structural impact of adding the requisite high-power wiring. Under these conditions the system required the addition of an auxiliary battery bank below the avionics computer, where there was only sufficient volume for a single battery pack (5.4Ah, 18.5V nominal). The auxiliary bank was wired to directly supply the motor controller which was located within the auxiliary motor mounting structure. The remaining critical issue was to address power delivery charge power to the auxiliary battery system as only the main bank were directly charged from the solar recharge system. The initial proposal was to re-direct a portion of the solar power to the auxiliary system but this idea was discarded as the small capacity of the auxiliary system would ultimately squander solar energy once it was recharged. The final solution was to connect supply energy from both of the main battery banks to the axillary system using the existing avionics power supply wires. This wiring plan, however, was not without challenges. Under the immense load of the auxiliary propulsion system (in excess of 60A at 20V) the single auxiliary battery pack would

always have a supply potential significantly lower than that of the main battery banks. Under these conditions the main battery banks would technically attempt to supply sufficient current to equalize the potential of all sources in the system, and hence would supply the bulk of the supply current, burning out the small gauge avionics supply wires.

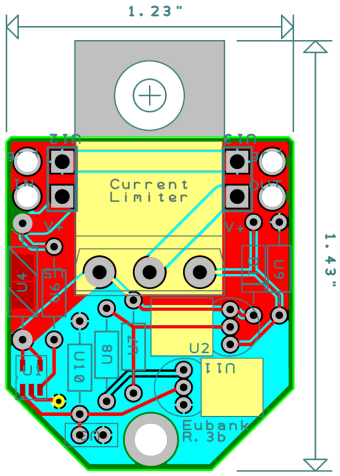
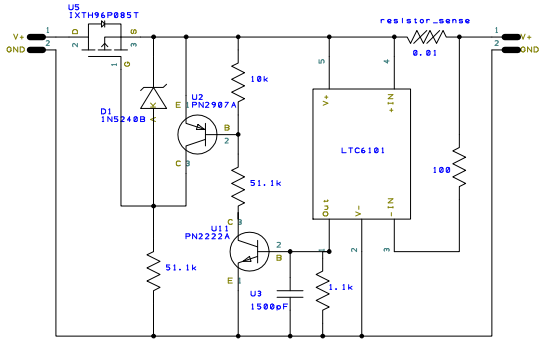



Figure A.4: Current Limiting Circuit Design




| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| E | D | C | B | A | Drawn<br>20 Apr 2010<br>Ryan Eubank | Check<br>27 Apr 2010<br>Curt Cooper | Projection<br>Do Not Scale | | |
| Drn | Drn | Drn | Drn | Project<br>FlyingFish / SolarBubbles | | | Client | | |
| Chk | Chk | Chk | Chk | Title<br>MOSFET High-Current Limiter | | | Filename<br>current_limit_3b.sch | Drawing No.<br>3b | Sheet<br>1 of 1 |

# BIBLIOGRAPHY


[1] S. Nicolaou, *Flying Boats & Seaplanes: A History from 1905*. Bideford: Bay View Books Ltd., 1996.

[2] B. D. Barth, *The End of an Era in Naval Seaplane Aviation: The Martin P5M Marlin*. Pacific Aero Press, 1994.

[3] O. of the Secretary of Defense, "Unmanned Aircraft Systems Roadmap, 2005-2030," tech. rep., United States, Department of Defense, Washington D.C., 2005.

[4] G. Pisanich and S. Morris, "Fielding an amphibious UAV: development, results, and lessons learned," in *Digital Avionics Systems Conference, 2002. Proceedings. The 21st*, vol. 2, p. 8C4, IEEE, 2002.

[5] J. K. Brown and B. Gilchrist, "Solar Harvesting System Considerations for the Persistent Ocean Surveillance (POS) Flying Fish Project," tech. rep., University of Michigan, Space Physics Research Laboratory, Oct. 2007.

[6] R. Eubank, E. Atkins, and S. Ogura, "Fault Detection and Fail-Safe Operation with a Multiple-Redundancy Air-Data System," in *Guidance, Navigation, and Controls Conference*, (Toronto, ON), AIAA, American Institute of Aeronautics and Astronautics, 02-06 Aug. 2010.

[7] S. Liden, "The Evolution of Flight Management Systems," in *Digital Avionics Systems Conference, Proceedings*, pp. 157–169, AIAA/IEEE, 1994.

[8] S. Fishbein, *Flight Management Systems: The Evolution of Avionics and Navigation Technology*. CT: The Smithsonian Institution, Praeger, 1995.

[9] R. Andoga, L. Fozo, L. Madarasz, and J. Povazan, "FADEC Control System for MPM 20 Engine," in *Applied Machine Intelligence and Informatics, 2009. SAMI 2009. 7th International Symposium on*, pp. 103–108, Jan. 2009.

[10] D. Briere, C. Favre, and P. Traverse, "A Family of Fault-Tolerant Systems: Electrical Flight Controls, from Airbus A320/330/340 to Future Military Transport Aircraft," *Microprocessors and Microsystems*, vol. 19, pp. 75–82, Mar. 1995.

[11] Y. C. Yeh, "Triple-Triple Redundant 777 Primary Flight Computer," in *Aerospace Applications Conference, Proceedings, IEEE*, vol. 1, pp. 293–307, IEEE, 2002.

[12] E. Atkins, R. Eubank, and A. Klesh, "A Reconfigurable Flight Management System for Small-Scale Unmanned Air Systems," in *AIAA Infotech@Aerospace Conference, Proceedings*, (Seattle, WA), AIAA, American Institute of Aeronautics and Astronautics, 06-09 Apr. 2009.

[13] D. Yeo, J. Henderson, and E. Atkins, "An Aerodynamic Data System for Small Hovering Fixed-Wing UAS," in *Guidance, Navigation, and Control Conference*, (Chicago, IL), American Institute of Aeronautics and Astronautics, 1801 Alexander Bell Dr., Suite 500 Reston VA 20191-4344 USA,, Aug. 2009.

[14] C. Cesnik, P. Senatore, W. Su, E. Atkins, C. Shearer, and N. Pitcher, "X-HALE: A Very Flexible UAV for Nonlinear Aeroelastic Tests," in *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Adaptive Structures Conference, Proceedings*, (Orlando, FL), AIAA/ASME/ASCE/AHS/ASC, 12-15 Apr. 2010. AIAA-2010-2715.

[15] R. Eubank, E. Atkins, and D. Macy, "Autonomous Guidance and Control of the Flying Fish Ocean Surveillance Platform," in *AIAA Infotech@Aerospace Conference, Proceedings*, (Seattle, WA), AIAA, American Institute of Aeronautics and Astronautics, 06-09 Apr. 2009.

[16] N. H. McClamroch, *Steady Aircraft Flight and Performance.* Princeton Univ. Press, 2011.

[17] W. Phillips, *Mechanics of Flight.* John Wiley & Sons Inc, 2004.

[18] A. Klesh, *Optimal Exploration Systems.* PhD thesis, Univ. of Michigan, Dept. of Aerospace Engineering, 2009.

[19] L. Dubins, "On curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[20] R. Dean and R. Dalrymple, *Water Wave Mechanics for Engineers and Scientists*, vol. 2. World Scientific Pub. Co. Inc., 1991.

[21] D. Savitsky and J. Koelbel, "Sea Keeping Considerations in Design and Operation of Hard Chine Planing Hulls," *Naval Architect*, vol. 2, pp. 55–59, 1979.

[22] B. N. Pamadi, *Performance, Stability, Dynamics, and Control of Airplanes.* Reston, VA: AIAA, 2 ed., 2004.

[23] R. Beard, D. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. McLain, and M. Goodrich, "Autonomous Vehicle Technologies for Small Fixed Wing UAVs," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 2, no. 1, pp. 92–108, 2005.

[24] D. Macy, R. Eubank, E. Atkins, L. Bernal, P. Washabaugh, G. Meadows, N. Wild, D. Smith, and H. Van Sumeren, "Flying Fish: A Persistent Ocean Surveillance Buoy with Autonomous Aerial Repositioning," in *AUVSI North America Conference*, AUVSI, Jun. 2008.

[25] J. Ziegler and N. Nichols, "Optimum Settings for Automatic Controllers," *Transactions of the ASME*, vol. 5, no. 11, pp. 759–768, 1942.

[26] V. Klein and E. Morelli, *Aircraft System Identification: Theory and Practice*. American Institute of Aeronautics and Astronautics, 2006.

[27] N. Aeronautics and U. S. Space Administration, "Aviation Safety Reporting System (ASRS) Database Incident Report Summaries," Jan. 2010.

[28] U. S. National Transportation Safety Board, "Aviation accident database investigation reports," Jan. 2010.

[29] U. S. Federal Aviation Administration, "Aviation safety information analysis and sharing: Accident/incident data system (aids) incident data records," Jan. 2010.

[30] F. S. Foundation, "Aviation safety network aviation safety database accident descriptions," 18 Jan. 2010.

[31] M. Coldsnow, "Alternative Methods to Standby Gain Scheduling Following Air Data System Failure," Master's thesis, Air Force Inst of Technology, Wright-Patterson AFB, Sep. 2009.

[32] S. A. McLaren, "Velocity Estimate Following Air Data System Failure," Master's thesis, Air Force Inst of Technology, Wright-Patterson AFB, Mar. 2008. Accession Number : ADA483268.

[33] O. o. I. Bureau d'Enquetes et d'Analyses (BEA and F. Analysis, "Interim Report on the Accident on 1st June 2009 to the Airbus A330-203 Registered F-GZCP Operated by Air France Flight AF 447 Rio de Janeiro - Paris," Preliminary f-cp090601ae, BEA, NTSB, CENIPA, AAIB, BFU, Jul. 2009.

[34] O. o. I. Bureau d'Enquetes et d'Analyses (BEA and F. Analysis, "Interim Report No. 2 on the Accident on 1st June 2009 to the Airbus A330-203 Registered F-GZCP Operated by Air France Flight AF 447 Rio de Janeiro - Paris," Tech. Rep. f-cp090601ae2, BEA, NTSB, CENIPA, AAIB, BFU, Dec. 2009.

[35] C. W. Johnson, "The Dangers of Interaction with Modular and Self-Healing Avionics Applications: Redundancy Considered Harmful," in *27th International System Safety Conference, Proceedings*, System Safety Society, Aug. 2009.

[36] G. Meadows, E. Atkins, P. Washabaugh, L. Meadows, L. Bernal, B. Gilchrist, D. Smith, H. VanSumeren, D. Macy, R. Eubank, B. Smith, and J. Brown, "The Flying Fish Persistent Ocean Surveillance Platform," in *AIAA Unmanned Unlimited Conference, Proceedings*, (Seattle, WA), AIAA, American Institute of Aeronautics and Astronautics, 06-09 Apr. 2009.

[37] D. Houck and L. Atlas, "Air Data Sensor Failure Detection," in *17th Digital Avionics Systems Conference, Proceedings*, vol. 1, (Bellevue, WA), p. D17, AIAA/IEEE/SAE, 31 Oct. - 07 Nov. 1998.

[38] T. Rohloff, S. Whitmore, and I. Catton, "Fault-Tolerant Neural Network Algorithm for Flush Air Data Sensing," *Journal of Aircraft*, vol. 36, no. 3, pp. 541–549, 1999.

[39] A. Calia, V. Poggi, and F. Schettini, "Air Data Failure Management in a Full-Authority Fly-by-Wire Control System," in *Computer Aided Control System Design, International Conference on Control Applications*, pp. 3277–3281, IEEE, Oct. 2006.

[40] C. H. Westhelle, "X-38 Backup Air Data System (AeroDAD)," in *AIAA Aerospace Sciences Meeting & Exhibit, 40 th*, (Reno, NV), AIAA, Jan. 2002.

[41] F. M. O. W. Colgren, R., "A Proposed System Architecture for Estimation of Angle-Of-Attack and Sideslip Angle," in *AIAA Guidance Navigation and Control Conference*, pp. 743–750, Aug. 1999.

[42] R. Kornfeld, R. Hansman, and J. Deyst, *The Impact of GPS Velocity Based Flight Control on Flight Instrumentation Architecture*. PhD thesis, Massachusetts Institute of Technology, 1999. Report Number: ICAT-99-5.

[43] J. Park and K. Ro, "A Prototype Design, Test and Evaluation of a Small Unmanned Aerial Vehicle for Short-range Operations," in *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, (Chicago, IL), pp. 1–8, AIAA, Sep. 2004.

[44] A. Rodriguez, E. Andersen, J. Bradley, and C. Taylor, "Wind Estimation Using an Optical Flow Sensor on a Miniature Air Vehicle," in *AIAA Conference on Guidance, Navigation, and Control*, 2007.

[45] M. Jafarizadeh, R. Hassannejad, M. Ettefagh, and S. Chitsaz, "Asynchronous Input Gear Damage Diagnosis Using Time Averaging and Wavelet Filtering," *Mechanical Systems and Signal Processing*, vol. 22, no. 1, pp. 172–201, 2008. Amsterdam, Elsevier.

[46] P. Chethan, "Blind Fault Detection Using Spectral Signatures," Master's thesis, Louisiana State University, 2003.

[47] J. Jiang, J. Yang, Y. Lin, C. Liu, and J. Ma, "An Adaptive PMU Based Fault Detection/Location Technique for Transmission Lines. I. Theory and Algorithms," *Power Delivery, IEEE Transactions on*, vol. 15, pp. 486–493, Apr. 2000.

[48] A. Willsky, "A Survey of Design Methods for Failure Detection in Dynamic Systems," *Automatica*, vol. 12, no. 6, pp. 601–611, 1976.

[49] Y. Liu, Y. Shen, and H. Hu, "A New Method for Sensor Fault Detection, Isolation and Accommodation," in *Instrumentation and Measurement Technology Conference, Proceedings of the 16th IEEE*, vol. 1, pp. 488–492, IEEE, May 2002.

[50] M. Omana and J. Taylor, "Fault Detection and Isolation Using the Generalized Parity Vector Technique in the Absence of a Mathematical Model," in *IEEE Conference on Control Applications*, (Singapore), pp. 1–3, IEEE, Oct. 2007.

[51] L. Klein, "A Boolean Algebra Approach to Multiple Sensor Voting Fusion," *IEEE transactions on Aerospace and Electronic systems*, vol. 29, pp. 317–327, Apr. 1993.

[52] L. Alvergue, X. Li, and J. Aravena, "Overinstrumented Systems," in *Region 5 Technical Conference, 2007 IEEE*, pp. 326–330, IEEE, Apr. 2007.

[53] F. Caliskan and C. Hajiyev, "Aircraft Sensor Fault Diagnosis Based on Kalman Filter Innovation Sequence," in *Decision and Control, Proceedings of the 37th IEEE Conference on*, vol. 2, pp. 1313–1314, IEEE, Dec. 1998.

[54] V. Barnett and T. Lewis, *Outliers in Statistical Data*. New York, NY: John Wiley & Sons, 3rd ed., 1994.

[55] D. Hall and J. Llinas, "An Introduction to Multisensor Data Fusion," *Proceedings of the IEEE*, vol. 85, pp. 6–23, Jan. 1997.

[56] I. Bloch, "Information Combination Operators for Data Fusion: A Comparative Review with Classification," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 26, no. 1, pp. 52–67, 1996.

[57] A. A. Thompson III, "Data Fusion for Least Squares," Tech. Rep. BRL-TR-3303, U. S. Army Laboratory Command, Army Ballistic Research Lab, Aberdeen Proving Ground, MD, Dec. 1991.

[58] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[59] G. Smith, S. Schmidt, and L. McGee, *Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity on Board a Circumlunar Vehicle*. National Aeronautics and Space Administration, 1962.

[60] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," *Design*, vol. 7, no. 1, pp. 1–16, 2001.

[61] J. Michalsky, "The Astronomical Almanac's Algorithm for Approximate Solar Position (1950-2050)," *Solar Energy*, vol. 40, no. 3, pp. 227–235, 1988.

[62] I. Reda and A. Andreas, "Solar Position Algorithm for Solar Radiation Applications," Tech. Rep. 5, National Renewable Energy Laboratory, Golden, CO, Jan. 2008. NREL/TP-560-34302.

[63] R. Perez, P. Ineichen, R. Seals, J. Michalsky, and R. Stewart, "Modeling Daylight Availability and Irradiance Components from Direct and Global Irradiance," *Solar Energy*, vol. 44, no. 5, pp. 271–289, 1990.

[64] J. Spencer, "Fourier Series Representation of the Position of the Sun," *Search*, vol. 2, no. 5, p. 172, 1971.

[65] M. Iqbal, *An Introduction to Solar Radiation*. Academic Press, Orlando, FL, 1983.

[66] J. Zimmerman, "Sun-Pointing Programs and Their Accuracy," *NASA STI/Recon Technical Report N*, vol. 81, p. 30643, 1981.

[67] E. Lorenz, "Empirical Orthogonal Functions and Statistical Weather Prediction," Tech. Rep. 1, Massachusetts Institute of Technology and Air Force Cambridge Research Center (US) Geophysics Research Directorate, Dec. 1956.

[68] A. Lorenc, "Analysis Methods for Numerical Weather Prediction," *Quarterly Journal of the Royal Meteorological Society*, vol. 112, no. 474, pp. 1177–1194, 1986.

[69] L. Richardson, *Weather Prediction by Numerical Process*. Cambridge Univ Pr, 2007.

[70] M. Xue, D. Wang, J. Gao, K. Brewster, and K. Droegemeier, "The Advanced Regional Prediction System (ARPS), Storm-Scale Numerical Weather Prediction and Sata Assimilation," *Meteorology and Atmospheric Physics*, vol. 82, no. 1, pp. 139–170, 2003.

[71] V. Johnson, A. Pesaran, and T. Sack, *Temperature-Dependent Battery Models for High-Power Lithium-Ion Batteries*. National Renewable Energy Laboratory, 2001.

[72] R. Eubank and E. Atkins, "Energy-Aware Flight Management and Control of the FlyingFish Solar-Powered Autonomous Seaplane," in *Unmanned Systems North America, AUVSI*, (Denver, CO), AUVSI, 24-27 Aug. 2010.

[73] M. Chen and G. Rincon-Mora, "Accurate Electrical Battery Model Aapable of Predicting Runtime and I-V Performance," *Energy Conversion, IEEE Transactions on*, vol. 21, pp. 504–511, Jun. 2006.

[74] L. Gao, S. Liu, and R. Dougal, "Dynamic lithium-ion Battery Model for System Simulation," *IEEE transactions on components and packaging technologies*, vol. 25, no. 3, pp. 495–505, 2002.

[75] V. Rao, G. Singhal, A. Kumar, and N. Navet, "Battery Model for Embedded Systems," in *VLSI Design, 2005. 18th International Conference on*, pp. 105–110, IEEE, 3-7 Jan. 2005.

[76] R. Kroeze and P. Krein, "Electrical Battery Model for use in Dynamic Electric Vehicle Simulations," in *Power Electronics Specialists Conference, 2008. PESC 2008. IEEE*, pp. 1336–1342, IEEE, 2008.

[77] R. Carnie, R. Walker, and P. Corke, "Image Processing Algorithms for UAV "Sense and Avoid"," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 2848–2853, May. 2006.

[78] C. Tomlin, J. Lygeros, and S. Shankar Sastry, "A Game Theoretic Approach to Controller Design for Hybrid Systems ," *Proceedings of the IEEE*, vol. 88, pp. 949–970, Jul. 2000.

[79] A. Richards and J. How, "Aircraft Trajectory Planning with Collision Avoidance Using Mixed Integer Linear Programming," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 3, pp. 1936–1941, 2002.

[80] F. J. M. Campo, *The Collision Avoidance Problem: Methods and Algorithms.* PhD thesis, Universidad Rey Juan Carlos, 2010.

[81] T. Williamson and N. Spencer, "Development and operation of the Traffic Alert and Collision Avoidance System (TCAS)," *Proceedings of the IEEE*, vol. 77, pp. 1735 –1744, Nov. 1989.

[82] R. Boisvert and V. Orlando, "ADS-Mode S System Overview," in *Digital Avionics Systems Conference, 1993. 12th DASC., AIAA/IEEE*, pp. 104–109, Oct. 1993.

[83] V. Orlando and P. Drouilhet, *Mode S Beacon System: Functional Description.* United States Federal Aviation Administration, Systems Research and Development Service and Lincoln Laboratory, 1983.

[84] D. Johnson and L. McGeoch, "The Traveling Salesman Problem: A Case Study in Local Optimization," *Local search in combinatorial optimization*, vol. 1, pp. 215–310, 1997.

[85] J. Kruskal, "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[86] M. Dorigo and L. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," *Evolutionary Computation, IEEE Transactions on*, vol. 1, pp. 53–66, Apr. 1997.

[87] M. Held and R. Karp, "A Dynamic Programming Approach to Sequencing Problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 10, no. 1, pp. 196–210, 1962.

[88] R. Karp, "Dynamic Programming Meets the Principle of Inclusion and Exclusion," *Operations Research Letters*, vol. 1, no. 2, pp. 49–51, 1982.

[89] R. Jonker and T. Volgenant, "Transforming Asymmetric into Symmetric Traveling Salesman Problems," *Operations Research Letters*, vol. 2, no. 4, pp. 161–163, 1983.

[90] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 ed., 2009.

[91] S. Russell, P. Norvig, J. Canny, J. Malik, and D. Edwards, *Artificial Intelligence: A Modern Approach*, vol. 74. Prentice hall Englewood Cliffs, NJ, 1995.

[92] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, pp. 100–107, Jul. 1968.