Implementation of an Air Supply Unit Control Scheme for the UC2AV (Unmanned Circulation Control Aerial Vehicle)

Cameron Rosen

University of Denver

Follow this and additional works at: https://digitalcommons.du.edu/etd

Part of the Artificial Intelligence and Robotics Commons, and the Electrical and Computer Engineering Commons

Recommended Citation

This Thesis is brought to you for free and open access by the Graduate Studies at Digital Commons @ DU. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons @ DU. For more information, please contact jennifer.cox@du.edu,dig-commons@du.edu.
Implementation of an Air Supply Unit Control Scheme for the UC²AV (Unmanned Circulation Control Aerial Vehicle)

A Thesis

Presented to

the Faculty of the Daniel Felix Ritchie School of Engineering and Computer Science

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Cameron Rosén

November 2016

Advisors: Kimon P. Valavanis, Ph.D. & Matthew J. Rutherford, Ph.D.
Abstract

The expanded prevalence of Unmanned Aerial Vehicles (UAVs) in recent years has created many opportunities to research novel applications for their use, enabled by the reduced cost, mission flexibility, and reduced risk that small-scale unmanned platforms provide in comparison to larger aircraft. Despite the versatility of unmanned aviation, limitations on payload size and weight, fuel and power capacity, and takeoff and landing infrastructure can restrict UAV applications, and have created a need for lift augmenting technologies that can reduce the impact of these limitations. Circulation Control (CC) is an active flow technique that has been proven as a method for augmenting the lift of fixed-wing aircraft. To enhance the performance and reduce the energy impact of a CC system on a UAV, a control system is required. This thesis summarizes research to develop, implement, and test both a Proportional-Integral-Derivative (PID) controller and a Fuzzy Logic (FL) controller to regulate the behavior of the CC system on the Unmanned Circulation Control Aerial Vehicle (UC²AV). Time domain frequency data in conjunction with system identification techniques are applied to model the dynamics of the system. A mathematical plant consisting of a Pulse Width Modulation (PWM) input and RPM, air velocity ($V_j$), and power consumption outputs is presented as a framework for controller development and testing prior to implementation on the physical CC system. Performance evaluation of the PID and FL controllers is conducted throughout a simulated flight envelope consisting of take-off, cruise, and landing phases. Response characteristics and power consumption during each phase is evaluated. Results obtained from experimentation validate the applicability
of both PID and FL control for regulating the behavior of the CC system, and aid in provid-
ing a power planning flowchart for optimizing the energy consumption of the augmented
lift system. The results indicate that the behavior of both controllers exhibits a correlation
to the simulation data above 67%. Additionally, both controllers present similar energy
usage characteristics, when applied throughout a simulated flight envelope. Results also
show that PID control displays a faster rise time and less overshoot than FL control in most
cases, but also a longer settling time. Overall the PID controller displays better regulation
of the dynamics of the CC system’s behavior. The use of the active CC regulation systems
described in this study provide opportunities for application to real-life UAV technology,
providing important advances to this rapidly growing technology.
Acknowledgements

I would like to express my sincere gratitude to the University of Denver for setting a foundation on which to embark on this journey. To both the Departments of Mechanical Engineering and Materials Science, and Electrical and Computer Engineering, I am thankful for your collaboration, support, and opportunities in academia. To my advisors, Dr. Kimon Valavanis and Dr. Matthew Rutherford, I am extremely grateful for your collegial guidance, wisdom, and understanding. To Dr. Konstantinos Kanistras, whose mentorship and guidance has been indispensable. I would also like to express special thanks to my parents Lee and Priscilla, for their unwavering support, patience, and positivity. And also to all of my friends, and those who have helped me along the journey represented in this work, I thank you.
# Table of Contents

1 Introduction
   1.1 Motivation ................................................. 2
   1.2 Problem Statement ........................................ 2
   1.3 Method of Approach ...................................... 3
   1.4 Summary of Contributions ................................. 4
   1.5 Thesis Organization ...................................... 5

2 Literature Review
   2.1 Circulation Control ...................................... 6
   2.2 PID and Fuzzy Logic Control Systems .................... 11
   2.3 PID and Fuzzy Logic Control of BLDC Motors, Compressors, and Energy Savings ........................................... 13
          2.3.1 BLDC Motor Control .................................. 13
          2.3.2 Comparisons between PID and Fuzzy Logic Controllers .................. 16
          2.3.3 Controller Applications in Power Optimization .................. 19
   2.4 Discussion ................................................ 21

3 System Description and Instrumentation ........... 23
   3.1 Air Supply Unit ............................................ 24
   3.2 Circulation Control System ................................ 26
   3.3 Instrumentation and Calibration ......................... 28
          3.3.1 RPM Sensor .......................................... 28
          3.3.2 Power Sensor ........................................ 30
          3.3.3 Pitot Probe ......................................... 31

4 System Identification ............................... 33
   4.1 Model Excitation and Data Collection ................... 35
   4.2 Model Structure, Fitting, and Validation ................ 37
          4.2.1 RPM Modeling and Validation .......................... 38
          4.2.2 Air Velocity ($V_j$) Modeling and Validation .......... 42
          4.2.3 Power Modeling and Validation .................... 43
   4.3 Accepted Model ............................................ 45
# 5 PID and Fuzzy Logic Controller Design and Implementation

5.1 PID Control ......................................................... 48
  5.1.1 PID Controller Simulation ................................. 50
5.2 Fuzzy Logic Control ............................................. 51
  5.2.1 Fuzzy Logic Controller Simulation ....................... 56
5.3 Controller Implementation ................................. 59

# 6 Results and Discussion

6.1 Takeoff Phase ................................................... 68
  6.1.1 PID Control During Takeoff Phase ....................... 70
  6.1.2 Fuzzy Control During Takeoff Phase .................... 71
  6.1.3 Energy Consumption During Takeoff .................... 72
6.2 Cruise Flight Phase ........................................... 75
  6.2.1 PID and FL Control During Cruise Flight .............. 77
  6.2.2 Energy Consumption During Cruise Flight ............... 81
6.3 Landing Phase .................................................. 82
  6.3.1 PID and FL Control During Landing Phase ............... 82
  6.3.2 Energy Consumption During Landing ..................... 84
6.4 Flight Envelope Energy Analysis ........................... 85
6.5 Discussion ...................................................... 88

# 7 Conclusions and Future Work

7.1 Conclusions ................................................... 90
7.2 Future Work ................................................... 97
List of Tables

3.1 Instrumentation/Sensor specifications for the CC system. .................. 28
5.1 Closed-loop response of independently increasing PID gains. ............. 49
5.2 Ziegler-Nichols PID tuning rules [1].............................................. 50
5.3 Table of linguistic rules for the error (e) and change in error (Δe) inputs of the Fuzzy Logic Controller. .................................................. 54
5.4 Response characteristics of 5 MF and 7 MF Fuzzy Logic controllers. ... 59
5.5 PID controller simulation and experimental response characteristics. ..... 62
5.6 FL controller simulation and experimental response characteristics. ...... 64
6.1 Takeoff performance data. ............................................................. 71
6.2 Response characteristics of the FL controller during takeoff. .......... 72
6.3 Energy (Joules) consumed for different motor initiation times. ........ 74
6.4 Fluctuations in airspeed and required changes in RPM to maintain a constant $C_\mu$. .......................................................... 76
6.5 Response characteristics of the PID and FL controllers to a 2 m/s increase in airspeed. ............................................................ 78
6.6 Power consumption between PID and FL controllers during a 2 m/s increase in airspeed. ....................................................... 79
6.7 Response characteristics of the PID and FL controllers to a 2 m/s decrease in airspeed. ............................................................ 80
6.8 Power consumption between PID and FL controllers during a 2 m/s decrease in airspeed. ....................................................... 81
6.9 Response characteristics of the PID and FL controllers during the simulated landing. ............................................................ 83
6.10 Power consumption between PID and FL controllers during the simulated landing. ............................................................ 83
6.11 Energy consumption for Mission-I and Mission-II, with varying scenarios. ... 88
7.1 Battery specifications for the battery power the CC system onboard the UC²AV. ............................................................ 96
7.2 Energy consumption, battery sizing, and mass savings for Mission-I and Mission-II, with varying scenarios. .................................. 96
List of Figures

2.1 Flow over a Coandă surface [2]. ........................................ 7
2.2 Circulation Control on the DEMON UAV [3]. ......................... 9
2.3 The UC²AV aircraft. .................................................... 11
2.4 Illustrations of Fuzzy membership functions as described by Zadeh [4]. 12
2.5 A comparison of theoretical and experimental data as presented by Hat et al. [5]. 15
2.6 Left: A graph of the results by Nasution, illustrating the possible energy savings when using a control system to regulate the speed of a variable speed compressor. Right: Table illustrating the response details between PI, PID, and FL controllers [6]. 16
2.7 A comparison of PID and FL controller response presented by Arulmozhiyal et al. [7]. .................................................. 17
2.8 A graph of the results by Neethu et al., illustrating the RPM response of a BLDC motor when using different controllers [8]. 19
2.9 A graph illustrating the reduction in energy consumption when using a Fuzzy controller in comparison to on/off control [9]. 20
2.10 Energy savings of various controllers on a compressor presented by Ekren et al. [10]. 21

3.1 Schematic of the CC system. ............................................ 24
3.2 The Anaconda UAV. ..................................................... 24
3.3 The Turnigy BLDC motor. .............................................. 25
3.4 The ASU’s housing, impeller, and motor [11]. .......................... 26
3.5 Plenum CAD design [11]. ............................................... 27
3.6 Left: CAD design of the CC system [11]. Right: Experimental setup of the CC system with detail showing pitot probe used for $V_j$ data collection. 27
3.7 Left: Comparison of the RPM data read using an Eagle Tree Systems ELogger and an Arduino-Nano. Right: Comparison of the RPM data read on an Arduino-Nano using a hall effect sensor and an Eagle Tree Systems RPM sensor. 29
3.8 Left: Current calibration curve. Right: Voltage calibration curve. 31
3.9 Pitot probe calibration schematic [12]. ............................... 32
3.10 Pitot probe calibration curve. ......................................... 32

4.1 Flowchart of the methodology used for creating a plant model of the CC system, adapted from work by Hat et al. [5]. 35
4.2 Sine-sweep (chirp) PWM command signal used for excitation of the CC system. 36
4.3 a) RPM, b) $V_j$, and c) power response of the CC system to a sine-sweep PWM input.

4.4 Proposed model for CC system plant.

4.5 Schematic of a DC motor [13].

4.6 MATLAB System ID toolbox GUI. 1. is where I/O data is input and preprocessed if necessary, 2. is where transfer function structures can be chosen, and 3. are where results are output and validated.

4.7 Comparison between RPM transfer function approximation and experimental data.

4.8 Comparison between $V_j$ transfer function approximation and experimental data.

4.9 Relationship between steady-state RPM and power consumption.

4.10 Comparison between the power transfer function approximation and experimental data.

4.11 Final model for CC system plant.

5.1 A MATLAB/Simulink block diagram used for simulation of the PID controller.

5.2 A schematic illustrating the architecture of a Fuzzy Logic controller [14].

5.3 Possible membership function geometries as presented by Zhao et al. [15].

5.4 Illustration of the triangular membership functions applied in this study.

5.5 Visual depiction of the Mamdani inference method [16].

5.6 Illustration of the centroid defuzzification method [14].

5.7 Visualization of the FL inputs to output.

5.8 The MATLAB Fuzzy Logic Toolbox GUI.

5.9 A MATLAB/Simulink block diagram used for simulation of the FL controller.

5.10 MATLAB/Simulink simulation of five MF and seven MF FL controllers.

5.11 The LabVIEW data acquisition GUI. 1. The controller inputs. 2. Controller outputs. 3. Real-time histogram. 4. Data acquisition and storage options.

5.12 Schematic Representation of the controller interfacing to the CC system.

5.13 Simulation and experimental response of the CC system with a PID controller subjected to an 8,000 to 26,000 RPM step input.

5.14 Simulation and experimental response of the CC system with a FL controller subjected to a 8,000 to 26,000 RPM step input.

5.15 Experimental response comparison of the CC system with a FL and PID controller, subjected to an 8,000 to 26,000 RPM step input.

6.1 Roadmap detailing the different possible phases of flight for the UC^2^AV.

6.2 Illustration of a mission incorporating aerial photography during cruise flight.

6.3 Illustration of takeoff phases.

6.4 Characterization of motor start-up sequence.

6.5 Left: Motor response with three sets of PID gains. Right: Motor response with three sets of ramp inputs using a PID controller.

6.6 Motor response with three sets of ramp inputs using a FL controller.

6.7 Left: Representation of motor RPM for Approach-I. Right: Representation for Approach-II.
6.8 Left: Power response of PID controller with three ramp inputs. Right: Power response of FL controller with three ramp inputs. 74
6.9 Left: RPM response of PID controller with three $C_\mu$ goals. Right: RPM response of FL controller with three $C_\mu$ goals. 78
6.10 Left: Power response of PID controller with three $C_\mu$ goals. Right: Power response of FL controller with three $C_\mu$ goals. 78
6.11 Left: RPM response of PID controller with three $C_\mu$ goals. Right: RPM response of FL controller with three $C_\mu$ goals. 80
6.12 Left: Power response of PID controller with three $C_\mu$ goals. Right: Power response of FL controller with three $C_\mu$ goals. 80
6.13 Left: RPM response of PID controller with three varying ramp inputs. Right: RPM response of FL controller with three varying ramp inputs. 83
6.14 Left: Power response of PID controller with three varying ramp inputs. Right: Power response of FL controller with three varying ramp inputs. 84
6.15 Left: Representative RPM profile for Mission-I, with CC during cruise flight. Right: Representative RPM profile for Mission-II, without CC during cruise flight. 86

7.1 Discharge voltage characteristics for LiPo batteries [17]. 92
7.2 Power planning flowchart for optimizing energy consumption of the ASU throughout the UC²AV’s simulated flight envelope. 94
7.3 The LiPo battery currently utilized for powering the CC system onboard the UC²AV. 95
Nomenclature

\( d \) 10 bit analog-to-digital value

\( A_f \) Amplitude

\( u \) Average RPM (RPM)

\( e_b \) Back electromotive force (V)

\( \Delta e \) Change in error

\( \Delta t \) Change in time (s)

\( K_t \) Coefficient of torque (V/(rad/s))

\( u_c \) Control output (microseconds)

\( t_c \) Cruise duration (s)

\( i \) Current (A)

\( \rho \) Density (kg/m\(^3\))

\( K_d \) Derivative gain

\( B \) Dynamic friction (Nm/(rad/s))

\( q_\infty \) Dynamic pressure (Pa)

\( E_{a-} \) Energy consumed during airspeed decrease (J)

\( E_{a+} \) Energy consumed during airspeed increase (J)

\( E_C \) Energy consumed during cruise (J)

\( E_L \) Energy consumed during landing (J)

\( E_T \) Energy consumed during takeoff (J)
\( e \)  Error (RPM)

\( K_{de} \)  Error change normalization gain

\( F_f \)  Final frequency (Hz)

\( T_f \)  Final time (s)

\( V_\infty \)  Free stream velocity (m/s)

\( FL \)  Fuzzy Logic

\( K_c \)  Fuzzy Logic control output gain

\( u_{crisp} \)  Fuzzy Logic crisp output

\( L \)  Inductance (H)

\( F_0 \)  Initial frequency (Hz)

\( T_0 \)  Initial time (s)

\( U_{PWM} \)  Input PWM (microseconds)

\( K_i \)  Integral gain

\( s \)  Laplace domain variable

\( \dot{m} \)  Mass flow rate (kg/s)

\( \mu \)  Membership function area

\( b \)  Membership function center

\( J \)  Moment of inertia (kg\( \cdot \)m\(^2\))

\( C_\mu \)  Momentum coefficient of blowing

\( p \)  Number of perturbations

\( D \)  Offset

\( Y_{V_j} \)  Output jet velocity (m/s)

\( Y_{RPM} \)  Output RPM (RPM)

\( P \)  Power (W)

\( K_p \)  Proportional gain
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>Proportional-Integral-Derivative</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse Width Modulation</td>
</tr>
<tr>
<td>R</td>
<td>Resistance (Ω)</td>
</tr>
<tr>
<td>$t_r$</td>
<td>Rise time (s)</td>
</tr>
<tr>
<td>RPM</td>
<td>Rotations Per Minute</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Rotational speed (rad/s)</td>
</tr>
<tr>
<td>$K_s$</td>
<td>Set point normalization gain</td>
</tr>
<tr>
<td>$x$</td>
<td>Set point value (RPM)</td>
</tr>
<tr>
<td>$P_{SS}$</td>
<td>Steady-state power consumption (W)</td>
</tr>
<tr>
<td>$T_s$</td>
<td>Step time (s)</td>
</tr>
<tr>
<td>$t$</td>
<td>Time (s)</td>
</tr>
<tr>
<td>$T$</td>
<td>Torque (Nm)</td>
</tr>
<tr>
<td>$H$</td>
<td>Transfer function</td>
</tr>
<tr>
<td>$H_{V_j}$</td>
<td>Transfer function of jet velocity</td>
</tr>
<tr>
<td>$H_{RPM}$</td>
<td>Transfer function of RPM</td>
</tr>
<tr>
<td>$P_u$</td>
<td>Ultimate period (s)</td>
</tr>
<tr>
<td>$S_u$</td>
<td>Ultimate sensitivity</td>
</tr>
<tr>
<td>$UC^2AV$</td>
<td>Unmanned Circulation Control Aerial Vehicle</td>
</tr>
<tr>
<td>$V_j$</td>
<td>Velocity of the jet (m/s)</td>
</tr>
<tr>
<td>$V$</td>
<td>Voltage (V)</td>
</tr>
<tr>
<td>$S$</td>
<td>Wing surface area (m²)</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In recent years, both military and civilian applications of Unmanned Aerial Vehicles (UAVs) have expanded dramatically. Additionally, due to the ongoing advances in technology and aviation, and the versatility of unmanned aviation, the growth of UAV operation is expected to continue well into the future. From an engineering perspective, UAVs offer the promise of mission flexibility, improved safety, and ease of operation. Current UAV applications, however, have limitations based on a number of factors – including payload size and weight, takeoff and landing infrastructure, and fuel and power capacity. These restrictions create a need for solutions that can effectively and efficiently mitigate these factors. One such solution that has been explored is the use of active flow control for lift augmentation. Circulation Control (CC) is an active flow technique that has been proven to be a method for augmenting lift of fixed-wing aircraft. Most research has focused on the lift benefits of a CC system, however, considerations with regard to energy efficiency and power consumption of CC systems have not yet been explored. To enhance the performance and reduce the energy impact of a CC system on a UAV, a control system is required. This thesis summarizes the research performed to develop, test, and implement a controller
to regulate the behavior of the CC system on the Unmanned Circulation Control Aerial Vehicle (UC$^2$AV).

1.1 Motivation

Research indicates that CC is an effective method for augmenting lift of fixed wing aircraft [2, 3, 11, 12, 18–26]. Furthermore, CC has been applied to an Anaconda UAV by Kanistras et al., and is shown to be an effective method for enhancing lift and reducing takeoff distance by as much as 53% [27, 28]. However, the power penalties and energy expenditure of the power source used for running the CC system itself have not yet been investigated.

The following open research questions motivate this investigation:

- *Is it beneficial to derive a mathematical model of the CC system?*

- *What type of controller (conventional model-based or model free) should be used to best regulate the CC system’s behavior?*

- *How can the power consumption of the CC system be optimized?*

1.2 Problem Statement

Despite the benefits of CC, specifically those observed on small-scale unmanned aircraft – i.e., improved lift, reduced takeoff distance, etc. – there are drawbacks with the addition of such a system, for instance the power needed to run it. Methods for reducing the drawbacks incurred by the introduction of a CC system, specifically power penalties and battery weight, have not yet been investigated. Mission specifics dictate different requirements on the CC system, including a reduction in takeoff and landing distances, or the
ability to transport an increased level of payload. Effectively regulating the CC system’s ability to enable these takeoff, landing, and payload goals, while minimizing the power impacts, presents a challenging issue. These drawbacks and challenges are used as a guide to develop and implement a control scheme capable of optimizing the use of the CC system with respect to power consumption.

The end goal of this research is the implementation of a controller to provide energy efficient regulation of the behavior of the CC system onboard the UC²AV, a mathematical model of the CC system, and a power planning flowchart providing information that an operator can use to appropriately size the payload and power systems onboard the UC²AV.

1.3 Method of Approach

To address the difficulties of regulating the behavior of the CC system onboard the UC²AV, and reducing the power penalties (and associated battery mass), the research methodology focuses first on simulation, then experimentation. Mathematical models of the CC system are created by gathering input and response data. This is achieved through calibrating RPM, airspeed, and power sensors, and interfacing them to a computer for data collection through an Arduino microcontroller. Modeling the compressor RPM, pneumatic output, and power characteristics of the CC system allows for the rapid testing of controllers without the risk of damage to a physical system. The research methodology also focuses on experimentation. The development of a testbed allows for the implementation of a Proportional-Integral-Derivative (PID) controller and Fuzzy Logic (FL) controller on the CC system in a real-time laboratory environment, as well as validation of the theoretical results. The CC system behavior is regulated during simulations of three main phases of an aircraft flight envelope: takeoff, cruise, and landing. Various controller gains are applied to achieve the desired rise time, settling time, and overshoot. For each phase of flight,
the power consumption of the system under various conditions is studied, allowing for the creation of a flowchart of energy optimality based on the aircraft mission.

1.4 Summary of Contributions

The primary contribution of this work is threefold: the first is the design and implementation of two control system architectures: PID and Fuzzy Logic, for regulating the behavior of the CC system onboard the UC²AV. The second is the characterization of the energy usage of the CC system during simulated takeoff, cruise, and landing phases of flight. The third is a power planning flowchart for minimizing the power consumption of the CC system. A summary of these contributions as well as the achievements attained during their development is cataloged here:

- Construct an abstracted mathematical model of the CC system onboard the UC²AV, providing a framework for controller simulation and design.

- Design a PID controller capable of accurately regulating the behavior of the CC system, and implement it on a microcontroller to enable real-time control of the physical system.

- Design a Fuzzy Logic controller also capable of regulating the behavior of the CC system, and further validate it by controlling the physical CC system through implementation on a microcontroller.

- Provide performance evaluations of the PID and FL controllers throughout a simulated flight envelope.

- Present a characterization of the power consumption of the CC system during the
takeoff, cruise, and landing phases of flight, to enable optimization of the system’s power utilization.

- Provide recommendations for the required battery capacity to power the CC system, based on the aircraft mission and a flowchart of energy consumption.

### 1.5 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 presents a review of the literature, where background information on CC systems is given, as well as applications of PID and Fuzzy Logic controllers and their use in regulating Brushless Direct-Current (BLDC) motors and compressors, as well as optimizing the energy use of a system. Chapter 3 details the CC system onboard the UC²AV and the setup and calibration of the instrumentation used to interface to this system. Chapter 4 provides information on system identification of the CC system, and Chapter 5 describes the development of the PID and FL controllers and their implementation. Chapter 6 presents the experimental results of testing on the CC system, while Chapter 7 presents the conclusions and recommendations for future work.
Chapter 2

Literature Review

Studies into the ability of active flow blowing techniques to augment lift of fixed-wing aircraft have been around since the late 1930’s [2, 3, 11, 12, 18–26], with applications in augmented lift and control surface technologies originating as far back as the 1970’s [18, 19, 22, 29]. Active control system management is the next step in the development of CC technology. Although active control of mechanical systems dates back to as early as the Industrial Revolution, control system applications in the real-time regulation of a CC system have not arisen in literature until recently [12]. This literature review is structured to provide an overview of the research on CC, beginning with mathematical simulation and scaled wind tunnel models of CC systems, outlining applications on full scale aircraft and UAVs, and moving to a review of literature related to PID and FL controllers and their applications in both BLDC motor and compressor regulation and energy optimization.

2.1 Circulation Control

Shown to be more effective than traditional lift augmentation devices (flaps, slats, vortex generators, etc.) [18,29], CC utilizes the Coandă effect by blowing high-energy air close
to the trailing edge of a wing through a slot [30, 31]. This sheet of high-energy airflow is blown over a Coandă surface, and remains attached due to a balance between the low static pressures created by the flow, and centrifugal forces of the curved surface (Figure 2.1) [2]. This balance moves the separation point of the flow towards the lower surface of the wing, which in turn results in an increase in lift [32]. The entire wing configuration is known as a Circulation Control Wing (CCW).

![Image of flow over a Coandă surface](image_url)

**Figure 2.1: Flow over a Coandă surface [2].**

Mathematical simulation and numerical investigations have allowed researchers to better understand the physics of CC prior to experimentation. Pfingsten et al. [30] use mathematical simulations to understand the flow around CC airfoils, and the authors’ results illustrate the difficulties of correctly modeling the flow associated with CC.

The effects of momentum and lift enhancement are studied by Nishino et al. [33], where three different jet nozzle thicknesses are studied. The momentum coefficient of blowing ($C_\mu$) is used as a metric for performance, and is a parameter critical to characterizing CC systems. Given by Equation (2.1), $C_\mu$ depends on the mass flow at the CCW’s slot exit ($\dot{m}_j$), the velocity at the slot ($V_j$), the free stream velocity ($V_\infty$), the dynamic pressure ($q_\infty = \frac{1}{2}\rho V_\infty^2$) and the wing’s surface area ($S$).

$$C_\mu = \frac{\dot{m}_j V_j}{q_\infty S} \quad (2.1)$$

7
Results indicate that smaller nozzle thicknesses lead to lower flow losses downstream, and better circulation around the airfoil. The momentum coefficient of blowing ($C_\mu$) is also utilized in other research as a metric for describing flow related to CC airfoils [24, 34, 35]. Extensive numerical research is conducted in a number of additional areas, including the effects of differently shaped trailing edges [34], the effect of slot heights on CC performance [36], and the possible benefits of pulsed blowing [35]. In general, the results of this mathematical modeling indicate the viability of various configurations in relation to CC performance characteristics. It is not possible to study CC solely with simulation, and so application of CC to physical systems is reviewed in the following paragraphs.

Much of the experimental research conducted on CC is focused on studying the aerodynamic effects in controlled laboratory environments, using wind tunnels and external compressed-air sources to supply the required airflow to the CCW [2, 22–24, 32]. This research also focuses primarily on how CC can be applied to full-scale aircraft. In the 1970’s, CC is studied by Englar [22] using a 1/8.5 scale A-6 demonstrator fixed-wing aircraft, in order to increase lift during takeoff and landing. Tests are performed in a wind tunnel, using a compressed air source to supply air to the wings. Results show the benefits of CC, with a lift coefficient 2.2 times larger than the conventional A-6 aircraft configuration.

CC is also studied in the 1970’s by Abramson [32], who uses a subsonic wind tunnel to find that increasing $C_\mu$ is correlated to an increase in the lift coefficient of the airfoil tested, and that smaller slot heights further improve these lift characteristics. Wind tunnels and compressed air are also used at the NASA Langley Research Center to study the effects of trailing-edge blowing by Alexander et al. [2], and also by Paciano et al. [23] to study the effects of flow uniformity at the outlet of the CCW plenum.

Work out of the University of Manchester [3] first used CC on a UAV system with the primary goal of testing flapless flight control as part of the Flapless Air Vehicle Integrated Industrial Research (FLAVIIR) project – a collaboration between ten British universities
over five years. The work focuses on a pneumatic system that is designed to allow for asymmetric airflow to each wing, allowing for roll control of the UAV. Compressed air canisters are first considered as a source of airflow for this system, but rejected due to the large mass impact. A modified turbocharger is chosen to provide the required airflow to the system, giving mass flow rates of 0.06 kg/s, and suitable for CC on the aircraft.

In work by Crowther [25], as part of the same project described above, a gas turbine combined with engine bleed techniques is used on an 80 kg, 85% scale of the DEMON UAV (Figure 2.2) aircraft to provide the necessary airflow for CC. Using Fluidic Thrust Vectoring and CC, the primary goal of the research is to demonstrate the ability to control an aircraft without conventional control surfaces. Using wind tunnel testing, results show that CC is an effective and efficient method for controlling aircraft attitude. As noted by Crowther, the mass cost of the entire CC system is 15% of the weight of the aircraft, and may be effected when scaled to full size, where the required $C_\mu$ will still need to be maintained, likely with a larger turbine and more engine bleed.

![Figure 2.2: Circulation Control on the DEMON UAV [3].](image)

Circulation Control is also studied on the Cruise-Efficient ESTOL (Extreme Short Take-Off and Landing) Transport Aircraft (CEETA) demonstrator UAV by Alley et al. [37]. The 179 lb aircraft is based on an 11% scale Boeing 737-300 aircraft. Two sets of wings are designed; one to emulate a conventional wing configuration, and the other with both leading
and trailing-edge blowing, with dual-radius flaps and engines mounted on the top side of the wing. To achieve the desired ESTOL performance goals of the aircraft, a combination of turbine-power lift, and CC are utilized. CC is implemented in the form of leading-edge and trailing-edge blowing, with high-energy airflow provided by an Electric Ducted Fan (EDF), which is a Commercial Off-The-Shelf (COTS) product used in the remote-controlled (RC) plane industry.

Most recently, Kanistras et al. [24], study the lift performance of several different airfoils to be used on UAVs, with results showing that CC increases lift, and that a 2:1 Coanda surface is the most effective across all of the tested airfoils. These studies into CC on UAVs are further investigated by Kanistras et al. [12], where a compact 3D printed centrifugal compressor (Air Supply Unit) capable of providing accelerated airflow to the CCW wing of an Anaconda UAV is suggested as method for creating a CC system. The Air Supply Unit (ASU) design is further developed [26], with the final compressor design weighing less than 100 g, and providing a maximum flow velocity of 71 m/s at 23,000 RPM. In order to regulate the speed of the compressor, and accurately measure output air velocity at the slot, a PID controller is introduced. This is the first mention in the literature, to the best of the author’s knowledge, of a control system being used to regulate the pneumatics of a CC system. Results for the ASU controller show that controllability is achievable, with an observed 2.0 s rise time during a simulated takeoff phase of flight, and an observed 6.1 s settling time during a simulated cruise phase. Integration of this CC system into the fuselage of an Anaconda UAV is addressed by Saka et al. [11], where the design of an ASU coupled to a plenum through various junctions and tubes is described. Flow uniformity and rate sufficient for CC are exhibited (up to 0.03 kg/s), and found suitable for flight testing on a UAV. The integration of this system (the UC²AV) is shown in Figure 2.3.
2.2 PID and Fuzzy Logic Control Systems

Active system control of CC systems is a critical next step in improving the usefulness of this active blowing technology. Various approaches to system control have been developed over the years, including what are now some of the most ubiquitous, known as the Proportion-Integral-Derivative (PID) controller [38], and Fuzzy Logic (FL) controller [39].

A PID controller is a closed-loop feedback control system that takes into account the error, accumulated error, and rate of change of error of a system [38]. PID controllers are model-based controllers: their performance is dependent on the system that they are controlling, and gains have to be adjusted to accommodate for this. Several methods exist for tuning, including a method developed by John Ziegler and Nathaniel Nichols, called the Ziegler-Nichols method [1]. Their work outlines methods for tuning the control gains of a system without a complete mathematical description of the system – a method that is useful when dealing with nonlinear, or complex systems. The procedure outlined in their work involves increasing the potential gain of the controller until the system reaches a steady-
state oscillation, and measuring both this gain value, and oscillation period. These values are then used as a guideline for deriving a set of proportional, integral, and derivative gains.

Fuzzy Logic control is another type of controller used for system control, taking advantage of the advancements in frequency-response methods. FL control utilizes the idea of fuzzy set theory [39] and is derived from the idea of “fuzzying” traditional Boolean logic (where the possible states of a system are either true, “1”, or false, “0”) to a form where states can range from values that are completely true to completely false. This is achieved by creating membership functions, in which the variables of a function or system can partially belong to several. This is illustrated in Figure 2.4.

![Figure 2.4: Illustrations of Fuzzy membership functions as described by Zadeh [4].](image)

As technologies have advanced, the implementation of controllers on microprocessors has increased. In modern systems, controllers are typically run on microcontrollers that are interfaced to the system being regulated [12, 40–43]. These microprocessors make the implementation of FL controllers possible – without them the computation that is needed would not be possible to emulate with a physical arrangement of electrical components.
2.3 PID and Fuzzy Logic Control of BLDC Motors, Compressors, and Energy Savings

The CC system requiring regulation in the research project being described here is developed by Kanistras and Saka [11, 26], utilizing a centrifugal compressor coupled to a BLDC motor to supply accelerated air to the trailing-edge of a circulation control wing. Research has shown that a BLDC motor can be used effectively to operate a compressor [11, 26, 44, 45], and that both PID and FL controllers can be used to adequately operate such a system, with respect to meeting certain performance criteria (Rise time, settling time, overshoot, etc.) and also power optimization [9, 10, 26, 46, 47]. However, certain challenges exist when implementing a controller on BLDC motors and compressors. BLDC motors can exhibit nonlinear qualities when operated at low or high RPMs, and the functionality of compressors can be difficult to model. In systems with complicated dynamics such as those presented, the gains of PID controllers can be difficult to tune. Additionally, constructing the proper membership functions for FL controllers can prove to be a challenge. Optimizing these systems for power consumption also presents challenges, since the speed and accuracy at which the system targets a set point can negatively impact the energy consumption.

2.3.1 BLDC Motor Control

BLDC motor speed regulation is a widely studied area in controls, and previous research has shown that PID and FL controllers can be used to control a BLDC motor’s speed in varied applications [5–7, 48]. In research by Yu et al. [48] and Hat et al. [5], only the unloaded response of the BLDC motor is studied. Yu et al. first use a heuristic approach to tune PID gains of a controller for a BLDC motor until a sufficient response is achieved.
This is used as a baseline for comparison against a PID controller that is optimized using the Linear Quadratic Regulator (LQR) function to optimize its performance. Both the traditional PID controller and the LQR optimized controller are applied to a computer model of a BLDC motor model to compare the two systems. The results of testing the theoretical models illustrate a reduction in settling time and percent overshoot with the optimized PID controller. The results are also verified experimentally when the controllers are applied to an actual BLDC motor, where an optimally tuned PID again outperform the heuristically tuned PID.

In research by Hat et al. [5], data are gathered on the open loop response of a BLDC motor, and a second-order transfer function model is created using the MATLAB/SIMULINK system ID toolbox. To create this model, a microcontroller is interfaced to both the motor (a Longs model 57BLF03) though a 20 A motor driver, and a computer. The microcontroller is set up to generate a PWM signal for commanding the motor speed, and to collect the motor speed information. A pseudo-random binary signal (PRBS) is used as the excitation function for this motor, and data sampled every 0.01 seconds. After feeding these data into the MATLAB/SIMULINK system ID toolbox, a transfer function model is chosen with a 91.44% fit to the experimental data. The quality of fit is quantified using a root mean square error (RMSE) method. This model is used to tune controller gains, which are then uploaded to a microcontroller to control the speed of the physical BLDC motor. The motor is run in an un-loaded state, and theoretical results are seen to adequately match experimental results with a low RMSE value of 8.6. The comparison between the theoretical and experimental data is also shown in Figure 2.5.
Figure 2.5: A comparison of theoretical and experimental data as presented by Hat et al. [5].

Adequate control of a BLDC motor and compressor is also shown in research by Nasution et al. [6]. The performance of P, PI, PD, and PID control systems with varying motor loads are evaluated when implemented on a variable speed compressor of an Air-Conditioning (AC) system, and the energy savings are compared to that of a traditional bang-bang (on/off) style controller that exists in most AC systems. The authors first design a PID controller that regulates the speed of the compressor, while measuring its electrical energy consumption using a PCI-1711/PCLD-8710 interface to a computer. Data is then compared between on/off conditions, several set frequency ranges, and variable, PID controlled speed. The PID settings are tuned using a trial and error method. The results show that a PID controller can stably regulate the temperature of a room, and also that energy consumption can be reduced. The gains optimal for energy usage are found to be $K_p = 3.3$, $K_i = 0.180$ and $K_d = 0.040$ for cases with thermal load and without load on the AC system. Energy consumption is reduced by 25% in comparison to the on/off controller for the no-load case.
Comparisons between PID and Fuzzy Logic Controllers

Comparisons between PID and FL controllers, and their performance when regulating a BLDC motor are studied in several areas of literature [7, 8, 49]. In research conducted by Arulmozhiyal et al. [7], controller performance is tested with various loads applied to a motor. The main focus of the paper is the application of several controllers used for speed regulation of a brushless DC motor. A conventional PID controller is used as a baseline for comparison, and compared against a Fuzzy PID controller. To test the controllers, the author models a BLDC motor in the MATLAB/SIMULINK work environment. The gains for the PID controller are obtained using the Zeigler-Nichols method ($K_p = 0.8$, $K_i = 48$, $K_d = 0.01$), and the fuzzy controller is designed as a two-input structure with coupled rules and triangular membership functions. To study the two controllers the rise time, overshoot, settling time, and steady-state error of the model’s response to various step responses is compared.
The results show that in general the FL controller outperforms the conventional PID controller when looking at these parameters. The settling time of the PID controller is 0.35 s, and 0.10 s for the FL controller when a reference speed of 1,500 RPM is set. The FL controller also outperforms the PID controller when a 5 Nm load is applied to the motor, settling in 0.15 s in comparison to 0.40 s. A comparison between the PID and FL controller is shown in Figure 2.7.

In continuation of the work by Arulmozhiyal et al., a conventional PID controller, Fuzzy PID controller, and Adaptive Fuzzy PID controller are compared in research by Kandiban et al. [49], using a BLDC motor model as the test system, and a conventional PID controller as a baseline for comparison. The gain values for this PID controller are obtained using the Zeigler-Nichols method. A Fuzzy PID controller is constructed using a Mamdani fuzzy logic model, with two inputs – error from the reference speed, and amount of error change (derivative of error). An Adaptive Fuzzy controller is also implemented, designed to adjust the gains incrementally based on error and change in error. Rise time, overshoot, and settling time are used as performance metrics, and results show superior performance by both of the fuzzy controllers, and that a constant motor speed can be maintained despite variations in the motor load.
In work by Salim [13], a PID, and fuzzy-based PID controller are implemented to control a DC motor using the LabVIEW work environment. A mathematical model of a motor is constructed based on the equations that describe its physics, and used to compare the effects of each controller. The PID controller gains are adjusted using the Zeigler-Nichols method, and the FL controller is configured to use the output signal, speed of error, and change of speed of error as the inputs to the fuzzy membership function. A peak time of 0.0048 s and rise time of 0.004 s is achieved with the FL controller, which provides better response characteristics than the PID controller, with a peak time of 0.008 s and rise time of 0.00053 s. Similar studies are also conducted by Mohan et al. [50], using PID and FL to regulate the speed of a BLDC motor. Two inputs are used for the FL controller: speed error, and change in speed error, and sent through standard triangular membership functions. Both controllers are simulated in MATLAB/SIMULINK with a model of a BLDC motor. With a 2,300 RPM setpoint, the FL controller exhibits better rise time, settling time, and overshoot (Rise time = 1.1 ms, settling time = 0.9 s, overshoot = 6.1 %), in comparison to a PID controller (Rise time = 1.8 ms, settling time = 1.4 s, overshoot = 8.4%).

PID and FL controllers are also compared in research by Neethu et al. [8]. The main focus of the authors’ research is the comparison between PI, PID, and Fuzzy Logic controllers, when used to regulate the speed of a BLDC motor. All three controllers are tested using a mathematical model of a BLDC motor, created in MATLAB/SIMULINK, with the settling time and overshoot used as performance metrics. Under loaded conditions, reference speeds of 1,000, 2,000, 3,000, and 4,000 RPM are used as the test cases for comparison. Results show better performance by the fuzzy logic controller, with both lower settling times and overshoot. Figure 2.8 shows a table of the responses of the studied controllers.
2.3.3 Controller Applications in Power Optimization

In addition to the other challenges related to CC (reducing mass impact, maintaining flow uniformity, etc.), reducing the power penalties due to the inclusion of such a system is also important. Controllers have been studied as a method to reduce the energy consumption of a system, and although much of this research relates to compressor control in cooling and refrigeration systems, this work has direct implications to CC system development.

As stated by the International Energy Agency (IEA) [51], an efficient way to improve the power efficiency of a motor is the introduction of a variable-frequency drive (VFD) [51, 52]. A VFD is any controller capable of adjusting the speed of a motor in response to changing loads, and is beneficial over on/off and fixed speed systems because it eliminates most partial load losses (throttles, dampers, etc.), while running at a speed that matches demand. The US Department of Energy [53] also notes that using VFD to match a system’s actual operating requirements can greatly reduce energy consumption. It is noted that a 10% reduction in the operating speed of pumps, fans, and compressors results in a 30% reduction in power.
In research by Aprea et al. [9], the authors show that a FL controller can reduce the energy consumption of a semi-hermetic reciprocating compressor used in refrigeration. Cooling loads in a room are simulated using electric heaters, and power consumption is measured using a wattmeter. A FL controller is designed to regulate compressor speed, with triangular membership functions constructed so that the controller is more sensitive as it reaches the reference temperature. The test chamber is subjected to a variety of conditions to simulate a working environment, including: opening and closing the chamber door, and subjecting the system to two “outdoor” temperatures of 32°C and 10°C. Results show an energy savings of about 13% with the application of FL control in comparison to the on/off control typically utilized in cooling.

Figure 2.9: A graph illustrating the reduction in energy consumption when using a Fuzzy controller in comparison to on/off control [9].

Energy savings are also illustrated by Ekren et al. [10, 47] by regulating the behavior of a variable speed compressor within a chiller system. In one study, the performance of three different controllers (PID, Fuzzy Logic, and Artificial Neural Network) are compared (Figure 2.10). After running the system through a cooling cycle, results show that the FL controller outperforms the PID controller by 1.4% when used for reducing power consumption. In another study, a similar chiller system is investigated, but this time comparing
the energy consumption of a variable speed compressor between FL and traditional on/off control. Here an energy savings of 17% is shown.

![Energy consumption graph](image)

Figure 2.10: Energy savings of various controllers on a compressor presented by Ekren et al. [10].

In research by Tekin et al. [45], control systems are applied to a fuel cell system as a systematic methodology for energy management. A compressor, BLDC motor, and fuel cell system are modeled to first design and test a PID and FL controller. A conventional PID controller is used, and the FL controller is constructed using five inputs: fuel cell voltage and its derivative, fuel cell current and its derivative, and the derivative of the power demand. The output of each controller is used to regulate the airflow through the system, with the results from these simulations showing a 12% reduction in energy use when using the FL controller in comparison to the PID controller.

### 2.4 Discussion

Review of the literature reveals that despite the early inception active flow blowing techniques in the 1930’s, there are still many advancements that can be made in the area of augmented lift. Much research has been conducted studying the accuracy of various mathematical models of CC, the effects of slot size and flow uniformity, and the relationship between $C_\mu$ and lift coefficients. However, only recently have CC systems been imple-
mented on an aircraft of any size, and in the literature that was investigated, only one case where real-time control is implemented to regulate such a system. Review of the literature also reveals the widespread application of PID and Fuzzy Logic controllers for regulating the speed of BLDC motors. In many cases, mathematical models of the system are created to simulate the controller performance prior to experimentation. Results from the literature illustrate that PID and FL controllers can adequately regulate speed, although FL controllers typically outperform conventional PID controllers with respect to rise time, settling time, overshoot, and energy consumption. Research also shows that compressors can be operated using BLDC motors, and furthermore, that compressors can be effectively regulated with both PID and Fuzzy Logic control.

Although most research focuses on the lift benefits of a CC system, considerations with regard to efficiency and power consumption of CC systems have not yet been explored. Additionally, the studies into power optimization presented in literature are primarily focused on increasing the efficiency of cooling systems (regulating the behavior of compressors), but have not been applied to pneumatic systems. While the fundamentals of system regulation are still applicable to CC systems, the lack of research in the area of CC power consumption illustrates the need for expanded knowledge in this area, including research into how PID and FL controllers can be applied to optimize power consumption.
Chapter 3

System Description and Instrumentation

This chapter provides an overview of the CC system within the UC²AV, its sub-components, and interfacing instrumentation. The Air Supply Unit (ASU) and details on its functionality and operational limits are described first. Then, the development and functionality of the Air Delivery System (ADS) is presented, followed by the calibration procedures of the sensors that gather data on the CC system.

The CC system is comprised of an ASU designed to provide the required mass flow of air, an ADS to deliver the accelerated airflow to the CCW’s slot with minimal losses, and instrumentation in order to both control the system, and gather information on the RPM, energy consumption, and mass flow output. The system is designed to be lightweight, pneumatically efficient, and deliver uniform flow across the slot of the wing in order to maximize lift augmentation. The ASU is first studied, as it is the primary dynamic system within the CC system. The ASU is the source of the accelerated airflow and power consumption, and is the subsystem that is directly controlled by adjusting inputs to the BLDC motor. The overall CC system is also investigated, since the effects (blockage and losses) the ADS imparts onto the ASU directly influence power consumption and controllability. Figure 3.1 illustrates a schematic of the CC system components.
3.1 Air Supply Unit

The ASU is a variable-speed centrifugal compressor designed to provide airflow to the ADS. Air is accelerated by the compressor using a BLDC motor. The compressor is designed to fit within the fuselage of an Anaconda UAV (Figure 3.2), which constrains the maximum volume envelope to a diameter of 180 mm and height of 150 mm [11].

![Figure 3.2: The Anaconda UAV.](image)

The motor is commutated by an electronic speed controller (ESC) capable of handling the power requirements of the motor, and power is supplied by a 4S (4-cell in series, 14.8 V nominal) Lithium-Polymer battery. The ESC is rated to handle a maximum of 90 A, and operates by switching a series of MOSFETs (metal-oxide-semiconductor field-effect transistors) to control the motor speed.
transistors) to energize the phases of the BLDC motor. This switching is handled by the electronics within the ESC, and is set by a Pulse-Width-Modulated (PWM) input command signal. The BLDC motor driving the compressor impeller is a 900 W Turnigy EDF L2855-900W (Figure 3.3).

![Figure 3.3: The Turnigy BLDC motor.](image)

Typically used in RC applications, the motor is capable of a maximum unloaded speed of 29,600 RPM. The physical limits present in the ASU are accounted for when integrating a controller: the operational RPM limits of the motor range from a minimum of about 7,000 RPM to a measured average maximum of 27,800 RPM. Below 7,000 RPM, the current applied to each phase by the ESC is insufficient to predictably operate the motor. The maximum RPM is governed by the total power output of the motor and load introduced by the compressor and ADS. Maximum RPM occurs when the torque created by the electricity through the motor phases is in equilibrium with the torque load from the compressor. At maximum RPM, a large amount of heat can build up in the motor, so it is run at a maximum of 26,000 RPM to protect possible damage to the internal components such as the windings. The maximum velocity air speed out of the ASU is dictated by the compressor design, losses within the unit, and maximum speed of the motor. Experimental data show that a maximum output of 0.03 kg/s is achieved [11]. Figure 3.4 shows the lightweight ASU housing, impeller, and motor, which together add 180 g to the CC system.
3.2 Circulation Control System

Accelerated airflow is transported to the CCW via the ADS. The ADS equally divides the airflow from the ASU between two flow paths – one for each wing of the UAV. To reduce the losses at the junction separating the airflow, many configurations were tested using CFD by Saka et al. [11], ultimately leading to a design with a minimal pressure drop of 500 Pa at the highest inlet velocity of 50 m/s. After the junction, the air travels through ducting to a plenum designed to provide flow uniformity to the CCW during flight. The velocity of this airflow at the plenum slot is known as $V_j$. The first iteration of the plenum design featured five internal diffuser vanes to provide uniform flow across the plenum span, and a slot height of 1.00 mm [26]. Air velocity at the slot is also found to be linearly related to the RPM of the ASU. Additional analysis and development of the plenum design also shows flow uniformity. Eleven vanes are added, and the slot height reduced to 0.40 mm (Figure 3.5), with testing of this plenum again showing a linear relation between $V_j$ and ASU RPM. The ADS system is 3D printed from Acrylonitrile Butadiene Styrene (ABS) plastic, which allows for precise geometries to be fabricated in a lightweight manner. Additive manufacturing techniques allow for fabrication of complex geometries, such as the vanes within the plenum, without the need for multiple parts and hardware which further
reduces weight. The entirety of the CC system (plenum, tubing and junctions, and ASU) weighs 650 g, excluding the weight of the battery required to power it [54].

Figure 3.5: Plenum CAD design [11].

The physical limitations of the ADS are represented primarily by the pneumatic losses it introduces, such as head losses caused by the inlets, outlets, bends, and diameter changes of the piping within the system, all of which are expected to reduce the ASU performance by approximately 25% [11]. The design of the pneumatic components is driven by weight limitations onboard the Anaconda UAV, as well as space limitations within the aircraft fuselage and NACA0015 airfoil CCW. The piping and travel distance of the accelerated air can also introduce delays between an input PWM command signal given to the ESC and the air flow response $V_j$. Together the ASU and ADS make up the CC system for this experimental application, as shown in Figure 3.6.

Figure 3.6: Left: CAD design of the CC system [11]. Right: Experimental setup of the CC system with detail showing pitot probe used for $V_j$ data collection.

A testbed consisting of the CC system mounted to a frame allows for real-time controller tuning and data collection in a configuration emulating that of the UC$^2$AV. The ASU is mounted within an Anaconda fuselage, and the ADS is set up as it would be within the
aircraft, with the plenums separated by a span of 500 mm and connected to the ASU with the junction and tubing that is also integrated into the UC\textsuperscript{2}AV.

### 3.3 Instrumentation and Calibration

In order to accurately regulate the behavior of the CC system, proper sensor calibration is necessary, since instrument uncertainty can lead to loss of system controllability and inaccurate data. Specifications for the three sensors are found in Table 3.1.

Table 3.1: Instrumentation/Sensor specifications for the CC system.

<table>
<thead>
<tr>
<th>Component</th>
<th>Manufacturer</th>
<th>Part Number</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPM Sensor</td>
<td>Eagle Tree Systems</td>
<td>RPM-BRS-V2</td>
<td>Pulse Sensing Range: 100 Hz to 50 kHz</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Max Voltage: 44.4 V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Max Amperage: 300 A</td>
</tr>
<tr>
<td>Power Sensor</td>
<td>AttoPilot</td>
<td>SEN-10644</td>
<td>Maximum Voltage: 51.8 V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Maximum Current: 178.8 A</td>
</tr>
<tr>
<td>Pitot Sensor</td>
<td>Freescale Semiconductor</td>
<td>MPXV7002DP</td>
<td>Pressure range: ±2 kPa</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Accuracy: 2% FS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sensitivity: 1 V/kPa</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Response time: 1 ms</td>
</tr>
</tbody>
</table>

#### 3.3.1 RPM Sensor

Two tests are run to validate the accuracy of using an Arduino-Nano microcontroller to capture RPM data, and the accuracy of the Eagle Tree systems RPM sensor. The RPM sensor attaches to two of the three leads of the BLDC motor, determines RPM based on the frequency of voltage supplied to these leads, and provides a PWM output signal corresponding to this value. The capability of the microcontroller to detect each pulse sent by the sensor is first validated by running the ASU motor and recording RPM data using both the microcontroller, and an Eagle Tree systems ELogger V4. The ELogger V4 is a data logging instrument designed for use in RC applications. It is capable of gathering
data from a variety of sensors, including power, current, airspeed, and RPM, at rates as high as 50 samples per second. The motor is run at several speeds to ensure accuracy is maintained throughout its operational range, and provides a good basis for comparing the RPM sensor reading accuracy of the Arduino-Nano to the ELogger V4. Figure 3.7 (Left) shows a noisier signal from the data gathered by the ELogger in comparison to that gathered by the microcontroller. However, the average RPM gathered at each steady state RPM is highly correlated with the raw data, with less than 1% error. The data shows that the microcontroller is properly capturing the signal pulses that are output by the RPM sensor.

The accuracy of the Eagle Tree RPM sensor is also verified by comparison to another RPM sensor. A Hall Effect sensor is used in conjunction with a permanent magnet mounted to the motor shaft. The changing magnetic flux from the rotating magnet is used to count the number of rotations and calculate RPM. Simultaneously, the Eagle Tree RPM sensor is affixed to the motor leads to also record RPM. As illustrated in Figure 3.7 (Right), RPM values at each steady-state interval show high correlation, with less than 0.1% error readings between the two sensors.

![Figure 3.7: Left: Comparison of the RPM data read using an Eagle Tree Systems ELogger and an Arduino-Nano. Right: Comparison of the RPM data read on an Arduino-Nano using a hall effect sensor and an Eagle Tree Systems RPM sensor.](image-url)
3.3.2 Power Sensor

To accurately measure the power consumption of the BLDC motor, an electrical power sensor is calibrated. The sensor is capable of measuring a maximum of 178.8 A, and 51.8 V and provides two 3.3 V scaled analog outputs proportional to the current and voltage. Current passes through the sensor from a range of 0 to 35 A by running the BLDC motor at various speeds. The amperage is read using the factory calibrated ELogger V4, and the corresponding analog voltage is measured by a microcontroller. Data is gathered by taking the average of 100 samples for current and analog readings, at six different motor RPMs. This is repeated a total of ten times to ensure accurate data. A linear function relating the analog output to the current is constructed (Figure 3.8 (Left)), with a maximum error of 1.63 A (4.87%). This relationship is described by Equation (3.1), and is used as the calibration curve relating the analog output of the sensor to the current through it.

\[ i = 0.1305 \times d + 0.8861 \]  (3.1)

While current to the motor can be actively controlled by adjusting the PWM signal, the voltage across the battery cannot. Instead, voltage is applied to the sensor with a calibrated Agilent E3630A triple-output power supply. The power supply voltage is set to a specified voltage ranging from 12-19 volts, and the corresponding sensor output voltage is recorded via a microcontroller. Voltage is adjusted in 0.5 V increments, and 100 analog data samples taken by the microcontroller at each setting. This voltage range is selected because it correlates to the typical operational voltage range of a 4S LiPo battery that is used to power the CC system under investigation. To ensure repeatability, this data collected during three separate trials. A linear relationship between the voltage and analog reading is inferred as shown in Figure 3.8 (Right), and described by Equation (3.2). This equation is used as the
calibration curve relating the output of the sensor to the voltage across it. A maximum error of 0.02 V (0.13%) is observed.

\[ V = 0.0737 \times d - 0.0128 \]  \hspace{1cm} (3.2)

![Figure 3.8: Left: Current calibration curve. Right: Voltage calibration curve.](image)

### 3.3.3 Pitot Probe

To measure \( V_j \) with accuracy, a pitot tube is built and calibrated. In order to minimize the error caused by jet blockage, the probe of the pitot tube must be equal or smaller to the slot height (h=0.4 mm). To reduce the probe size, the probe diameter is reduced, and the length is extended, using brass tubing. The small brass tubing reduces the outer diameter of the probe to 0.5 mm, with an inner diameter of 0.3 mm. The pitot is calibrated using a Flotek 360 wind tunnel with a maximum velocity of 27 m/s, as described by Kanistras et al. [12].

The pitot probe is calibrated with the wind tunnel’s conventional pitot tube and manometer. The manometer records the dynamic pressure within the 6 in x 6 in x 18 in the testing section of the wind tunnel. The pitot probe collects the total and static pressures through silicon tubes which are carried to a pressure transducer outside of the wind tunnel. The pressure transducer (Freescale Semiconductor MPXV7002) outputs an analog voltage value
proportional to the difference between total and static pressure (dynamic pressure), and this analog voltage value is read by a microcontroller. The pressure that is measured from the wind tunnel testing is plotted against the analog output of the transducer, and is shown in Figure 3.10.

\[ q_\infty = 8.2153 \times d - 0.7679 \]  

(3.3)

The function relating the analog output with the dynamic pressure \( q_\infty \) is linear, and shown in Equation (3.3). Velocity is then calculated as shown in Figure 3.9.

![Figure 3.9: Pitot probe calibration schematic [12].](image)

![Figure 3.10: Pitot probe calibration curve.](image)
Chapter 4

System Identification

This chapter describes the process and benefits of creating a model of a physical system, known as System Identification (SID). The methods of identification are described, including how they are applied to the CC system in this research through a proposed methodology. A comparison of simulation and experimental data is provided, and an accepted CC system model is presented.

System identification is the process of abstracting a physical system into a set of mathematical equations to model its behavior [55–57]. These equations can take a variety of forms, including linear and non-linear, and differential and non-differential equations, and relate the input signal to the output of a system. When differential equations are used to relate the system input and output, they are referred to as transfer functions. This model of the system is known as a “plant” and describes the open-loop characteristics of a physical system [55–57]. Implementing a mathematical simulation-based approach can offer many advantages to testing on a physical system. Tuning a controller heuristically on a physical system can be expensive, damage the hardware, or create unsafe operating conditions [5]. Building a plant model allows for the simulation of various operating conditions and controllers without the risk of damaging a physical system. The expected outcome of system
identification is to provide a framework for the testing and simulation of various controllers (including PID and Fuzzy Logic) without the need of implementation on hardware. Additionally, in this case the system models should be able to emulate the behavior of the CC system so that RPM, $V_j$, and power usage can be estimated.

There are two primary approaches that can be used to model a system. One approach involves using the physical properties of the system to construct mathematical equations that express the dynamics. This is an effective method when all of the properties are known, and all of the equations that describe the system can be solved. This method is used many places in literature, including by [13, 50, 56, 58–61] to create simulation models of BLDC motors in the MATLAB/SIMULINK, or LabVIEW environment. However, in many cases, a number of the physical properties needed to accurately represent a model in simulation are unknown. In the case of compressors, the complicated fluid interactions that occur during operation typically require Computational Fluid Dynamic (CFD) modeling, non-linear energy-transfer equations [62–64], or empirically derived characteristics to aid in modeling [46]. In the case of the BLDC motor under study, many of the physical properties are unknown (such as the dynamic friction constant, rotor inertia, phase resistance, and phase inductance), some of which (such as rotor inertia) require dismantling and measuring pieces of the motor, a task that is not often feasible.

When the equations governing a system are difficult to solve, or physical parameters are unknown, a second approach for system modeling – using the open-loop input and output (I/O) data – aids in creating a model approximating the physical plant. This I/O method allows for the creation of transfer functions to fit the dynamics of the system. For the present study, this second method is used to develop a plant to describe the functional characteristics of the CC system.
The methodology described in Figure 4.1, adapted from research by Hat et al. [5] is applied to create an acceptable system identification approximation of the CC system, using the second (I/O) approach described above.

Figure 4.1: Flowchart of the methodology used for creating a plant model of the CC system, adapted from work by Hat et al. [5].

4.1 Model Excitation and Data Collection

The first step of SID is model excitation and data gathering. Excitation functions describe the input signals that are sent to the plant in order to quantify the open-loop time-domain response. A variety of excitation functions can be utilized to evaluate I/O data and plant models, including a pseudo random binary signal [5], square wave, step input [5, 8], and sweep signals [55,57]. While square wave inputs can help gather the steady-state characteristics of a system, it insufficiently models the dynamics of a system; it is essentially a series of step responses. These step responses only truly characterize how a system responds to instantaneous changes. Instead, sine wave sweep signals (also known as a “chirp”
signal) are often applied as an input function because they allow for an understanding of the response of the system to a range of frequencies [55]. The sine sweep signal is designed as expressed by Equation (4.1), and shown in Figure 4.2.

\[
PWM = A_f \sin(2\pi(F_0 + \beta \cdot t^2)) + D, \quad \beta = (F_f - F_0)/T_f^0
\]  

(4.1)

Figure 4.2: Sine-sweep (chirp) PWM command signal used for excitation of the CC system.

This identification method relies on a control PWM input, and the output responses of RPM, \(V_j\), and power usage, to create a system plant. A third-order median filter is applied to the output response data to remove some of the errant noise within the sensor readings, as this can disrupt the accuracy of the abstracted model. Figure 4.3 shows the CC system’s RPM, \(V_j\), and power use output response to a 0-3 Hz (instantaneous frequency) sine-wave chirp signal.
Figure 4.3: a) RPM, b) $V_j$, and c) power response of the CC system to a sine-sweep PWM input.

4.2 Model Structure, Fitting, and Validation

After data is collected, various model structures are chosen and tested. The structure of a model refers to number of poles and zeros of the transfer function, in the case of linear
model approximations, but may also refer to a description of a nonlinear system. Figure 4.4 illustrates the proposed model for the system plant. In this model, each output is related to the single PWM command input through a transfer function or equation.

![Proposed model for CC system plant.](image)

**Figure 4.4: Proposed model for CC system plant.**

### 4.2.1 RPM Modeling and Validation

To fit a transfer function to the RPM data of the BLDC motor, examination of the mathematical equations that describe the physics are employed first. BLDC motors have three stator windings and permanent magnets to create rotational motion. These separate windings can create complicated, non-linear effects such as torque “cogging” that is noticeable at low RPM. However, when the mutual inductance and physical properties between stator windings are negligible, the dynamics can be described similar to those of a DC motor [57]. An illustration of this electrical circuit is shown in Figure 4.5.
The mathematical equations that describe the physics of this system can be obtained using Kirkoff’s Voltage Law for the electrical portion of the motor. The mechanical side can be obtained by summing the torque that is produced. Equation (4.2) details this relationship [57].

\[
V = L \frac{di}{dt} + Ri + e_b \\
e_b = K_t \omega \\
T = \frac{d\omega}{dt} + B\omega \\
T = K_t i
\] (4.2)

A transfer function representing the dynamics of a DC motor RPM is derived by combining these equations and expressing them in terms of the frequency domain by applying the Laplace transform, yielding a 2-pole, 0-zero transfer function (Equation (4.3)). This model structure is employed as a starting point for system identification, similar to other research [5, 13, 65].

\[
H(s) = \frac{K_t}{(Ls + R)(Js + B) + K_t^2}
\] (4.3)
Once a model structure is chosen, I/O data are characterized using MATLAB System Identification Toolbox. This toolbox provides a graphical user interface (GUI) for processing I/O data in either the frequency or time domain using functions within MATLAB. Figure 4.6 describes this GUI.

The Instrument Variable (IV) identification method is selected as the regression approach for solving the transfer function with the collected I/O data. In comparison to State Variable Filters, Subspace State Estimation, and Generalized Poisson Moment regression approaches, the IV approach tends to provide consistent solutions independent of sensor noise [56]. A second order transfer function (2-pole, 0-zero) is found to provide a sufficient fit for the RPM of the BLDC motor in relation to the PWM command input, and is described by the transfer function in Equation (4.4).
To validate the equation, the step input response between experimental data and the response predicted by the transfer function is compared. The motor is run at steady-state, with a PWM pulse width input of 1500 microseconds, which is then stepped to 1750 microseconds. This same step response is modeled in MATLAB/SIMULINK and compared to the experimental response. The models closely match, although the response of the simulation is slightly slower than that of the experimental data, and the simulation underestimates the steady-state RPM by about 0.63%, which is a close approximation. Using a Normalized Root Mean Square Error (NRMSE) for comparison, the fit between the two datasets over the studied timeframe is 71.95%, and is shown in Figure 4.7. Here noise from the experimental data can be seen, in comparison to the noise-free model. A closer fit was not found with the different transfer functions that were tested. Some error in the response and steady-state RPM is expected, since small non-linearities in physical models cannot be entirely accounted for with the linear approximation that a transfer function provides. However, the small steady-state error illustrates that the model gives a reasonable approximation for compressor speed.
4.2.2 Air Velocity ($V_j$) Modeling and Validation

A process similar to that used for creating an RPM model is applied to create a mathematical model approximating the air velocity at the plenum slot, $V_j$. Understanding the delays between the input PWM command to $V_j$ allows for $C_\mu$ to be predicted. The speed of air is related to the RPM of the compressor, and this information is considered when choosing the starting point for the form of the transfer function to describe $V_j$. Several transfer function structures are explored, with a 2-pole, 0-zero function (like that for RPM) providing the most accurate fit. This transfer function is presented in Equation (4.5).

$$H(s)_{V_j} = \frac{Y(s)_{V_j}}{U(s)_{PWM}} = \frac{3.485}{s^2 + 9.979s + 93.82}$$ (4.5)

A comparison between the experimental and simulation data for $V_j$ is created in the same method as for RPM. The motor is run at steady-state, with a PWM pulse width input of 1500 microseconds, which is then stepped to 1750 microseconds. This same step response is modeled in MATLAB/SIMULINK and compared to the experimental response.
When comparing this simulation model approximation to experimental data, it is seen that the transfer model described by Equation (4.5) overshoots the experimental data, but otherwise accurately models the $V_j$ behavior. A closer fit model is not found when testing other transfer functions. Comparing the two responses with the NRMSE method shows a fit of 67.52%, and an acceptable steady-state error of 1.03%, and is shown in Figure 4.8. These data also illustrate inaccuracies between experimental data with noise, and the theoretical, noise-free approximation.

![Figure 4.8: Comparison between $V_j$ transfer function approximation and experimental data.](image)

**4.2.3 Power Modeling and Validation**

Modeling the power consumption of the BLDC motor also provides benefits. An accurate model aids in estimating the power consumed by the CC system, without the need to physically operate the system. This is useful for estimating the power consumption by the system in varying aircraft missions, without any danger of over-draining the battery – a scenario that can permanently damage LiPo batteries. Comparing the PWM input and power output shows that power consumption is related to motor acceleration, as well as RPM. A variety of higher-order transfer function structures are applied to capture these
dynamics, however, none of those tested are able to provide a suitable fit for estimating power consumption. Using steady-state data, it is seen that RPM and power are related by a 2nd order quadratic function, as shown by Equation (4.6) and Figure 4.9. Multiplying the output of the RPM transfer function by this relationship creates an acceptable estimation for the power usage of the BLDC motor that still generally captures the dynamics.

\[ P = (8.3721 \times 10^{-7})u^2 - 0.0138u + 83.705 \]  

(4.6)

![Figure 4.9: Relationship between steady-state RPM and power consumption.](image-url)

The experimental and simulation data for power consumption is created utilizing the same method as for RPM and \( V_j \). The motor is run at steady-state, with a PWM pulse width input of 1500 microseconds, which is then stepped to 1750 microseconds. Again, the power response is modeled in MATLAB/SIMULINK and compared to the experimental response. This comparison is illustrated in Figure 4.10. In the figure, oscillations in the power consumption can be seen, and are attributed to sensor noise, as well as the switching of MOSFETs within the ESC [66]. The simulation does not capture the high-frequency dynamics of the power consumption that occur from motor acceleration – these non-linear
behaviors are not captured by the linear approximation transfer function. However, the model does provide an accurate estimation of the steady-state power consumption, with a steady-state error of 0.27% over the investigated time interval.

![Graph showing comparison between power transfer function approximation and experimental data.]

Figure 4.10: Comparison between the power transfer function approximation and experimental data.

### 4.3 Accepted Model

The accepted plant model for the CC system is presented in Figure 4.11. This plant is used as a starting point for creating PID and FL controllers capable of appropriately regulating the behavior of the CC system. The closed-loop simulation response with these controllers is later compared to experimental data in Chapter 6.
Figure 4.11: Final model for CC system plant.
Chapter 5

PID and Fuzzy Logic Controller Design and Implementation

To optimize the CC system’s behavior for power consumption and $C_r$ regulation, a controller is required. Without real-time control, CC system parameters such as motor RPM or $V_j$ cannot be accurately managed. The ability to quickly and accurately adjust the output of the CC system to dynamic changes (variations in airspeed and compressor load), is necessary for optimizing the CC system’s effectiveness, and is not possible using an open-loop system.

This chapter discusses the design and implementation of both a PID and a Fuzzy Logic (FL) controller on an Arduino microcontroller. A PID controller is chosen for this study due to its widespread use, ability to tune without complete system knowledge, and capability to rapidly test various gains. A FL controller is also chosen due to its large applicability, ability to handle nonlinearities, and good performance, even when the mathematical models of a plant are uncertain [14, 67]. The PID controller is first detailed, starting with the fundamentals of how the controller operates, then the use of simulation to further develop the controller is outlined. The design of the FL controller is then discussed – with a descrip-
tion of the fundamental operation, as well as simulation. This is followed by an explanation of the implementation of PID and FL controllers on a microcontroller, the subsequent experimental data collection, and a comparison of the simulation data to experimental data.

The performance of the controllers is evaluated by three parameters; the rise time, settling time, and overshoot, each of which is determined as an average of six runs. Rise time is defined as the time taken for the system to go from 10 to 90% of the steady-state value. Settling Time is the time taken for the system to reach a steady-state error within 2%, and overshoot is the percentage of error the system experiences if the set point is exceeded.

5.1 PID Control

PID controllers function by analyzing the error between a system’s output, and desired system’s output, known as the set point. The control signal is the sum of three error terms: the P-term, which is proportional to the error, the I-term, which is the integral of the error over time, and the D-term, which is the derivative of the error. The PID control algorithm relating the control signal to the error and controller gains is described in Equation (5.1). This equation is used as the basis for relating the control signal supplied to the ESC and the error between the motor RPM and set point. The importance of each error term is adjusted by changing the coefficient for that term, known as a gain. In Equation (5.1), these gains are represented by $K_p$, $K_i$, and $K_d$, known as proportional, integral, and derivative gains, respectively.

$$u_c(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \tag{5.1}$$

Each gain has a distinct effect on the response of the closed-loop system. Increasing $K_p$ reduces rise time, but can also increase overshoot. It also does not account for steady-
state error that can occur. An increase in $K_i$ eliminates this steady-state error, but further increases overshoot in the response. Increasing the derivative gain $K_d$ has the benefit of decreasing the overshoot in the response, but can also reduce the rise time. Table 5.1 summarizes the effects of each of these gains on the rise time, overshoot, settling time, and steady-state error of a system response [68].

Table 5.1: Closed-loop response of independently increasing PID gains.

<table>
<thead>
<tr>
<th>Gain</th>
<th>Rise Time</th>
<th>Overshoot</th>
<th>Settling Time</th>
<th>Steady State Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_p$</td>
<td>Decrease</td>
<td>Increase</td>
<td>Small Change</td>
<td>Decrease</td>
</tr>
<tr>
<td>$K_i$</td>
<td>Decrease</td>
<td>Increase</td>
<td>Increase</td>
<td>Eliminate</td>
</tr>
<tr>
<td>$K_d$</td>
<td>Small Change</td>
<td>Decrease</td>
<td>Decrease</td>
<td>No Change</td>
</tr>
</tbody>
</table>

Tuning these gains to achieve a satisfactory system response typically requires a heuristic approach, or simulation. To avoid a trial and error process, several methods exist for determining acceptable gains of a system, including the Cohen-Coon method [69], and Ziegler-Nichols method [1]. The Cohen-Coon method is an open loop method whereby gains are determined using the response and delay of the step-response of a system [69]. This method is not considered favorable in this application however, since conventional wisdom indicates that the PID gains it produces are more suitable for systems with large delays [70]. Therefore, the Ziegler-Nichols method is used to create a starting point for stable CC system control. The procedure outlined by this method involves increasing the potential gain of the controller until the system reaches a steady-state oscillation, and measuring both this gain value (ultimate sensitivity, $S_u$), and oscillation period (ultimate period, $P_u$). These values are then used as a guideline for deriving a set of proportional, integral, and derivative gains [1]. How each of these gains relates to $S_u$ and $P_u$ varies, depending whether a P, PI, or PID control scheme is used, and is detailed in Table 5.2.

Initial gains are created for PID control of the CC system using this method. The potential gain is increased until steady-state oscillation is found to occur at $K_p = 0.12$. The
oscillation period is 0.465 s and these values are used for $S_u$ and $P_u$, yielding initial PID gains of $K_p = 0.072$, $K_i = 0.232$, and $K_d = 0.058$. Upon implementation, it is observed that the addition of a derivative gain imparts a slower response, and instability to the system. Thus, this term is removed, effectively creating a PI control scheme for regulating the CC system. The PI gains used as a starting point for control of the CC system are $K_p = 0.054$ and $K_i = 0.387$, and provide sufficient response. However, further tuning of these gains is necessary to reduce the rise time of the system.

Table 5.2: Ziegler-Nichols PID tuning rules [1].

<table>
<thead>
<tr>
<th>Controller Type</th>
<th>$K_p$</th>
<th>$K_i$</th>
<th>$K_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.5$S_u$</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>PI</td>
<td>0.455$S_u$</td>
<td>$P_u/1.2$</td>
<td>—</td>
</tr>
<tr>
<td>PID</td>
<td>0.65$S_u$</td>
<td>$P_u/2.0$</td>
<td>$P_u/8$</td>
</tr>
</tbody>
</table>

5.1.1 PID Controller Simulation

Simulation in the MATLAB/Simulink environment is performed to further tune the PI gains of the controller. Using the plant model created from SID, a controller model is simulated. By creating a representation of the PID controller, plant, and feedback loop, different inputs are applied to predict and study the steady-state and transient response of the CC-system. Figure 5.1 shows the MATLAB/Simulink representation of this simulation.

![Figure 5.1: A MATLAB/Simulink block diagram used for simulation of the PID controller.](image-url)
From simulation, gains of $K_p = 0.09$ and $K_i = 1.4$ are determined, providing a balance between a quick response and minimal overshoot. These gains differ from those produced using the Ziegler-Nichols method due to the response that is desired. While the Ziegler-Nichols method provides a rapid way of producing a reliable set of gains, this conservative approach is not always appropriate in systems where a fast response time is desired. The gains determined in simulation are also applied to the real-time controller of the physical system.

### 5.2 Fuzzy Logic Control

Fuzzy Logic control is a rule-based control strategy that emulates the intuitive human understanding of a system. It is based on a set of linguistic rules to regulate a system, rather than real numbers. To illustrate this concept, an example for regulating a vehicle’s speed by a human is presented by Passino: “If speed is lower than the set point, then press down further on the accelerator pedal.” [14]. This idea of expressing the behavior of a system using linguistic variables is the primary operating principle of FL control.

Fuzzy Logic controllers operate in three main steps: Fuzzification, Inference, and Defuzzification [4, 39, 50, 71]. This process is illustrated in Figure 5.2.

![Figure 5.2: A schematic illustrating the architecture of a Fuzzy Logic controller [14].](image)
Fuzzification is the process of converting a real number (or “crisp”) value to a linguistic variable. This is achieved using what is known as Membership Functions (MFs), which describe the linguistic space in which a crisp value lays. The linguistic names of MFs are typically expressed as “positive big”, “negative small”, or similar terms. A crisp input value can belong to multiple membership functions at a time, and these functions provide a graphical representation of the degree of this membership. A variety of MF shapes can be chosen for the fuzzification step, as long as it can be used as a continuous probability function. Possible MFs include: triangular, trapezoidal, Gaussian, sigmoid, and others, as illustrated by Figure 5.3 [15,72]. Each of these provides differences in performance, which are important to consider.

Figure 5.3: Possible membership function geometries as presented by Zhao et al. [15].
From the range of possible MFs, a triangular MF type is chosen, as research shows better performance with this shape in comparison to trapezoidal and Gaussian functions. Research by Ali et al. [72] and Monicka et al. [73] show better steady-state behavior and lower error when using triangular MFs. Additionally, it is suggested by Monicka et al. [73] that functions with a smaller nucleus (central focal point), such as triangular, which has a single-point nucleus, exhibit smaller steady-state error, and that the computational efficiency of simple MFs like triangular functions are better for real-time implementation [72]. Better control of motor speed is also exhibited by triangular MFs in relation to trapezoidal, Gaussian, sigmoidal, and other MFs by Zhao et al. [15]. Review of the research indicates that triangular membership functions are used almost exclusively for the control of BLDC and DC motors [7, 13, 49, 50, 71, 74, 75].

In addition to shape, the number of MFs is also important to consider. A larger number of MFs correlates to a higher resolution controller – a controller with a finer level of control. However, a larger number of MFs implies more fuzzy rules and larger computational and memory requirements [50], which can negatively impact the controller speed when implemented on a microcontroller. Adequate control of a motor is exhibited in research by FL controllers with both five [13, 15, 71, 74], and seven [7, 8, 49, 50, 72, 73, 75] MFs, so both are simulated in MATLAB/Simulink for this study in order to find a balance between accuracy and computation speed, and is described in greater detail later. Ultimately, a five MF set is chosen for fuzzification, as simulation results show no large degradation in controller capabilities (Figure 5.4).

Two inputs – error, and the change of error – are chosen as inputs to the FL controller, and are shown in literature to provide adequate information to the controller for regulation of BLDC and DC motors [7, 13, 15, 49, 50, 73]. Figure 5.4 illustrates the MFs for the FL controller inputs. These MFs are normalized between -1 and 1 and evenly distributed throughout the range.
The second step performed by the FL controller is inference. This is the application of fuzzy rules from a human knowledge base, and is expressed with linguistic variables. A set of rules is applied, in the form of IF-THEN statements, that relate the IF (“antecedent”) linguistic variable to a THEN (“consequent”) rule. Derived from the membership functions others have applied to control BLDC motors the set of fuzzy rules in Table 5.3 is employed [7, 13, 15, 49, 50, 73].

Table 5.3: Table of linguistic rules for the error (e) and change in error (Δe) inputs of the Fuzzy Logic Controller.

<table>
<thead>
<tr>
<th>e \ Δe</th>
<th>NB</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NS</td>
<td>NS</td>
<td>Z</td>
</tr>
<tr>
<td>NS</td>
<td>NB</td>
<td>NS</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
</tr>
<tr>
<td>Z</td>
<td>NS</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PS</td>
</tr>
<tr>
<td>PS</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PS</td>
<td>PB</td>
</tr>
<tr>
<td>PB</td>
<td>Z</td>
<td>PS</td>
<td>PS</td>
<td>PB</td>
<td>PB</td>
</tr>
</tbody>
</table>

Defuzzification is the final step in FL control, and relates the degree of membership the inputs have to the output MFs back to a crisp (numeric) value. This step can be calculated using Sugeno or Mamdani type inference [7, 49, 73, 76]. Sugeno inference does not rely on output MFs, but rather an equation that creates a crisp value output based on the “rule strength” of each input – that is, how well it fits an input MF [76]. However, due to its intuitive nature and widespread use, the Mamdani inference method is applied here. The
Mamdani method determines the degree of membership the inputs have, and relates that to a combined area of the output MFs. This is best illustrated visually, and shown in Figure 5.5.

Using the Mamdani method, computing a crisp value from the output membership functions can be applied in several ways, including the centroid (center of gravity), center average, mean of maxima, and numerous others methods [77]. The centroid method is most popular, and is applied here [14]. This method computes the area of the output MFs to which the fuzzified inputs belong, finds the center of this area, and relates it to a specific, crisp, number as a control output. Equation (5.2) shows how a crisp output is computed using the centroid method, where \( b \) is the value of the center of each MF, and \( \int \mu_i \) is the area under each membership function.

\[
\begin{align*}
    u_{crisp} &= \frac{\sum_i b_i \int \mu_i}{\sum_i \int \mu_i} \\
    & \quad \text{(5.2)}
\end{align*}
\]

This method is further illustrated in Figure 5.6, and is beneficial for use in physical models because it limits the range of outputs to the system; the centroid is never outside the range of the membership functions [14].
5.2.1 Fuzzy Logic Controller Simulation

Simulation of the FL controller allows for testing prior to implementation on a microcontroller, and provides validation that the proposed fuzzy rule set is capable of regulating the BLDC motor. To simulate the FL controller, the MATLAB Fuzzy Logic toolbox is
employed. The toolbox allows for the creation of FL controllers using a GUI that applies various MATLAB functions. Shown in Figure 5.8, this interface allows the user to specify any number of inputs and outputs, the number and shape of the MFs, and the method in which fuzzy sets are combined and a crisp output obtained.

![Fuzzy Logic Toolbox GUI](image)

**Figure 5.8**: The MATLAB Fuzzy Logic Toolbox GUI.

This designed controller is simulated in the MATLAB/Simulink environment, similar to the method used for simulation of the PID controller. Figure 5.9 shows the schematic of the controller, plant, and feedback. $K_s$ and $K_{dc}$ are used to normalize the inputs to the -1 to 1 range of the input MFs, and $K_c$ is used to scale the controller output to the 900-2100 microsecond PWM pulse width range that the plant is identified from.
In simulation, two-input FL controllers are tested with both five MF and seven MF sets for each of the inputs, creating 25, and 49 fuzzy rule sets, respectively. These are both run in simulation to compare their ability to accurately control the CC system model plant. A step input representative of the maximum change the ASU motor may experience, from 0 to 26,000 RPM is applied (Figure 5.10). The two controllers perform similarly, with almost identical rise times. The seven MF controller has slightly better transient performance with a lower settling time, but does experience a small amount of overshoot. Due to these similarities, a five MF Fuzzy Logic control is selected to reduce the computation time.

Figure 5.9: A MATLAB/Simulink block diagram used for simulation of the FL controller.
Figure 5.10: MATLAB/Simulink simulation of five MF and seven MF FL controllers.

Table 5.4: Response characteristics of 5 MF and 7 MF Fuzzy Logic controllers.

<table>
<thead>
<tr>
<th>Fuzzy Logic Controller</th>
<th>Rise Time (s)</th>
<th>Settling Time (s)</th>
<th>Overshoot (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 MF</td>
<td>0.1342</td>
<td>0.8500</td>
<td>0.0000</td>
</tr>
<tr>
<td>7 MF</td>
<td>0.1413</td>
<td>0.7101</td>
<td>0.7430</td>
</tr>
</tbody>
</table>

5.3 Controller Implementation

Real-time control of the CC system is achieved through the use of an Arduino-Nano microcontroller interfaced to the ESC, and the ASU and ADS sensors detailed in Chapter 3. In order to communicate with the microcontroller, a LabVIEW data acquisition program is designed. The program provides a GUI for adjusting parameters on the controller, as well as collecting time, RPM, setpoint, power, and $V_j$ data in real-time using the microcontroller’s serial port. Data is collected with each loop of code on the microcontroller and stored in a file on the host computer. Figure 5.11 further describes this GUI.
The microcontroller utilized for the study is interfaced to the CC system with an RPM sensor, pitot probe, and power sensor. The output control signal is a PWM signal that is fed to the ESC of the ASU motor, which in turn modulates the frequency of electrical energy supplied to the motor phases, and therefore regulates RPM. A schematic of this architecture is presented in Figure 5.12.

![Schematic Representation of the controller interfacing to the CC system.](image)

Given that microcontrollers are discrete time devices – that is, computation occurs at specific intervals and time is treated as a sequence of quantities rather than as a continuous function – control systems must be designed to compute control signals in a discrete time manner. The PID controller code is designed to loop indefinitely on the microcontroller,
and computes the control signal within each loop. The RPM is read over a 10 ms interval and compared to the set point command to calculate the error value. During each loop the proportional, integral, and derivative error is computed. Proportional error is calculated by multiplying the error $K_p$. The integral error is created by multiplying the error by the $K_i$ gain and elapsed time, and adding that value to the integral error calculated from the previous loop. This continuous summing of error is a discrete time integral and shown in Equation (5.3). The derivative error, although unused here, is calculated by multiplying the error by $K_d$, subtracting from the previous error, and dividing by the elapsed time (Equation (5.4)). The three errors: proportional, integral, and derivative are then summed to create the control signal. This controller runs at a rate of 20 Hz on the microcontroller.

$$\sum_{i=1}^{k} e(t_i) \Delta t$$  \hspace{1cm} (5.3)

$$\frac{e(t_i) - e(t_{i-1})}{\Delta t}$$  \hspace{1cm} (5.4)

The behavior of the PID controller and simulated CC system is compared to experimental tests of the control system in actual use, and a good correlation is found between the simulation and experimental results. During these tests, the motor is run at a steady-state 8,000 RPM, and then a step function to 26,000 RPM is commanded. Figure 5.13 shows the comparison of the average experimental response of six trials, as well as the simulation response. The simulation exhibits less response delay than the experimental data, but both exhibit similar rise time characteristics. The simulation settling time is lower than that of the experimental data, due to oscillations caused by sensor noise that are not represented in simulation. Applying a filter can reduce noise, but also introduce a phase shift that slows down the response of the controller [78]. These response characteristics are further detailed in Table 5.5. Over the 3 second interval studied, the fit between both responses is found to
be 79.78%, using the NRMSE method. A perfect fit is not expected, due to differences that are not captured between the linearized plant model approximation and physical system, however the comparison shows that the gains determined in simulation can be transferred to the physical system and provide a similar response.

![Graph showing simulation and experimental response](image)

**Figure 5.13:** Simulation and experimental response of the CC system with a PID controller subjected to an 8,000 to 26,000 RPM step input.

**Table 5.5: PID controller simulation and experimental response characteristics.**

<table>
<thead>
<tr>
<th></th>
<th>Rise Time (s)</th>
<th>Settling Time (s)</th>
<th>Overshoot (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation</td>
<td>0.1414</td>
<td>0.9126</td>
<td>0.0000</td>
</tr>
<tr>
<td>Experimental</td>
<td>0.1696</td>
<td>1.5206</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

The FL controller is also implemented on the microcontroller. In a discrete time environment it operates in much the same way as in simulation. As the code runs the controller, it repeatedly calculates the error (e), and change in error (Δe), and determines the magnitude in which each correlate to their respective input MFs. This is done by comparing the value of the inputs to the set of MFs in which they belong. The rule base is applied and these values are transferred to their respective output MFs. The center of the area the val-
ues belong to is then calculated (centroid method), using Equation (5.2), and a crisp value obtained. The controller code is designed to loop indefinitely, and the five MF controller computes the fuzzification, inference, and defuzzification at a rate of 24 Hz – 11% faster than the seven MF version.

Figure 5.14 shows the comparison between the average experimental response of six trials, and the simulation response. The simulation response shows no delay when given the step input of 26,000 RPM, while the experimental response shows an average 108 ms delay. Despite this, the experimental data shows a rise time that is shorter than the simulation. The experimental results also exhibit a shorter settling time in comparison to that of the simulation data.

![Figure 5.14: Simulation and experimental response of the CC system with a FL controller subjected to a 8,000 to 26,000 RPM step input.](image)

Both data sets show resonable correlation between their responses, and the fit between both is found to be 66.69%, using the NRMSE method. This is due to the response delay in the experimental system response that is not present in the simulation data. The response characteristics between the two data sets show similar rise times, but a shorter settling time
for the experimental data. A perfect fit is not expected, again due to intricacies in the physical system that the simplified plant model is unable to capture, as well as the behavior of discrete time systems that is not represented in simulation. Despite the differences between simulation and experimental responses, the FL controller is shown to adequately control the ASU motor.

Table 5.6: FL controller simulation and experimental response characteristics.

<table>
<thead>
<tr>
<th></th>
<th>Rise Time (s)</th>
<th>Settling Time (s)</th>
<th>Overshoot (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation</td>
<td>0.5542</td>
<td>1.1602</td>
<td>0.0000</td>
</tr>
<tr>
<td>Experimental</td>
<td>0.4208</td>
<td>0.7169</td>
<td>0.9476</td>
</tr>
</tbody>
</table>

Figure 5.15: Experimental response comparison of the CC system with a FL and PID controller, subjected to an 8,000 to 26,000 RPM step input.

Based on the scenarios tested in this aspect of the study, each controller performed similarly in response to an 8,000 to 26,000 RPM step response, as illustrated in Figure 5.15. The rise time of the PID controller is almost 2.5 times shorter than the FL controller. However, the FL controller exhibits a settling time that is almost half as long as the PID controller,
and the difference in overshoot is negligible. Additionally, in the data collected, the FL controller experiences less oscillation during steady-state. The controllers performance with respect to energy and power consumption during different flight scenarios is described in Chapter 6.
Chapter 6

Results and Discussion

This chapter discusses the application of both of the PID and FL controllers to the UC2AV’s CC system. Lab testing is performed to simulate the operation of the CC system during three phases of flight (takeoff, cruise and landing), using the testbed and instrumentation described in Chapter 3. By studying the energy consumption during the phases of flight, recommendations are made, creating a power planning flowchart for optimizing the CC system’s power consumption, dependent on the mission.

Numerous missions are possible during the flight of UC2AV, some of which necessitate the use of CC. Figure 6.1 provides an illustration of the possible flight scenarios of the UC2AV during takeoff, cruise, and landing.

![Figure 6.1: Roadmap detailing the different possible phases of flight for the UC2AV.](image)

66
During takeoff, a CC system serves two purposes – to either reduce takeoff distance using the augmented lift capability of the CC system, or to use the augmented lift benefit to carry payload above the maximum takeoff weight (MTOW) that is initially designed for.

During cruise flight, operating the CC system is not necessarily required. For instance, in the scenario where the extra lift is used to reduce takeoff distance, CC control may not be needed to maintain lift during cruise. However, if applied to increase the aircraft payload weight capacity, CC will be required during cruise flight to maintain sufficient lift.

While in the cruise phase of flight, alternate flight situations may also arise requiring CC. Aerial photography can benefit from slower airspeeds when actively acquiring photos, and the augmented lift capabilities of the UC²AV allow for this. In this situation, maximum lift augmentation would be required to minimize the airspeed while maintaining sufficient lift to prevent stall. Figure 6.2 illustrates this flight scenario. Another possible flight scenario is payload capture. During cruise flight, the augmented lift capabilities of CC can allow for the UC²AV to slow while collecting additional payload above the initial MTOW of the aircraft without CC.

Figure 6.2: Illustration of a mission incorporating aerial photography during cruise flight.

While landing, a CC system serves the same purposes as for takeoff. In addition, if the UC²AV is carrying weight above what the Anaconda was initially designed for, CC would be employed to maintain lift during landing.
The energy use of the ASU during each of these flight phases is investigated through the application of a PID controller and FL controller to regulate the behavior of the CC system such that energy consumption is reduced.

### 6.1 Takeoff Phase

Optimization of the CC system’s power consumption is first approached using a controller during a simulated takeoff. Takeoff is studied in two phases; the **ground phase**, which begins when takeoff is initiated, and ends when the aircraft becomes airborne, and the **air phase**, which is the portion of flight from when the aircraft leaves the ground until it reaches an altitude of 15 m (50 ft). Figure 6.3 illustrates the takeoff phases.

![Illustration of takeoff phases.](Image)

During the takeoff phase, it is determined that $C_\mu$ must reach a maximum value (corresponding to maximum lift enhancement, and also maximum ASU motor RPM) before the UAV becomes airborne [27]. A step function described by Equation (6.1), and a ramp function described by Equation (6.2) are applied as RPM set point inputs to the controller to simulate methods of initiating CC during takeoff.

$$x(t) = \begin{cases} 8000 & t < T_s \\ 26000 & t \geq T_s \end{cases} \quad (6.1)$$
In Equation (6.1), the set point is 8,000 RPM prior to time $T$, and steps to 26,000 RPM after. In Equation (6.2) the set point is 8,000 RPM until the initial time of the ramp function ($T_0$), at which point the function rises to a value of 26,000 RPM over the rise time ($t_r$), and is then maintained.

Previous flight test results have shown that the ground phase takes approximately 6 seconds [27]. Thus, for the linear ramp function input, ramp rise times ($t_r$) of 0, 3, and 6 seconds are chosen. For both input functions, data are collected over six runs, and the average of the RPM response and energy usage is calculated. The energy usage for the remainder of the takeoff phase is calculated by integrating the steady-state power use of the motor over time until the time the UAV becomes airborne.

Each time the motor is run, a start-up sequence activates to reliably bring the motor to its minimum operating point. This sequence consists of a brief period of defined controller output until an RPM of 8,000 is reached, at which point the start-up sequence is terminated and the controller engaged. The characteristics of this sequence are illustrated in Figure 6.4.

![Characterization of motor start-up sequence.](image-url)
6.1.1 PID Control During Takeoff Phase

A PID controller is applied to the aforementioned takeoff scenarios to investigate which control scheme best optimizes the power consumption of the CC system. The response of the PID controller and CC system behavior are tuned, producing three different settling times all within the duration of the takeoff ground phase. The PID controller gains are determined experimentally and are presented in Table 6.1.

The data summarized in Table 6.1 show the performance of the CC system with a step input applied to three different sets of PID controller gains, and also with three different ramp inputs applied. In the case where a ramp with a 0 second rise time is applied, data for a step input is considered; an instantaneous ramp function is a step function.

The left plot in Figure 6.5 shows the response of the CC system to a step input from 8,000 RPM (the idle speed of the motor, and speed immediately following the boot-up sequence) to 26,000 RPM, for three sets of gains. The right plot in Figure 6.5 shows the response when three different inputs are given: a step from 8,000 to 26,000 RPM, a ramp from 8,000 to 26,000 RPM over a span of three seconds, and a ramp from 8,000 to 26,000 RPM over a span of six seconds.

Figure 6.5: Left: Motor response with three sets of PID gains. Right: Motor response with three sets of ramp inputs using a PID controller.
Table 6.1: Takeoff performance data.

<table>
<thead>
<tr>
<th>Case</th>
<th>Gains</th>
<th>Rise Time (s)</th>
<th>Settling Time (s)</th>
<th>Overshoot (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Step Input (8,000 - 26,000 RPM)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1: Step</td>
<td>$K_p : 0.001, K_i : 0.18$</td>
<td>3.0173</td>
<td>5.5759</td>
<td>0.00</td>
</tr>
<tr>
<td>Case 2: Step</td>
<td>$K_p : 0.01, K_i : 0.29$</td>
<td>1.3824</td>
<td>2.8628</td>
<td>0.08</td>
</tr>
<tr>
<td>Case 3: Step</td>
<td>$K_p : 0.09, K_i : 1.4$</td>
<td>0.1664</td>
<td>0.4541</td>
<td>4.15</td>
</tr>
<tr>
<td><strong>Ramp Input (8,000 - 26,000 RPM)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1: 0 s Ramp</td>
<td>Same as Case 3: Step</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2: 3 s Ramp</td>
<td>$K_p : 0.07, K_i : 1.1$</td>
<td>2.4483</td>
<td>3.7783</td>
<td>1.23</td>
</tr>
<tr>
<td>Case 3: 6 s Ramp</td>
<td>$K_p : 0.07, K_i : 1.1$</td>
<td>4.9257</td>
<td>6.3659</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### 6.1.2 Fuzzy Control During Takeoff Phase

The application of a FL controller during takeoff is also investigated. The three different set points that are input to the PID controller are again utilized for the FL controller: a step from 8,000 to 26,000 RPM, a ramp from 8,000 to 26,000 RPM with $t_r = 3$ s, and a ramp from 8,000 to 26,000 RPM with $t_r = 6$ s. The plot in Figure 6.6 shows the CC system’s response to the three different ramp inputs, where the FL controller adequately tracks the set point. In each case the overshoot is minimal, and settling time is similar to the ramp input $t_r$. The data summarized in Table 6.2 shows the response characteristics of the CC system in further detail. Different gains for a single step input are not explored as they were for the PID controller, since investigations make it clear that the ramp input at the rise times of 3 and 6 seconds use less energy than their step input counterparts tuned to have similar settling times.
Figure 6.6: Motor response with three sets of ramp inputs using a FL controller.

Table 6.2: Response characteristics of the FL controller during takeoff.

<table>
<thead>
<tr>
<th>Case</th>
<th>Rise Time (s)</th>
<th>Settling Time (s)</th>
<th>Overshoot (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1: 0 s Ramp</td>
<td>0.4208</td>
<td>0.7169</td>
<td>0.8901</td>
</tr>
<tr>
<td>Case 2: 3 s Ramp</td>
<td>2.3767</td>
<td>3.0317</td>
<td>0.6974</td>
</tr>
<tr>
<td>Case 3: 6 s Ramp</td>
<td>4.8612</td>
<td>6.0016</td>
<td>1.1818</td>
</tr>
</tbody>
</table>

Comparing the response of both the PID and FL controllers to the three ramp inputs shows that the PID controller responds with faster rise times for case 1 and case 2, and faster settling times for all cases. Fuzzy logic control exhibits less overshoot in general, especially when subjected to a step input. Here the overshoot for FL control is less than 1%, and is larger than 4% for the PID controller. Despite these differences, both controllers respond similarly, each reaching steady-state shortly after the ramp rise time.

6.1.3 Energy Consumption During Takeoff

Two approaches are utilized to study the overall energy consumption during takeoff: I.) the ASU motor and controller are initiated at the beginning of takeoff (prior to UAV acceleration), and II.) when the ASU motor is set to maximum motor RPM (and $C_\mu$) at
the time the UAV is expected to leave the ground. These approaches are illustrated in Figure 6.7, which shows a representative RPM profile of the motor during takeoff.

In Approach-I, the takeoff sequence is assumed to consist of the motor start-up, immediately followed by one of the input scenarios detailed in Table 6.1 or Table 6.2 as well as maximum thrust by the UAV’s propeller, then, the ASU motor is run at maximum RPM until the end of the takeoff phase. In Approach-II takeoff is initiated when maximum thrust is given to the UAV propeller, the ASU motor initiated at a time such that it reaches maximum RPM right at the expected time of liftoff, and run at maximum RPM until the termination of the takeoff phase. When the motor reaches the maximum RPM of 26,000, the average power consumption is 260.34 W in steady-state.

Based on previous experimental data, the total time of takeoff is found to be 15 seconds [27]. This data, as well as the power usage of the motor at maximum RPM, is used to calculate the energy consumption for the remainder of takeoff. Total energy consumption is calculated as the sum of the start-up energy, case-specific energy, and energy used to reach an altitude of 15 m (50 ft). The left plot in Figure 6.8 shows the power consumed for the three ramp scenarios using the PID controller, and the right shows the power consumption for the same ramp scenarios with the FL controller. The step response causes the largest power spike in both controllers, reaching almost 750 W for the PID controller. In both
plots the power consumption can be seen to converge to the steady-state value previously mentioned, as the motor RPM stabilizes. Oscillation in power is also seen, due to the switching of the MOSFETs within the speed controller, as mentioned in Chapter 4. The calculated amount of energy consumed for each step and ramp input, and each takeoff approach is detailed in Table 6.3.

Table 6.3: Energy (Joules) consumed for different motor initiation times.

<table>
<thead>
<tr>
<th>Case</th>
<th>Approach-I: Energy Consumption (Joules)</th>
<th>Approach-II: Energy Consumption (Joules)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID Controller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1: Step</td>
<td>3474.58</td>
<td>3080.55</td>
</tr>
<tr>
<td>Case 2: Step</td>
<td>3506.70</td>
<td>2429.61</td>
</tr>
<tr>
<td>Case 3: Step</td>
<td>3965.46</td>
<td>2261.29</td>
</tr>
<tr>
<td>Case 1: 0 s Ramp</td>
<td>Same as Case 3: Step</td>
<td></td>
</tr>
<tr>
<td>Case 2: 3 s Ramp</td>
<td>3471.90</td>
<td>2631.93</td>
</tr>
<tr>
<td>Case 3: 6 s Ramp</td>
<td>3138.77</td>
<td>2973.69</td>
</tr>
<tr>
<td>Fuzzy Logic Controller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1: 0 s Ramp</td>
<td>3984.20</td>
<td>2271.36</td>
</tr>
<tr>
<td>Case 2: 3 s Ramp</td>
<td>3503.67</td>
<td>2470.55</td>
</tr>
<tr>
<td>Case 3: 6 s Ramp</td>
<td>2342.66</td>
<td>2803.66</td>
</tr>
</tbody>
</table>

When the motor is initiated at the beginning of the takeoff phase (Approach-I), the controller scheme resulting in minimal energy consumption of the cases attempted is a 6 second ramp input (Case 3: 6 s Ramp). With PID control, the CC system consumes
3138.77 J, and with FL control 2342.66 J is consumed during the takeoff phase. However, when Approach-II is applied, the optimal control scheme consists of a step input, with controller gains set to provide the fastest response time (Case 3: Step/Case 1: 0 s Ramp). With PID control, 2261.29 J of energy is consumed by the CC system, and 2271.36 J with FL control. Examination of the PID controller shows that the use of ramp inputs uses less energy than the application of various gains to achieve the same rise time. When examining the various ramp cases and approach methods between the two controllers, it is seen that the energy consumption is similar. In many of the cases, the difference in energy consumption between the controllers is less than 200 J. The close correlation of these data imply that both PID and FL controllers affect the behavior of the CC system’s energy consumption in the same way during takeoff simulation. The results also show that independent of PID or FL control, initiating the ASU motor and CC system as late as possible, while still meeting the RPM and \( V_j \) requirements for takeoff (Approach-II) is more energy efficient than initiating the motor at the beginning of the takeoff phase (Approach-I). Additional considerations aside from energy consumption must also be taken to decide the optimal control scheme. Effects that cannot be accurately measured, including flow adherence to the flap; if flow has become fully developed; and \( C_\mu \) interaction with ground effects will require further in-flight testing, because they may influence the optimal time of the motor initialization.

6.2 Cruise Flight Phase

The next phase of flight that is investigated with regard to power optimization is cruise flight. Cruise flight is the primary phase of flight in terms of both duration and energy consumption. It is the phase in which altitude, angle of attack, and airspeed is maintained at a constant value to maximize aerodynamic and energy efficiency of the aircraft. During cruise flight of the UC\(^2\)AV, operation of the CC system is not necessarily required. How-
ever, when the augmented lift capabilities of CC are used to increase payload capacity, CC is required to maintain lift throughout cruise.

During the operation of CC during cruise flight, the largest factor is maintaining a constant level of lift augmentation. This is achieved by regulating $V_j$ such that a constant $C_\mu$ value is achieved (Equation (2.1)). The typical airspeed of the UC$^2$AV during cruise flight is 20 m/s [28], and is used for the value of $V_\infty$ during simulation of this flight phase. Similar airspeed values are also used by Osborne et al. [79] to simulate the flight of an Aerosonde UAV. In research by Brezoescu et al. [80], variable wind gusts of 2 m/s are applied to a 1.42 m wingspan UAV model to demonstrate controller robustness. This 2 m/s variation in airspeed due to gusts also correlates well to the variation in data collected by Kanistras [28], and so is used to model perturbations in the UC$^2$AV’s airspeed.

Three $C_\mu$ values are chosen to represent the maximum, intermediate, and minimum operating conditions of the CC system’s ASU motor. For nominal airspeeds of 20 m/s, $C_\mu$ values of 0.016, 0.011, 0.005 correlate to operational ASU motor speeds of 22,700, 18,000, and 10,600 RPM, respectively. An ASU motor RPM of 22,700 correlates to $V_j = 70$ m/s, 18,000 RPM to $V_j = 57$ m/s, and 10,600 RPM to $V_j = 37$ m/s, using Equation (2.1). To maintain a specified $C_\mu$ with $V_\infty = 20$ m/s despite ±2 m/s perturbations in airspeed, the controller must quickly respond by increasing or decreasing the motor RPM. Table 6.4 shows the $C_\mu$ value, nominal motor RPM, and required change in motor RPM to maintain a constant $C_\mu$ with airspeed variations.

Table 6.4: Fluctuations in airspeed and required changes in RPM to maintain a constant $C_\mu$.

<table>
<thead>
<tr>
<th>$C_\mu$ Value</th>
<th>Nominal RPM</th>
<th>+2 m/s Airspeed RPM</th>
<th>-2 m/s Airspeed RPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: $C_\mu = 0.016$</td>
<td>22,700</td>
<td>26,000</td>
<td>20,100</td>
</tr>
<tr>
<td>2: $C_\mu = 0.011$</td>
<td>18,000</td>
<td>20,100</td>
<td>15,900</td>
</tr>
<tr>
<td>3: $C_\mu = 0.005$</td>
<td>10,600</td>
<td>12,000</td>
<td>10,000</td>
</tr>
</tbody>
</table>
6.2.1 PID and FL Control During Cruise Flight

The RPM data in Table 6.4 is utilized to construct step inputs for the CC system’s controller. The ASU motor is first run at steady-state at the nominal RPM, and then stepped up or down to the value needed to maintain a constant $C_\mu$ during a gust. Both PID and FL controllers are investigated separately using these inputs, and the response created by averaging six runs.

The RPM and power response to an increasing step for each controller is shown in Figure 6.9 and Figure 6.10. The left plots illustrate the PID controller response, and the right plots illustrate the FL controller response. Both controllers exhibit adequate tracking; however, the PID controller has a faster rise time and less overshoot than the FL controller in all cases. Despite the shorter rise times, the PID controller has a longer settling time except for the $C_\mu = 0.005$ case. These response characteristics are further detailed in Table 6.5. At higher RPMs the PID controller oscillates more than the FL controller, however, the transient response of the FL controller at lower RPMs is much slower than that of the PID controller. As seen in Figure 6.10, the power response between the two controllers is similar, however, the short rise time of the PID controller translates into high power consumption during acceleration, which is seen by the large spikes in power. Despite these spikes in power consumption, the energy consumed while the CC system’s motor reaches steady-state is about the same for both controllers except for the $C_\mu = 0.005$ case, as detailed in Table 6.6. During control at low RPM, the slow transient response of the FL controller translates into more than double the energy usage to reach steady-state than the PID controller.
Figure 6.9: Left: RPM response of PID controller with three $C_\mu$ goals. Right: RPM response of FL controller with three $C_\mu$ goals.

Figure 6.10: Left: Power response of PID controller with three $C_\mu$ goals. Right: Power response of FL controller with three $C_\mu$ goals.

Table 6.5: Response characteristics of the PID and FL controllers to a 2 m/s increase in airspeed.

<table>
<thead>
<tr>
<th>$C_\mu$ and Step Value</th>
<th>PID Controller</th>
<th>Fuzzy Logic Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rise Time (s)</td>
<td>Settling Time (s)</td>
</tr>
<tr>
<td>1: $C_\mu = 0.016$</td>
<td>0.0821</td>
<td>1.6844</td>
</tr>
<tr>
<td>Step: 22,700–26,000 RPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2: $C_\mu = 0.011$</td>
<td>0.1184</td>
<td>1.4353</td>
</tr>
<tr>
<td>Step: 18,000–20,100 RPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3: $C_\mu = 0.005$</td>
<td>0.1308</td>
<td>1.0133</td>
</tr>
<tr>
<td>Step: 10,600–12,000 RPM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A decreasing step function following the RPM steps in Table 6.4 is also studied. The response of the CC system with this input is applied to both controllers, and the average
Table 6.6: Power consumption between PID and FL controllers during a 2 m/s increase in airspeed.

<table>
<thead>
<tr>
<th>Step Value</th>
<th>PID Controller Energy Consumption (Joules)</th>
<th>FL Controller Energy Consumption (Joules)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: $\mu = 0.016$ Step: 22,700–26,000 RPM</td>
<td>227.53</td>
<td>276.05</td>
</tr>
<tr>
<td>2: $\mu = 0.011$ Step: 18,000–20,100 RPM</td>
<td>194.70</td>
<td>191.33</td>
</tr>
<tr>
<td>3: $\mu = 0.005$ Step: 10,600–12,000 RPM</td>
<td>43.60</td>
<td>108.56</td>
</tr>
</tbody>
</table>

response of six tests is shown in Figure 6.11, where it is seen that the PID controller exhibits no overshoot during this scenario. An absence of overshoot is desirable in terms of energy efficiency, because no additional energy is required to reach the set point. Table 6.7 details the response characteristics of both controllers. The PID control of the CC system during the three $\mu$ tracking scenarios exhibits faster rise times, and no overshoot in comparison to the FL controller. The PID controller also outperforms with regard to shorter settling times in all cases except the decreasing step response from 22,700–20,100 RPM. Figure 6.12 shows the power response of both controllers. The response is similar at steady state, however the PID controller shows a larger drop in power consumption when the motor is decelerating. This is due to differences in the output command PWM signal between the controllers. Table 6.8 details the power consumed while each controller adjusts the ASU motor speed to account for the airspeed decrease. The data shows that the PID controller uses less energy in two of the three cases (case 2 and 3) to reach the new steady-state RPM.
Figure 6.11: Left: RPM response of PID controller with three $C_{\mu}$ goals. Right: RPM response of FL controller with three $C_{\mu}$ goals.

Table 6.7: Response characteristics of the PID and FL controllers to a 2 m/s decrease in airspeed.

<table>
<thead>
<tr>
<th>$C_{\mu}$ and Step Value</th>
<th>PID Controller</th>
<th>Fuzzy Logic Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rise Time (s)</td>
<td>Settling Time (s)</td>
</tr>
<tr>
<td>1: $C_{\mu} = 0.016$</td>
<td>0.6518</td>
<td>1.8402</td>
</tr>
<tr>
<td>Step: 22,700–20,100 RPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2: $C_{\mu} = 0.011$</td>
<td>0.1910</td>
<td>1.6317</td>
</tr>
<tr>
<td>Step: 18,000–15,900 RPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3: $C_{\mu} = 0.005$</td>
<td>0.1674</td>
<td>0.9812</td>
</tr>
<tr>
<td>Step: 10,600–10,000 RPM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.12: Left: Power response of PID controller with three $C_{\mu}$ goals. Right: Power response of FL controller with three $C_{\mu}$ goals.
Table 6.8: Power consumption between PID and FL controllers during a 2 m/s decrease in airspeed.

<table>
<thead>
<tr>
<th>$C_\mu$ and Step Value</th>
<th>PID Controller Energy Consumption (Joules)</th>
<th>FL Controller Energy Consumption (Joules)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: $C_\mu = 0.016$</td>
<td>274.60</td>
<td>246.20</td>
</tr>
<tr>
<td>Step: 22,700–20,100 RPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2: $C_\mu = 0.011$</td>
<td>127.39</td>
<td>165.41</td>
</tr>
<tr>
<td>Step: 18,000–15,900 RPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3: $C_\mu = 0.005$</td>
<td>26.86</td>
<td>161.77</td>
</tr>
<tr>
<td>Step: 10,600-10,000 RPM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.2.2 Energy Consumption During Cruise Flight

These aforementioned data only illustrate the controller response characteristics and energy consumption for a single perturbation in airspeed – from either 20–22 m/s, or 20–18 m/s. However, during flight these perturbations are likely to occur frequently. Multiplying the energy consumption by the number of expected perturbations and accounting for the steady-state power consumption creates a reasonable estimate for the energy consumption during cruise flight. Equation (6.3) details how energy consumption can be calculated, where $E_{a+}$ and $E_{a-}$ are the energy consumed during an increase and subsequent decrease in airspeed and are chosen from Table 6.6 and Table 6.8 for the same $C_\mu$ and controller. The energy consumed during this step up and step down perturbation is then multiplied by the expected number of perturbations. The steady state power, $P_{ss}$, is chosen as 30.90 W at 10,600 RPM, 109.25 W at 18,000 RPM, or 189.69 W at 22,700 RPM, dependent on the $C_\mu$ value the controller is chosen to maintain. The steady-state power is then multiplied by the time the airspeed is expected to be constant.

$$Energy During Cruise = p \times (E_{a+} + E_{a-}) + P_{ss} \times t_c \quad (6.3)$$

When examining the response of both controllers, it is seen that PID control of the CC system exhibits better performance in comparison to FL control, in terms of rise time, settling time, overshoot, and energy consumption in almost all of the investigated cases.
These data illustrate that PID control outperforms FL control for simulations of cruise flight of the UC²AV.

6.3 Landing Phase

Landing is the final phase of the UC²AV flight envelope that is investigated. Landing is defined to have two phases: preparation for landing, and final descent. Landing is similar to takeoff because maximum lift augmentation (maximum \( C_{\mu} \) and RPM) is also required. Additionally, the benefits of augmented lift are applied in the same way as takeoff: either to reduce landing distance, or to land while carrying additional payload.

Preparation for landing is defined as the time the pilot aligns the aircraft with the runway during the final approach. For simulation purposes, the duration of this maneuver is assumed to be 3 s, and during this time the ASU motor must reach the maximum RPM of 26,000. After aircraft alignment, the time taken for the aircraft to touchdown and stop is simulated as 6 seconds. An entire landing sequence is simulated by a ramp input of \( t_r = 3 \text{ s} \), starting from the nominal RPM at cruise (10,600, 18,000, 22,700 RPM, or off) and reaching 26,000, then maintaining maximum RPM for a duration of 6 seconds – the time observed for the UC²AV to complete a landing, for a total landing time of 9 seconds. The ramp input from the CC system in an off state, to maximum RPM is not tested again here, since it is mimics the same behavior of the 3 second ramp that is investigated during a simulated takeoff.

6.3.1 PID and FL Control During Landing Phase

The PID and FL controllers are also applied to the landing phase of flight to determine which performs better with regard to response characteristics, and energy consumption.
The RPM response of the CC system to the simulated landing inputs for both controllers is illustrated in Figure 6.13.

From Figure 6.13 and Table 6.9, it can be seen that the both the PID and FL controllers respond similarly to the simulated landing. As expected, all reach steady-state shortly after the 3 s ramp. However, in all cases the FL controller experiences shorter settling times than that of the PID controller, with an average settling time that is 0.59 s faster. Despite the shorter settling times, experimental data also show greater overshoot in all cases by the FL controller in comparison to the PID controller.

Table 6.9: Response characteristics of the PID and FL controllers during the simulated landing.

<table>
<thead>
<tr>
<th>$C_n$ and Ramp Value</th>
<th>Rise Time (s)</th>
<th>Setting Time (s)</th>
<th>Overshoot (%)</th>
<th>Rise Time (s)</th>
<th>Setting Time (s)</th>
<th>Overshoot (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: $C_n = 0.016$</td>
<td>2.3650</td>
<td>3.9603</td>
<td>0.0502</td>
<td>2.9000</td>
<td>3.4248</td>
<td>0.9802</td>
</tr>
<tr>
<td>Ramp: 22,700–26,000 RPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2: $C_n = 0.011$</td>
<td>2.4128</td>
<td>3.9119</td>
<td>0.0000</td>
<td>2.4023</td>
<td>3.2637</td>
<td>0.2001</td>
</tr>
<tr>
<td>Ramp: 18,000–26,000 RPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3: $C_n = 0.005$</td>
<td>2.5202</td>
<td>3.6881</td>
<td>0.0000</td>
<td>2.3454</td>
<td>3.1841</td>
<td>0.2648</td>
</tr>
<tr>
<td>Ramp: 10,600–26,000 RPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6.3.2 Energy Consumption During Landing

The CC system’s power response with both controllers is illustrated by Figure 6.14. The power consumption detailed is similar between the two controllers – similar steady-state values are reached during the same span of time. The power response of the system with PID control does however experience a higher level of oscillation, and is due to the larger PWM output commands in comparison to the FL controller. The analysis of power consumption over the entirety of the landing phase is detailed in Table 6.10. The table shows little variation in power consumption between the two controllers, with a maximum difference of 183.40 J over the 9 second landing time. The similarities in power consumption are expected; if the controllers are accurately tracking the ramp set point, then the RPMs and therefore power consumption for each should be similar for each given case. Additionally, results show that more power is consumed when ramping from a higher starting RPM. This is also expected, as the power needed to run the motor at higher RPMs is greater.

Figure 6.14: Left: Power response of PID controller with three varying ramp inputs. Right: Power response of FL controller with three varying ramp inputs.
Table 6.10: Power consumption between PID and FL controllers during the simulated landing.

<table>
<thead>
<tr>
<th>$C_\mu$ and Ramp Value</th>
<th>PID Controller Energy Consumption (Joules)</th>
<th>FL Controller Energy Consumption (Joules)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: $C_\mu = 0.016$</td>
<td>2552.50</td>
<td>2503.30</td>
</tr>
<tr>
<td>Ramp: 22,700–26,000 RPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2: $C_\mu = 0.01$</td>
<td>2261.90</td>
<td>2078.50</td>
</tr>
<tr>
<td>Ramp: 18,000–26,000 RPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3: $C_\mu = 0.005$</td>
<td>2023.80</td>
<td>2176.70</td>
</tr>
<tr>
<td>Ramp: 10,600–26,000 RPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4: $C_\mu = —$</td>
<td>1069.88</td>
<td>908.50</td>
</tr>
<tr>
<td>Ramp: 0 (off)–26,000 RPM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.4 Flight Envelope Energy Analysis

Synthesizing the results from the takeoff, cruise, and landing phases of flight allows for insight into the energy consumption across the entire flight envelope. The application of varying flight scenarios allows for the creation of a control scheme that reduces the energy consumed by the CC system’s ASU.

From the data collected on energy consumption over the various flight phases of the flight envelope an equation can be developed which summarizes the energy consumption outcome. Equation (6.4) presents a method for calculating the overall energy consumption of the UC²AV during flight. The energy consumed during takeoff, $E_T$, is chosen based on Table 6.3, the energy consumed during cruise, $E_C$ is calculated based on Equation (6.3), and the energy required for landing, $E_L$ is chosen from the appropriate scenario in Table 6.9.

$$Total\ Energy\ Consumed = E_T + E_C + E_L$$

Although results show similar energy consumption characteristics between the PID and FL controllers, advantages for the use of a PID controller are also indicated. Better tracking performance with a PID controller in relation to the FL controller makes it ideal for accu-
rately responding to controller inputs. The faster rise time and smaller overshoot of the PID controller are important factors to considering when optimizing the power consumption of the system, therefore the PID controller is selected to examine best and worst case flight scenarios.

Among possible flights, there are two dominant missions that arise: Mission-I, requiring CC throughout the flight envelope, and Mission-II, operating the CC system only during takeoff and landing, and conserving battery power for possible critical situations that may arise during cruise. There are several scenarios where Mission-I is likely, including the transport of a package. In a scenario where the package weight is heavier than the initially designed takeoff weight, CC control is employed throughout the flight envelope to maintain the required lift. Mission-II is likely to occur in a scenario where runway infrastructure is limited. Operating the CC system during takeoff without additional weight allows for shorter takeoff distances, and this same method can also be applied during landings that necessitate short aircraft landing distances. In this scenario, operation of the CC system is not required during cruise. The application of the PID controller to regulate the behavior of the aircraft during these missions is important in minimizing the power penalties of the ASU. Scenarios are presented for each mission, illustrating how adjustment of the CC system’s behavior can minimize its power consumption. Figure 6.15 shows representative ASU motor RPM profiles for these possible missions.

![Mission-I & Mission-II RPM profiles](image)

Figure 6.15: Left: Representative RPM profile for Mission-I, with CC during cruise flight. Right: Representative RPM profile for Mission-II, without CC during cruise flight.
For Mission-I, a scenario with a large energy impact is imagined as: transporting a heavy payload during a breezy day, with a 10 minute flight duration (Scenario-I). This involves initiating the ASU to maximum RPM with a step function at the beginning of takeoff (takeoff Approach-I), maintaining the required lift to transport the payload during cruise with a $C_\mu = 0.016$ and 20 gusts/perturbations to airspeed, and landing by returning the ASU to maximum RPM. Using Equation (6.4), this energy consumed during this worst case is $115209.58 \text{ J}$.

Applied to the same mission, a scenario with a smaller energy impact could consist of transporting a light payload during a calm day, with a 10 minute flight duration (Scenario-II). The time for the aircraft to leave the ground is well known and the ASU initiated corresponding to this (takeoff Approach-II), a minimal $C_\mu = 0.005$ is required to maintain lift and zero perturbations in airspeed are experienced, and landing is initiated from the $C_\mu$ value applied during cruise flight. Application of these scenarios to Equation (6.4) yields $22532.96 \text{ J}$. This is best case scenario requires 80% less energy than the amount of energy required in the worst case.

For Mission-II, a large energy impact scenario is imagined as: A 10 minute flight with poor runway infrastructure requiring a short takeoff and landing, with an unknown projected time required for takeoff (Scenario-I). This scenario requires initiating the ASU to maximum RPM with a step function at the beginning of takeoff (takeoff Approach-I), and returning the ASU to maximum RPM to complete the landing phase. Using the appropriate energy consumption, the data consumption for this scenario is $5035.34 \text{ J}$, using Equation (6.4).

A lower energy impact scenario for Mission-II over a 10 minute flight can be modeled as also requiring a short takeoff and landing, but with knowledge of the time required for takeoff, and initiating the ASU accordingly (Scenario-II). In this scenario the ASU is operated with takeoff Approach-II, turned off during cruise, and re-initiated during land-
ing. Using Equation (6.4), the energy consumption for this case is 3331.17 J. Table 6.11 summarizes these results from this flight envelope analysis.

Table 6.11: Energy consumption for Mission-I and Mission-II, with varying scenarios.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission-I, Scenario-I</td>
<td>3965.46</td>
<td>108691.62</td>
<td>2552.50</td>
<td>115209.58</td>
</tr>
<tr>
<td>Mission-I, Scenario-II</td>
<td>2261.29</td>
<td>18247.87</td>
<td>2023.8</td>
<td>22532.96</td>
</tr>
<tr>
<td>Mission-II, Scenario-I</td>
<td>3965.46</td>
<td>0.00</td>
<td>1069.90</td>
<td>3331.17</td>
</tr>
<tr>
<td>Mission-II, Scenario-II</td>
<td>2261.29</td>
<td>0.00</td>
<td>1069.90</td>
<td>5035.34</td>
</tr>
</tbody>
</table>

6.5 Discussion

In summary, both of the PID and FL controllers are shown to provide an adequate level of control of the CC system throughout a simulated flight envelope. During the simulated takeoff phase, it is seen that the PID controller generally outperforms the FL controller for each of the simulated ramp input cases. Ramp inputs are found to consume less power than different sets of gain tunings, and Approach-II for takeoff is found to be more energy efficient than Approach-I. However, the better performance of the PID controller is not found to directly translate into energy savings. A comparison of both the PID and FL controllers reveals that the PID controller uses marginally less energy in Case 1 and Case 2 when Approach-I is used, and marginally less energy in Case 3 when Approach-II is used. In the other three cases, data shows that the FL controller is marginally more energy efficient.

A comparison of both controllers during cruise reveals faster rise times by the PID controller in almost all cases, and less overshoot when both increasing and decreasing RPM to track $C_{\mu}$, in comparison to the FL controller. Additionally, the settling times between the
two controllers are demonstrated as similar or shorter for the PID controller during this $C_\mu$ tracking. Power consumed by the PID controller is less in four of the six cases studied, in relation to the FL controller. Additionally, in the cases where the FL does use less energy, it is slightly lower than that consumed by the PID controller, with a difference of less than 29 J.

During landing, the data show almost no difference between the tracking capabilities of each controller. Similar rise times, settling times, and overshoot are exhibited. These similarities lead to similar energy consumption characteristics by the two controllers as well. The maximum difference in energy consumption between the PID and FL controller is found to be 183.40 J over the 9 second simulated landing.

An analysis of the power consumption of the CC system’s ASU over the entire flight envelope with the use of a PID controller reveals how the energy consumption varies. Operating the CC system during cruise, and at higher RPMs results in larger amounts of energy consumption.
Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this research, both a PID controller and FL controller are presented as a means of accurately regulating the behavior of the CC system onboard the UC²AV. The purpose of the CC system is to provide lift augmentation on demand, dependent on the aircraft’s mission. The work shown here presents a mathematical model of the CC system as a framework for future controller development, and introduces a power planning flowchart for the energy consumption of the CC system with regard to various flight scenarios, using either a PID or FL controller to achieve the desired behavior.

A detailed study is conducted to develop a mathematical model of the CC system. Time domain frequency data in conjunction with system identification techniques are applied to model the dynamics of the system. A mathematical plant consisting of a PWM input and RPM, $V_j$, and power consumption outputs is presented as a framework for controller development and testing prior to implementation on the physical CC system.

The design of both a PID controller and FL controller are presented. Initial PID gain tunings are chosen based on the Zeigler-Nichols method, and further developed in simu-
lation. Five MF and seven MF Mamdani inference FL controllers are also designed and compared in simulation, where the response characteristics are found to be similar. Both PID and FL designed controllers are implemented on a microcontroller, and the response characteristics compared to one another. The implemented controllers are also compared to simulation, with both controllers exhibiting a correlation to the simulation data above 67%.

In comparing the PID and FL controller’s performance, the PID control displays a rise time that is almost 2.5 times faster than the FL controller. The PID controller also exhibits zero overshoot, but a settling time 2.1 times longer than that of the FL controller. The fast response and minimal overshoot of the PID controller are desirable in CC regulation, where a quick and accurate response to dynamic changes is crucial.

Thorough investigations between the PID and FL controller and the takeoff, cruise, and landing phases of flight are also presented. The CC system’s RPM response, as well as power consumption is characterized. Here the PID controller is selected to study the power consumption of an entire flight envelope due to its faster rise times and minimal overshoot in comparison to the FL controller. The results for takeoff, cruise, and landing are combined to understand and optimize the power consumption of the CC system throughout the entirety of the flight envelope. The results show that reducing the power consumption of the CC system is largely dependent on the performance requirements imposed upon it. Initiating the ASU later during takeoff, and minimizing or negating the lift augmentation requirements during cruise drastically decrease the energy consumed. The energy expenditure during the proposed worst case mission is 30 times higher than the best case mission.

Overall energy consumption is presented in Joules, however, battery manufacturers describe battery capacity in units of mAh. Joule is a unit of energy, while mAh is a measure of electrical charge. Joules are related to mAh by Equation (7.1), where voltage is the nominal voltage of the battery – 14.8 V for the 4S battery applied in this research.
\[ mAh = \frac{Joules}{Volts} \times \frac{1000 mA}{3600 seconds} \]  \hspace{1cm} (7.1)

Lithium-Polymer batteries can be permanently damaged if over-discharged, so care must be taken to prevent this. Below a potential of 3.0 V, the battery cells become irreversibly damaged [81]. Examining discharge curves for LiPo batteries show that discharging to 80\% of the stated capacity results in a safe final discharged voltage of 3.3 V to 3.6 V, as shown in Figure 7.1. By increasing the calculated mAh usage over the entirety of the UC\(^2\)AV flight envelope by 25\%, a battery capacity that compensates for the discharge characteristics of the LiPo battery power source is created.

![Figure 7.1: Discharge voltage characteristics for LiPo batteries [17].](image)

The knowledge of battery capacity requirements, combined with characterization of the energy expenditure during the investigated phases of flight, is aggregated to create a flowchart for optimizing the power usage of the CC system. Figure 7.2 illustrates this power planning flowchart, and details the CC system’s energy expenditure for a flight representative of a typical mission the UC\(^2\)AV is likely to experience: a 10 minute flight, utilizing the best case for Approach-I of takeoff, with 10 airspeed perturbations during cruise, using the PID controller. Energy consumption is presented in relation to a mission with a
low energy impact: when takeoff Approach-I is applied, CC is not used during cruise, and the CC system is reinitated for landing. Battery capacities and weights are presented as outputs. Other cases can also be extrapolated, using the energy data for each scenario.
Figure 7.2: Power planning flowchart for optimizing energy consumption of the ASU throughout the UC²AV’s simulated flight envelope.
As an example from the flowchart, the energy consumption during a 10 minute flight, from initiation to the end of landing, can be calculated using the data presented in Chapter 6, and Equation (6.4). Assuming the use of a PID controller, a 6 second ramp and Approach-I for takeoff, $C_\mu = 0.011$ tracking and 10 airspeed perturbations/gusts during cruise, and landing, the total energy consumption by the CC system can be modeled by Equation (7.2).

\[
\text{Takeoff} \quad \text{Cruise} \quad \text{Landing} \\
3138.77 \, J + \, 64454.69 \, J + \, 2261.90 \, J = 69299.49 \, J
\]  

(7.2)

The energy consumption for this scenario is 69299.49 J. This translates to 1300.67 mAh, and a battery capacity of 1625.83 mAh. This energy consumption is represents an increase of 1546.60% over the presented minimal energy-use case with CC off during cruise, of 4208.65 J. The closest COTS battery capacity greater than that of the presented scenario is 1750 mAh, weighing 175 g\(^1\) – a mass savings of 319 g in comparison to the 5100 mAh, 494 g\(^2\) battery currently powering the UC\(^2\)AV’s CC system (Figure 7.3). The specifications of this battery are presented in Table 7.1.

![Image of battery](image_url)

Figure 7.3: The LiPo battery currently utilized for powering the CC system onboard the UC\(^2\)AV. \(^2\)

---

\(^1\)HobbyKing, HobbyKing.com

\(^2\)ReadyMadeRC, ReadyMadeRC.com
The power consumption, recommended battery capacity and associated weight, and mass savings from the battery currently onboard the UC²AV for the missions and scenarios presented in Chapter 6 is detailed in Table 7.2. When the demands on the CC system are low, as in Mission-I, Scenario-II and Mission-II, Scenario-II, and when CC is only used turning takeoff and landing (Mission-II), results show low levels of power consumption, which is expected. In these cases a single battery can be utilized in multiple flights before recharging.

Table 7.2: Energy consumption, battery sizing, and mass savings for Mission-I and Mission-II, with varying scenarios.

<table>
<thead>
<tr>
<th>Mission</th>
<th>Total Energy Consumption (J)</th>
<th>Total Energy Consumption (mAh)</th>
<th>Required Battery Capacity (mAh)</th>
<th>COTS Capacity (mAh)</th>
<th>Mass (g)</th>
<th>Mass Savings (g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission-I, Scenario-I</td>
<td>115209.58</td>
<td>2162.34</td>
<td>2702.93</td>
<td>2800</td>
<td>285</td>
<td>209</td>
</tr>
<tr>
<td>Mission-I, Scenario-II</td>
<td>22532.96</td>
<td>422.92</td>
<td>528.64</td>
<td>850</td>
<td>95</td>
<td>399</td>
</tr>
<tr>
<td>Mission-II, Scenario-I</td>
<td>5035.34</td>
<td>94.51</td>
<td>118.13</td>
<td>500</td>
<td>74</td>
<td>420</td>
</tr>
<tr>
<td>Mission-II, Scenario-II</td>
<td>3331.17</td>
<td>62.32</td>
<td>78.15</td>
<td>500</td>
<td>74</td>
<td>420</td>
</tr>
</tbody>
</table>

In conclusion, the application of either PID or FL system controllers is a feasible solution for regulating the behavior of the CC system onboard the UC²AV. Both provide adequate control of the CC system’s behavior, and exhibit similar energy consumption characteristics throughout a simulated flight envelope. However, of these two controllers the PID controller outperforms the FL controller with regard to dynamic system behavior regulation, with fast rise times and less overshoot in a majority of the investigated cases. Additionally, the application of a power planning flowchart illustrates the potential for mass
savings, with a 42% reduction in battery mass for a simulated scenario with a large energy impact.

7.2 Future Work

While the work presented here has demonstrated the ability of both PID and FL controllers to regulate the behavior of the UC²AV’s CC system, many opportunities for advancing this research remain.

The system identification plant model presented provides a framework for the application of additional controllers and control methods. The application of optimized controllers, such as an LQR optimized PID controller, neural-network trained fuzzy logic controller, adaptive controller, or genetic algorithm optimized controller present opportunities for further improvements in regulation of the system’s behavior, and optimization of power consumption. Additionally, investigations into active control of the CC system during alternate flight scenarios provide insight into the power impacts, and possible missions that could be conducted. The lift augmentation that CC provides is beneficial in numerous scenarios including: aerial photography, in-flight refueling, parcel delivery or acquirement, or water tanker filling. Understanding and modeling the demands of the CC system during these scenarios allows for the creation of a higher fidelity model of the energy consumption. Future work integrating the controller into the electronics of the UC²AV enables a higher level of automation. Collecting telemetry and attitude data automatically by communicating with the aircraft’s IMU (inertial measurement unit) and flight controller can allow the control system to autonomously decide the phase of flight the aircraft is experiencing and adjust the CC system’s output accordingly. Finally, further advances are possible by conducting real-time flights with active control of the CC system. Collecting data on a
system-level enables further knowledge into the impact on efficiency, and the benefits of CC with respect to lift augmentation.
Bibliography


