UNIVERSITY OF CALIFORNIA SAN DIEGO

Feedback Control Driven Mechanical Design Optimization

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

James Robert Strawson

Committee in charge:

Professor Thomas Bewley, Chair
Professor Falko Kuester, Co-Chair
Professor Mauricio De Oliveira
Professor John Tae Hyeon Hwang
Professor Ryan Kastner

2018

The dissertation of James Robert Strawson is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California San Diego

2018

DEDICATION

This dissertation is dedicated to my beloved wife, Peian, who taught me that achievements are pointless without someone to share them with. It is also dedicated to my parents, who provided the support and encouragement for me to pursue my goals.

EPIGRAPH

Simplify, then add lightness.

—Colin Chapman

TABLE OF CONTENTS

Signature Page . . . iii

Dedication . . . iv

Epigraph . . . v

Table of Contents . . . vi

List of Figures . . . ix

List of Tables . . . xi

Acknowledgements . . . xii

Vita . . . xv

Abstract of the Dissertation . . . xvi

Chapter 1 Introduction . . . 1
1.1 Motivation . . . 1
1.2 State of the Art: Embedded Platforms for Robotics R&D and Education 2
1.3 Presented Ecosystem for Embedded Platforms for Robotics R&D and Education . . . 3
1.4 State of the Art: Multirotor Design and Control . . . 4
1.5 Organization of Material . . . 5

Chapter 2 "Robot Control" A New Ecosystem for Embedded Control & Robotics Education . . . 7
2.1 Beginnings . . . 7
2.1.1 Microcontroller-Based Development Environment: Strengths and Weaknesses . . . 7
2.2 Lowering the Barrier-of-Entry . . . 11
2.3 Robotics Cape and Supporting Hardware Decisions . . . 12
2.4 BeagleBone Blue . . . 19
2.5 Coherent Software Environment: librobotcontrol . . . 21
2.5.1 Modules . . . 21
2.5.2 Examples . . . 23
2.5.3 Documentation . . . 23
2.5.4 Project Template . . . 25
2.6 Three Reference Designs . . . 29
2.6.1 EduRover . . . 29
2.6.2 EduMiP . . . 33

2.6.3 EduMAV . . . 35
2.7 Impact . . . 36
2.8 Acknowledgements . . . 37

Chapter 3 Multirotor Layout Optimization . . . 38
3.1 Introduction . . . 38
3.2 Quantifying Performance . . . 39
3.2.1 Frame Layout Definition . . . 39
3.2.2 Rotor Orientation Definition . . . 41
3.2.3 Force Matrix . . . 43
3.2.4 Mixing Matrix . . . 46
3.2.5 Control Force Authority Vectors . . . 49
3.2.6 Objective function used in optimization . . . 51
3.3 Optimization Approach . . . 52
3.3.1 Search Space . . . 52
3.3.2 Global Minimum Computation . . . 53
3.4 Qualitative Observations . . . 54
3.4.1 Inward Tilt . . . 54
3.4.2 Isomerism in Solution . . . 58
3.4.3 Comparison with Related Work . . . 60
3.5 Conclusions . . . 62
3.6 Acknowledgements . . . 62

Chapter 4 Mechanical Design and Analysis . . . 63
4.1 General Design . . . 63
4.1.1 Monocoque Approach and Material Selection . . . 64
4.1.2 Monolithic vs Modular Design Tradeoffs . . . 65
4.1.3 Motion Capture Marker Placement . . . 66
4.1.4 Modular Mounting System and Custom Electronics . . . 67
4.2 Motor and Propeller Testing . . . 69
4.2.1 Experiment . . . 69
4.2.2 Results . . . 72
4.3 Modal Analysis of Frame . . . 73
4.3.1 Finite Element Model and Boundary Conditions . . . 73
4.3.2 Results and Validation . . . 74
4.4 Fluid Flow Analysis . . . 76
4.4.1 Testing Scenarios and Parameters . . . 77
4.4.2 Results and Qualitative Analysis . . . 79
4.5 Conclusions . . . 81
4.6 Acknowledgements . . . 81

Chapter 5 Control and Performance Evaluation 82
5.1 Control Software Architecture 82
5.1.1 Feedback Thread Components 84
5.1.2 Feedback Linearization 85
5.2 Experimental Model Validation 88
5.2.1 Experimental Results 90

Chapter 6 Concluding Remarks 91
6.1 Principle Contributions 91
6.1.1 Robotics Prototyping 91
6.1.2 Robotics Education 92
6.1.3 Layered Manufacturing 93
6.1.4 Multirotor Design 93
6.2 Future Work 93
6.3 Final Thoughts 94

Bibliography 95

LIST OF FIGURES

Figure 1.1: Three design phases for the fully-actuated EduMAV hexacopter design. . . 5

Figure 2.1: Arduino-Based MiP from UCSD's 2012 MAE143c course offering. 9
Figure 2.2: MiP, EduMiP's commercially available counterpart manufactured and sold by WowWee Toys. 10
Figure 2.3: Robotics Cape expansion board for the BeagleBone Black series with connectivity and components labeled. 12
Figure 2.4: Open-source schematic for the Robotics Cape as made available for reference by users and as used during the MAE144 curriculum as an example of several common digital circuits and as an example of standard practices in schematic design. 14
Figure 2.5: BeagleBone Blue with features shared with Robotics Cape highlighted in blue. Image courtesy of RenaissanceRobotics.com and included with permission from its creator. 19
Figure 2.6: Organization of API modules of the librobotcontrol library. 22
Figure 2.7: Sample of the style and detail of the function documentation provided for every single function in librobotcontrol. 24
Figure 2.8: EduRover 4-wheel steering 4-wheel drive vehicle with hyper-articulate steering. 30
Figure 2.9: CAD rendering of underside of EduRover demonstrating drivetrain and steering layout. 31
Figure 2.10: CAD rendering of wheels turned to crab mode to allow sideways motion. . . 31
Figure 2.11: EduRover with wheels turned in crab mode demonstrating orientation for balancing. 32
Figure 2.12: CAD rendering of EduRover's hub containing motor, gearbox, steering arms, and kingpin suspension. 32
Figure 2.13: Current BeagleBone-Based EduMiP 33
Figure 2.14: EduMAV render, front view. 35

Figure 3.1: CAD rendering of the airframe used in the case study, showing the coordinate system and rotor indexing. 39
Figure 3.2: Geometric representation of optimized airframe showing thrust vectors along rotor axes. The single red line along pointing in the positive X direction originating at the origin indicates the direction of forward flight. 42
Figure 3.3: Geometric representation of optimized airframe showing thrust vectors along rotor axes. The single red line along pointing in the positive X direction originating at the origin indicates the direction of forward flight. 43
Figure 3.4: DJI S1000 multirotor frame demonstrating low center of mass and inward tilt of rotors. 55
Figure 3.5: Optimum configuration for center of mass 1m above the rotor plane. 56
Figure 3.6: Optimum configuration for center of mass 1m below the rotor plane. 57

Figure 3.7: Isomer of the previously presented optimum solution with reversed rotor rotations. 59
Figure 3.8: Rotor orientation and corresponding performance characteristics of the CyPhy LVL1 drone rotor orientations applied to our case study frame dimensions. 61

Figure 4.1: 3D model of hexacopter design with and overall frame diameter of 45cm and 12.7cm diameter propellers. 63
Figure 4.2: Various 3D printed EduMAV prototypes. 64
Figure 4.3: Comparison of earlier monocoque frame design (left) incorporating sealed electronics mounting compartment, compared with the minimalistic final design (right). 65
Figure 4.4: Model with Intel Aero compute board and RealSense RGB-D camera payload module mounted. 67
Figure 4.5: Propeller test stand used for performance evaluation 70
Figure 4.6: Propeller test stand and results 71
Figure 4.7: Results of mesh sensitivity study 74
Figure 4.8: Displacement plots of first 3 vibrational modes of the FE shell model . . . 75
Figure 4.9: Prototype 3D printed model undergoing static stiffness validation. 76
Figure 4.10: Simplified model for CFD. Red disks are at the front (nose) of the airframe. X axis points right (starboard), Y axis points upwards, and Z axis points rearwards. 77
Figure 4.11: Streamlines for Each CFD Test Scenario 80

Figure 5.1: Modules of the RC Pilot flight controller 83
Figure 5.2: Thrust curve for motor propeller combination used. 86
Figure 5.3: EduMAV in flight inside Vicon Motion Capture System. 87
Figure 5.4: Isometric view of case study airframe highlighting the rotor angles. 88
Figure 5.5: Top view of case study airframe with optimized angles as used for experiment. 89

LIST OF TABLES

Table 3.1: Available Forces With Optimum and CyPhy Layouts 60

Table 4.1: Motors and Propellers Tested 69
Table 4.2: Force components in Newtons. X axis points right (starboard), Y axis points upwards, and Z axis points rearwards. 77

Table 5.1: Experimental Results 90

ACKNOWLEDGEMENTS

Thank you, Thomas Bewley, for introducing me to robotics, and to the spaghetti wiring that triggered several years of OCD PCB and software design that formed my discipline as an engineer. Also, thank you for your significant personal efforts in pushing the adoption of my work by BeagleBoard, MathWorks, and LabVIEW. It gives me great satisfaction to see my open-source efforts used by such a wide audience.

Thank you, Falko Kuester, for providing the outstanding laboratory and equipment that formed my home on campus during my studies. More importantly, thank you for constructing the unique interdisciplinary environment that has introduced me to so many unique minds and helps every student under your supervision grow as an academic and as a person.

I must note that one of the most valuable skills I have acquired through my time as a grad student, the ability to teach, explain, and coherently convey ideas to any audience, is a direct result of the dozens of hours of classroom time that both Thomas Bewley and Falko Kuester handed over to me to lead. Spending time at the front of a classroom is as much a learning experience as sitting in the audience. It is an opportunity very few are privileged with, and for that I am eternally grateful.

Thank you to my committee members and professors for providing guidance when I request it, answering questions when I have them, and for doing so without hesitation or complaint at often ridiculous hours of the day. I could not ask for a better faculty, and because of this, UCSD will always be a significant part of who I am.

Thank you, Tom Wypych, for somehow managing to be a mentor, teacher, peer, and friend all at once. You've taught me lessons about electronics, radios, cars, and every other subject that a top-notch engineer should be familiar with. The result of this is perhaps the most important lesson you've inadvertently taught me which is the immense value in being able to speak confidently and accurately on any subject in the company of any of the miriad of people that we encounter in life.

Thank you, Dominique Meyer, for providing much-needed enthusiasm, energy, and

support. You are a model student and researcher who constantly reminds me how hard-working and ambitious we should all strive to be, and in doing so, brighten my overall view of humanity.

Finally, I would like to recognize and extend my gratitude to all of my coauthors: Clark Briggs, Thomas Bewley, Pengcheng Cao, Falko Kuester, and Danny Tran. They helped me conduct much of the research and construct much of the content herein and deserve my sincerest gratitude.

Chapter 2, in part, is a reprint of the material as it appears, titled "Leveraging Open Standards and Credit-Card-Sized Linux Computers in Embedded Control & Robotics Education" SciTech 2015. Bewley, Thomas; Briggs, Clark; Strawson, James. The dissertation author was a coauthor of this material and was responsible for the design and manufacturing of the electronics & hardware, as well as teaching a significant portion of the course materials and writing of the text.

Chapter 3, in part, has been submitted for publication of the material titled "Rotor Orientation Optimization for Direct 6 Degree of Freedom Control of Multirotors" as it may appear in IEEE Transactions on Robotics 2019. Strawson, James; Bewley, Thomas; Kuester, Falko. The dissertation author was the primary investigator and author of this material.

I would like to thank Yifan Xu, Meng-Fu Chiang, Mingchen Mao, Pengcheng Cao, and He Huang for their dedicated work conducting the motor and propeller testing. I would also like to thank Danny Tran for his many late nights making Abaqus behave and helping me analyze the fluid flow characteristics of the design.

Chapter 4, in part, has been submitted for publication of the material titled "Multirotor Airframe Design with Rotor Orientations Optimized for Fully Actuated Feedback Control" as it may appear in IEEE Transactions on Robotics 2019. Strawson, James; Tran, Danny; Cao, Pengcheng; Bewley, Thomas; Kuester, Falko. The dissertation author was the primary investigator and author of this material.

Chapter 5, in part, is currently being prepared for submission for publication of the

material, titled "Feedback Control Performance and Design for Multirotors with Optimally Tilted Rotor Orientations" in IEEE Transactions on Robotics 2019. Strawson, James; Bewley, Thomas; Kuester, Falko. The dissertation author was the primary investigator and author of this material.

VITA

2012	B. S. in Mechanical Engineering, University of California, San Diego
2017	M. S. in Engineering Sciences (Mechanical Engineering), San Diego
2018	Ph. D. in Engineering Sciences (Mechanical Engineering), University of California, San Diego

ABSTRACT OF THE DISSERTATION

Feedback Control Driven Mechanical Design Optimization

by

James Robert Strawson

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2018

Professor Thomas Bewley, Chair
Professor Falko Kuester, Co-Chair

The current crop of outdoor-focused quadrotors struggle to explore tight, GPS denied, and vision impaired environments while managing self-induced turbulence and keeping the environment and UAV itself safe from collisions. Mainstream designs typically arrange their propellers in or near the same plane, resulting in an under-actuated system that must roll and pitch in order to move laterally.

This dissertation describes the design, analysis, and construction of a soft multirotor airframe with the capability of in-plane maneuverability and decoupled 6DOF control which allow for low profile sensor payload integration without the need for a gimbal as well as predictable

and safe flight in confined spaces such as tunnels and collapsed buildings. All aspects of the design are described, beginning with the electronics package itself.

The author observes the disappointingly low use of small embedded Linux platforms in robotics education. As an alternative to the ubiquity of microcontroller-based development boards such as Arduino, this work presents the use of the Robotics Cape and BeagleBone Blue along with their associated software and hardware ecosystem in both a prototyping and education environment. This ecosystem was initially designed and produced to facilitate the aforementioned multirotor's construction, yet continues to be used as part of the UCSD MAE curriculum for the benefit of others.

Following the design of the multirotor's electronics package, this dissertation presents a method for optimizing rotor angles as a function of the frame parameters and the desired performance characteristics. It compares the performance of an optimized configuration with an existing commercial hexacopter, and quantifies the improved control authority of the optimized design.

The multirotor concept is then implemented as a monocoque airframe, designed through a presented technique for rapidly iterating the airframe shell thickness based on modal analysis using finite shell elements. Finally, the real-world performance of the platform is evaluated by examining several close-quarters flight scenarios using CFD, through analytical performance characterization of the supporting flight controller, and through physical measurement of the constructed multirotor's control authority.

Chapter 1

Introduction

1.1 Motivation

Remotely operated as well as fully autonomous aerial robotic systems have demonstrated their unique utility for remote imaging and sensing, communications, as well as payload delivery, with applications in precision farming, real-estate, cinematography, freight delivery, infrastructure assessment, emergency response, disaster and post-disaster reconnaissance as well as search and rescue, to name a few. Particularly in the context of search and rescue, robotic systems face a broad range of challenging and often hostile environmental conditions, while capturing the actionable data needed to create a common operating picture for first responders, decision makers and the public alike.

Unmanned aerial vehicles (UAVs) have shown promise in these environments and have been readily adopted to enable rapid sensor deployment and data gathering from a birds-eye perspective. However, the next generation of semi- and fully-autonomous aerial vehicles should be able to safely operate in close proximity to impacted environments and damaged structures, in tight spaces, around hazardous conditions and materials, as well as first responders and individuals in distress. The "one size fits all" approach to UAV design, as such, needs to evolve to one that

allows for task and objective specific design, informed and driven by the control theory that ultimately governs their dynamic behavior and capabilities.

1.2 State of the Art: Embedded Platforms for Robotics R&D and Education

Modern disruptive technologies such as 3D CAD software, 3D printing, laser cutting, PCB design software, mail-order PCB fabrication & low-volume PCB assembly, smartphone and tablet SDKs, high-capacity lithium batteries, a huge variety of COTS sensors, and sub-$50 credit-card-sized Linux computers like the Raspberry Pi and BeagleBone Black, are slowly broadening the capabilities available to roboticists and academics. This broadening availability of low-cost manufacturing capabilities and ongoing spin-off of cutting-edge low-power consumer cellphone technology into the robotics space, necessitate a major interdisciplinary revamping in the teaching of embedded control & robotics. Consequently, it also broadens the range of skill sets required to design, develop, construct, and test robotic prototypes in an industrial or academic environment.

As industry and academia as a whole valiantly pursues novel robotic solutions to the applications described above, among others, it is clear that any time spend debugging one-off electrical and software prototypes detracts from time spent pursuing truly novel ideas. The author observes students at UCSD and other Universities devoting a continuously growing amount of time to learning how to utilize the aforementioned disruptive technologies and debugging increasingly complex electronics in the hope that these technologies will enable their concepts. Meanwhile, they unintentionally devote their time to being a technician, rather than a researcher.

Furthermore, Universities typically disjoin the study of challenging technical fields into fragmented departments, like mechanical and aerospace engineering, electrical engineering, computer science, control & cybernetics, and human factors. Modern applications in robotics

require the rejoining of these traditionally isolated disciplines of study. The effective design and control of robotic vehicles requires a fundamentally interdisciplinary perspective that is ill served by keeping the teaching of its constituent technical components disjoint. There is an emerging need for educational institutions to distill and relate these constituent disciplines, and the remarkable recent advancements therein, to a new generation of roboticists.

1.3 Presented Ecosystem for Embedded Platforms for Robotics R&D and Education

To help counteract this trend, we have developed a development ecosystem and supporting educational materials which aim to provide a systematic, integrated introduction to the fundamental technologies and techniques available, focusing on control & coordination algorithms, open standards & tools, and software architectures that may be broadly used. This ecosystem is designed to be applicable and influential both at the maker level, in high schools and beyond, and at the "professional" level, in college and industry. The result is an unprecedented, interdisciplinary, highly motivating learning experience based on agile high-function robotic vehicle prototypes that students continue to hack, extend, reference, and learn from long after the completion of formal classwork.

The ecosystem itself is comprised of three main components. First is an interoperable pair of supporting electronics packages, the Robotics Cape and BeagleBone Blue, designed to provide out-of-the-box working solutions to the vast majority of functions required for robotics prototyping. Second, is a single software package, librobotcontrol, which provides a straightforward interface to utilize the electronics hardware and simultaneously encourages best-practice coding techniques through its included example code and API design. Finally, the ecosystem presents three small and unique robots which, together, demonstrate the complete functionality of the ecosystem, as well as provide working robotic platforms with interesting dynamics that can,

and do, form the focus of a robotics curriculum. One of these three robotic platforms, a flying hexacopter named EduMAV, is the focus of the second half of this dissertation.

1.4 State of the Art: Multirotor Design and Control

Small UAVs with vertical take-off and landing (VTOL) capabilities are particularly suited for search and rescue efforts, allowing for easy field deployment, while providing a means to capture the context and content needed for situational awareness. These multi-rotor UAVs commonly use a quad, hex or ocot-rotor design, requiring pitch and roll motion to maneuver in six-degree-of-freedom (6DOF) space. In disaster and post-disaster environments, where the UAV might have to hover close to a structure, transition into it through a small opening, traverse corridors, while operating close to floors, walls, ceilings or other hard surfaces, maneuverability is a major concerns. Among the challenges, are that (1) pitching and rolling requires additional space around the UAV to allow freedom of motion, (2) an articulated gimbal is necessary to keep the sensor payload level during flight which adds physical bulk to the UAV and increases power consumption and (3) maneuverability and predictability is severely reduced due to self-induced turbulence and prop wash.

Others have proposed and demonstrated the controllability of multirotors with off-axis tilted rotors capable of in-plane maneuvering and increased position control bandwidth [Wor] [SSK+15]. This work furthers the concept by demonstrating an optimization method for determining which angles should actually be used when designing for a set of performance and manufacturing goals, as well as quantifying benefits and side-effects of the angled design when placed in the types of close-quarter environments for which the concept is supposed to improve performance in.

1.5 Organization of Material

Chapter 2 of this dissertation details the robotics development ecosystem. This includes electrical component choices, software architecture, software documentation, the design of the first two reference platforms, EduMiP and EduRover, and finally the embedded control systems course curriculum outline for which this ecosystem is designed to support. The third and final robotics reference design, EduMAV, is significantly more intricate, academically interesting, and provides the focus of this dissertation spanning chapters three through five.

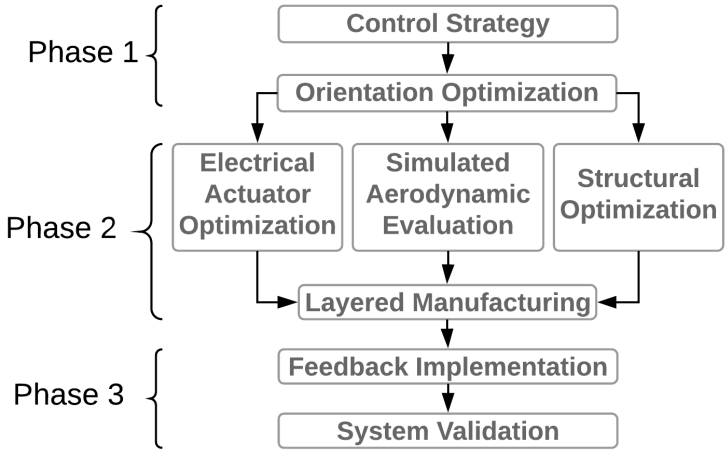


Figure 1.1: Three design phases for the fully-actuated EduMAV hexacopter design.

The design of EduMAV is logically separated into three phases, described in chapters three, four, and five respectively. The dependency chain of this design process is visualized in Figure 1.1.

Phase one characterizes a desired control strategy for EduMAV and then proposes a multi-objective design optimization (MDO) approach ([CJEDF+94] [Sid82]) which accounts for the size and mass distribution of a given multirotor frame, and optimizes a set of 6 rotor orientations to provide exceptional control authority over all six degrees of freedom of the vehicle. The user of this approach may tune the priorities assigned to the control authority over each of the six different degrees of freedom in the formulation of the optimization problem for specific applications. A

case study is developed in parallel with the optimization formulation to demonstrate possible numerical results and qualitative demonstrations of the impact design considerations have on optimum hexacopter rotor layout.

The second phase described (1) component layout in conjunction with utility-driven design choices, (2) motor & propeller characterization, (3) monocoque airframe stiffness & modal analysis, and (4) aerodynamic performance simulation and analysis.

The third and final phase describes the proposed feedback control system, its implementation as it relates to the robotics development ecosystem from chapter two, and validates the final performance as it relates to the optimization algorithm's prediction through a physical experiment.

Chapter 2

"Robot Control" A New Ecosystem for Embedded Control & Robotics Education

2.1 Beginnings

This new ecosystem, as developed and presented, is inspired by both the strengths and weaknesses of UCSD's 2012 offering of MAE143c. In this course offering, both undergraduate and graduate students across multiple engineering disciplines are taken on the journey of assembling, modeling, and programming a small mobile inverted pendulum or "MiP" for short. The author participated in this course and created the MiP pictured in figure 2.1.

2.1.1 Microcontroller-Based Development Environment: Strengths and Weaknesses

Critically, the kit of parts provided to assemble this robot cost $150 per student, roughly the same as a textbook. This low price point is essential to accessibility of this curriculum to a wide audience and provides incentive for its adoption. The kit primarily consisted of the following.

- Arduino Nano 8-bit microcontroller development board
- Solderless breadboard and jumper wires
- Analog accelerometer and gyroscope breakout boards
- H-bridge breakout board
- 9V Battery
- Miniature motor and gearbox units
- Laser-cut frame
- Home-made optical rotary encoders
- Various required hardware and accessories

The choice of the Arduino Nano was one of complacency with the status quo. Arduinos are based around the 8-bit AVR microcontroller series from Atmel, a series released in 1996, 16 years prior to the 2012 offering of MAE143c. They are extremely restrictive in that they are extremely low compared to modern ARM processors, offer no operating system, file system, networking. The slow speed became one of the serious limitations faced by students in this class as the simple counting of quadrature encoder positions infringed on the reliable operation of the very feedback control algorithms the class primarily sought to teach.

This speed limitation is reflected in the microcontrollers chosen for use in the new generation of robotic toys and internet-of-things devices commercially available. As an example, we describe EduMiP's commercially available counterpart, "MiP", manufactured and sold by WowWee Toys in partnership with UCSD's own Fluid Control and Coordinated Robotics Laboratory of which this dissertation's author and chair, Thomas Bewley, are both members. This commercial variant, pictured in figure 2.2, makes the use of an Arm Cortex M0 processor to

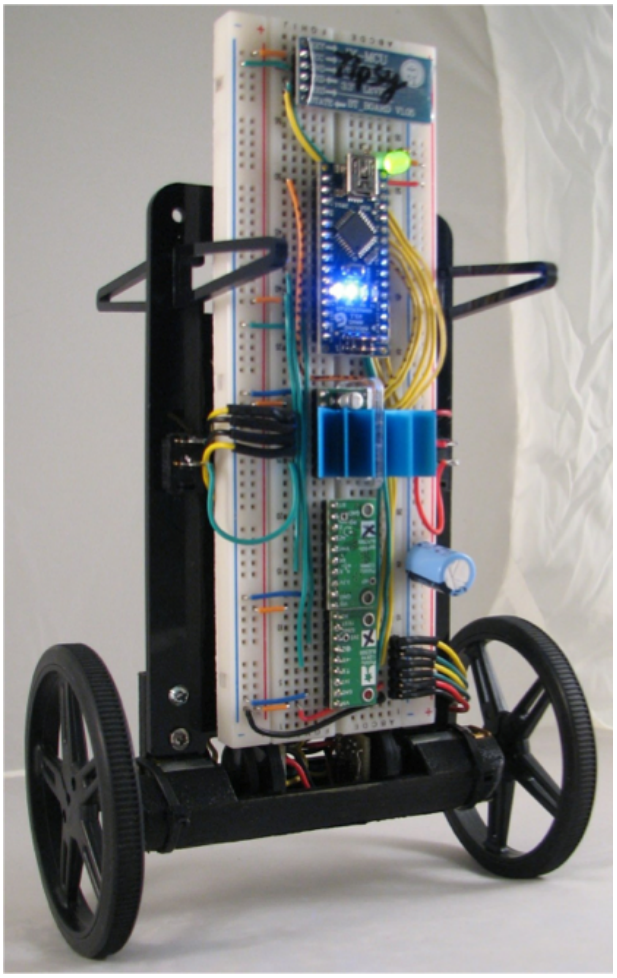


Figure 2.1: Arduino-Based MiP from UCSD's 2012 MAE143c course offering.

provide the additional speed and connectivity to support features such as Bluetooth, audio, more intricate modes of driving and operation, as well as utilization of more advanced sensors such as an infrared gesture recognition sensor. Such modern processors and sensors are the state of the art for the design and manufacturing of low-cost commercial robots. The author stipulates that robotics education should reflect this adoption of rapidly moving technology, not lag behind.

Despite their limitations, Arduinos remain fantastically popular due to three things: their low cost, open-source nature, and most critically, a simple programming environment that lowers the barrier-of-entry to newcomers wishing to learn about, or simply prototype, robotic systems. The slow adoption of Arm processors and embedded-Linux into the education space can largely

Figure 2.2: MiP, EduMiP's commercially available counterpart manufactured and sold by WowWee Toys.

be attributed to the increased complexity of the hardware and programming environment over the Arduino ecosystem. In the development of our own ecosystem, we retain these strengths and develop a programming environment with an arguably lower barrier-of-entry to robotics development than Arduino's.

2.2 Lowering the Barrier-of-Entry

A first step in lowering the barrier-of-entry to a development environment is to recognize the development steps that frustrate students the most and detract from the overall learning experience. In the 2012 offering of MAE143c, roughly %50 of students failed to complete the construction of their MiP and make it fully operational by the end of the quarter. This is largely due to hardware failures that leave students unable to complete intermediate assignments, often through no fault of their own, and increase the amount of debugging and dirty work that both the instructors and students must do simply to proceed to more critical course materials and concepts. The largest points of failure are electrical connections on the solderless breadboard. While clean wiring and rapid prototyping with breadboards are useful skills, we argue the vast amount of time wasted by students in practice detracts far too greatly from what is a mechanical engineering course.

Another downside of the solderless breadboard concept is the necessity of breakout boards for each integrated circuit used in the prototype. These breakout boards are necessarily more expensive than the integrated circuit by itself, often significantly so. They also add one more manufacturer to the supply chain, increasing the difficulty in assembly and distributing kits to students. Instructors can not be expected to restructure a course at the last minute because of unavailability of parts.

Another downside of using individual breakout boards is the lack of coherency in the supporting software platform. Students, instructors, or both, must be expected to source disjoint supporting documents and software for every single component. This further adds disarray and complication to the development process and ultimately detracts from the focus on core course material. This is particularly frustrating for students with little or no software background which is to be expected in what is, again, a mechanical engineering course.

2.3 Robotics Cape and Supporting Hardware Decisions

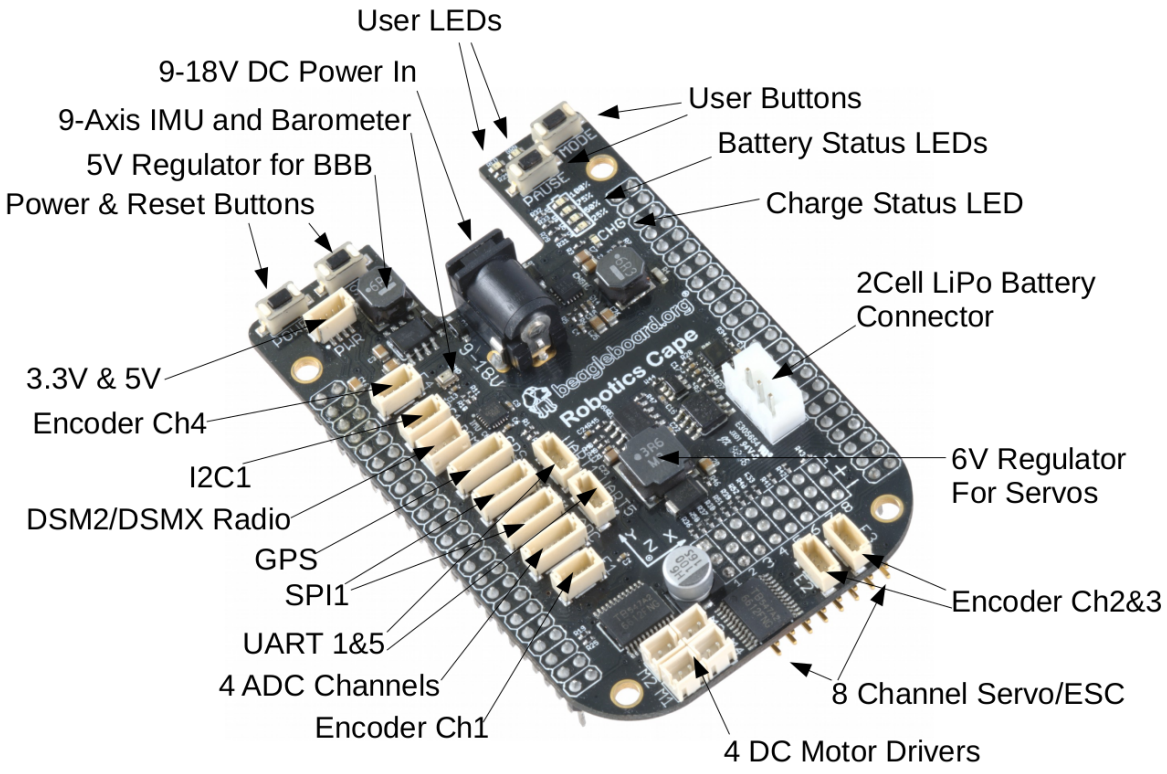


Figure 2.3: Robotics Cape expansion board for the BeagleBone Black series with connectivity and components labeled.

The solution we provide to these issues is to provide a pair of interoperable development boards that incorporate all of the required electrical functionality for the development of a wide range of small mobile robotics platforms. The two boards provide equivalent functionality in different form factors.

At the conclusion of the first offering of MAE143c, there were two primary equivalents to the Arduino supporting embedded-Linux, the Raspberry Pi and BeagleBone Black. The BeagleBone provides significantly higher connectivity for sensors and robotics-applicable integrated circuits. It is also an entirely open-source hardware platform which aligns with the aforementioned strengths that encourage a healthy development environment. We therefore elect to expand

the BeagleBone community with a daughter board known as the Robotics Cape as pictured in figure 2.3 and on the EduMiP pictured in figure 2.13.

Even though this PCB removes the hand-wiring of circuits from the course assignments, the design of digital circuits is still incorporated into the course material, and is actually expanded to include the functionality of GPIO inputs for use of buttons, LEDs, PWM control of H-bridges, and quadrature encoder counting. Any time now not wasted debugging faulty connections in the solderless breadboard can now be focused on learning these new circuits, as well as studying the schematic actually by the Robotics Cape. This teaching of best-practices in schematic design is arguably a more applicable skill in the modern world than debugging breadboards. This schematic in its entirety is pictured in figure 2.4.

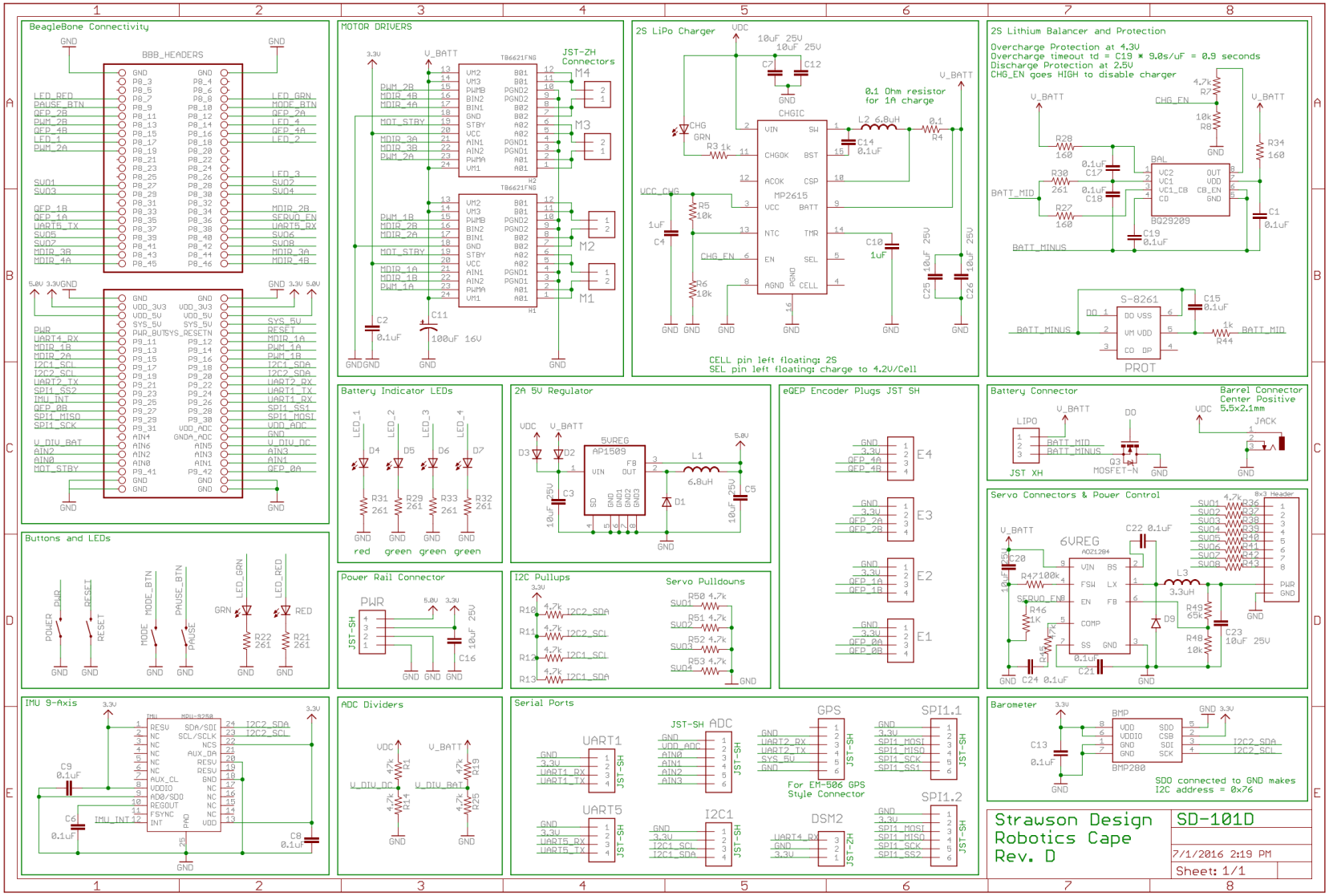


Figure 2.4: Open-source schematic for the Robotics Cape as made available for reference by users and as used during the MAE144 curriculum as an example of several common digital circuits and as an example of standard practices in schematic design.

After much deliberation and discussion with instructors and students, the supporting feature list is constructed as follows:

- 2 Cell Lipo Charging, Balancing, and Protection
- Battery Charge Indicator LEDs
- 4 H-Bridge DC Motors Controllers 1.2A each
- 8-Channel Servo/ESC Output Enabled by PRU
- 6V 4A Regulated Power Supply to Protect Servos
- 9-Axis IMU: Invensense MPU-9250
- Barometer: Bosch BMP280
- 5V 2A Switching Regulator for Robust BBB Power Supply
- 8-18V DC Input Jack to Power BBB and Charge LiPo Battery
- 4 Quadrature Encoder Inputs Enabled by eQEP and PRU
- GPS Input with EM-406/EM-506 Style UART Connector
- Headers for I2C UART SPI ADC PWM GPIO
- Supports DSM2 and DSMX Satellite Radios
- Easy to Access Buttons and LEDs

This set of components and hardware functionality is chosen to reduce the amount of wiring, and therefore points of failure, of the later-discussed three reference robots as much as possible. Wiring of the EduMiP, for example, requires only plugging in of easily-removable connectors for the battery, two motors, and two quadrature encoders. It also seeks to incorporate

a set of components that support the most common functions required by robots such as driving motors, servos, power management, and orientation sensing.

One of the major hassles in prototyping, maintaining, demonstrating, and using robots and consumer products such as remote controlled toys is the charging and maintenance of batteries. Lithium batteries, desirable for their capacity and current supply ability, are particularly dangerous when used carelessly. On multiple occasions, students of the 2012 offering of MAE143c experienced short-circuits while transporting their MiP in backpacks often full of assorted metal hardware and spare wires. This was not catastrophic due to the use of small and safe 9V batteries. This would be an unacceptable with lithium batteries.

To bring the modern capabilities of lithium batteries to the education space in a safe manner, we include a charger, cell voltage balancer, and under/over voltage protection circuits to the Robotics Cape to provide a self-contained battery management system much like that of a laptop. This allows a two-cell series lithium battery to be connected to the Robotics Cape semi-permanently via the industry-standard 3-pin JST XH balance connector available on the vast majority of hobby focused lithium battery packs. This provides power to the main processor, sensors, motors, servos, and any other ancillaries that may require power, while at the same time being balanced and protected by the on board circuitry. By incorporating the charger into the development board itself, the necessity for a separate charger and for the continuous connection and disconnection of a battery pack by the user are removed entirely. This drastically reduces the effort required to bring lithium battery technology to small robotic systems, but also reduces the cost and increases the safety of the system while doing so by self-containing all power circuitry into one circuit board protected by conformal coating to avoid potential accidental short circuits.

As trivial as button and LED circuits may seem, they are nearly universally required by robots for basic human interaction, and yet are conspicuously missing from commercial development boards such as Arduino, BeagleBone Black, and the Raspberry Pi. We recognize the immense value that is derived from a student being able to complete the robotics equivalent of

a "Hello World" program on a headless development board by interacting with buttons and LEDs. These two circuits also contribute to the safety aspect while developing robots by giving users a simple method for visually indicating a programming error when running software in a headless fashion via a simple red LED flash, and by allowing one button to be purposed for shutting down robot functions safely such as turning off motors. While the exact utility of these two circuits is left to the implementation by the programmer, we elect to strongly encourage these safe robotics development practices by implementing them for the user by default in the supporting software package's project template.

For orientation sensing, we carefully elect to include the Invensense MPU-9250 for two reasons. Firstly, because it conveniently incorporates a 3-axis accelerometer, gyroscope, and magnetometer in one package which can communicate over an I2C bus which is shared with the Cape's Barometer and BeagleBone's EEPROM, freeing other buses for full control by the user. Secondly because it includes a small microprocessor dedicated to orientation estimation and gesture recognition called the Digital Motion Processor or DMP. As taught in the course material, userspace applications are not well-suited to servicing sensors and interrupts at high frequencies. This inclusion of the DMP provides reliable orientation estimation and gesture detection even when the main processor is under heavy load causing the primary userspace application controlling the robot to be unable to service the sensor in a timely fashion. To complement the IMU we also include a Bosch BMP280 barometer to support altitude estimation for aerial robotics. This is a popular barometer for its excellent documentation and fair balance between cost and accuracy.

To further support aerial and ground platforms alike, 8 output channels are provided to drive servos and brushless motor drivers typically found in quadrotors and RC planes. Since many servos are not capable of handling the maximum 8.4V output from the driving 2-Cell lithium battery and brushless motor drivers do not require power at all on the signal connector, power to the servo connector rail is provided by a software-controllable 6V switching regulator

which was thoroughly stress tested and proven reliable with abusive high current steady loads and PWM-switching loads as generated by hobby servos.

To support ground vehicles such as EduMiP and EduRover, four H-bridges provide bidirectional control of small DC motors up to 1.2A continuous directly off the 2-cell lithium battery connector. The current rating choice was difficult as users will frequently try to overload motor channels due to an unwillingness to read documentation, follow instructions, or put in the effort to implement dedicated external motor drivers driven by a serial port or similar. In the end, the 1.2A H-bridges were chosen to keep within the practical limits of heat dissipation of a board this size and the safe current limits of the battery connection.

2.4 BeagleBone Blue

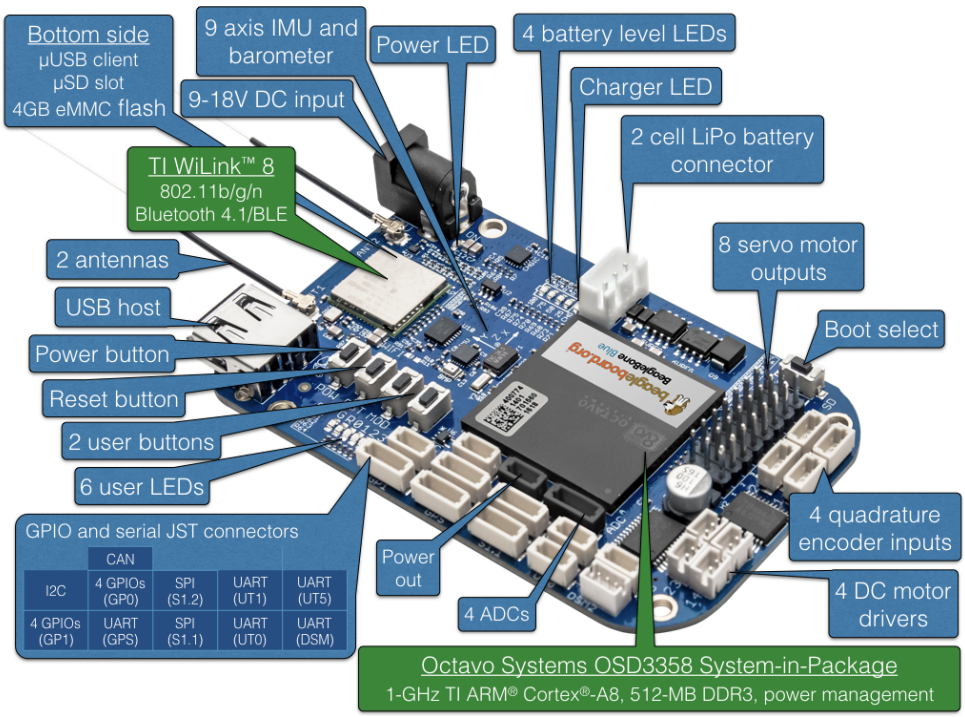


Figure 2.5: BeagleBone Blue with features shared with Robotics Cape highlighted in blue. Image courtesy of RenaissanceRobotics.com and included with permission from its creator.

The there are many benefits of the mother/daughterboard arrangement. Firstly, the Robotics Cape, as a daughterboard, can build upon the existing hardware and software developed by the BeagleBone community. It also appeals to an audience of existing users of the BeagleBone series by presenting a low-cost accessory to a product many already own and have experience with. Separating functionality into two boards also reduces the cost of repair should a single board fail due to user or manufacturing error. From the mechanical engineering perspective, the form factor of two fiberglass PCBs, and motherboard and daughterboard, sandwiched together by friction-fit header pins actually provides significant structural rigidity. This particular advantage is utilized fully in the construction of EduMiP as pictured in figure 2.13 which uses the sandwiched PCBs as the primary structural backbone of the robot.

However, in the interest of cost, weight, and elegance, the author and this dissertation's chair, Thomas Bewley, partnered with the BeagelBoard.org foundation to design and produce an updated version of the BeagleBone Black which incorporates all of the robotics-focused circuitry of the Robotics Cape. This is product, the BeagleBone Blue, is pictures in figure 2.5 and manages to be lighter and cheaper than the combination of a BeagleBone Black and Robotics Cape together. This product appeals more to new users who do not already have a BeagleBone Black, and to those seeking to reduce weight as much as possible, the author included.

2.5 Coherent Software Environment: librobotcontrol

As evident from the vast popularity of the Arduino platform, a coherent and well documented software development environment is critical to the adoption of the platform, as well as the desirability and effectiveness of the platform. Despite its general clarity and easy of use, the Arduino IDE provides an extremely limited number of functions to the user. This is partly due to the lack of hardware features of the development boards. Sadly, newcomers learning to program must spend time scouring the internet for example code explaining how to interface with their chosen sensors and hardware, or they must write drivers themselves.

Instead, we elect to provide a complete and thorough C API that interfaces with all of the provided hardware, as well as providing timing, math, multithreading, and a host of other functions optimized for the platform to ensure a smooth development experience as well as teach good practices along the way. The result of this effort is the librobotcontrol Debian package containing over 114,000 lines of code which is now distributed with every BeagleBoard development board and has been adopted as the official hardware support package for BeagleBone IO interfacing.

2.5.1 Modules

The library itself is broken into independent modules which can be individually included in a user's program. A complete list of these modules is presented in figure 2.6. A complete description of all of library is beyond the scope of this dissertation, however, full details and source code can be found at http://strawsondesign.com/docs/librobotcontrol/. All function names begin with the "rc_" prefix followed by the module name to make searching the documentation and finding related functions straightforward.

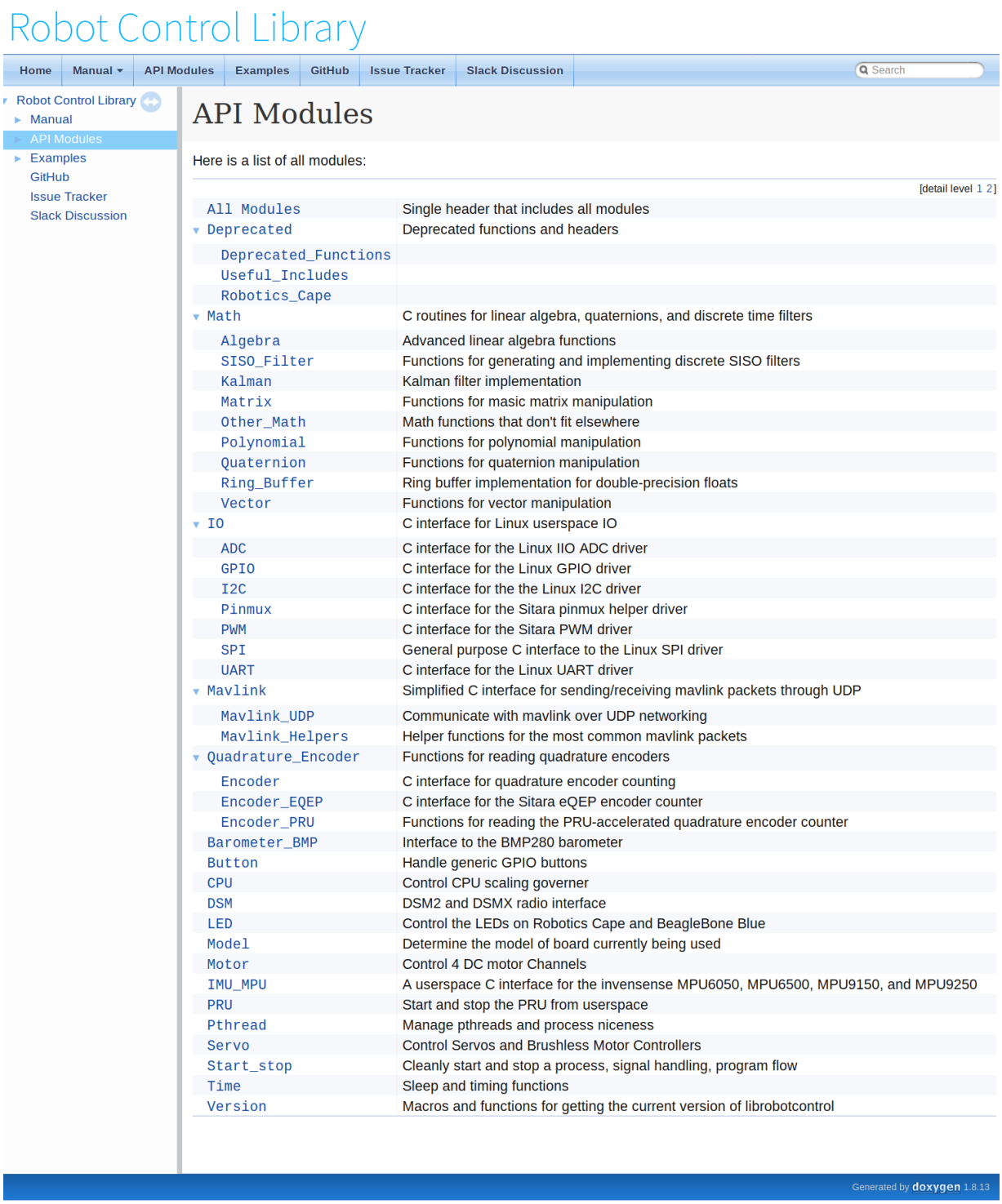


Figure 2.6: Organization of API modules of the librobotcontrol library.

2.5.2 Examples

Each module of the library has one or more associated test or example programs which come distributed with the binary and source packages. These highlight one of the critical advantages of developing robotics applications in userspace instead of on bare metal which is the ability to quickly test and verify hardware and functions by utilizing simultaneously installed executable binaries.

- rc_altitude.c
- rc_balance.c
- rc_benchmark_algebra.c
- rc_bind_dsm.c
- rc_blink.c
- rc_calibrate_accel.c
- rc_calibrate_dsm.c
- rc_calibrate_escs.c
- rc_calibrate_gyro.c
- rc_calibrate_mag.c
- rc_check_battery.c
- rc_cpu.c
- rc_dsm_passthrough.c
- rc_kill.c
- rc_model.c
- rc_spi_loopback.c
- rc_test_adc.c
- rc_test_algebra.c
- rc_test_bmp.c
- rc_test_buttons.c
- rc_test_dmp.c
- rc_test_dmp_tap.c
- rc_test_drivers.c
- rc_test_dsm.c
- rc_test_encoders.c
- rc_test_encoders_eqep.c
- rc_test_encoders_pru.c
- rc_test_escs.c
- rc_test_filters.c
- rc_test_kalman.c
- rc_test_leds.c
- rc_test_matrix.c
- rc_test_mavlink.c
- rc_test_motors.c
- rc_test_mpu.c
- rc_test_polynomial.c
- rc_test_pthread.c
- rc_test_servos.c
- rc_test_time.c
- rc_test_vector.c
- rc_uart_loopback.c
- rc_version.c

2.5.3 Documentation

The entire librobotcontrol package including example programs, tutorials, and the library functions itself, is completely documented with the help of Doxygen. This documentation is made available by the author at http://strawsondesign.com/docs/librobotcontrol/. This method is

chosen because the documentation itself is constructed from the comments contained in the code which it is documenting. The result is that good coding practices result in very little additional effort being required for full code documentation, and any changes in the code itself are reflected in the documentation automatically without the need to keep two disjoint files in sync.

Every function in the library has a complete description along with hyperlinks to example source code which demonstrate their use and to various type definitions.

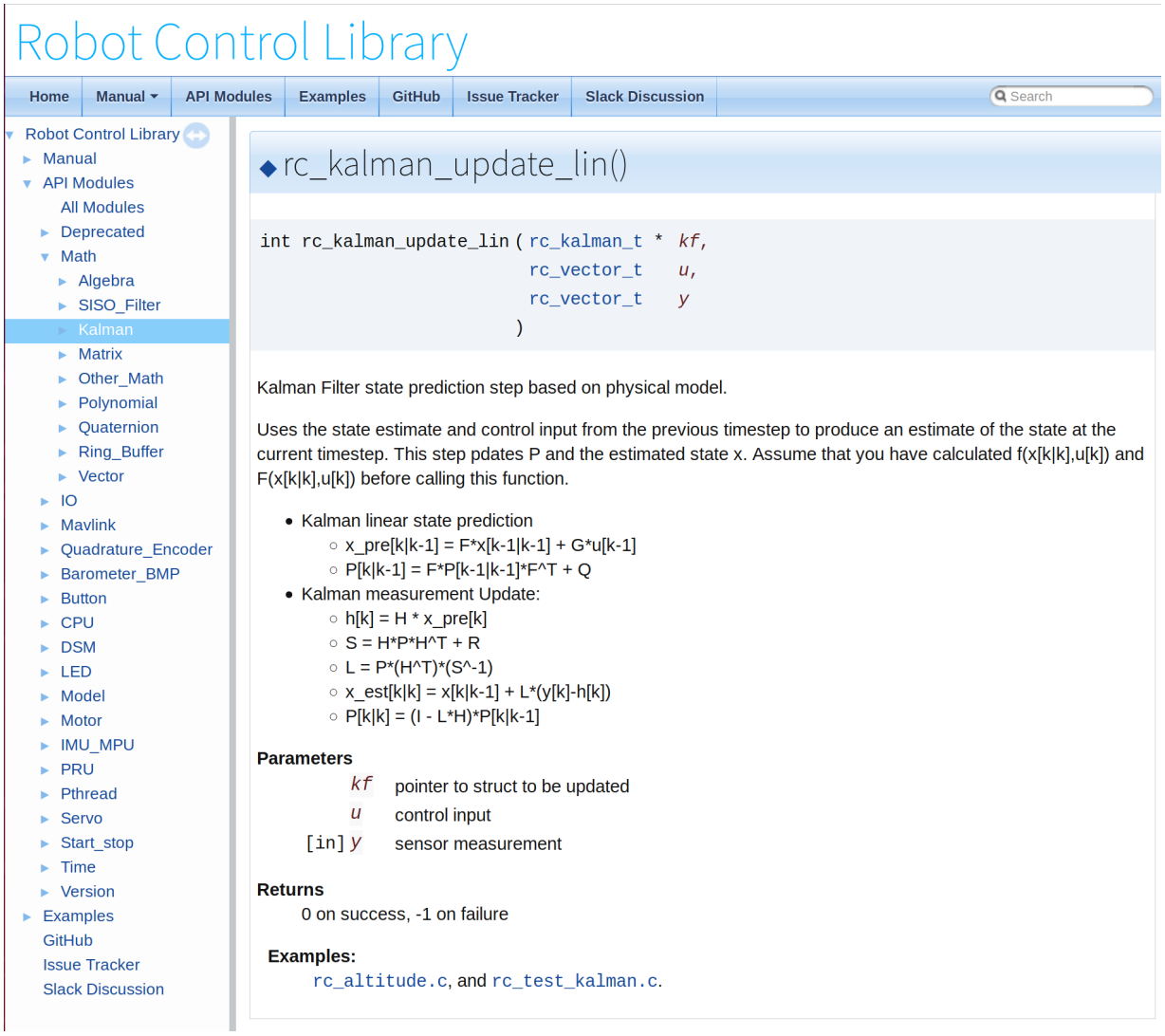


Figure 2.7: Sample of the style and detail of the function documentation provided for every single function in librobotcontrol.

2.5.4 Project Template

A critical first step in learning a new programming language or developing in a new environment is the "Hello World" program. The first class in which the Robot Control environment is presented walks the students through the Robot Control project template and the steps it takes to build upon a basic "Hello World" to create a safely operating robot controller in this environment.

The critical components of this template are as follows:

- Ensure no existing instances are running and make new PID file
- Start a signal handler routine
- Initiate a global program flow state variable to coordinate multiple threads
- Assign a button on the development to control the state variable and shut down the program
- Sleep during loops
- Shut down all hardware cleanup before program exist

While these programming principles are not revolutionary, they are presented in a particularly concise manner in this embedded controls course in the context of a robot control program to students with little or no programming background. This template program, along with the architecture of the Robot Control library, is focused on promoting these best practices to help students learn and produce the best product possible during the course.

```
/**
 * @file rc_project_template.c
 *
 * This is meant to be a skeleton program for Robot Control projects. Change
 * this description and file name before modifying for your own purpose.
 */

#include <stdio.h>
#include <robotcontrol.h> // includes ALL Robot Control subsystems
```

```
10
11 // function declarations
12 void on_pause_press();
13 void on_pause_release();
14
15
16 /**
17  * This template contains these critical components
18  * - ensure no existing instances are running and make new PID file
19  * - start the signal handler
20  * - initialize subsystems you wish to use
21  * - while loop that checks for EXITING condition
22  * - cleanup subsystems at the end
23  *
24  * @return     0 during normal operation, -1 on error
25  */
26 int main()
27 {
28   // make sure another instance isn't running
29   // if return value is -3 then a background process is running with
30   // higher privaledges and we couldn't kill it, in which case we should
31   // not continue or there may be hardware conflicts. If it returned -4
32   // then there was an invalid argument that needs to be fixed.
33   if(rc_kill_existing_process(2.0)<-2) return -1;
34
35   // start signal handler so we can exit cleanly
36   if(rc_enable_signal_handler()==-1){
37     fprintf(stderr,"ERROR: failed to start signal handler\n");
38     return -1;
39   }
40
41   // initialize pause button
42   if(rc_button_init(RC_BTN_PIN_PAUSE, RC_BTN_POLARITY_NORM_HIGH,
43             RC_BTN_DEBOUNCE_DEFAULT_US)){
44     fprintf(stderr,"ERROR: failed to initialize pause button\n");
45     return -1;
46   }
47
48   // Assign functions to be called when button events occur
49   rc_button_set_callbacks(RC_BTN_PIN_PAUSE,on_pause_press,on_pause_release);
50
```

```
  // make PID file to indicate your project is running
  // due to the check made on the call to rc_kill_existing_process() above
  // we can be fairly confident there is no PID file already and we can
  // make our own safely.
  rc_make_pid_file();


  printf("\nPress and release pause button to turn green LED on and off\n");
  printf("hold pause button down for 2 seconds to exit\n");

  // Keep looping until state changes to EXITING
  rc_set_state(RUNNING);
  while(rc_get_state()!=EXITING){
    // do things based on the state
    if(rc_get_state()==RUNNING){
      rc_led_set(RC_LED_GREEN, 1);
      rc_led_set(RC_LED_RED, 0);
    }
    else{
      rc_led_set(RC_LED_GREEN, 0);
      rc_led_set(RC_LED_RED, 1);
    }
    // always sleep at some point
    rc_usleep(100000);
  }

  // turn off LEDs and close file descriptors
  rc_led_set(RC_LED_GREEN, 0);
  rc_led_set(RC_LED_RED, 0);
  rc_led_cleanup();
  rc_button_cleanup();  // stop button handlers
  rc_remove_pid_file(); // remove pid file LAST
  return 0;
}


/**
 * Make the Pause button toggle between paused and running states.
 */
void on_pause_release()
{
```

```
  if(rc_get_state()==RUNNING) rc_set_state(PAUSED);
  else if(rc_get_state()==PAUSED) rc_set_state(RUNNING);
  return;
}

/**
* If the user holds the pause button for 2 seconds, set state to EXITING which
* triggers the rest of the program to exit cleanly.
**/
void on_pause_press()
{
  int i;
  const int samples = 100; // check for release 100 times in this period
  const int us_wait = 2000000; // 2 seconds

  // now keep checking to see if the button is still held down
  for(i=0;i<samples;i++){
    rc_usleep(us_wait/samples);
    if(rc_button_get_state(RC_BTN_PIN_PAUSE)==RC_BTN_STATE_RELEASED) return;
  }
  printf("long press detected, shutting down\n");
  rc_set_state(EXITING);
  return;
}
```

2.6 Three Reference Designs

To support this robotics ecosystem and its associated curriculum, we present a trio of robots that serve as reference designs, testing platforms, and physical devices for students to interact with and take ownership of. All three target different education levels as their complexity of control progresses. Their design is tightly connected to the component choices and layout of the Robotics Cape and BeagleBone Blue as their concept and rough design parameters were envisioned during the conceptual design phase of the Robotics Cape as motivating applications.

2.6.1 EduRover

The first of the trio is EduRover, a four-wheel drive and four-wheel steerable vehicle designed to target late high school and early collage students interested in introductory robotics. That platform is chosen for this audience because it leverages interesting kinematics possible with four-wheel steering without requiring knowledge of dynamic behavior to make functional.

This platform provides the motivation for the Robotics Cape's four H-bridge motor drivers, four quadrature encoder inputs, DSM radio control input, and 6V voltage regulator for safe operation of hobby-grade servos. It leverages the same motors, charger, battery, and some hardware as EduMiP to maintain consistency within the ecosystem for the manufacturer and consumers alike.

Like the other two platforms, we provide sample code that demonstrates functionality of EduRover. In this case, the sample code supports four main steering modes that demonstrate what is possible with the 120 degree range of motion in the steering axis of each of the wheels.

Figure 2.8: EduRover 4-wheel steering 4-wheel drive vehicle with hyper-articulate steering.

- Regular 4-wheel steering providing tightest turn radius with forward motion.
- Crab motion allowing sideways motion and parallel parking maneuvers. (Figure 2.10)
- Spin mode allowing rotation around the chassis centroid.
- Parallel lane-change mode where the front and back wheels turn identically.

In each mode, Ackerman steering angles are calculated in software before being sent to the four steering servos. This is a concept that can easily be taught before students learn dyanamics and differential equations, yet the effect of wheel scrub and Ackerman steering can easily be tuned and observed with this system.

Thanks to the IMU on-board the Robotics Cape, EduRover and its accompanying sample software will automatically steer its wheels such that two tires will touch the ground in a normal fashion when the unit is rotated 90 degrees up onto any pair of wheels. A feedback loop, much like EduMiP, then keeps EduRover balanced on any of its four sides. This maneuver does have to

be done by manually picking it up, the motors and gearboxes are designed with enough torque that the unit has successfully demonstrated driving up to a wall and lifting itself 90 degrees up the wall face before automatically starting the balancing feedback loop.

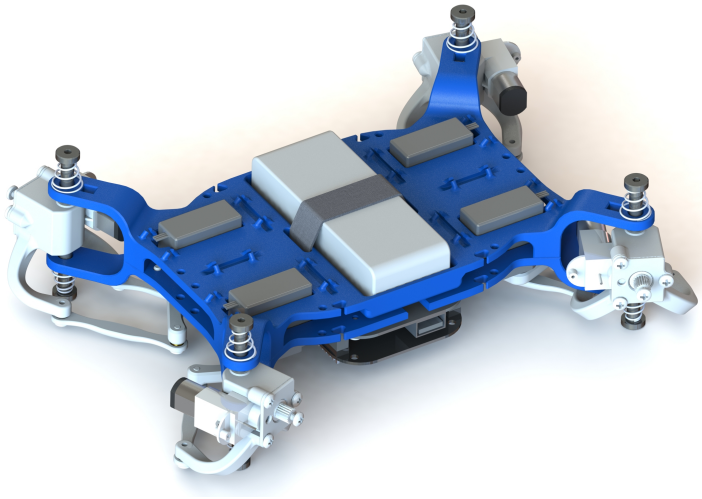


Figure 2.9: CAD rendering of underside of EduRover demonstrating drivetrain and steering layout.

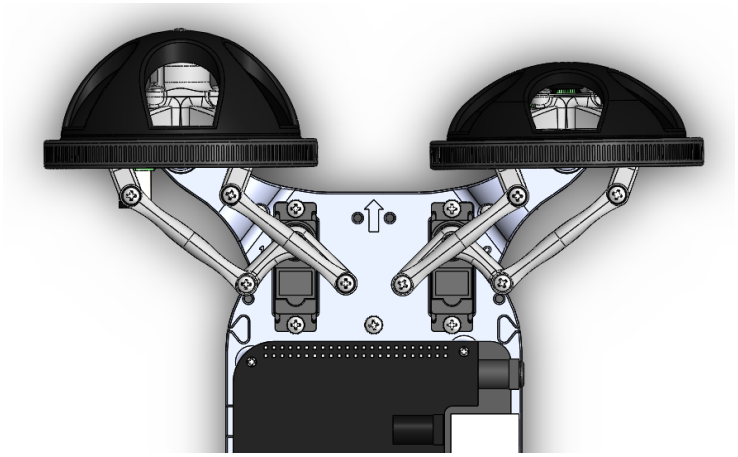


Figure 2.10: CAD rendering of wheels turned to crab mode to allow sideways motion.

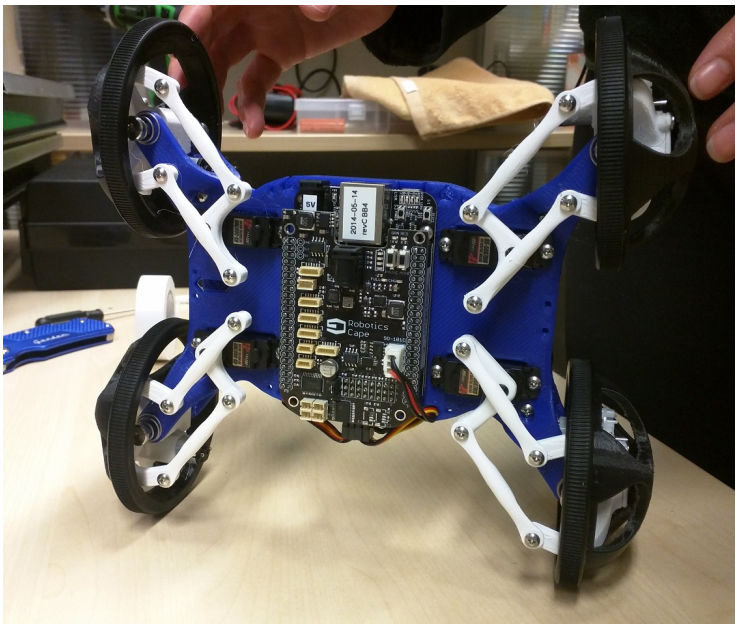


Figure 2.11: EduRover with wheels turned in crab mode demonstrating orientation for balancing.

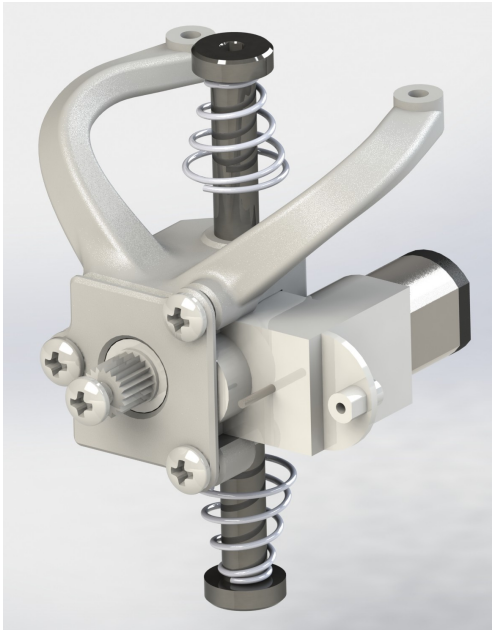


Figure 2.12: CAD rendering of EduRover's hub containing motor, gearbox, steering arms, and kingpin suspension.

2.6.2 EduMiP

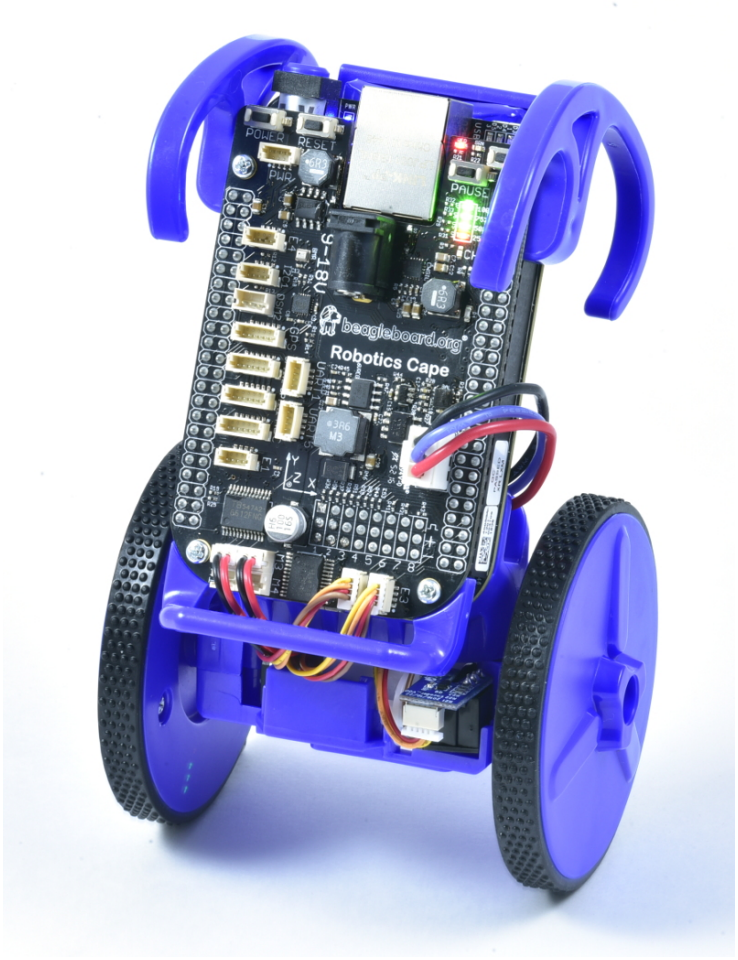


Figure 2.13: Current BeagleBone-Based EduMiP

EduMip is the cornerstone reference design of this ecosystem and the primary topic of the UCSD course MAE144: Embedded Control and Robotics. As discussed briefly before, it is an unstable, nonminimum-phase âĂIJmobile inverted pendulumâĂİ, with dynamics similar to the standard test problem (a.k.a. âĂIJplantâĂİ) of a pendulum swinging freely from a cart as it moves along a track, but is much more compact, economical, and fun. We have found EduMiP to be quite useful and versatile for teaching feedback control theory at the professional level, and have implemented several different types of controllers to stabilize it, including classical (SISO)

control strategies in the successive loop closure (SLC) framework, state-space control strategies, and adaptive control strategies.

EduMiP is about 6âĂİ tall and 4âĂİ wide, with injection molded structural members and incorporates the robust wheel assembly used in the consumer âĂIJMiPâĂİ toy (Figure 2.2), which was developed and manufactured in partnership between our lab and WowWee Robotics. EduMiP is rather simple to assemble and extend, with small wire bundles to connect the cape to the two motors, to the two encoders, and to the LiPo battery in its minimal configuration. At the maker level, EduMiP is useful to motivate more advanced (college-level) investigations in dynamic modeling and feedback control; a reference control solution, which makes the vehicle self-upright from horizontal and balance in the upright configuration, is provided for a rewarding out-of-the-box experience.

2.6.3 EduMAV

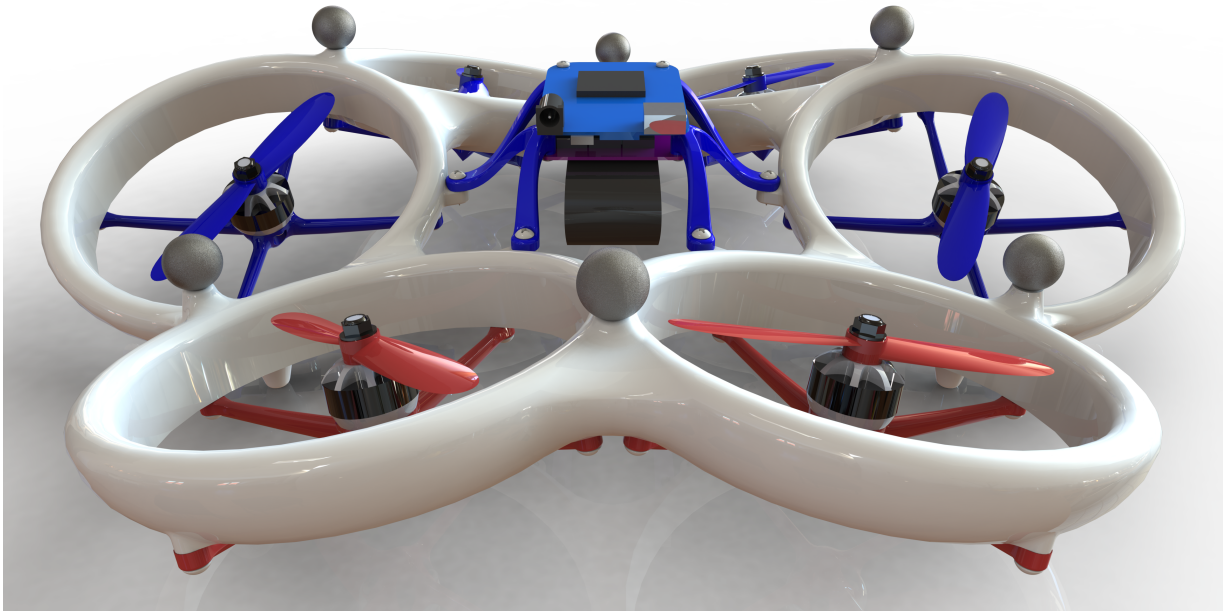


Figure 2.14: EduMAV render, front view.

EduMAV, or Educational Micro Air Vehicle, is the final of the three reference designs and targets graduate research and R&D in 3D dynamics. The remainder of this dissertation focuses on the optimization, design, and control of EduMAV so we will conclude this section here to avoid repetition.

2.7 Impact

There are a number of essential topics in robotics that can be taught with an exceptionally effective, hands-on, exploratory approach using the family of vehicles and accompanying hardware/software described here, including:

- Discrete-time control and stability augmentation of continuous-time stable and unstable systems.
- Multivehicle coordination algorithms.
- Multithreaded software architectures leveraging open-source standards on powerful low-cost Linux computers.
- PCB design software (e.g. Eagle) coupled with mail-order PCB fabrication and low-volume PCB assembly.
- 3D CAD software (e.g. SketchUp or Solidworks) coupled with 3D printing and laser cutting for rapid prototyping.
- Smartphone and tablet SDKs, and their use in the development of a general app for control of robots.

What began as the development of a simple controls lab for a one-quarter senior-level course at UCSD has grown into a meticulous representative embodiment of a wide range of key topics in robotics, including multithreaded software architectures, embedded controls, rapid prototyping, custom PCB fabrication, integration with smartphones & tablets, and design-for-large-scale-manufacturing. The resulting educational program provides a unique backdrop for motivating makers with many aspects of STEM that they might otherwise be unaware of, and provides students in college and industry with compelling capstone projects in robotics that tie

together key component technologies and concepts, and provide clear best-practice example realizations for the solution of many common problems in robotics

The result is an unprecedented, interdisciplinary, highly motivating learning experience based on agile high-function robotic vehicle prototypes that students continue to hack, extend, reference, and learn from long after the completion of formal classwork. Universities typically disjoin the study of challenging technical fields into fragmented departments, like mechanical & aerospace engineering, electrical engineering, computer science, control & cybernetics, and human factors. Modern applications in robotics require the rejoining of these traditionally isolated disciplines of study. The effective design and control of robotic vehicles requires a fundamentally interdisciplinary perspective that is ill-served by keeping the teaching of its constituent technical components disjoint. There is an emerging need for educational institutions to distill and relate these constituent disciplines, and the remarkable recent advancements therein, to a new generation of roboticists. Towards this end we have developed this program, which aims to provide a systematic, integrated introduction to the fundamental technologies and techniques available, focusing on control & coordination algorithms, open standards & tools, and software architectures that may be broadly used.

2.8 Acknowledgements

Chapter 2, in part, is a reprint of the material as it appears, titled "Leveraging Open Standards and Credit-Card-Sized Linux Computers in Embedded Control & Robotics Education" SciTech 2015. Bewley, Thomas; Briggs, Clark; Strawson, James. The dissertation author was a coauthor of this material and was responsible for the design and manufacturing of the electronics & hardware, as well as teaching a significant portion of the course materials and writing of the text.

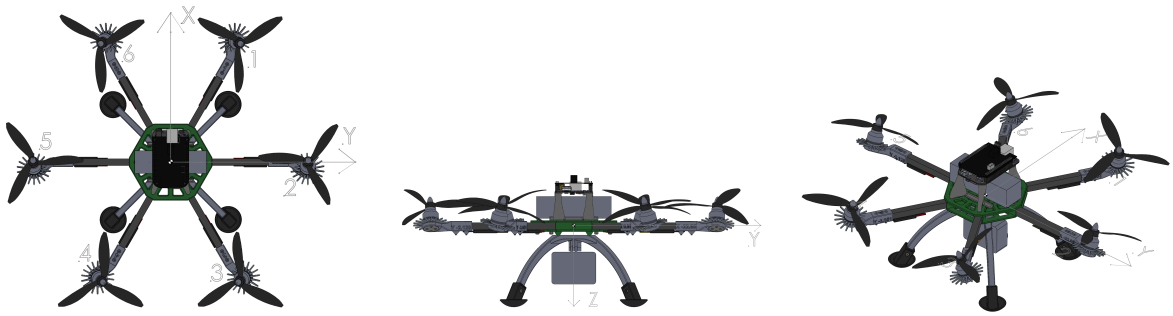
Chapter 3

Multirotor Layout Optimization

3.1 Introduction

The need to pitch and roll to stabilize the spatial position of a traditional multirotor has two major drawbacks. First, two- or three-axis gimbals must be used to keep any on-board cameras steady while hovering, which draws power and adds unnecessary weight and complexity. Second, controlling the six degrees of freedom (6DOF) of the vehicle in an underactuated setting, with only the traditional four control inputs (roll, pitch, yaw, and lift) necessitates more advanced control strategies, such as successive loop closure control or backstepping [FSZ+13], and results in a slower system response.

Shimizu et al. [SSK+15] demonstrated that careful arrangement of at least six propellers in a multirotor frame is sufficient to generate independent control authority over all six degrees of freedom. CyPhy Works [Wor] proposed a hexacopter capable of level forward flight, mitigating the need for a camera gimbal. In both efforts, the rotor angle selection was apparently based on intuition rather than systematic thrust vector optimization.



(a) (b) (c)

Figure 3.1: CAD rendering of the airframe used in the case study, showing the coordinate system and rotor indexing.

3.2 Quantifying Performance

As mentioned above, Shimizu et al. [SSK+15] demonstrated direct independent control of the six degrees of freedom of a hexacopter (three thrust vectors along the axes of the hexacopter's body coordinate system, and three moments about these axes) by modulation of its six control inputs, given that the rotor angles of the hexacopter are tilted appropriately. Given a set of rotor locations, the approach discussed in the present paper optimizes the set of rotor orientations to maximize a weighted sum of the available moments and forces about and along each axis. These maximum forces and torques are the real-world outputs of the system when control inputs are applied independently to a multirotor while in an equilibrium hover state, and takes into account the real-world upper and lower saturations of the control inputs to the individual motors. It is shown that the optimized result is generally left-right anti-symmetric. A performance metric that sums maximum control authority along each of the orthogonal control directions is used, with possibly different weights placed on the positive and negative directions.

3.2.1 Frame Layout Definition

To simplify the equations of motion, this paper uses the convention of centering the body coordinate system at the center of mass of the multirotor, using the conventional aerospace NED

coordinate system, with Z pointing out the bottom of the multirotor frame, X pointing forward, and Y pointing to the right. Roll, pitch, and yaw angular directions follow the right-hand rule about X, Y, and Z respectively. All forces in this paper are reported in Newtons, with moments in Newton meters, and angles in radians.

The center of pressure on each of the propellers is assumed to be in the centroid of the propeller itself, with the applied force acting colinearly with the rotor axis. Since multirotor frames typically use outrunner-style brushless motors with the motor mounted to the frame beneath the propeller, it is important to design and construct the frame by locating the centroids of the propellers themselves, not the motor mount bases. The positions of these six propeller centroids in space, relative to the center of mass of the frame, are given by six vectors, $c(1)$ through $c(6)$, each of length three.

For our case study, the propeller positions form an evenly spaced circle in a plane 30.7mm above the center of mass as derived from our CAD model; the majority of existing multirotor frames implement such evenly spaced arrangements. However, the method described here accommodates much more general arrangements of the rotors, even including asymmetric layouts. In section 3.2.4, it is shown that controllability of a particular configuration can be verified by ensuring that the system's force matrix has full rank. A less common V-style hexacopter arrangement, popular for supporting wide camera angles, is suitable in this regard, and can benefit significantly from the optimization method suggested herein.

The layout of the case-study considered in this work is shown in top, back, and isometric views in Fig.3.1. Note that the six rotors are indexed clockwise about the origin starting at the front right. The NED coordinate system used is also marked in this figure.

3.2.2 Rotor Orientation Definition

Since the the force generated by each propeller is assumed to be applied to the propeller's physical centroid, it is convenient to define the orientation of each rotor axis as a unit vector starting at the propeller centroid location and extending in the direction of the force applied by the propeller. Each rotor then has two degrees of freedom to be optimized, namely, the tilt in the X and Y directions from a nominal vertical orientation. For a 6-rotor hexacopter frame, this results in 12 parameters to optimize over. The x and y components of these six unit vectors relative to their origin are organized into two vectors v_x and v_y.

If v_x and v_y contained all zeros, then all rotors would be pointing upwards in the negative Z direction. The final optimized hexacopter orientation found in the present work is

$$\begin{aligned} v_x &= [0.233, 0.031, -0.195, 0.195, -0.031, -0.233], \\ v_y &= [-0.173, 0.391, -0.218, -0.218, 0.390, -0.171]. \end{aligned} \tag{3.1a}$$

A visualization of these vectors is shown in Fig.3.2. The final result of this paper is presented here in the introduction to assist the reader in visualizing the system under examination.

For completeness, the corresponding six components in the Z direction of the unit force vectors are also organized in vector v_z. Since these are derived from the x and y components they cannot be manipulated while optimizing and are not part of the search space. However, they are used when constructing the force matrix. The v_z corresponding to the v_x and v_y given in (3.1) are given by

$$v_z = -[0.957, 0.920, 0.956, 0.956, 0.920, 0.957]; \tag{3.1b}$$

note that all values are negative, as the force vectors point upwards, and NED coordinates orient the Z axis downwards.

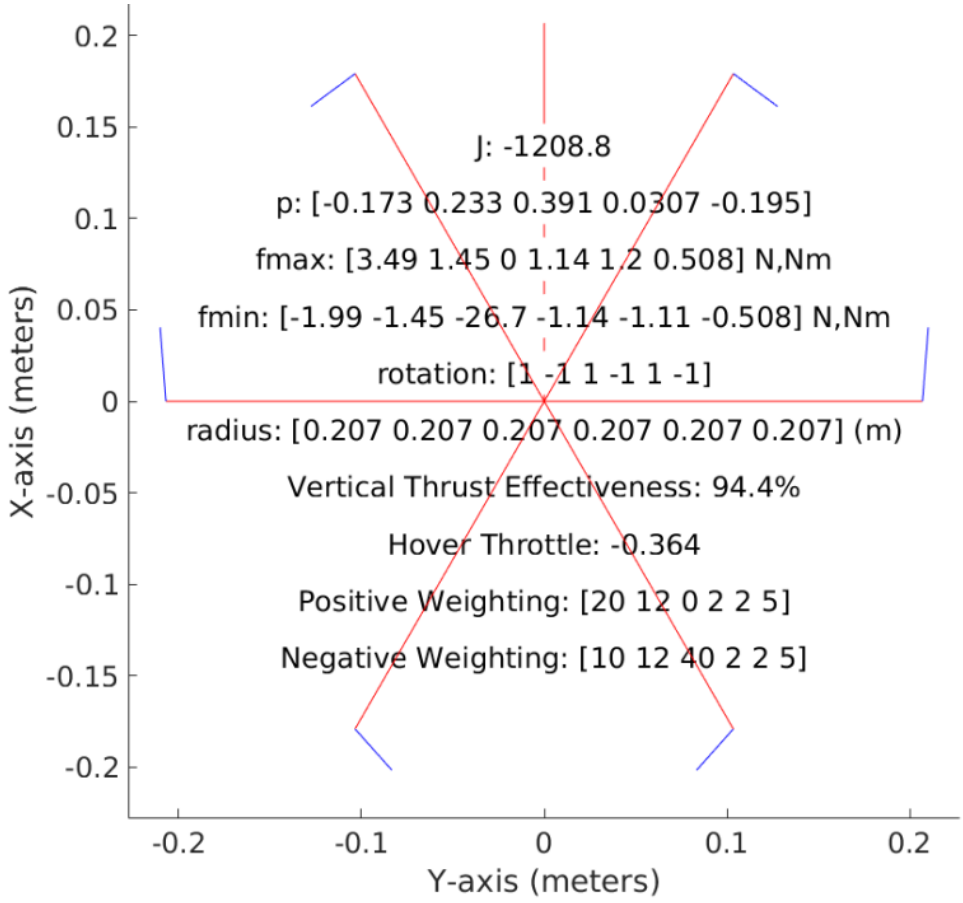


Figure 3.2: Geometric representation of optimized airframe showing thrust vectors along rotor axes. The single red line along pointing in the positive X direction originating at the origin indicates the direction of forward flight.

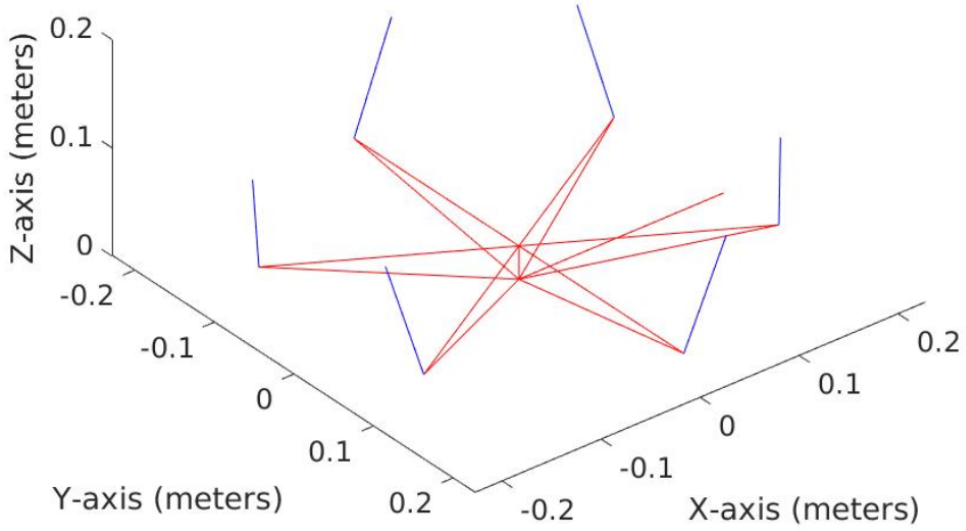


Figure 3.3: Geometric representation of optimized airframe showing thrust vectors along rotor axes. The single red line along pointing in the positive X direction originating at the origin indicates the direction of forward flight.

3.2.3 Force Matrix

The 6×6 force matrix F defines the contributions from each rotor to each of the three forces and three moments on the body. Representing this matrix accurately requires experimentally-derived properties of the motor and propeller combination (the rotor), specifically the maximum force f_{max} and maximum torque τ_{max} that each rotor generates under steady load. The vast majority of commercial motors and propellers in the multirotor industry today are unidirectional, so they can be said to accept a normalized control input u from 0 (off) to 1 (max), where a positive control input u_i of 1 to each rotor generates a positive steady-state force and torque of f_{max} and τ_{max}. For our case study these maximum values and the total mass of the frame m_f are

$$f_{max} = 4.72N, \quad \tau_{max} = .0775Nm, \quad m_f = 0.992kg. \tag{3.2}$$

There is freedom in the mechanical design to choose the direction of rotation of each rotor; these directions are summarized here as rotation vector r with entries of -1 indicating

clockwise rotor rotation, and $+1$ indicating counterclockwise when viewed from above the frame. While the motor torques on the frame are small as compared with the moments generated by the rotor thrusts, they are not entirely negligible, and it is shown in §3.4.2 that reversing the propeller spin directions can in some cases flip the optimized orientation of the rotors. For our case study, we use a rotation vector where the front two rotors spin inwards and the remaining rotors alternate directions around the frame, as is typical in multirotor frames; that is, assuming six rotors,

$$r = [1, -1, 1, -1, 1, -1] \tag{3.3}$$

The force matrix F can now be constructed by summing together the the force and moment from each rotor along and about each body axis. Each row i corresponds to contributions from an individual rotor and the columns are the resulting forces and moments (in units of N and Nm) resulting from a unit control input to rotor i. This structure is defined, in the case of six rotors, as follows.

$$F = \begin{bmatrix} f_x^1 & f_y^1 & f_z^1 & \tau_{roll}^1 & \tau_{pitch}^1 & \tau_{yaw}^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_x^6 & f_y^6 & f_z^6 & \tau_{roll}^6 & \tau_{pitch}^6 & \tau_{yaw}^6 \end{bmatrix} \tag{3.4}$$

Since the rotor orientations are conveniently defined as unit vectors we can sum together the components of each rotor's force and moment in each direction to construct the entries of the

force matrix F. This may be extended to larger numbers of rotors by adding additional rows to F.

$$\begin{aligned}
f_x^i &= v_x^i f_{max} \\
f_y^i &= v_y^i f_{max} \\
f_z^i &= v_z^i f_{max} \\
\tau_{roll}^i &= (c_z(i)v_y^i - c_y(i)v_z^i)f_{max} - \tau_{max} r^i v_x^i \\
\tau_{pitch}^i &= (c_x(i)v_z^i - c_z(i)v_x^i)f_{max} - \tau_{max} r^i v_y^i \\
\tau_{yaw}^i &= (c_x(i)v_y^i - c_y(i)v_x^i)f_{max} - \tau_{max} r^i v_z^i.
\end{aligned} \tag{3.5}$$

3.2.4 Mixing Matrix

The next step is to construct a mixing matrix which will be used in the implemented flight controller. Mixing is the process of generating individual control signals to each of the rotors which result in orthogonal control inputs along the 3 axes and around them in roll, pitch and yaw. This allows SISO feedback controllers to be used for each of the orthogonal degrees of freedom [WB10]. For a linear control model, each SISO controller's output is mixed to generate signals to all six rotors. These motor signals are then summed to generate the final control signals to all motors. For a quadrotor model with four inputs and outputs, the mixing matrix is easy to determine by inspection. This six degree of freedom case is a bit more involved, as discussed below.

The first step is to take the inverse of the force matrix, which of course is only possible if it is full-rank. To extend this method to non-square F (more than six rotors), it is possible to use the Moore-Penrose pseudoinverse to find the lowest-energy solution.

$$M = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_6 \end{bmatrix} F^{-1} = \begin{bmatrix} m_x^1 & \dots & m_x^6 \\ m_y^1 & \dots & m_y^6 \\ m_z^1 & \dots & m_z^6 \\ m_{roll}^1 & \dots & m_{roll}^6 \\ m_{pitch}^1 & \dots & m_{pitch}^6 \\ m_{yaw}^1 & \dots & m_{yaw}^6 \end{bmatrix} \tag{3.6}$$

Next, d_1 through d_6 are chosen to scale the rows of M in such a way that a control input of $+1$ or -1 in any given direction saturates at least one of the motor inputs on its upper or lower bound during steady equilibrium hover conditions. This is done to ensure that the outputs of the SISO feedback controller are scaled appropriately when implemented. To accomplish this, it is necessary to find what inputs to the motors are required for an equilibrium hover state. This will

vary with different payloads, and even different levels of battery charge for a given payload, so a nominal condition is chosen with the battery at half-charge and the mass of a "typical" payload included with the mass of the frame. In this case study, the maximum rotor force was determined on a test-stand at a nominal battery charge of 14.8V.

As the center of mass is at the origin of the coordinate system, the condition for steady hover is a force of $-9.81 \cdot m_f$ in the Z direction, and zero forces and moments in the remaining directions. The motor control signals s_h required for steady hover may be found by solving the linear system in (3.7a) with the solution for our case study given by (3.7b):

$$f_h = F^T s_h = [0, 0, -9.81 m_f, 0, 0, 0]^T, \tag{3.7a}$$

$$\Rightarrow \quad s_h = [0.364, 0.364, 0.364, 0.364, 0.364, 0.364]^T. \tag{3.7b}$$

This result shows that the optimized case given in (3.1) requires 36.4% throttle to all motors to hover. This is a result of the radially symmetric frame layout. Note that the property of requiring equal throttle across all motors for steady hover is observed for symmetric frame layouts, and when a global minimum has been reached during the optimization. Other valid and usable rotor orientations for this frame, resulting from intuition or from incomplete minimization in the optimization of the orientations, do not necessarily exhibit this property.

Next, each row of the mixing matrix is scaled such that, when any single minimum or maximum control force or torque is applied during steady-state hover, no motor control signal s_i exceeds the range $[0, 1]$. In the asymmetric directions, X and pitch, it is likely that one direction may allow more control authority than the other. Thus, the rows of the mixing matrix are scaled such that the direction allowing greater control authority saturates a motor when a control input of -1 or $+1$ is applied. The choice of control input range from -1 to $+1$ only serves to scale the mixing matrix for convenience of implementation, and does not affect the results of this optimization.

Note also that the minimum design force in the lift direction ($-Z$) is taken as zero, as it is impossible to achieve less than this if unidirectional rotors are used; thus, the control actuation u_Z (only) is confined to the $[-1,0]$ range (negative because it corresponds to force in the upward direction). For the case study, the inverses of the force matrix and the appropriately scaled mixing matrix are:

$$F^{-1} = \begin{bmatrix} -0.08 & 0.18 & -0.10 & -0.10 & 0.18 & -0.08 \\ 0.25 & 0.01 & -0.24 & 0.24 & -0.01 & -0.25 \\ -0.04 & -0.04 & -0.04 & -0.04 & -0.04 & -0.04 \\ -0.20 & -0.32 & -0.29 & 0.29 & 0.32 & 0.20 \\ 0.33 & -0.02 & -0.30 & -0.30 & -0.02 & 0.33 \\ 0.68 & -0.72 & 0.70 & -0.70 & 0.72 & -0.68 \end{bmatrix}$$

$$M_{cs} = \begin{bmatrix} -0.27 & 0.64 & -0.34 & -0.34 & 0.64 & -0.27 \\ 0.36 & 0.02 & -0.35 & 0.35 & -0.02 & -0.36 \\ -1.00 & -1.00 & -1.00 & -1.00 & -1.00 & -1.00 \\ -0.23 & -0.36 & -0.33 & 0.33 & 0.36 & 0.23 \\ 0.39 & -0.03 & -0.36 & -0.36 & -0.03 & 0.39 \\ 0.34 & -0.36 & 0.35 & -0.35 & 0.36 & -0.34 \end{bmatrix}$$

3.2.5 Control Force Authority Vectors

Finally, we create two vectors f_{min} and f_{max} of, in each component, the forces (in N) and moments (in Nm) resulting from applying the minimum and maximum control inputs to each channel, independently, while at steady hover. In the real world, the implemented flight controller is unlikely to apply full control input to just one channel at a time while hovering. However, this exercise gives us a handy method of quantifying the minimum and maximum possible force and torque in each component. To find these, we construct two vectors u_{min} and u_{max} consisting of the saturation limits found during the mixing matrix scaling operation,

$$\begin{aligned} u_{min} &= [-0.57, -1.00, -1.00, -1.00, -0.93, -1.00], \\ u_{max} &= [1.000, 1.000, 0.000, 1.000, 1.000, 1.000]. \end{aligned} \tag{3.8}$$

To find the resulting forces generated by these control limits, we simply multiply by the mixing matrix and the force matrix for each direction. Note that, due to (3.6), this can be reduced to multiplying by the row scaling factors d_1 through d_6.

$$\begin{aligned} f_{min} &= F^T M^T u_{min} = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_6 \end{bmatrix} u_{min}^T, \\ f_{max} &= F^T M^T u_{max} = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_6 \end{bmatrix} u_{max}^T, \\ f_{min} &= [-1.99, -1.45, -26.7, -1.14, -1.11, -0.51]^T, \\ f_{max} &= [3.49, 1.45, 0, 1.14, 1.20, 0.51]^T. \end{aligned} \tag{3.9}$$

Note specifically that these are the minimum and maximum *total* forces and torques applied by the rotors to the airframe for the optimized configuration illustration in Figure 3.2; the value of

f_h required to maintain hover, as given in (3.7a), is $g\,m_f = -9.74\ N$ in the third component and zero in the other components. Note that, due to the symmetry of the frame, the control authority along Y and about X and Z are symmetric, but the authority in the others directions are not. It is discussed in §3.2.6, how to manipulate this. Also note that, since the rotors are unidirectional, the maximum thrust in the positive Z direction is 0, corresponding to turning off all rotors.

3.2.6 Objective function used in optimization

To optimize a set of rotor orientations, a criterion for the optimization is defined as a scalar function J to be minimized. We propose a linear weighted sum of the components of f_{min} and f_{max} such that

$$J = \sum_{n=1}^{6} w_{min} \cdot f_{min} - \sum_{n=1}^{6} w_{max} \cdot f_{max}. \tag{3.10}$$

For this study, based on our desired flight characteristics, we chose the weights in this expression as

$$\begin{aligned} w &= [w_x, w_y, w_z, w_{roll}, w_{pitch}, w_{yaw}], \\ w_{min} &= [10, 12, 40, 2, 2, 5], \\ w_{max} &= [20, 12, 0, 2, 2, 5]. \end{aligned} \tag{3.11}$$

The resulting frame is intended for use as an indoor imaging platform, and therefore the emphasis is placed on the hover force in w_{min}, and on forward flight in w_{max}. For the three angular directions, it is desired to be able to turn about the yaw axis quickly, so a higher weighting is placed on the yaw torque than the roll and pitch torques.

3.3 Optimization Approach

Presented so far is a scalar performance metric for evaluating the performance of a given set of rotor orientations, given the specification and layout of the rotors themselves. We now constrain the search space on these orientations as appropriate, in order to search quickly for the optimum set of orientations.

3.3.1 Search Space

For a given rotor specification and layout, there are twelve possible degrees of freedom to orient the six rotors. For this case study, the frame restricted to be symmetric from left to right; more specifically, the rotor positions and orientations of rotors 1, 2, and 3 are mirrored across the XZ plane to define rotors 4, 5 and 6. This reduces the search space to 6 variables.

After running several optimizations, we discovered that the sum of forward-facing X components of all rotors equaled 0 at the global minimum. The same is automatically true in the Y direction, due to the mirror constraint mentioned earlier,

$$0 = \sum_{n=1}^{6} v_y^i, \quad 0 = \sum_{n=1}^{6} v_x^i. \tag{3.12}$$

This property can be exploited to reduce the search space further, to only 5 variables, by imposing these sums as constraints. The search vector p can thus be defined as the following orientation components of rotors 1 through 3.

$$p = [v_x^1, v_y^1, v_x^2, v_y^2, v_y^3] \tag{3.13}$$

From this reduced search space, we can still populate the other rotor orientation vector components

by imposing the mirror constraint and the sums in (3.12),

$$\begin{gathered} v_x^6 = v_x^1,\ \ v_x^5 = v_x^2,\ \ v_x^4 = v_x^3,\ \ v_x^3 = -(v_x^1 + v_x^2) \\ v_y^6 = -v_y^1,\ \ v_y^5 = -v_y^2,\ \ v_y^4 = -v_y^3, \end{gathered} \tag{3.14}$$

3.3.2 Global Minimum Computation

Due to the nonlinear saturation process in constructing the mixing matrix, the search space is littered with local minima and often fails to converge for unreasonable starting points. Since the computation is not intensive, the global minimum can easily be found by starting Matlab's fminsearch function at evenly spaced starting points across the search space. Since the unit vector components are derived from p, the range of possible starting points must be carefully selected by constraining the five components of p as follows to avoid rotor orientation vectors with length > 1.

$$1 > p_1^2 + p_2^2, \quad 1 > p_3^2 + p_4^2, \quad 1 > p_5^2 + (p_1 + p_3)^2. \tag{3.15}$$

Across 10,000 evenly spaced starting points in the range defined by (3.15), fminsearch arrives at the same global minimum roughly ten times. Since the majority of these starting points are nonsense, the solving process can be sped up further by constraining the starting points to a range near an initial reasonable layout. This will accelerate the user-interactive process of refining the objective function weights or changing frame parameters for a given use case.

For the case study example, starting from 500 evenly spaced starting points in the range ± 0.15 from a known good p, produces the same global minimum 140 times. Based on this result, for quick solves while changing frame parameters, it is recommended to run over 500 starting points which takes less than 15 seconds on an Intel i7-3820 quad-core processor. The full search over 10,000 points takes less than 5 minutes when taking advantage of multithreading.

3.4 Qualitative Observations

Many commercial and home-built multirotors tilt and twist the rotor directions relative to the frame in an attempt to intuitively improve performance in roll, pitch, and yaw. With the model outlined here, it is possible to quantitatively compare designs. Furthermore, it becomes clear what the qualitative effects are on rotor angles by changing frame parameters.

3.4.1 Inward Tilt

A trend in the multirotor market over the past few years has been to tilt the rotors inwards toward the center of the frame. This rotates the force vector of each rotor in such a way as to move the moment created by roll and pitch control inputs closer to the center of mass. This applies only in the traditional layout where the center of mass is below the rotor plane such as the DJI S1000 [dji18]. Additionally this helps keep the propellers out of view of the camera during aggressive maneuvers.

By removing all weight on control authority in the X and Y axis, the optimization method here can also be used to find an inward tilt for traditional multirotor frames which gives a balance of hover efficiency and roll/pitch authority.

However, when X and Y authority is desired for a 6DOF control system, the opposite tilt is observed in the optimization results. For example, Fig.3.5 and Fig.3.6 display the optimized angles for the case study where all weighting in the X and Y axis have been removed and the center of mass location has been moved to 1m above and 1m below the rotor plane, respectively. This demonstrates that the location of the center of mass is indeed a driving factor for the trend of commercial multirotors to tilt the rotors inwards.

The reason that a generally outwards tilt is observed in the optimum solution when the center of mass is below the rotor plane is derived from the inversion of the force matrix F to get the mixing matrix M. Each row of M gives a set of control signals s_i to the motors which

Figure 3.4: DJI S1000 multirotor frame demonstrating low center of mass and inward tilt of rotors.

generates an orthogonal force or moment about the center of mass and not about the rotor plane. In the case where the center of mass is not in the rotor plane, a net force applied in the rotor plane would also generate a moment about the center of mass.

The inversion of F compensates for this such that the rows of M corresponding to the directions X and Y command the motors in such a way as to generate a force-moment couple about the center of the rotor plane that results in a purely orthogonal force as desired by the controller. By tilting the rotors outwards sufficiently, the rotors primarily contributing to available forces in the X and Y directions are oriented such that they also contribute to canceling the unwanted moment about the center of mass, improving the overall force available before any single motor signal saturates.

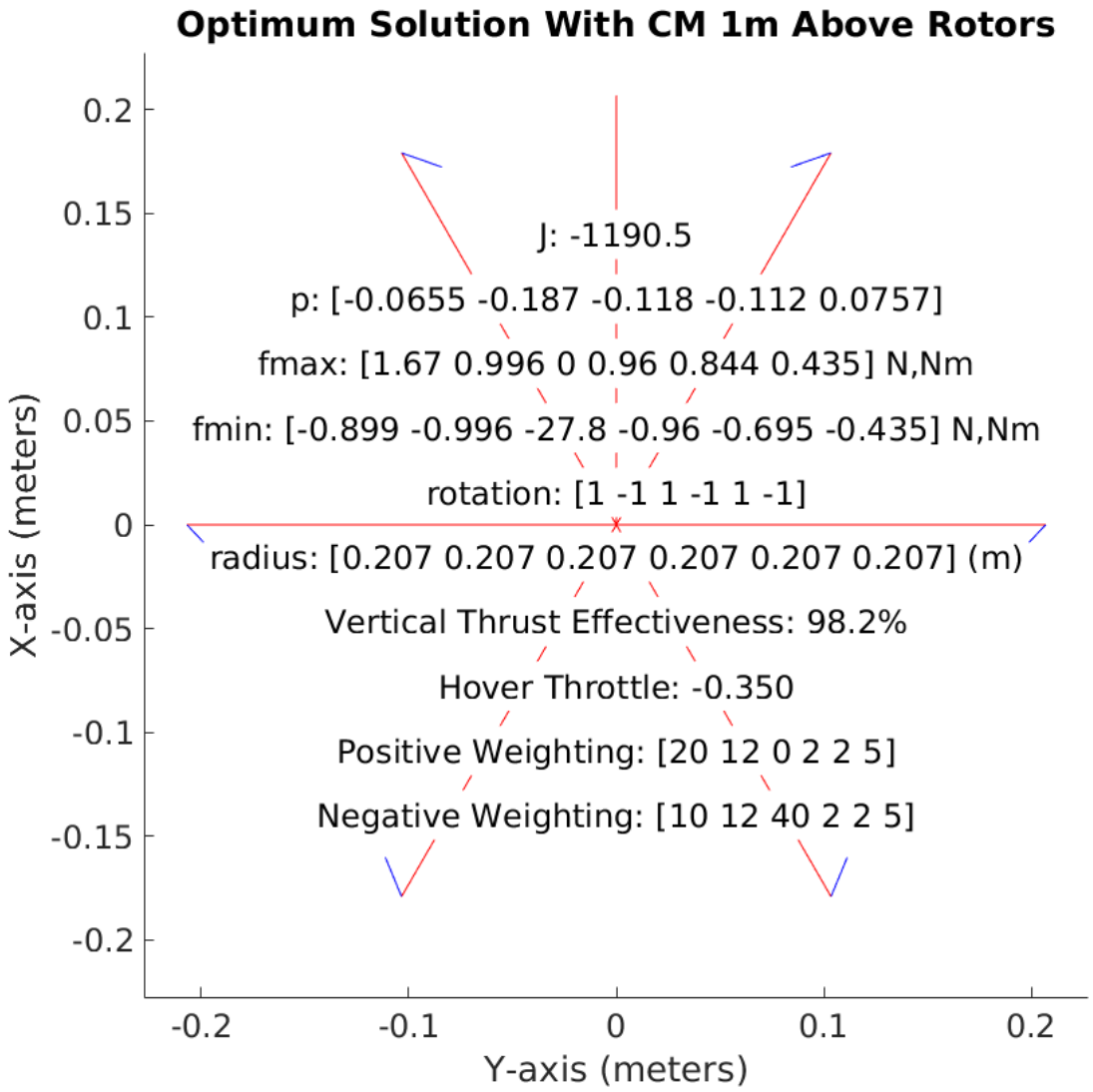


Figure 3.5: Optimum configuration for center of mass 1m above the rotor plane.

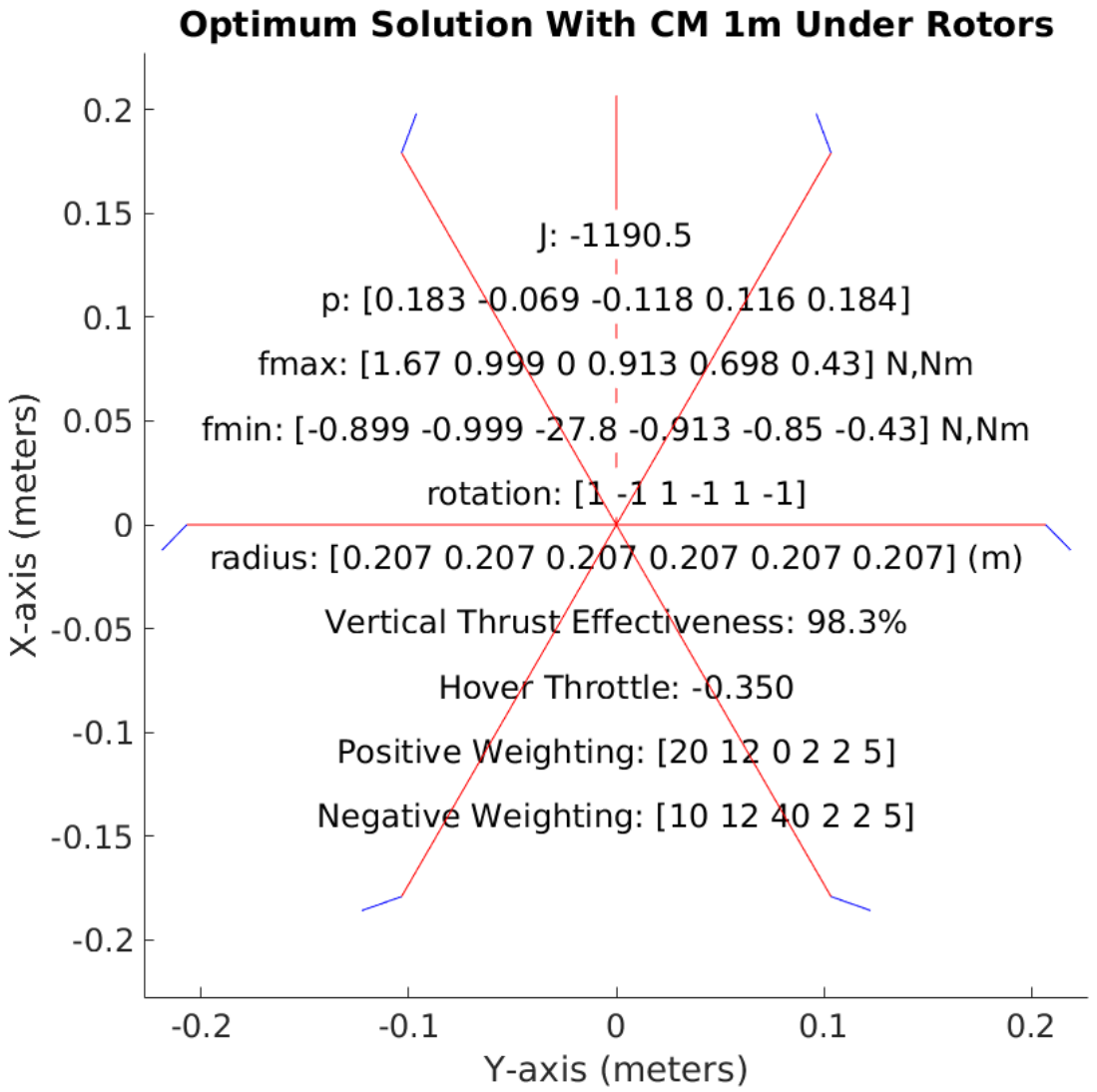


Figure 3.6: Optimum configuration for center of mass 1m below the rotor plane.

3.4.2 Isomerism in Solution

Due to the left/right mirror symmetry across the XZ plane imposed by the case study orientation constraints, there are two easily observable isomers in the solution set which are mirrors of each other across the YZ plane. The isomer of the original solution from Fig.3.2 can be seen in Fig.3.7 which displays an identical performance metric but with reversed control authority vectors. This can be generated by reversing the rotor rotation directions and flipping w_{min} and w_{max} in all directions but Z.

The available thrust in the forward X direction is the most heavily asymmetric control authority so one isomer can be reliably chosen over another by making the X direction weighting asymmetric. This would likely be done to conform to aesthetic or mechanical design constraints, or simply to favor steady forward flight over backwards flight.

The motor torque applied to the frame is quite small compared with the moments generated by the rotor thrust, but given symmetric weightings w_{min} and w_{max} a reversal of all rotor spin directions will result in an optimization that reliably favors one isomer over another. By assuming the reaction torque of the motors on the frame is zero and keeping symmetric weighting, the isomers will have identical scalar performance metrics and the optimizer will find exactly two global minima.

Knowing this, the frame designer should pick a rotor rotation arrangement that is conducive to the isomer which is most desirable to manufacture. To do this we suggest setting the weightings initially symmetric and experiment with rotor rotations to find the direction that produces the desired result. Then start adding asymmetric weightings if one direction is more heavily desired for the design objectives.

Other rotation arrangements are possible as motor thrusts dominate the frame dynamics and we do not rely on motor torque for yaw control authority as is the case with planar multirotor frames. However, through our experimenting we have not found an arrangement that offers better performance than the traditional alternating pattern.

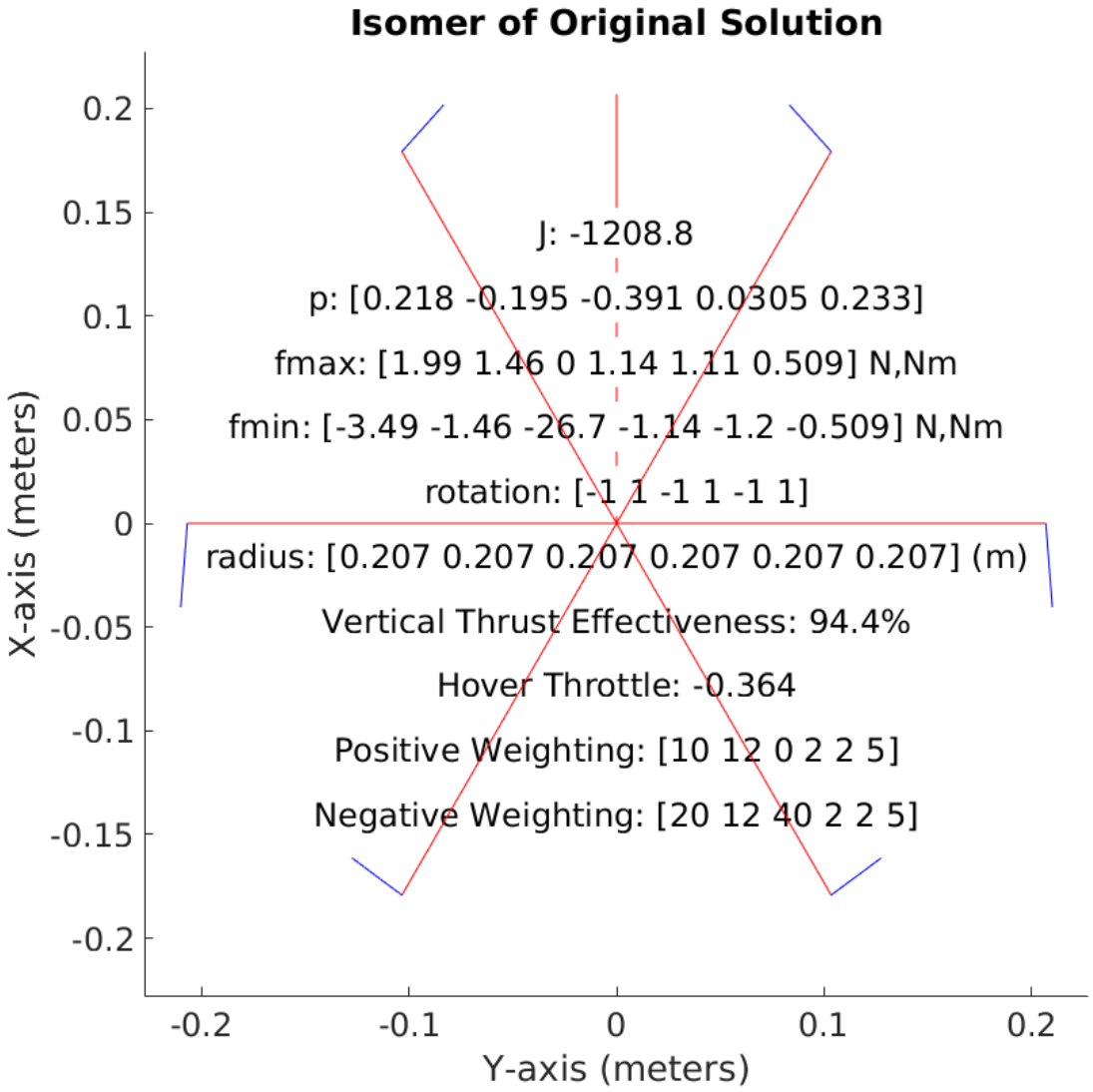


Figure 3.7: Isomer of the previously presented optimum solution with reversed rotor rotations.

3.4.3 Comparison with Related Work

A design similar to that considered here, called the LVL1 hexacopter, was offered by CyPhy Works. Based on a concept CAD rendering from their website, we derived the rotor orientations of their 6DOF frame and evaluated the likely performance with our theoretical model. To get a performance comparison, we apply their rotor orientations to the case study frame dimensions and motor parameters which are similar to that of the LVL1 hexacopter. The layout can be seen in Fig.3.8 and is compared against our optimized solution from Fig.3.2.

Examining the performance characteristics in table 3.1 it can be seen that the optimized solution has greater maximum control authority than the CyPhy Layout in every direction, with a better efficiency in hover.

Table 3.1: Available Forces With Optimum and CyPhy Layouts

	Optimum	CyPhy
X(N)	3.49	2.29
-X(N)	-1.99	-1.32
Y(N)	1.45	1.39
-Z(N)	-26.7	-25.9
roll(Nm)	1.14	0.69
pitch(Nm)	1.20	0.71
-pitch(Nm)	1.11	-1.01
yaw(Nm)	0.51	0.40
Vertical Thrust Effectiveness	94.4%	91.6%

The performance discrepancy can be explained by two main factors. Firstly, the rotors tilt inwards a large amount which was likely borrowed from other multirotor designs which have been optimized for pitch and roll control. This also hurts the available upward thrust and the vertical thrust effectiveness significantly. Here the vertical thrust effectiveness is defined as the sum of control inputs required for steady hover divided by that which would be required if all rotors pointed straight upwards. Only a frame with all rotors pointing straight up would achieve 100% vertical thrust effectiveness by that metric.

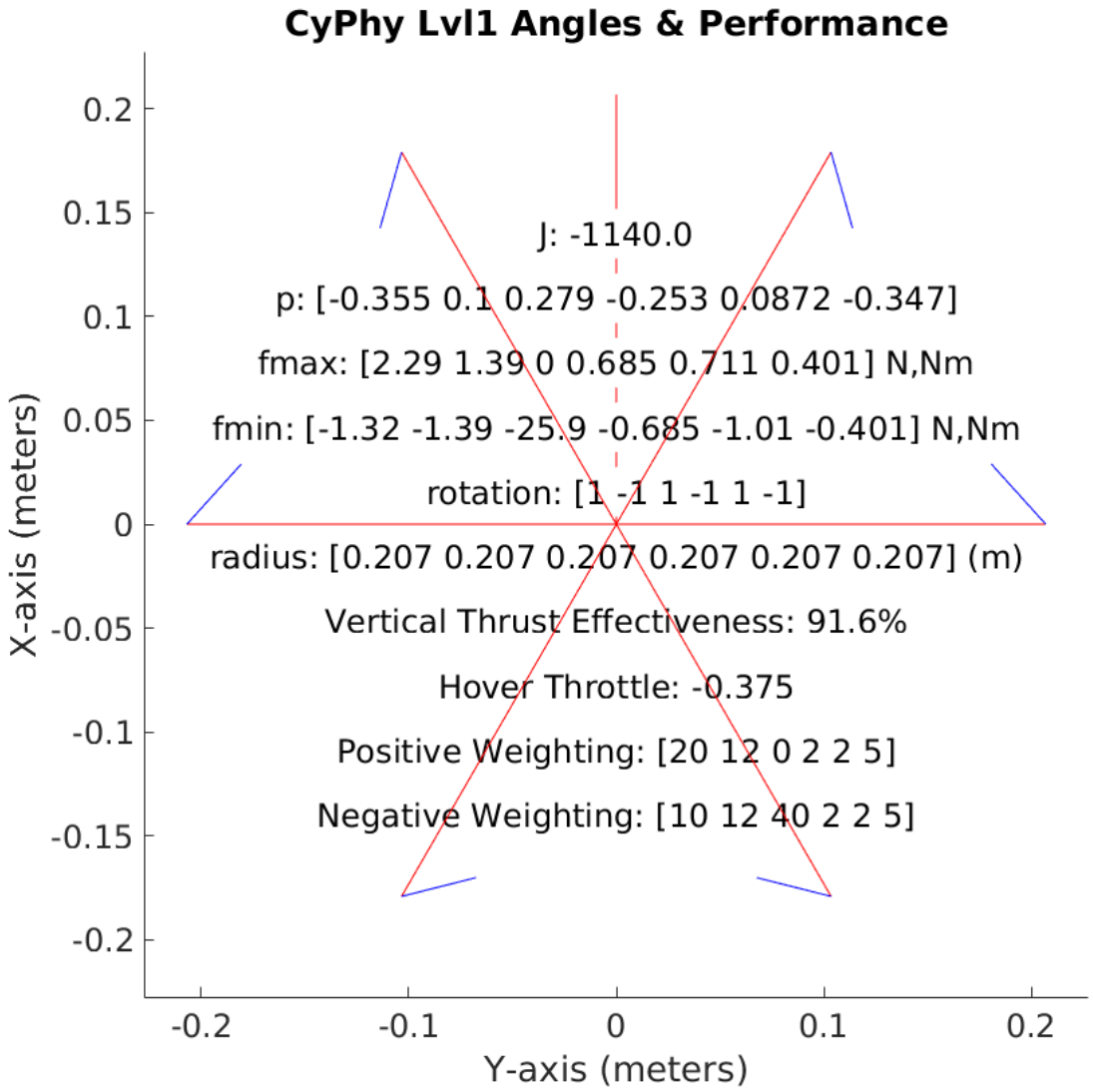


Figure 3.8: Rotor orientation and corresponding performance characteristics of the CyPhy LVL1 drone rotor orientations applied to our case study frame dimensions.

Secondly, by converting the global direction vectors of the angles to local coordinates for the end of each arm, it can be seen that each rotor twists inwards 14.5 degrees and twists about its arms 15.5 degrees. This seems to indicate that the angles were chosen by intuitively tuning 2 parameters instead of searching over the full range of possible rotor orientations as described here.

Finally note that the designers chose the basic layout with the left and right rotors pointing forwards, which results in the isomer that is capable of more thrust in the forward X direction than backward. We also favor this isomer as we assume the hexacopter will generally be in forward flight and it keeps the propellers further from a forward-facing camera's field of view.

3.5 Conclusions

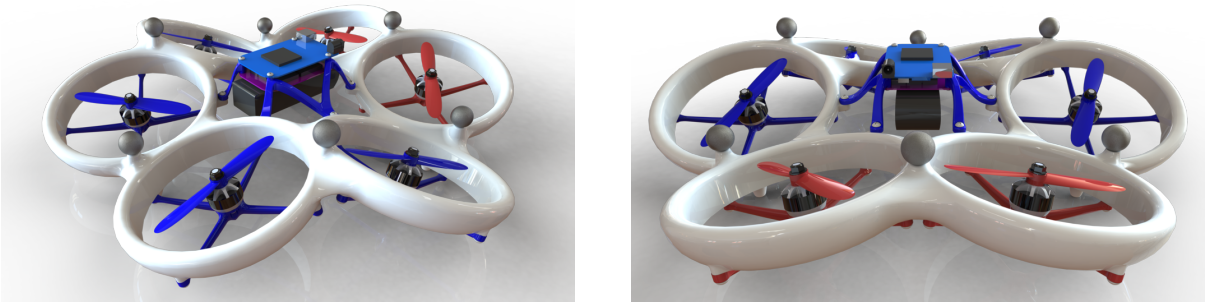
This chapter demonstrates a method for evaluating and optimizing the performance of a multirotor with direct control authority in all six degrees of freedom. The performance advantage of this control strategy is shown and compared against existing technologies. Furthermore, the qualitative and quantitative effects of multirotor airframe design on optimum rotor orientations and real-world performance are illustrated in theory and in practice.

3.6 Acknowledgements

Chapter 3, in part, has been submitted for publication of the material titled "Rotor Orientation Optimization for Direct 6 Degree of Freedom Control of Multirotors" as it may appear in IEEE Transactions on Robotics 2019. Strawson; James, Bewley; Thomas, Kuester; Falko. The dissertation author was the primary investigator and author of this material.

Chapter 4

Mechanical Design and Analysis



(a) (b)

Figure 4.1: 3D model of hexacopter design with and overall frame diameter of 45cm and 12.7cm diameter propellers.

4.1 General Design

The primary design goal is to create a small and configurable sensor platform for agile flight in confined spaces such as caves, tunnels, and collapsed buildings. However, the design must be parametric to allow scaling for significantly larger or smaller lift capacities. It must also allow rapid adaptation for new sensors and electronics payloads.

The vast majority of commercial multirotor designs use a central hub with radial arms

Figure 4.2: Various 3D printed EduMAV prototypes.

supporting each motor, even in the case where the propellers are ducted or shrouded [HSA+15]. The authors propose, instead, a ducted multirotor where the ducting itself forms the primary load-bearing structure. This requires careful material selection and modal analysis to be viable, but results in a highly performant and elegant design with very low part count. The resulting protective ductwork is extremely resilient to impacts resulting from its material selection and it being the primary structural component as opposed to an afterthought in the design.

4.1.1 Monocoque Approach and Material Selection

The monocoque airframe is a hollow nylon shell that forms the shape of six propeller ducts in a hexagonal arrangement as shown in the final design in Fig.4.1. In addition to supporting the motors, the shell acts as an impact-resilient crash structure to protect the payload and allow

for collisions during flight in confined environments.

It is constructed on a large-format FDM printer [3DP] out of Taulman Nylon 910 due to its fantastic impact resiliency [Tau], safety when printing indoors [KLR+01]. 3D printing the monocoque shell allows a completely hollow structure while still being a single piece. With injection molding this would require at least two separate parts and roto-molding would require a prohibitively complex arrangement of molds for this particular shape. Most importantly, additive manufacturing allows for very complex curved surfaces which is necessary due to the to arrangement of rotor angles optimally for 6DOF fight.

The shell can be seen in various prototype forms in Fig.4.2. These are all constructed on a 3DP-X1000 series FDM printer by 3D Platform which was chosen for its ability to print up to 1m x 1m x 50cm parts at equivalent resolutions to desktop 3D printers [3DP].

4.1.2 Monolithic vs Modular Design Tradeoffs

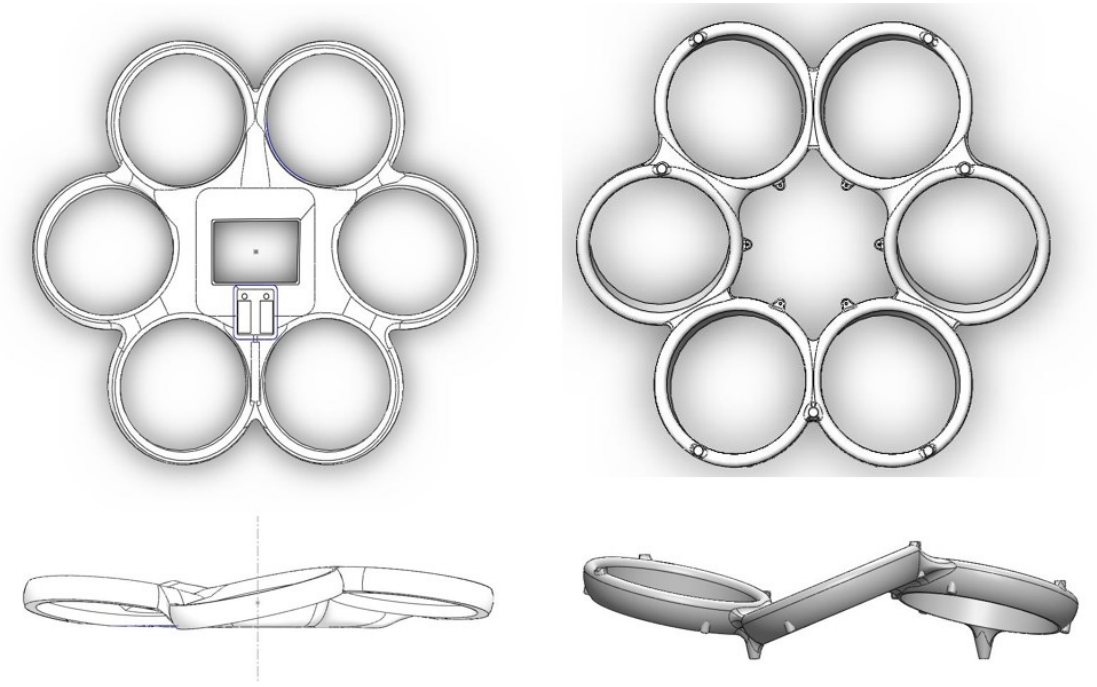


Figure 4.3: Comparison of earlier monocoque frame design (left) incorporating sealed electronics mounting compartment, compared with the minimalistic final design (right).

A major design benefit of a monocoque frame is the ability to include multiple design features such as component mounting points into a single part, reducing the overall part count and number of fasteners required. However, this often comes at the expense of mass and design flexibility. This trade off was reached early in the design process as it became evident that incorporating a sealed electronics enclosure into the monocoque frame increased the surface complexity to the point that the parametric design would no longer scale easily with propeller size, nor allow rapid design revisions supporting different electronics and sensor packages. The earlier design can be seen side-by-side with the final design in Fig. 4.3.

4.1.3 Motion Capture Marker Placement

A simple and scalable design feature that remained incorporated into the monocoque shell is the inclusion of mounting points for retroreflextive markers as visible in Fig.4.1. These markers are necessary for tracking the position and orientation of the airframe in a motion capture system for tuning and development of the flight controller. In the interest of allowing multiple unique platforms to fly simultaneously in a motion capture system for swarm research, the frame was designed with six irregularly positioned marker mounting positions such that any 3 markers define a unique geometry that can be identified by the motion capture software [Lim] [VIC].

$$p = \frac{n!}{n_1!n_2!} = \frac{n!}{m!(n-m)!} \tag{4.1}$$

The theorem of permutations with repetition provides the means to calculate the number of possible marker arrangements on the same frame. For n total possible objects with n_1 identical objects of type 1 and n_2 identical objects of type 2, this theorem simplifies to the first expression of Equ.4.1. This can be simplified to state that the number of permutations p of n objects with n_1 identical objects of type 1 and n_2 identical objects of type 2 is given by equ.4.1. Letting n be the number of mounting locations, m be the number of markers, and types 1 & 2 be the presence or

absence of a marker in a specific mounting location, the number of possible unique permutations is given by the second expression of Equ.4.1.

The minimum requirement of three markers for the motion capture system allows 20 permutations. However, using four markers allows for more robust object detection and still allows up to 15 multirotors to be uniquely identified, in addition to the six possible arrangements using five markers and the single case where all six locations are populated [Lim].

4.1.4 Modular Mounting System and Custom Electronics

Since sensors and payloads vary so greatly in size and dimension, custom brackets must be 3D printed for each payload. Thanks to the ubiquity of home desktop printers this is no longer problematic. Six mounting points are provided on the monocoque shell that are also used to mount the primary electronics control module and can be shared with any desired payload module design. The CAD renderings in Fig.4.4 show an example payload module with and Intel Aero compute board and RealSense RGB-D depth sensing camera for low-light navigation in GPS-denied environments.

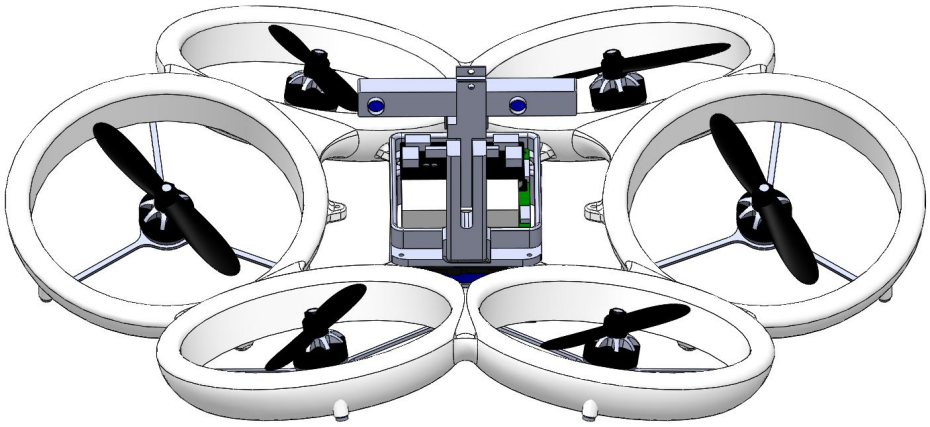


Figure 4.4: Model with Intel Aero compute board and RealSense RGB-D camera payload module mounted.

The primary control module as pictured in Fig. 4.1 consists of two custom PCBs, and a 2250mAh 4-cell Lithium Polymer battery. The control board is a BeagleBone Blue which houses a 1Ghz ARM processor, Wifi/Bluetooth module, and all sensors required for flight control. This hardware was developed as an open-source collaboration between the authors and the BeagleBoard.org foundation. It provides the option of two flight controllers: Ardupilot and the RC_Pilot flight controller [Str18c]. The second (lower) PCB is a custom 6-channel brushless motor driver board designed by the authors for this project and is available online [Str18a].

4.2 Motor and Propeller Testing

The motors and propellers used in a multirotor are perhaps the most critical components to ensuring efficient and reliable flight. This testing sought to find the optimal combination of commercially available components suited for this airframe. The overall desired size of the airframe restricts the maximum propeller diameter to the common and widely-available five-inch standard, from which eight different propellers and twelve suitably-sized brushless motors were selected for testing.

Table 4.1: Motors and Propellers Tested

Motors	**Propellers**
EMAX RS2205S 2600kv	Cyclone T5040C 3blade
Lumenier 1806-13 2500kv	Cyclone T5045C 3blade
Lumenier MX2206-9 2450kv	Gemfan 5x4.5 3blade
Lumenier RX2204-14 2300kv	Gemfan 5x4 2blade
Lumenier RX2205-12 2400kv	HQProp 5x4 4blade
Lumenier RX2206-11 2350kv	Lumenier 5x4.5 2blade
Lumenier RX2206-13 2000kv	Lumenier 5x4 3blade
Tiger Motor 1804-20 2400kv	Lumenier 5x5.3 3blade
Tiger Motor F40 III 2400kv	
Tiger Motor F40 Pro 2400kv	
Tiger Motor F30 2800kv	
Tiger Motor F30 2300kv	

4.2.1 Experiment

Each motor and propeller combination was attached to an RC Benchmark 1520 propeller thrust stand [inc] affixed with a duct and motor mount mimicking those of the final monocoque frame pictured in Fig.4.5. Each combination was then spun up to ten different throttle positions equally spaced between zero and 100% throttle where five samples were taken of current draw, rotational speed, and thrust over two seconds before being averaged and recorded. This process is fully automated for consistency.

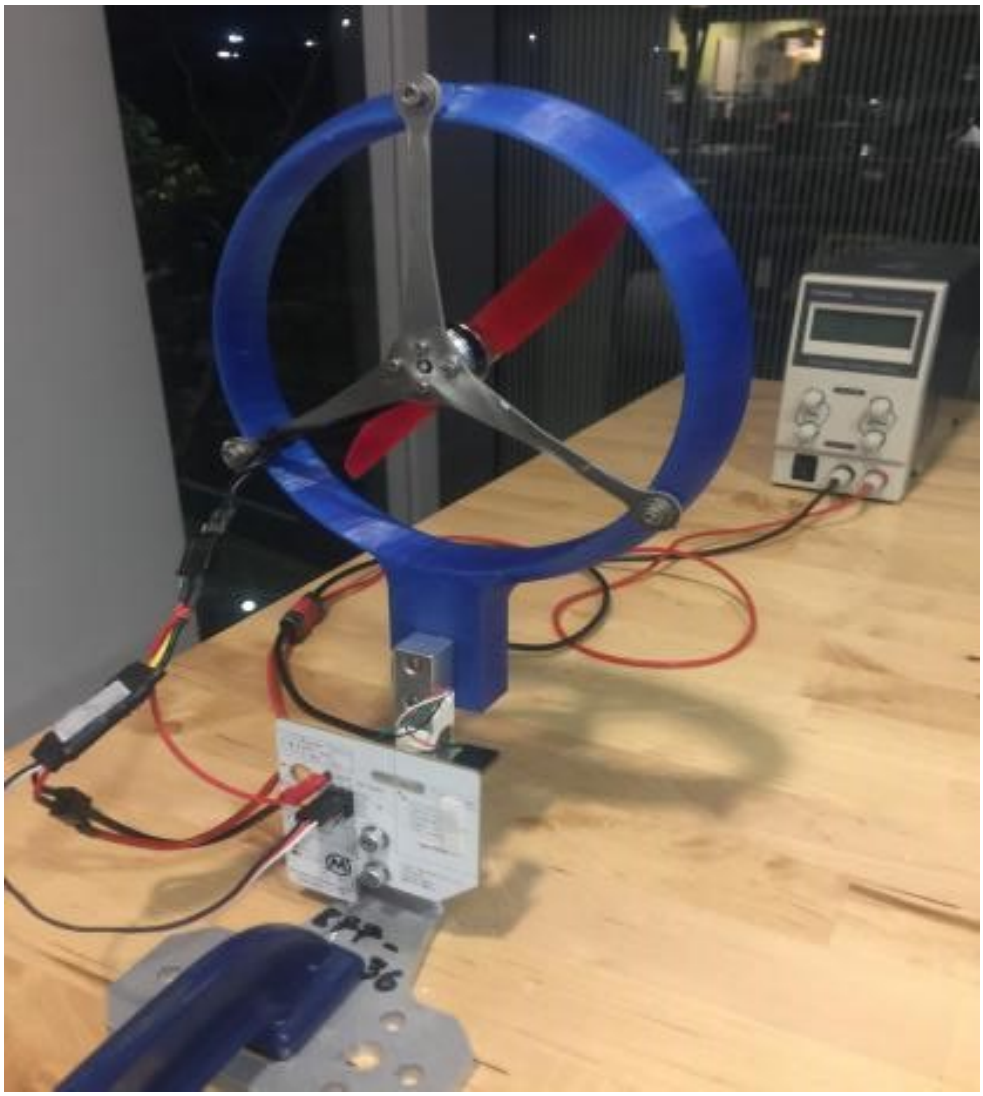
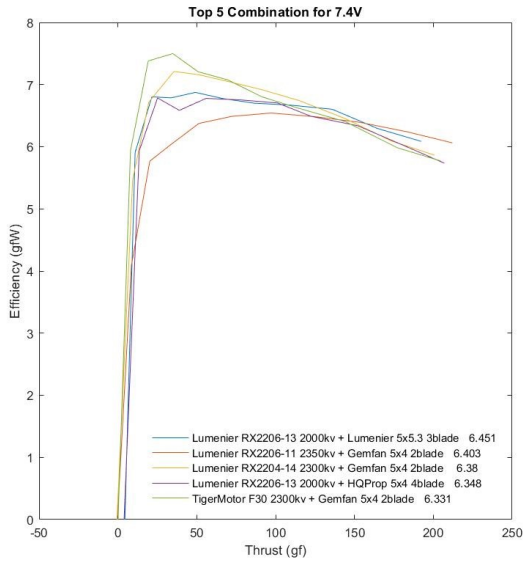
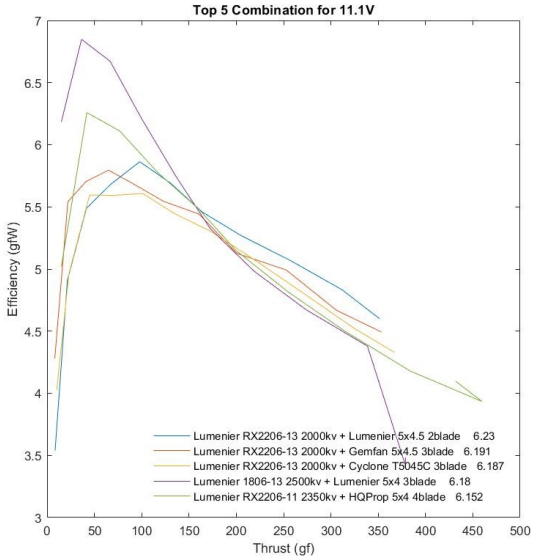


Figure 4.5: Propeller test stand used for performance evaluation



(a) Efficiency curves of 5 most efficient motor and propeller combinations at 7.4V



(b) Efficiency curves of 5 most efficient motor and propeller combinations at 11.1V

Figure 4.6: Propeller test stand and results

The data collection process was repeated at both 7.4v and 11.1v mimicking the nominal voltage for both two and three-cell Lithium Polymer battery packs. This resulted in 192 individual test runs. For safety, the motors are powered by a current-limiting 10A power supply and several experiments reached this current limit, protecting the motor controller and preventing motor overheating and potential short circuits.

4.2.2 Results

Complete results, efficiency plots, and raw experiment data for every test conducted are available at https://github.com/StrawsonDesign/motor_propeller_testing.

An early estimate of the payload and overall airframe mass suggests that 1.47N thrust per motor would be required to hover, therefore efficiency in W/N at this thrust level formed the primary objective when choosing the motor and propeller combination. Many combinations result in excessive noise, vibration, and overheating and were flagged a non-viable during testing. These behaviors tend to produce poor numeric performance and flagged combinations did not appear high on a list sorted by efficiency.

Fig.4.6a and Fig.4.6b present the efficiency curves for the top five most efficient combinations at 1.47N of thrust as calculated by interpolating between the two nearest sample points.

For this particular test and objective, the Lumenier RX2206-13 2000kv motor proved most efficient at both voltages when combined with a Lumenier 5x5.3 3-blade propeller at 7.4V and a Lumenier 5x4.5 2-blade propeller at 11.1V. While the most efficient combination for the 7.4V test was 3.6% more efficient the ideal combination of the 11.1V test, the two-blade propeller was chosen for the prototype build as the 7.4V testing was severely limited in maximum thrust and would impinge on the final system's controllability.

The Lumenier RX2206-13 2000kv motor and 5x4.5 2-blade propeller continue to perform well even when driven at 14.8V without excessive vibration, overheating, or current draw. This allows for the option to run 4-cell lithium battery packs without issue.

4.3 Modal Analysis of Frame

The overall shape of the hollow monocoque frame is determined by the duct size and profile. This leaves the thickness of the shell to be determined through careful structural modeling. Due to the overall design, the loading on the motor mounts and frame are minimal. Instead, potential interaction of the feedback controller with vibrational modes of the structure are of primary concern. The design goals include a feedback controller with a crossover frequency between 8Hz and 10Hz, it is therefore desirable to have the primary vibrational modes at least one order of magnitude faster to avoid interaction.

4.3.1 Finite Element Model and Boundary Conditions

To facilitate rapid adjustment and computational solution of the model, an initial CAD model of the monocoque shell was drawn entirely of surfaces to facilitate modeling with S4R shell elements. The FE model can then be adjusted for thickness very easily without changing the CAD model. To mimic the attachment of a heavy and rigid payload module, all 6 mounting points on the inner surface of the shell are fixed in place. Finally, six point-masses of 30g each are added to the motor mounting locations on the frame to replicate the effects of the motors on the vibrational modes of the frame. The FE model was given a homogeneous modulus of elasticity as $72,932 PSI$ as provided by the material manufacturer's specification [Tau].

A mesh sensitivity analysis is performed over several element sizes to confirm that the resulting natural frequencies are feasible in a finite element environment and to determine a mesh size that balances accuracy with compute time. The study evaluates mesh sizes ranging from 10mm to 1mm and the authors concluded that a mesh size of 3mm formulates a suitably converged result with a reasonable compute time of 12.6 seconds to solve the model.

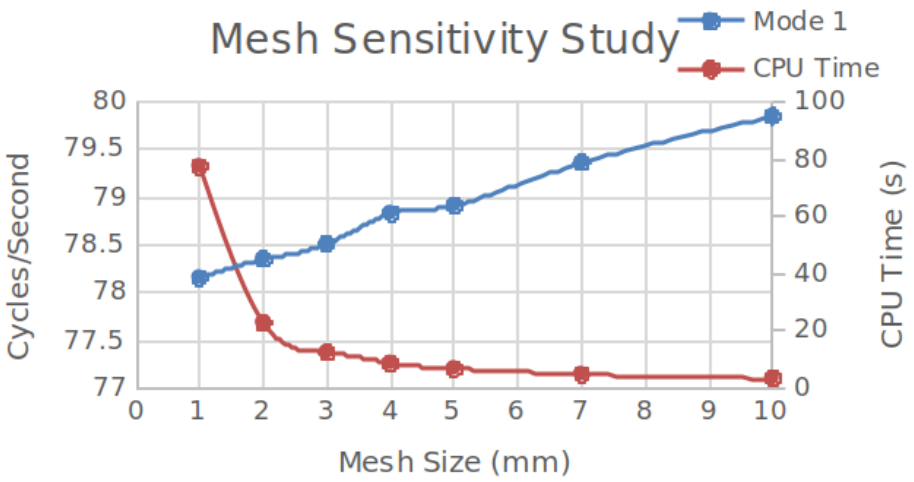
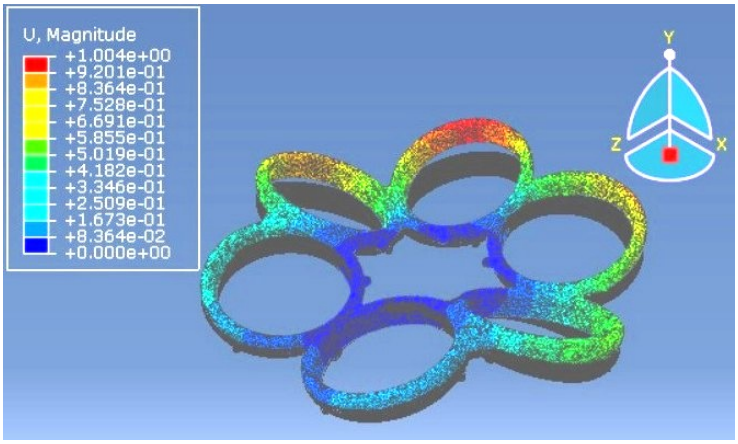


Figure 4.7: Results of mesh sensitivity study

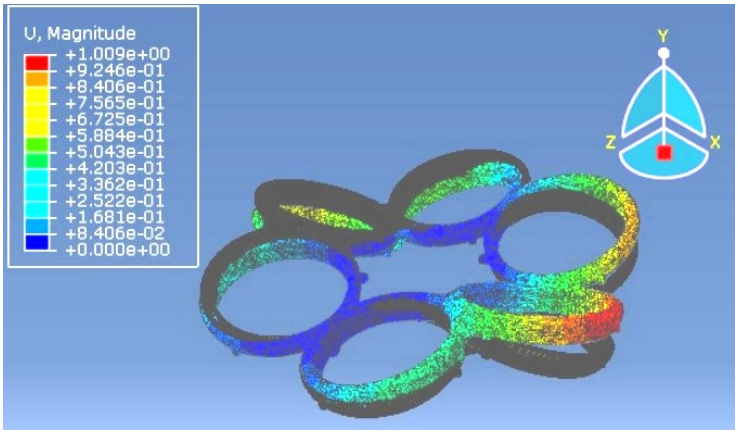
4.3.2 Results and Validation

Since the 3DP printer chosen to manufacture the frame has extrusion nozzle sizes of 0.25, 0.4, 0.6, and 0.8mm available, these are chosen as the shell thicknesses for analysis [3DP]. The 0.4mm shell has its first three vibrational modes at frequencies of 82.1, 83.4, and 97.6Hz which are right at the design goal. Displacement plots indicating the directions of flex for these three modes are presented in Fig.4.8.

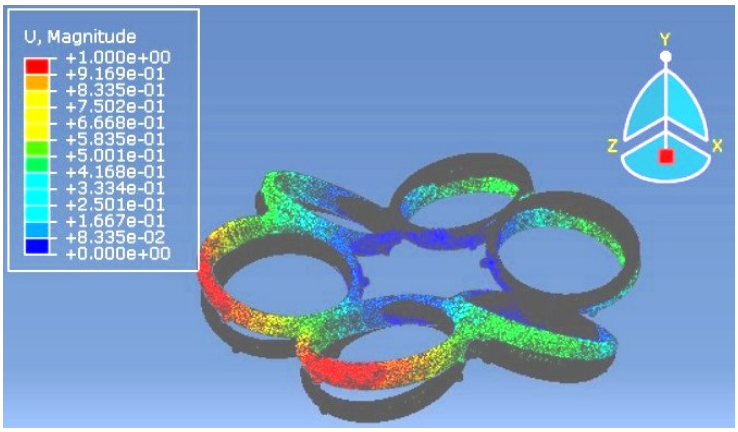
To validate the solid model, we perform a static analysis of the stiffness in the direction of deformation when the shell is under its primary (slowest) vibrational mode. We support the mounting points of the shell rigidly then use a force gauge to deform the body with 6mm of deflection where the force gauge is applied. This required 11.5N of force which is only 3.5% from the 11.9N predicted force required to deform the FE model under static point load.



(a) Mode 1 82.1Hz



(b) Mode 2 83.4Hz



(c) Mode 3 97.6Hz

Figure 4.8: Displacement plots of first 3 vibrational modes of the FE shell model

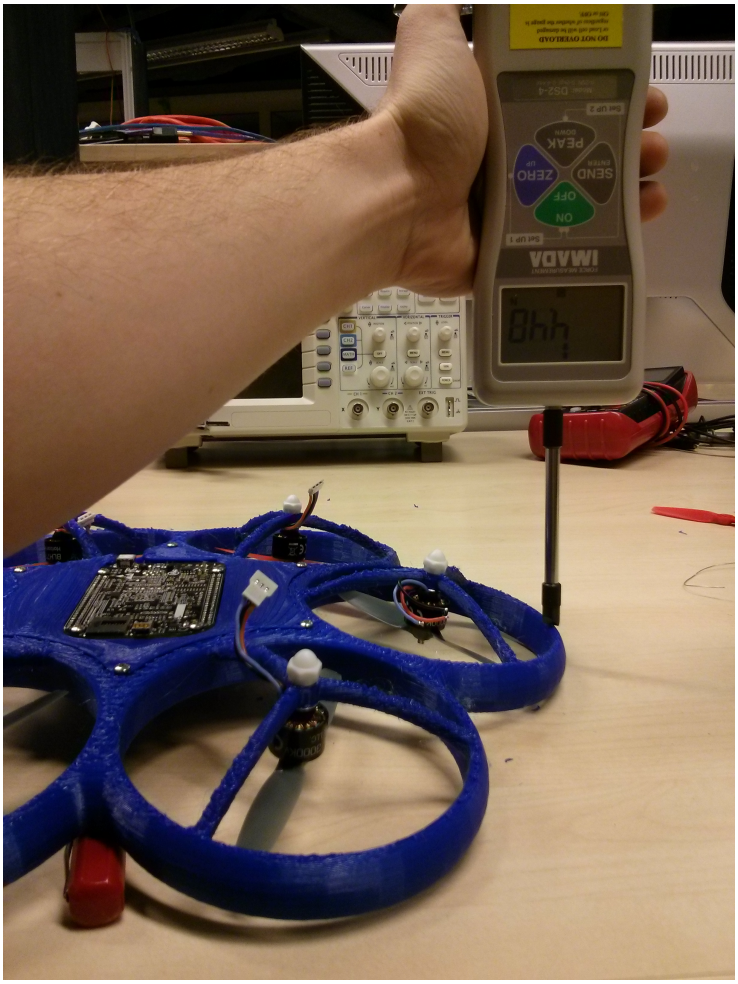


Figure 4.9: Prototype 3D printed model undergoing static stiffness validation.

4.4 Fluid Flow Analysis

Intentionally flying a toy quadcopter extremely close to a wall will usually result in it being drawn toward the wall due to Bernoulli's principle causing reduced pressure near the wall [PMK+13] [RCR14] [GSJP17] [GKPC05]. We hypothesize and verify that a side product of the tiled rotor layout would be a small stabilizing effect causing the airframe to tend away from vertical surfaces without requiring control input.

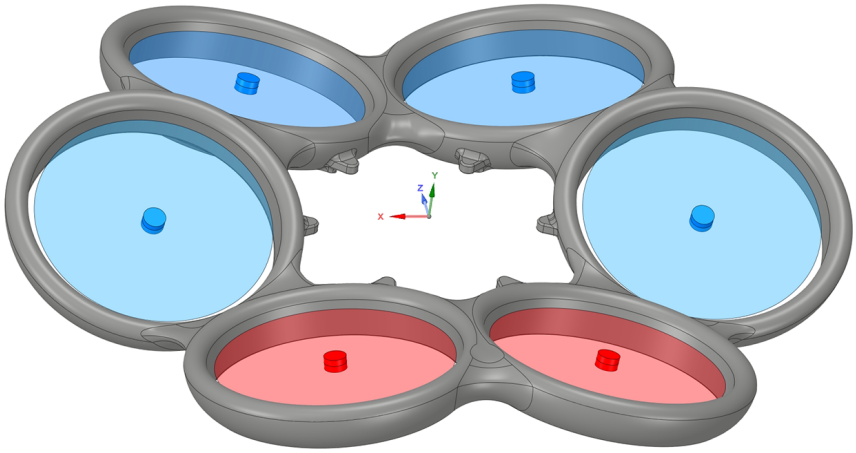


Figure 4.10: Simplified model for CFD. Red disks are at the front (nose) of the airframe. X axis points right (starboard), Y axis points upwards, and Z axis points rearwards.

Table 4.2: Force components in Newtons. X axis points right (starboard), Y axis points upwards, and Z axis points rearwards.

Test Cases	$\mathbf{F_x}$	$\mathbf{F_y}$	$\mathbf{F_z}$
Open Air	0.013	7.073	0.113
Right Side 6cm From Wall	-0.206	6.547	0.658
Nose 6cm From Wall	-0.005	7.100	0.101
6cm Above Ground	-0.168	6.668	0.178
6cm Below Ceiling	-0.014	6.258	0.071
Centered in 1mx1mx6m Tunnel	0.020	7.425	0.013

4.4.1 Testing Scenarios and Parameters

Five near-object flight scenarios and an open-air baseline scenario are simulated using ANSYS Fluent to predict the change in forces acting on the airframe in a variety of plausible conditions.

- Flight in open air
- 6cm above ground
- 6cm below a ceiling
- 6cm between nose and wall

- 6cm between right edge and wall
- Centered in a 1m x 1m x 6m tunnel

All simulations are configured to use incompressible air at $1atm$ and $300K$. The far-field boundary of all fluid domains are located (where applicable) a distance of 25 duct radii away from the airframe in all directions except downwards in which a distance of 50 duct radii was chosen, slightly larger than [YLP16] since computational time was less of a concern. In all simulations, the hexacopter is treated as a rigid body with fixed orientation and flow is evaluated at steady-state.

Each disk is modeled as an infinitely thin porous disk with the same swept area as the chosen Lumenier 5x4.5 propeller. A uniform axial velocity distribution of incompressible air over the disk is assumed in accordance with actuator disk theory [Kec12].

It should be noted that the duct diameter is not small enough to significantly improve propulsive efficiency. A force of 125gf, obtained from our test bench, was distributed over the disks to specify a uniform discontinuous constant pressure jump; this represents the motors performing at about 22% of the available 567gf of available thrust in equilibrium hover. Finally, our disk models did not impart any tangential velocities to their generated wake. By applying the actuator disk theory, we were able to obtain a steady-state solution [Kec12] for each of our test cases.

The fluid domain for each test case is created by performing boolean subtraction operation using the hexacopter's airfame and propeller hubs as tool bodies and an enclosing box as the target body. The topology of the box and actuator disks are merged to ensure the generation of a conforming mesh for continuous air flow. All fluid domains are discretized into unstructured tetrahedrons and the actuator disks are meshed using a structured quadrilateral polar array configuration.

A region enclosing 5cm above, 60cm below, and 17cm around the airframe is meshed with

five-times higher density than extremities of the domain to better capture the steeper gradients near the airframe and walls. The element count for the test cases ranged between 2.4 to 3 million.

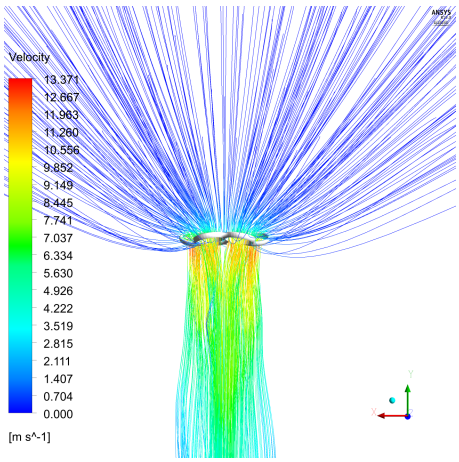
The Reynolds-Averaged Navier-Stokes turbulence model and Shear Stress Transport (SST) k-ω, are chosen to access the aerodynamic forces exerted on the hexacopter. The PREssure STaggering Option (PRESTO) pressure interpolation scheme and Monotone Upstream-Centered Schemes for Conservation Laws (MUSCL) interpolation methods are used with the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm to resolve all simulations.

4.4.2 Results and Qualitative Analysis

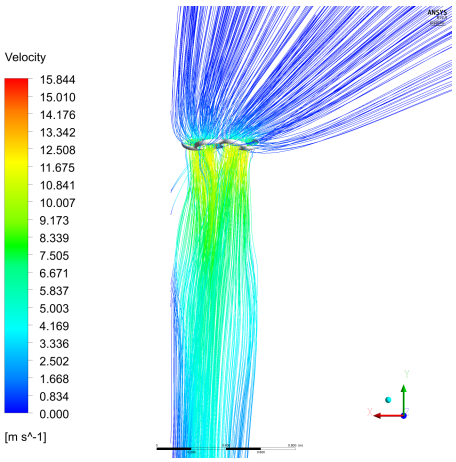
Table 4.2 presents the net forces experienced by the airframe when a fixed airflow required to support a $700g$ takeoff mass is applied to the actuator disks. Note that the frame is symmetric left-to-right but intentionally asymmetric forward-to-aft since the rotor angles are calculated with the methods presented in [SBKon]. This predictably results in a near-zero force in the right-facing X axis for the open-air baseline case, for which the fluid flow can be seen in Fig. 4.11a.

In the scenario where the airframe's right side is against a wall, a beneficial additional stabilizing force of $0.2N$ away from the wall is predicted by the model. This partly confirms the hypothesis for this scenario, the streamlines for which are depicted in Fig. 4.11b. The simulation scenario with the airframe's nose against a wall does provide $0.1N$ stabilizing force away from the wall, however this is roughly the same forced experienced in the open-air baseline. As such, any trimming of the airframe attitude by the feedback controller would negate this. Note that this near-zero change in force is still vastly favorable to being pulled towards the wall.

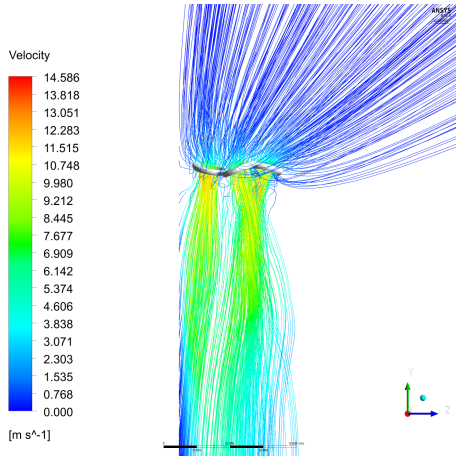
When the airframe is $6cm$ above ground, as depicted in Fig. 4.11d, there is significant air recirculation and irregular flow due to in ground effect (IGE) causing a decrease in lift. Hovering this close to the ground is extremely unstable as explored in [GSJP17] and [GKPC05] and provides additional justification for the higher frequency response afforded by the fully actuated 6DOF control scheme as presented in [SBKon].



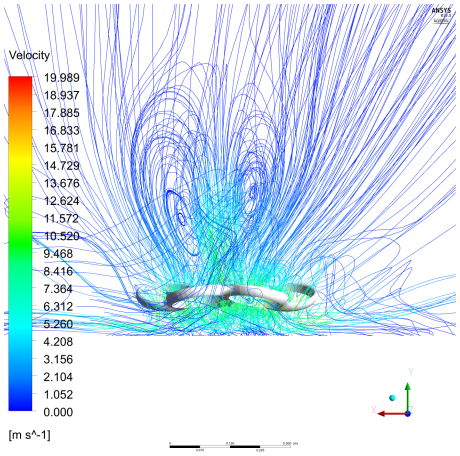
(a) Open Air



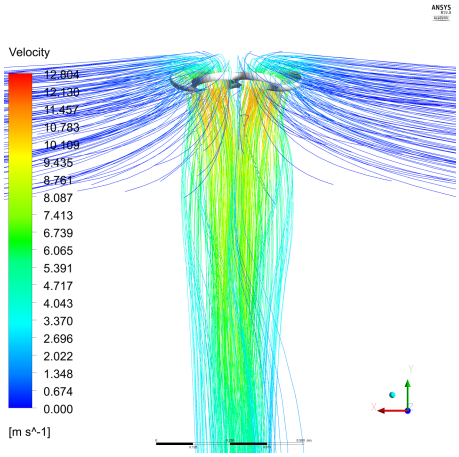
(b) Right Side 6cm From Wall



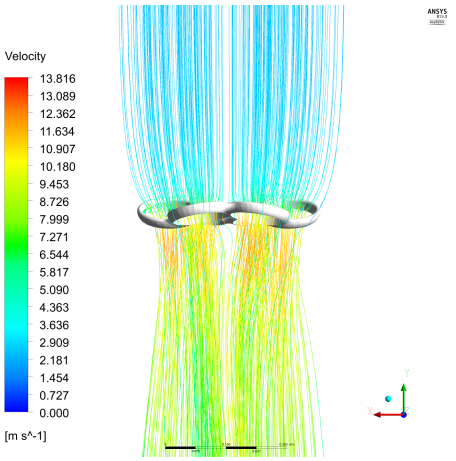
(c) Nose 6cm From Wall



(d) 6cm Above Ground



(e) 6cm Below Ceiling



(f) Centered in 1m x 1m x 6m Tunnel

Figure 4.11: Streamlines for Each CFD Test Scenario

4.5 Conclusions

This chapter demonstrates the overarching design and analysis of a monocoque frame hexacopter focused on utilizing the rotor angle optimization method presented in chapter 3. It demonstrates that highly complex surface bodies such as a monocoque frame lend themselves well to rapid design revisions utilizing shell FE analysis and that this 3D printed structure performs similarly to its FE stiffness model. Finally, it demonstrates that a tilted rotor layout can provide beneficial stabilizing fluid interactions in some confined flight scenarios.

4.6 Acknowledgements

I would like to thank Yifan Xu, Meng-Fu Chiang, Mingchen Mao, Pengcheng Cao, and He Huang for their dedicated work conducting the motor and propeller testing. I would also like to thank Danny Tran for his many late nights making Abaqus behave and helping me analyze the fluid flow characteristics of the design.

Chapter 4, in part, has been submitted for publication of the material titled "Multirotor Airframe Design with Rotor Orientations Optimized for Fully Actuated Feedback Control" as it may appear in IEEE Transactions on Robotics 2019. Strawson, James; Tran, Danny; Cao, Pengcheng; Bewley, Thomas; Kuester, Falko. The dissertation author was the primary investigator and author of this material.

Chapter 5

Control and Performance Evaluation

In this final chapter, we look at the control architecture of the rc_pilot flight control software and how it demonstrates proper usage and available functionality of the librobotcontrol ecosystem from Chapter 2. We also experimentally measure the performance of the 6DOF control authority optimized and predicted in Chapter 3.

5.1 Control Software Architecture

Like EduRover and EduMiP, we provide a reference control program to demonstrate functionality of EduMAV which is purely written in C using the librobotcontrol library. It is designed primarily around 6DOF control utilizing the optimized rotor layout of EduMAV, but also supports standard Quad, Hex, and Octocopter arrangements.

To coincide with the miltithreading techniques presented in the associated course materials, this reference program separates major functions into their own threads separated by priority. There are six threads, each highlighted in green in Figure 5.1. The first and primary thread launched on program creation executes main() and only serves to initialize hardware and software subsystems and to spawn other system threads before waiting for the shutdown signal.

The Graphical Visualization and Debugging Thread is the only thread allowed to print to

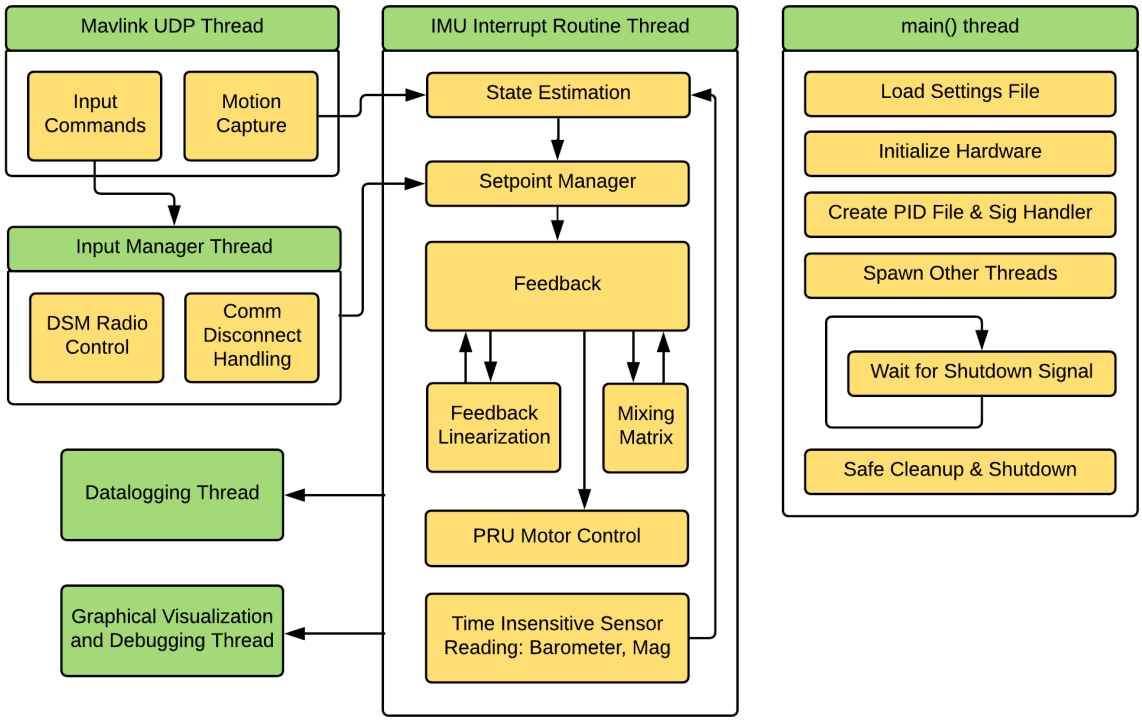


Figure 5.1: Modules of the RC Pilot flight controller

stdout and provides a terminal interface that prints any variables or states that have been requested in the settings file. This lightweight approach provides the flexibility to debug new or existing code that simply cannot be afforded with heavy ground control stations such as qgroundcontrol which are limited to displaying fixed mavlink telemetry packets. By limiting this thread to writing to stdout, graphical consistency is preserved and relatively heavy print functions are removed from more time-critical threads.

While debugging through the Mavlink prototcol is discouraged, a thread is dedicated to interpreting and sending Mavlink packets through UDP utilizing either the ethernet or WiFi capabilities of the BeagleBone Platform. This thread is entirely internal to the librobotcontrol library and is therefore easy to use in other student projects utilizing this ecosystem, reducing the barrier to entry for simple tasks such as telemetry. In the rc_pilot program, Mavlink is specifically

used for broadcasting Heartbeat and health information such as battery and flight status. Perhaps more importantly for research, it also listens to standard Mavlink motion-capture packets which are broadcast by an accompanying Vicon Motion Capture system interface program we wrote to inform robots of their location and orientation at up to 100hz. This provides the vitally accurate position information for feedback loop tuning and position-hold control. The Vicon system can be seen in the background of Figure 5.3 portarying EduMAV in flight.

The system also accepts Mavlink packets for remote control which are interpreted alongside Spektrum DSM packets in the input manager thread. This thread is also managed entirely by the libRobotControl library as part of its DSM module which is also used by the EduMiP and EduRover reference programs, emphasizing portability and reliability of the ecosystem software.

A datalogging thread utilizes a dual-buffer system that flushes log data to disk at a user-defined rate around 1Hz to minimize file IO. Data is copied into the buffers every loop by the feedback controller with a direct memory copy to keep the logging thread utilizing as few resources as possible.

5.1.1 Feedback Thread Components

Finally, and most critically, the IMU Interrupt Routine thread houses the sensor readings, state estimation, and feedback control. This is the same interrupt routine thread framework utilized by EduMiP and EduRover and provided by librobotcontrol. Critically, its timing is based off of a GPIO interrupt provided by the sensor itself after each self-sample of the gyroscope and accelerometer. Latency of these two sensors is critical for position and attitude estimation so by letting the sensor sample itself on a deterministic clock and then inform the controller when new data is ready, the time between sample and feedback is reduced as much as possible. This architecture is also very forgiving of hiccups in the interrupt response time due to the operating system's parallel processes as the sensor samples themselves remain deterministic.

Less critical sensors such as the magnetometer and barometer are read after the critical

feedback loop and at rates slower than the accelerometer and gyro as they govern only the low frequency components of attitude and position estimation. This also helps ensure hiccups in these sensor readings do not induce any more variation in the rate of the state estimator or feedback controller.

Another optimization step is the implementation of the PPS (Pulse Period Signaling) ESC signal generation on the BeagleBone's PRU or Programmable Realtime Unit. This implementation allows the main feedback thread to instantly tell the PRU to start a motor command and while the PRU is still issuing the motor commands the feedback thread can move onto sampling other sensors. This setup supports both traditional 1000-2000 microsecond pulse signaling as well as the newer latency-reducing OneShot 125 protocol and is also built into the libRobotControl library for use in other projects.

5.1.2 Feedback Linearization

Part of the Feedback Control module requires mapping linear SISO control outputs in each of the 6 degrees of freedom to the motors. This requires both a mixing matrix, as described in Chapter 3, and a static model of the motor thrust behavior.

The linear model and mixing matrix M assume that the thrust from each motor is proportional to the control inputs. This is not the case with hobby-grade brushless motor and propeller combinations. To account for this it is necessary to linearize the output by correcting for the motor thrust curve. We do this by taking 10 evenly spaced thrust measurements to generate a thrust curve for that specific motor and propeller combination. Nominal battery voltage during flight is 14.8V for this case study. The resulting curve is shown in Fig.5.2.

Within the flight controller we map a desired thrust to an actual signal s_i that will be sent to the brushless motor controller by linearly interpolating between the two nearest points of our experiment. This is a computationally inexpensive process and keeps the system behaving as linearly as possible. Multiple maps are included in the rc_pilot source code as options depending

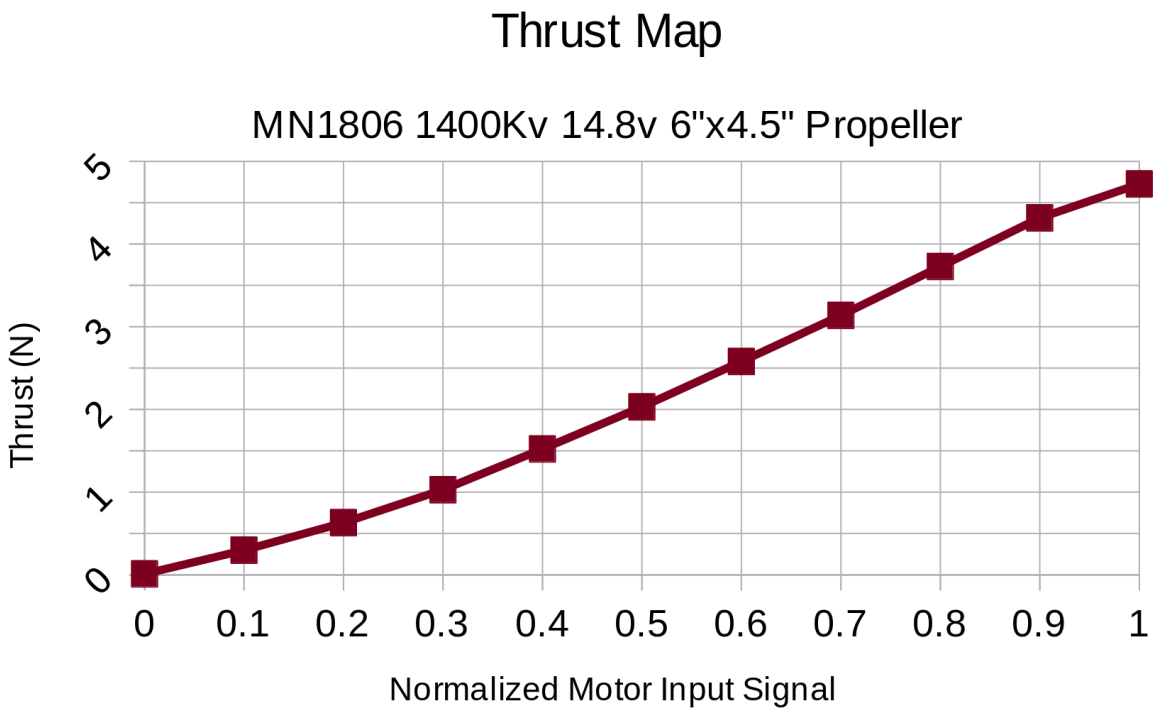


Figure 5.2: Thrust curve for motor propeller combination used.

on which motor and propeller combinations are in use. These are selectable in the settings and configuration file, alongside a linear map for unknown motor models.

Generation of specific SISO controllers for multirotor flight is a well-understood problem, for further information and specifics we direct the reader to the full implementation in source code at https://github.com/StrawsonDesign/rc_pilot.

Figure 5.3: EduMAV in flight inside Vicon Motion Capture System.

5.2 Experimental Model Validation

For this case study we designed and constructed a hexacopter frame with interchangeable 3D printed motor mounts. We then used a configurable parametric CAD model for the motor mounts which allowed us to generate motor mount models with arbitrary motor orientation in just a few minutes. This meant we could test many rotor orientations on the same frame in a very short period of time.

We utilized an RCbenchmark thrust stand [RCB] to characterize the individual motor and propeller combinations and also directly measure the available thrust in X and Y. We also used a 2kW DC power supply to drive the single-motor test and the full frame test to ensure varying battery voltages did not effect the result. The constructed frame is pictured in Fig.5.5 and Fig.5.4.

Figure 5.4: Isometric view of case study airframe highlighting the rotor angles.

Figure 5.5: Top view of case study airframe with optimized angles as used for experiment.

5.2.1 Experimental Results

The control force vectors f_{min} and f_{max} from Eq.3.9 predict a maximum control input of $3.49N$ in the $+X$ direction, $1.99N$ in the $-X$ direction, and $1.45N$ in the $\pm Y$ direction. We simulated a hover scenario with all rotors spun up to the hover vector s_h while the entire unit was fixed on the same thrust test stand used to characterize the motors. We then applied thrust in the $\pm X$ and $\pm Y$ directions. The results are listed in table 5.1.

Table 5.1: Experimental Results

	$+X$	$-X$	$+Y$	$-Y$
expected (N)	3.95	-1.99	1.45	-1.45
experiment (N)	3.75	-2.32	1.66	-1.64
error	13.1%	16.6%	14.5%	13.2%

While we are pleased to have achieved higher measured thrust than the model predicts, we must understand where the error comes from. Firstly, our model assumes each rotor provides thrust directly along its axis and does not account for aerodynamic effects due to proximity to other rotors. Secondly, we noticed the motors would generate about 20% greater thrust when cold than when hot due to the increased impedance of hot copper coils. While we did allow sufficient time for the motor temperatures to stabilize, each motor is under a different load when performing the whole-frame test which does not necessarily match the conditions of the single-motor test. The thermal and aerodynamic properties of the single-motor mount also cannot be guaranteed to be equal to those of the constructed hexacopter frame. Finally, the thrust gauge and motors used are hobby-grade and may not lend themselves to high consistency or repeatability.

We are satisfied that these results are sufficiently consistent and close to our model predictions that we can proceed to use the models presented thus so far to design a 6DOF controller and compare performance between different rotor configurations and orientations.

Chapter 6

Concluding Remarks

This dissertation outlines the development of a novel robotics development ecosystem, librobotcontrol, and its three supporting reference designs, EduRover, EduMiP, and EduMAV. One of these reference designs, EduMAV, is chosen for the subject of rigorous design optimization through quasi-static feedback performance modeling, FEA modal analysis, fluid flow analysis, and experimental performance evaluation.

6.1 Principle Contributions

Here we identify key ideas presented in this dissertation which are likely to persist and leave a lasting impact in the fields of robotic prototyping, robotics education, Layered Manufacturing, and multirotor design for aerial imaging.

6.1.1 Robotics Prototyping

With the advent of low-cost credit card sized Linux computers, it is now possible for roboticists to prototype and even manufacture robots leveraging operating systems and hardware that opens all of the benefits operating systems have to offer such as networking, disk IO, and

networking. However, no such programming environment had been produced that harnessed this technology with the simplicity and low barrier to entry that is found with existing MCU environments such as Arduino.

In the course of this dissertation, we not only conceptualized, but constructed, manufactured, distributed, and refined a new ecosystem for robotics prototyping. This ecosystem has already taken hold in the world of hobbyists, academia, and industry as the BeagleBone Blue and Robotics Cape continue to become more popular platforms for prototyping.

Furthermore, Mathworks have adopted the libRobotControl ecosystem and officially support a Matlab Simulink package for the BeagleBone Blue and Robotics Cape hardware. They also make available an example Simulink Program demonstrating the functionality of EduMiP. This demonstrates that this dissertation's contribution to the robotics prototyping world is not just conceptual, but tangible today at the time of writing.

6.1.2 Robotics Education

This libRobotControl ecosystem furthermore remains a cornerstone of UCSD's robotics education as MAE 144, Embedded Control and Robotics and has also been featured in MAE280b, State Space Control. Undergraduate and Graduate students in the UCSD Robotics Lab and the Culteral Heritage Engineering Institute (CHEI) also continue to use libRobotControl along with the EduMiP, EduRover, and EduMAV reference designs for their own pursuits in robotics research. The EduMiP reference design is also available to other Universities who, hopefully, will adopt this curriculum as they observe how influential this has been in invoking pasion for robotics in the ever-growing MAE144 UCSD course.

6.1.3 Layered Manufacturing

The design of EduMiP promoted a working relationship with 3D Platform, manufacturer of the 3D printer used in its construction. This relationship has resulted in significant improvements to the hardware, software, and operational techniques of 3D Platform's printers as the author refined the print process through many iterations of EduMAV design. Specifically, the drying of nylon Filament in a vacuum and FDM parameters required for successful printing of a relatively large yet thin and hollow moncoque shell are now documented and distributed materials that other laboratories will benefit from.

6.1.4 Multirotor Design

As market demand and embedded system technology drives multirotor camera platforms smaller and smaller, it is inevitable that camera gimbals will become even less desirable on these platforms. The concept and solution for optimizing rotor angles presented in this dissertation will likely be adopted by those wishing to shrink aerial sensor platforms and fly in constrained environments while simplifying the mechanical design with as few moving parts as possible.

6.2 Future Work

As discussed in Chapter 3, the presented optimization method for rotor orientation and layout is easily extensible to more rotors, rotors in different layouts, and even multiple equilibrium points. Due to its extensibility, this methodology provides the foundation for design optimization of other platforms such as multi-mode VTOL/Horizontal Flight aircraft, as well as other rotor-propelled designs such as underwater ROVs. The author has already demonstrated in software that this method extends to Octocopters as well as optimizing for hover at multiple orientations such as looking forward and down, and hopes that others will continue to do the same.

6.3 Final Thoughts

I am joyful to conclude this dissertation knowing that the techniques and products of this work are not simply ideas on paper, but are already being adopted and in use by others in my field with both more and less experience than myself. I truly believe it is the greatest satisfaction of the academic to pursue original ideas and see them flourish.

It has been a privilege to work with such excellent faculty, researchers, and resources through the course of developing this material. I thank you again to everyone involved.

Bibliography

[3DP] 3DPlatform. 3dplatform homepage. https://3dplatform.com/. Accessed: 2017-02-09.

[BCS11] C. Bills, J. Chen, and A. Saxena. Autonomous mav flight in indoor environments using single image perspective cues. In *2011 IEEE International Conference on Robotics and Automation*, pages 5776–5783, May 2011.

[BSLRR+17] H. Bavle, J. L. Sanchez-Lopez, A. Rodriguez-Ramos, C. Sampedro, and P. Campoy. A flight altitude estimator for multirotor uavs in dynamic and unstructured indoor environments. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1044–1051, June 2017.

[CJEDF+94] Evin J. Cramer, Jr. J. E. Dennis, Paul D. Frank, Robert Michael Lewis, and Gregory R. Shubin. Problem formulation for multidisciplinary optimization. *SIAM Journal on Optimization*, 4(4):754–776, 1994.

[CZK15] Sungjoon Choi, Q. Zhou, and V. Koltun. Robust reconstruction of indoor scenes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5556–5565, June 2015.

[dji18] dji. Dji spreading wings s1000 uav, Sept 2018.

[FSZ+13] H. C. T. E. Fernando, A. T. A. De Silva, M. D. C. De Zoysa, K. A. D. C. Dilshan, and S. R. Munasinghe. Modelling, simulation and implementation of a quadrotor uav. *Industrial and Information Systems (ICIIS), 2013 8th IEEE International Conference on*, pages 207–212, Dec 2013.

[GKPC05] Balakrishnan Ganesh, Narayanan Komerath, Devi Prasad Pulla, and Albert Conlisk. Unsteady aerodynamics of rotorcraft in ground effect. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*, page 1407, 2005.

[GPM17] J. I. Giribet, C. D. Pose, and I. Mas. Fault tolerant control of an hexacopter with a tilted-rotor configuration. In *2017 XVII Workshop on Information Processing and Control (RPIC)*, pages 1–6, Sept 2017.

[GSJP17] Nicolas Gourdain, Deepali Singh, Thierry Jardin, and Sébastien Prothin. Analysis of the turbulent wake generated by a micro air vehicle hovering near the ground with a lattice boltzmann method. *Journal of the American Helicopter Society*, 62(4):1–12, 2017.

[HOB17] T. Haus, M. Orsag, and S. Bogdan. A concept of a non-tilting multirotor-uav based on moving mass control. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1618–1624, June 2017.

[HSA+15] S. Al Habsi, M. Shehada, M. Abdoon, A. Mashood, and H. Noura. Integration of a vicon camera system for indoor flight of a parrot ar drone. In *2015 10th International Symposium on Mechatronics and its Applications (ISMA)*, pages 1–6, Dec 2015.

[inc] Tyto Robotics inc. Rc benchmark propeller thrust stand.

[Kec12] Rolf-Erik Keck. A numerical investigation of nacelle anemometry for a hawt using actuator disc and line models in cfx. *Renewable Energy*, 48:72 – 84, 2012.

[KLR+01] George Kriek, Nelson Lazear, Verne Rhodes, Joe Barnes, John Bollmeier, Jane Chen Chuang, Michael W. Holdren, Anthony S. Wisbith, Jennifer Hayward, and Diane Pietrzyk. Development of emission factors for polyamide processing. *Journal of the Air & Waste Management Association*, 51(7):1001–1008, 2001.

[KSG18] E Kasminskiy, J Strawson, and H Gaudet. Vicon to mavlink over udp communication software, Jun 2018.

[Lim] Vicon Motion Systems Limited. Vicon system reference, revision 1.4.

[LKL+13] Hyeonbeom Lee, Suseong Kim, Hyon Lim, H. J. Kim, and Daewon Lee. Control of an octa-copter from modeling to experiments. *Robotics (ISR), 2013 44th International Symposium on*, pages 1–5, Oct 2013.

[LLC] Strawson Design LLC. Strawson design flight control source code. https://github.com/StrawsonDesign/fly. Accessed: 2017-02-09.

[Mar83] Robert D. Martin. *Human Brain Evolution in an Ecological Context*. American Museum of Natural History, 1983.

[MAV] MAVLink. Introduction.

[MHO14] ÃŸ Magnussen, G. Hovland, and M. Ottestad. Multicopter uav design optimization. *Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on*, pages 1–6, Sept 2014.

[PMK+13] Caitlin Powers, Daniel Mellinger, Aleksandr Kushleyev, Bruce Kothmann, and Vijay Kumar. *Influence of Aerodynamics and Proximity Effects in Quadrotor Flight*, pages 289–302. Springer International Publishing, Heidelberg, 2013.

[RCB] RCBenchmark. Rcbenchmark product homepage. https://www.rcbenchmark.com. Accessed: 2017-02-09.

[RCR14] David Conal Robinson, Hoam Chung, and Kris Ryan. Computational investigation of micro rotorcraft near-wall hovering aerodynamics. *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014.

[RI99] James K. Rilling and Thomas R. Insel. The primate neocortex in comparative perspective using magnetic resonance imaging. *Journal of Human Evolution*, 37(2):191–223, 1999.

[SBKon] James Strawson, Thomas Bewley, and Falko Kuester. Rotor orientation optimization for direct 6 degree of freedom control of multirotors. *IEEE Transactions on Robotics*, Manuscript submitted for publication.

[Sid82] J.N. Siddall. *Optimal Engineering Design: Principles and Applications*. Dekker Mechanical Engineering. Taylor & Francis, 1982.

[Sof] IntelÂő Software. IntelÂő aero platform developer kits.

[Sof15] IntelÂő Software. Utility for changing laser camera parameters (ivcam v0.5), Jan 2015.

[SSK+15] T. Shimizu, S. Suzuki, T. Kawamura, H. Ueno, and H. Murakami. Proposal of 6dof multi-copter and verification of its controllability. In *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 810–815, July 2015.

[Str18a] James Strawson. Carrier board for 6 brushless motor controllers in beaglebone form factor, Jun 2018.

[Str18b] James Strawson. J. strawson robotics control library, Aug 2018.

[Str18c] James Strawson. Rc_pilot 6dof flight controller, Aug 2018.

[Tau] Taulman3D. Taulman 910 nylon polymer.

[VIC] VICON. Vicon datastream sdk 1.7.0 developer's manual, vicon motion systems ltd. 2016.

[WB10] Brian T Whitehead and Stefan R Bieniawski. Model reference adaptive control of a quadrotor uav. In *AIAA Guidance, Navigation, and Control Conference*, pages 2–5, 2010.

[Wor] CyPhy Works. Cyphy works lvl1 drone kickstarter campaign. https://www.kickstarter.com/projects/1719668770/cyphy-lvl-1-drone-reinvented-for-performance-and-c. Accessed: 2017-02-09.

[YLP16] Steven Yoon, Henry C Lee, and Thomas H Pulliam. Computational analysis of multi-rotor flows. In *54th AIAA Aerospace Sciences Meeting*, page 0812, 2016.

[ZWW16] J. Zou, C. Wang, and Y. M. Wang. The development of indoor positioning aerial robot based on motion capture system. In *2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE)*, pages 380–383, Nov 2016.