UNIVERSITY OF CALIFORNIA SAN DIEGO

Design, Deployment, Navigation, and Control of Mobile Robots for Perception and Sensor Data Collection

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

in

Engineering Sciences (Mechanical Engineering)

by

Pengcheng Cao

Committee in charge:

Professor Thomas Bewley, Chair
Professor Falko Kuester, Co-Chair
Professor Raymond de Callafon
Professor Mauricio de Oliveira
Professor John T. Hwang

2024

The Dissertation of Pengcheng Cao is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

DEDICATION

This dissertation is dedicated to my beloved parents, my father Xuedong Cao and mother Lanqin Xu, for their years of love and support that has been really essential for the every milestone achieved in my studies since I was a kid.

It is also dedicated to my beloved wife and soulmate, Mengqian Hang, for her indefinite and unconditional love, support, and care without which I could never become the person I am today.

EPIGRAPH

I just wished to hereby examine into all that concerns Heaven and the human, and penetrate the changes of the past and present, completing all as the work of one family.

—Sima Qian

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

To my dearest parents, **Xuedong Cao and Lanqin Xu**, for your unconditional love and support over my years of pursuit of goals in my education and career. It is also certain that without your disciplining and teaching in how to become a good person, how to face my ups and downs and many other how-toes, I would not have come this far. I owe you my greatest gratitude, forever.

To my dearest wife and soulmate, **Mengqian Hang**, for your indefinite and unconditional love, support, and care. It is you that make me understand what the true love is. It is you that make me understand all challenges of life can be tackled if there is someone always by my side. Thank you for being my love, soulmate, and best friend of this life. I am yours and you are mine.

To my advisor, **Prof. Falko Kuester**. It is you who have been providing and advising on exciting hand-on projects and research opportunities for generations of students who are interested in robotic systems, additive manufacturing, remote sensing and etc. It is also you who opened doors for me to start my research career and have been kindly providing all kinds of support all over the years. You are the most noble person I have worked with in my career and I want to thank you for having faith in me and your countless occasions of supporting.

To my advisor, **Prof. Thomas Bewley**. It is you who introduced me to the world of robotics and dynamic systems and controls. You are the most brilliant professor by whom I had the chance to be advised and from whom I learned about the high standards of creativity in engineering research. And by working with you as teaching assistant I learned about what makes a great instructor and how to be responsible for the entire cohort as you always are. I want to thank you for being my role model in academia.

To my esteemed committee members, **Prof. John Tae Hyeon Hwang**, **Prof. Mauricio de Oliveira**, and **Prof. Raymond de Callafon**. All of you have been

teaching very enlightening and inspiring classes from which I have learned a great deal in the fields of design optimization, aerospace engineering, system identification, and general dynamic systems and controls. Also thank you for your years of support and guidance as my committee members.

To **Prof. Tara Hutchinson** for your invaluable instructions and inputs during multiple years of drone imagery study at the UCSD NHERI Large High Performance Outdoor Shake Table. Also for your great personality, curiosity, and academic spirit which deeply touches and influences me in a very positive way.

To **Prof. Mohammad Yousef Bahadori**, **Prof. Renkun Chen**, **Prof. William M. McEneaney**, and **Prof. Sonia Martinez Diaz** for providing me with opportunities to working with you as instructional assistant. Thank you for your trust and it was an honor to have worked with you and strived to deliver quality and well-organized MAE courses to our students. And special thanks to **Prof. Mohammad Yousef Bahadori** for your kind and encouraging remarks, enjoy your retirement!

To **Prof. Nikolay Atanasov** and **Prof. Henrik Christensen** for your fun and enlightening robotics courses ECE 276 and CSE 276 series. I have been really grateful that I could take courses with you who are two of the very prominent robotics engineers and researchers of our time. Your courses are very well-organized, in-depth, and insightful, and fascinate me to pursue a career in robotics engineering. For this I owe you my gratitude forever.

To **Prof. Dominique Rissolo**, **Eric Lo**, **Hilda Lozano Bravo**, **Tanner Norton**, **John Driscoll**, **Scott McAvoy**, **Nathan Hui**, **Chris McFarland**, **Vid Petrovic**, **Lovella Cacho**, **Crystal Liu**, **Krista Meinke**, **Joana Halnez**, **Sandra de Sousa**, **Brittany Norris**, **Kyung Brown** and many others. All of you have not only become my valuable colleagues to work with in CHEI Lab, Qualcomm Institute, and MAE department, but also kind and supportive friends during my study. Thank you for your expertise, as

well as for kindness and warm hearts.

To my labmates and friends in CHEI Lab including **Dr. James Strawson**, **Dr. Dominique Meyer**, **Dr. Luca De Vivo Nicoloso**, **Dr. Joshua Pelz**, **Dr. Theoharis Papatrechas**, **Prof. Caterina Morganti**, **Kunal Gupta**, **Dr. Hugh Runyan**, **Danylo Dorhobytsky**, **Zhaoliang Zheng**, **Zhaoyang Ma**, **Alexander Mai**, **Danny Tran**, **Joseph Phillips**, **Jonathan Klingspon**, **Shreyas Niradi**, **Robin Wang**, **Yash Dodhia**, **Moloy Das**, and **Shriya Shetty**. To my labmates and friends in FCCR Lab, including **Dr. Kurt Talke**, **Dr. Daniel Yang**, **Dr. Eric Sihite**, **Muhan Zhao**, **Ricardo Gorinstein**, **Xuebin Zhu**, **Chase Lazar**, **Everbrook Zhou**, and **Dr. Joonyoung Jang**. To my friends and co-authors in Prof. Tara's Lab, **Ruipu Ji**, **Shokrullah Sorosh**, **Prof. Gloria Faraone**, and **Prof. Xiang Wang**. And to my friends and collegemates, **Prof. Haochen Quan**, **Dr. Yangsheng Hu**, **Dr. Zhouyi Liao**, **Dr. Changfu Li**, **Dr. Li Tan**, **Dr. Shangjie Frank Ma**, **Dr. Yang Wang**, **Dr. Naiqing Zhang**, **Dr. Mingsong Jiang**, **Dr. Haoyang Leng**, **Dr. Wei Zhou**, **Ruiqi Wang**, **Ricardo Felez-Val**, **Dr. Victor Gandarillas**, **Jiayao Yan**, **Dr. Bingran Wang**, **Tianyi Lu**, **Zhizhen Qin**, **Dr. Pengcheng Wu**, **Haitong Chen**, **Mingyi Li**, and many others. You have provided me with great scientific insight and support for many of my projects. You have also enriched my 5-year student life at UC San Diego with many precious and delightful moments together. Thank you for everything.

To my teammates and friends from SE 207/281, **He Huang**, **Yifan Xu**, **Mingchen Mao**, **Alexandre Guibert**, and **Robert Chambers**. Thank you for working with me on multiple interesting 3D-printable UAV projects and making them become solid research items.

To my friends from RoboGrads including **Dr. David Paz-ruiz**, **Dr. Akanimoh 'Sanmi Adeleye**, **Dr. Jacob Johnson**, **Ehsan Ziaee**, **Prof. Alyssa Kubota**, **Dr. Sachiko Matsumoto**, **Hengyuan Zhang**, **Sander Tonkens**, **Nikhil Shinde**, **Yuheng**

Zhi, and many others for our many treasured and delightful moments together. And special thanks to **Prof. Henrik Christensen** for consistently supporting RoboGrads and its board members.

To UC San Diego Extended Studies and its Summer Program Managers **Anais Perez** and **Namphuong Pullman**, and all the young, brilliant and vibrant students whom I had the opportunity to teach and work with . It is you who made my dream come true as being an instructor in college. I really enjoyed my career as a course organizer and instructor and want to thank you for having faith in me who was actually a first-time teacher.

To **Dr. Zhe Shen** from the University of Tokyo of Japan, **Dr. Julian Galvez-Serna** from Queensland University of Technology of Australia, **Prof. Hassan Alkomy** from University of New Brunswick of Canada, and **Yufei Wu** from University of Tennessee, Knoxville for our cross-institutional collaborations and friendships.

Finally, to everyone else, family, friends, and colleagues that have spoken words of encouragement. The combined support of yours is what has made this possible.

Next, I would like to acknowledge the co-author contributions and all the published, unpublished, or in-review materials and research articles incorporated in this dissertation in the sequence of chapters.

Chapter 2, in full, is a reprint of the material as it appears, titled "Decoupled translational and rotational flight control designs of canted-rotor hexacopters", in *Proceedings of AIAA Scitech 2021 Forum.* Cao, Pengcheng; Strawson, James; Bewley, Thomas; Kuester, Falko. The author of the dissertation was the primary investigator and first author of this article.

Chapter 3, in full, is a reprint of the material as it appears, titled "Mission-Oriented Trajectory Optimization for Search-and-Rescue Multi-rotor UAVs in Cluttered and GPS-Denied Environment", in *Proceedings of AIAA Aviation 2022 Forum.* Cao, Pengcheng;

Hwang, John T; Bewley, Thomas; Kuester, Falko. The author of the dissertation was the primary investigator and first author of this article.

Chapter 4, in full, is a reprint of the material as it appears, titled "QuadGlider: Towards the Design and Control of a Bio-Inspired Multi-Modal UAV with Compliant Wings", in *Proceedings of 2023 IEEE Aerospace Conference.* Cao, Pengcheng; Phillips, Joseph; Bewley, Thomas; Kuester, Falko. The author of the dissertation was the primary investigator and first author of this article.

Chapter 5, in full, is a reprint of the material as it appears, titled "BeagleRover: An Open-Source 3D-Printable Robotic Platform for Engineering Education and Research", in *Proceedings of AIAA Scitech 2022 Forum.* Cao, Pengcheng; Strawson, James; Zhu, Xuebin; Zhou, Everbrook; Lazar, Chase; Meyer, Dominique; Zheng, Zhaoliang; Bewley, Thomas; Kuester, Falko. The author of the dissertation was the primary investigator and first author of this article.

Chapter 6, in full, is a reprint of the material as it may appear, titled "Cluster-based Dynamic Object Filtering via Egocentric Motion Detection for Building Static 3D Point Cloud Maps", in *Proceedings of 2023 IEEE Robotic Computing Conference.* Cao, Pengcheng; Bewley, Thomas; Kuester, Falko. The author of the dissertation was the primary investigator and first author of this article.

Chapter 7, in full, is a reprint of the material as it may appear, titled "UAV-Based Video Analysis and Semantic Segmentation for SHM of Earthquake-excited Structures", in *Proceedings of 2024 World Conference on Earthquake Engineering.* Cao, Pengcheng; Ji, Ruipu; Ma, Zhaoyang; Sorosh, Shokrullah; Lo, Eric; Driscoll, John; Norton, Tanner; Wang, Xiang; Hutchinson, Tara; Pei, Shiling; Kuester, Falko. The author of the dissertation was the primary investigator and first author of this article.

VITA

2009–2012	High School Diploma, Suzhou High School of Jiangsu Province, China
2012–2015	Bachelor of Engineering, Shandong University, China
2015–2016	Exchange Program and Practicum, Ulm University & Heimholtz Institute Ulm, Germany
2016–2018	Master of Science, Department of Mechanical and Aerospace Engineering, University of California San Diego
2018	Manufacturing Engineer, Value Windows & Doors, Duarte, CA
2018–2024	Doctor of Philosophy, Department of Mechanical and Aerospace Engineering, University of California San Diego

PUBLICATIONS

Pengcheng Cao, Ruipu Ji, Zhaoyang Ma, Shokrullah Sorosh, Eric Lo, John Driscoll, Tanner Norton, Xiang Wang, Tara Hutchinson, Shiling Pei, and Falko Kuester. "UAV-Based Video Analysis and Semantic Segmentation for SHM of Earthquake-excited Structures (In Review)". In: *Proceedings of 2024 World Conference on Earthquake Engineering*. The International Association for Earthquake Engineering (IAEE), 2024.

Pengcheng Cao, Thomas Bewley, and Falko Kuester. "Cluster-based Dynamic Object Filtering via Egocentric Motion Detection for Building Static 3D Point Cloud Maps." In: *Proceedings of 2023 Seventh IEEE International Conference on Robotic Computing (IRC)*, pp. 368-372. IEEE, Dec 2023.

Alexandre T Guibert*, Robert J Chambers*, **Pengcheng Cao**, H Alicia Kim, Shengqiang Cai, and Falko Kuester. "Gripping Aerial Topology Optimized Robot (GATOR)". In: *Proceedings of 2023 IEEE Aerospace Conference*, pp. 01-10, IEEE, May 2023.

Pengcheng Cao, Joseph Phillips, Thomas Bewley, and Falko Kuester. "QuadGlider: Towards the Design and Control of a Bio-Inspired Multi-Modal UAV with Compliant Wings". In: *Proceedings of 2023 IEEE Aerospace Conference*. pp. 1–17, IEEE, May 2023.

Pengcheng Cao, John T Hwang, Thomas Bewley, and Falko Kuester. "Mission-Oriented Trajectory Optimization for Search-and-Rescue Multi-rotor UAVs in Cluttered and GPS-Denied Environments". In: *Proceedings of AIAA AVIATION 2022 Forum*, p. 3999, AIAA, June 2022.

Pengcheng Cao, James Strawson, Xuebin Zhu, Everbrook Zhou, Chase Lazar, Dominique Meyer, Zhaoliang Zheng, Thomas Bewley, and Falko Kuester. "BeagleRover: An Open-Source 3D-Printable Robotic Platform for Engineering Education and Research". In: *Proceedings of AIAA SCITECH 2022 Forum*, p. 1914, AIAA, Jan 2022.

James Strawson, **Pengcheng Cao**, Danny Tran, Thomas Bewley, and Falko Kuester. "Monocoque Multirotor Airframe Design with Rotor Orientations Optimized for Direct 6-DoF UAV Flight Control". In: *Proceedings of AIAA AVIATION 2021 FORUM*, p. 2431, AIAA, June 2021.

James Strawson, **Pengcheng Cao**, Thomas Bewley, and Falko Kuester (2021). "Rotor orientation optimization for direct 6 degree of freedom control of multirotors". In: *Proceedings of 2021 IEEE Aerospace Conference*, pp. 1–12, IEEE, March 2021.

Pengcheng Cao, James Strawson, Thomas Bewley, and Falko Kuester. "Decoupled translational and rotational flight control designs of canted-rotor hexacopters". In: *Proceedings of AIAA Scitech 2021 Forum*, p. 1058, AIAA, Jan 2021.

ABSTRACT OF THE DISSERTATION

Design, Deployment, Navigation, and Control of Mobile Robots for Perception and Sensor Data Collection

by

Pengcheng Cao

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2024

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

Aerial robots, including rotary-wing and fixed-wing unmanned aerial vehicles or UAVs, have shown great capabilities in surveying as well as search and rescue from above. However, either rotary-wing or fixed-wing UAVs have nearly insoluble flaws. In order to overcome the under-actuating nature of multi-rotor UAVs, Chapter 2 proposes modeling

methods and control schemes for fully-actuated hexacopters. Additionally, rotary-wing robots suffer from limited battery life as well as lack of fail-safe mechanism upon losing motors, while fixed-wing robots lacks the ability to take off and land vertically. Therefore, Chapter 4 proposes a bio-inpired hybrid aerial robot to extend mutli-rotor flight time and fail-safe capability and provide fixed-wing glider with vertical take-off and landing or VTOL capability. Moreover, to extend the flight time and optimize the energy consumption of multi-rotor UAVs, Chapter 3 proposes a multi-disciplinary design optimization based flight trajectory optimizer involving linear rotor inflow models to reduce flight time or energy consumption of specific missions.

In terms of unmanned ground vehicles or UGVs used for perception and mapping, there has been a research gap to provide a low-cost, highly agile over-actuated chassis design. Chapter 5 proposes a 3D-printable double Ackermann steering chassis design with 2-wheel standing and balancing capability to fill in this gap. Chapter 6, on the other hand, proposes the system design of a UGV capable of performing perception and mapping in a limited lighting, unstructured, and GPS-denied environment based on a nevertheless nonholonomic chassis, where primary concern becomes the reliability in performing real-time mapping and preservation of solely static environment.

The last but not least topic discussed in this dissertation is to promote the role of UAV imagery in earthquake response. In Chapter 7 we combine the traditional UAV plan view perspective with north and east elevation view video data to provide motion estimation in all 6 degrees of freedom, as well as proposing Video Transformer for motion tracking.

All in all, with attempts to expand and promote the designs, deployment and control schemes of both aerial and ground mobile robots, this dissertation strives to provide case study results and state-of-the-art methods for future robotics studies.

Chapter 1

Introduction

1.1 Motivation

Research on mobile robots, particularly Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs), is driven by the quest to extend human capabilities and reach in various environments. UAVs, with their ability to fly and maneuver in three-dimensional space, offer unique perspectives and access that are otherwise unattainable, making them invaluable in tasks such as aerial surveillance, disaster management, and agricultural monitoring. UGVs, on the other hand, excel in ground-based operations like autonomous transportation, military reconnaissance, and hazardous material handling, where human presence is risky or inefficient. The synergy between UAVs and UGVs can revolutionize industries by providing comprehensive coverage, enhanced data collection, and greater operational efficiency. This research not only pushes the boundaries of robotic technology but also promises to transform how we interact with and manage our physical world.

Recent years have witnessed the rapid development of multirotor UAVs with advanced on-board and real-time autonomous capabilities. The multirotor UAVs offer major advantages with low risk to human pilots and higher confidence and success rate in performing the military tasks of aerial surveillance, inspection, and reconnaissance in

perilous, adverse, and cluttered environment [Ken12]. As for civilian usages, there have been significant market share and research efforts for multirotor UAVs in applications of aerial imaging [NR14][Wit+14], aerial manipulation [RLO18][San+20], and structural inspection [JV13][Wat+20] etc. As adopted by the majority of the off-the-shelf multirotor UAVs, the standard multirotor designs typically align their propeller axes in parallel, resulting in an under-actuated system whose pose cannot track an arbitrary trajectory overtime in 6 degrees of freedom (DoFs) [Shi+15].

For search-and-rescue (SAR) mission purposes, various UAV researches and designs have been proposed and some are already in use to assist the SAR task forces [PPK17]. However, there has not been a comprehensive work in design optimization and guidance, navigation, and control (GNC) system design for fully-acuated multirotor UAVs used in SAR missions. In this case, The author would like to plan a Ph.D. thesis on this topic. First, the author would like to formulate a design optimization methodology for multirotor UAVs. The UAV is designed to be fully-actuated with specific tilted rotor layouts in order to enhance its maneuverability while operating close to floors, walls, ceilings or other hard surfaces [Str18a]. In addition, the thrusters and battery characteristics are optimized according to the mission to increase battery life and energy efficiency. Second, the flight controller is to be designed for the fully-actuated UAV such that it can robustly and accurately track 3D trajectories and rapidly delivered to the target position in the building where SAR effort is needed. Last but not least, the author want to work on a multi-sensor fusion system for the UAV to realize situational awareness incorporating higher-level autonomy (level 4 or higher of GNC capabilities according to NIST ALFUS framework[Hua+05] as tabulated in [Ken12]).

As for applications with UGVs, on one hand, one can rarely find a small-sized, 4-wheel-driven mobile robot which can serve as a chassis design for extraterrestrial or off-road rovers. And one can hardly find the literature deploying mobile robots with

complex kinematics design fro educational and research uses.

On the other hand, one of the significant challenges in utilizing LiDAR data for mapping and navigation is the presence of dynamic objects within the environment [LT10]. All of the SOTA methods, however, require a prior map to be created before filtering out dynamic objects points in the post-processing. There has been no outstanding findings or methods in removing dynamic objects on the fly when a prior map is generated known to the authors.

At last, in terms of the deployment of UAVs in seismic and structural studies, imagery and visual sensing have proven to be effective to assist tracking and reconstructing the dynamic response and assess the post-earthquake structural health of earthquake excited buildings as illustrated in some previous works [Wan+20] [Wan+22]. However, previous methodologies were limited to viewing the assessed specimen or structure from a single perspective only or failed to combine the information from multiple views to obtain motion estimation for all 6 degrees of freedom (DoFs). Also, target-based motion tracking methods make it difficult to estimate the motions of all other non-targeted regions of structures. Natural features on structures need to be extracted for a more general feature tracking.

1.2 Literature Review

In this section, we provide a brief literature review of various aspects of robotics research discussed and covered in this dissertation. The purpose of this literature review is to further provide context extended from the Motivation so that the critical problems raised here will concern some of the most targeted research gaps.

1.2.1 Tilt-rotor Multicopter Flight Control

Multirotors are currently one of the most prevalent types of Unmanned Aerial Vehicle (UAV) platforms. They have been widely used for various research, commercial, industrial, or hobbyist purposes due to their unique abilities, such as vertical takeoff and landing, hovering, and mechanical simplicity [Ras+20b]. Thus, they are frequently deployed and utilized for applications in areas including aerial imaging [Wyp+14] and manipulation [RLO18]. Among all multi-rotors, although the vehicles with the configuration of 4 rotors, namely quadrotors, are advantageous due to its simplicity of design and manufacturing, the 6-rotor configuration drones, namely hexrotors, perform better in reliability due to the tolerance to the failure of one or two motors [GSG16], and generally with greater agility due to more thrust provided by 2 extra motors.

Despite the advantages of multirotors, however, the standard multirotor designs typically employ a parallel rotor-axes configuration which arranges their propellers in the same plane, resulting in an under-actuated system whose pose cannot track an arbitrary trajectory overtime in 6 degrees of freedom (DoFs) [Shi+15]. The under-actuated nature of a traditional multirotor can be interpreted as its inability to exert forces parallel to its lateral plane of the body frame [Raj+15], requiring these lateral forces to be generated by rolling and pitching.

Apart from advancing the quadrotor design and dynamics, there have been modeling works of fully-actuated hexrotor UAVs prior to the proposed model in this proposal. Rajappa et al. [Raj+15] modelled their fully-actuated hexrotor via deviating the rotor axes from the vertical axes by optimized canted and dihedral angles with preliminary design optimization. Jiang and Voyles [JV13] modeled the fully-actuated hexrotor dynamics providing force closure for a dexterous aerial vehicle for mobile manipulation tasks. Some researchers have already attempted to overcome the under-actuation drawback of

quadrotors by presenting the designs and dynamics modelling of fully-actuated quadrotors. Nemati and Kumar [NK14b], Ryll et al. [RBG12] proposed and modeled the tilting rotor quadrotor which provides the added advantage of over-actuation, made possible by additional actuated controls, as compared to a traditional quadcopter without titling rotors. Some others proposed, modeled and tested the quadcopters with four extra propellers in order to decouple the attitude and translation dynamics [Sal+08], or to reject disturbances [FN17].

1.2.2 Multirotor UAV Navigation

In addition, The flight control problem for multirotor UAVs has attracted the attention of many researchers specializing on areas of control and robotics. Although some of the multirotor UAVs rely on manual flight control by receiving control commands sent by human pilots via remote controls or other external systems, the flight control problem in the scope of this thesis is limited to automatic flight control. Automatic control can be defined as the process of manipulating the inputs to the UAV system to obtain a desired effect on its outputs without or with minimum intervention of a human pilot in the control loop[Ken12]. Most of the applied automatic control schemes and techniques in the literature can be categorized into [CWL09] [Pen+09]: (1) learning-based methods [She+18] [HW20] [Liu+18], (2) Optimzation-based control methods [QH14] [Ahm+08] [Pen+09], (3) model-based control methods [Teu+10] [Lop+16] [KAB16].

Last but not least, designing an autonomous UAS with higher levels of autonomy requires an appropriate navigation system for sensing, state estimation, environment perception, and situational awareness. The multi-sensor fusion system has been adopted and applied in many UAV systems for positioning [Pen+20] [LF18], pose estimations [Mas+18], detecting and avoiding obstacles [Lyu+18] [RBA15] etc to help realize situation awareness, and is also used in geographical studies [Mas+18].

1.2.3 Bio-inspired UAV Designs

In fact, various novel bio-inspired UAV designs with compliant membrane wings were proposed and studied thanks to bio-inspirations. Gerdes et al. [Ger+14] developed the Robo Raven which is a bird-inspired flapping-wing UAV which is able to actuate each of the two highly compliant wings independently. Abdulrahim et al. [AGL05] presented a micro aerial vehicle design with morphing membrane wings to validate the roll control scheme of twisting the wing tips instead of using ailerons. These two designs indicate that membrane wings can be successfully applied on micro to small UAVs to generate lift but each incorporates only one single flight or locomotion mode. Shin et al. [SPP18][SPP19] presented a multi-modal robot inspired by *Pteromyini* or flying squirrels which is able to achieve both locomotion modes of walking and gliding while can also control its angle of attack using robotic legs and tail during the flight. This design, however, cannot take off and land vertically or hover in its airborne mode. Karasek et al. [Kar14] [Ros+17] presented the designs of several robotic hummingbirds capable of performing near-hover and forward flapping flight. Ma et al. [MFW12][Ma+13] presented the design and control of Havard Robobee which is a coin-sized micro aerial robot which can perform vertical take-off, hovering, and maneuvering of its flight directions. However, although flapping wings could have higher efficiency than did propellers in hover propellers [Chi+20], flapping-wing UAVs consumed significantly higher power than fixed membrane-wing UAVs during cruise or glide since flapping would increase the induced drag considerably [Sac16].

1.2.4 Low-cost or 3D-printable UGV Designs

Micro robot platforms have been involved in pre-college and college STEM educations since 1960s. In 1969, Seymour Papert first developed the idea of using Logo programming language and Turtle robots to teach mathematics [RO+90] to pupils. During the last 50 years, the development of educational robots has been boosted by both the

advancements in technologies and increasing market demands from both parents and educational institutes. Some of the most successful educational robots include Lego Mindstorms EV3 [RL14] to teach robotic motions and programming, humanoid robot NAO [SR13] to interact with children, and MakeBlock mBot [mak18] as one of the low-cost STEM robots. For students during their college education in engineering, the Renaissance Robotics and UCSD Coordinates Robotics Lab have developed the EduMIP robot [Str18b] which is a 2-wheel mobile inverted pendulum driven by a BeagleBone Blue SoC computer[bea21].

Apart from educational robotics, 3D printing has also become an emerging area with both fast progressing research and applications. For robotics, 3D printing is implemented mainly for novel design and fast prototyping. In educational robotics, de Souza and Elisiário [SE19] designed multiple 3D-printable robotic projects to educate teenagers in STEM disciplines. In addition, Gonzalez et al. [Gon+12] and Lapeyre et al. [Lap+14] studied 3D-printable mobile and humanoid robots for educational uses, respectively. In soft robotics, Gul el al. [Gul+17] implemented fused deposition modeling (FDM) based 3D printing and post-processing to manufacture a cylindrical actuator as a steerable catheter. In aerial robots, Strawson et al. [Str+21b] [Str+21a] and De Vivo et al. [DTK19] attempted on implementing large-scale 3D printers to manufacture aerodynamically functional components with structural and mechanical optimization for multirotor and fixed-wing unmanned aerial vehicles (UAVs), respectively.

1.2.5 Dynamic Object Filtering in UGV Mapping

One of the significant challenges in utilizing LiDAR data for mapping and navigation is the presence of dynamic objects within the environment [LT10]. On one hand, during the point cloud registration, dynamic objects may introduce inconsistencies in static maps by occluding the static part of environment [Hah+03] and drifting away the odometry to cause misalignment when registering multiple scans into a single map [Yan+18]. On the

other hand, regarding map construction, dynamic objects can cause ghost trail effect and therefore leave residual or noise points to become part of the final generated static map [LHM21]. To overcome this challenge, researchers have proposed a number of state-of-the-art algorithms, including ERASOR [LHM21], Removert [KK20], and Peopleremover [SN18]. All of these methods, however, require a prior map to be created before filtering out dynamic objects points in the post-processing. There has been no outstanding findings or methods in removing dynamic objects on the fly when a prior map is generated known to the authors.

1.2.6 Use of UAV Imagery in Seismic Studies

Recently, UAV-based video analysis has attracted the interest of researchers due to its cost-effectiveness. Wang et al. (2022) [Wan+22] performed UAV-based video analysis to monitor the earthquake response of full-scale structures. Horizontal displacements of the structure in two directions were obtained by tracking the square reference targets on the rooftop from the plan-view video applying a sub-pixel edge detector [Tru+13]. The camera pose and orientation were recovered by world-to-image correspondences of stationary targets. Han et al. (2022) [HWF22] used circular reference targets and laser spot projected from a stationary highly penetrating laser lamp as reference points for the videos captured by the UAV to monitor the in-plane displacement of small-scale and full-scale structures. The motion of the UAV parallel and perpendicular to the plane of the structure was eliminated by the "physical to pixel" scaling relationship of the distance between the geometric centers of the dots on the marker. The structural displacement was obtained by the marker displacement relative to the laser spot after scaling factor conversion.

In real structural health monitoring tasks, placing reference targets on periphery of structures may not be possible due to the consideration of integrity and aesthetics of

structures. Also, target-based motion tracking methods make it difficult to estimate the motions of all other non-targeted regions of structures. Natural features on structures need to be extracted for a more general feature tracking. Wang et al. (2020) [Wan+20] applied speed-up robust features (SURF) algorithm [Bay+08] and Canny edge detector [Can86] to identify the region-based pixel features to monitor the earthquake response of a 6-story full-scale structure. Sun et al. (2022) [Sun+22] proposed an analysis approach for point cloud transformed from the UAV captured image, which was validated by the displacement monitoring of a small-scale building model. The principle of region growth was applied to point cloud segmentation to obtain the marking points. Weng et al. (2021) [Wen+21] proposed a homography-based method to monitor the structural displacement from video taken by a UAV-mounted camera. Feedforward neural network (FNN) was applied to obtain the image coordinates of selected features of the structure on its stationary position, which were further used as homography features. The current video-based displacement monitoring methods show good performance in the accuracy of measurement. However, prior efforts primarily focused on measuring 3-DoF in-plane displacements and rotations. There is still a research gap for measuring the full 6-DoF motion response using video-based displacement monitoring.

1.3 Critical Problems

Therefore, 6 critical problems this dissertation attempts to address are listed as follows:

- For fully-actuated multirotor UAVs, can an navigation flight controller be designed to robustly and accurately track 3D trajectories while decoupling the attitude and linear velocity control?
- Can a mission-oriented multi-disciplinary design optimization (MDO) problem be

formulated to optimize the multi-rotor flight trajectory considering advanced flight dynamics with rotor inflow model?

- Can we propose a bio-inspired UAV design with hybrid modality of both rotary-rotor drone and fixed-wing glider to provide fail-safe mechanism against motor failure?

- Can we design a low-cost 3D-printable UGV which can serve for both educational and field survey purposes?

- Can we develop a fast dynamic object filtering algorithm and make it a robust software package to filter out dynamic objects like moving pedestrians, bikers, scooters, and cars for real-time 3D point cloud mapping by UGVs or UAVs?

- Last but not least, can we propose a novel UAV-based video analysis schemes utilizing all three-view perspectives to estimate 6-DoF structural motions? Additionally, can you apply the novel Visual Transformer model to our research?

1.4 Dissertation Structure

The structure of this dissertation is as follows. This Chapter, i.e. Chapter 1, provides an introduction and literature review for different aspects involved in this paper, as well as raising the three critical problems that this dissertation strives to provide solutions towards.

In order to overcome the under-actuating nature of multi-rotor UAVs, Chapter 2 proposes modeling methods and control schemes for fully-actuated hexacopters.

Moreover, to extend the flight time and optimize the energy consumption of multi-rotor UAVs, Chapter 3 proposes a multi-disciplinary design optimization based flight trajectory optimizer involving linear rotor inflow models to reduce flight time or energy consumption of specific missions.

Additionally, rotary-wing robots suffer from limited battery life as well as lack of fail-safe mechanism upon losing motors, while fixed-wing robots lacks the ability to take off and land vertically. Therefore, Chapter 4 proposes a bio-inpired hybrid aerial robot to extend mutli-rotor flight time and fail-safe capability and provide fixed-wing glider with vertical take-off and landing or VTOL capability.

In terms of unmanned ground vehicles or UGVs used for perception and mapping, there has been a research gap to provide a low-cost, highly agile over-actuated chassis design. Chapter 5 proposes a 3D-printable double Ackermann steering chassis design with 2-wheel standing and balancing capability to fill in this gap.

Chapter 6, on the other hand, proposes the system design of a UGV capable of performing perception and mapping in a limited lighting, unstructured, and GPS-denied environment based on a nevertheless nonholonomic chassis, where primary concern becomes the reliability in performing real-time mapping and preservation of solely static environment.

The next topic discussed in this dissertation is to promote the role of UAV imagery in earthquake response. In Chapter 7 we combine the traditional UAV plan view perspective with north and east elevation view video data to provide motion estimation in all 6 degrees of freedom, as well as proposing Video Transformer for motion tracking.

Chapter 8 provides a conclusion and summary of the entire dissertation, as well as the proposed answers to the critical problems and future research directions.

Chapter 2

Modeling and Flight Control of Tilt-rotor Hexacopters

2.1 Chapter Abstract

The mainstream designs of the multi-rotor UAVs typically arrange their motors and propellers near or in the same plane, imposing limitations on their control maneuvers due to under-actuation. The limited control maneuvers cause the multi-rotors to have to pitch and roll in order to perform lateral translations. These multi-rotor aerial vehicles also lose the capability of keeping their center-of-mass stationary in the air while rotating by a certain pitch or roll angle due to the thrusts generated along rotated vertical axis in the body frame. This paper presents the mathematical model of a hexacopter with rotor-axes tilted to enhance the control maneuverability of conventional multi-rotor UAVs. Based on the Newton-Euler formalism of dynamics, we propose the path tracking controller and the tilt-hovering controller for canted-rotor hexacopters. First, the path tracking controller decouples lateral movements from pitch and roll rotations of the vehicle. Second, the tilt-hovering controller stabilizes the position of the center of mass of the vehicle as well as its orientation while it is rotated by a certain pitch or roll angle. Numerical simulations are performed to corroborate the mathematical model and control designs, and to compare with the performance of conventional multi-rotor UAVs.

2.2 Chapter Nomenclature

$\mathcal{F}_E$	=	Earth-fixed inertial frame.
$\mathcal{F}_B$	=	hexacopter body frame.
$\mathcal{F}_{m_i}$	=	local frame of i-th rotor (motor-propeller group).
O_B	=	hexacopter center of mass (CoM) and origin of $\mathcal{F}_B$.
O_E	=	origin of $\mathcal{F}_E$, coincident with O_B before hexacopter takes off.
O_{m_i}	=	center of each rotor and origin of $\mathcal{F}_{m_i}$.
$\mathbf{R}_{EB}$	=	rotation matrix from $\mathcal{F}_B$ to $\mathcal{F}_E$.
$\mathbf{T}_{EB}$	=	body rotation angles transformation matrix from $\mathcal{F}_B$ to $\mathcal{F}_E$.
$\mathbf{R}_{Bm_i}$	=	rotation matrix from $\mathcal{F}_{m_i}$ to $\mathcal{F}_B$, where $i = 1 \dots 6$.
$\mathbf{p}_{Bm_i}$	=	the positions of O_{m_i} in body frame $\mathcal{F}_B$, where $i = 1 \dots 6$.
ϕ_i	=	rotor-axis rotation angle around X-axis of $\mathcal{F}_{m_i}$, where $i = 1 \dots 6$.
θ_i	=	rotor-axis rotation angle around Y-axis of $\mathcal{F}_{m_i}$.
ψ_i	=	the i-th rotor arm's yaw angle with respect to the body X-axis.
V_i	=	unit vector indicating orientation of i-th rotor-axis, $V_i \in \mathbb{R}^3$.
Φ	=	3-tuple of (ϕ_1, ϕ_2, ϕ_3).
Θ	=	3-tuple of $(\theta_1, \theta_2, \theta_3)$.
Ω	=	the 6-by-1 force vector of the squares of each rotor's rotary speed.
ξ	=	$[x_E, y_E, z_E]^T$, the linear displacement/position vector in $\mathcal{F}_E$.
$\mathbf{I}_B$	=	the moments of inertia matrix in $\mathcal{F}_B$, $\mathbf{I}_B \in \mathbb{R}^{3\times3}$.
η_B	=	the vehicle's angular velocity in $\mathcal{F}_B$, $\eta_B \in \mathbb{R}^3$.
η_E	=	the vehicle's angular velocity in $\mathcal{F}_E$, $\eta_E \in \mathbb{R}^3$.
K_ξ	=	the gain matrix for translational control.
K_R	=	the gain matrix for rotational control.

2.3 Introduction

Multi-rotors are currently one of the most prevalent types of Unmanned Aerial Vehicle (UAV) platforms applied to various research, commercial, industrial, or hobbyist purposes. They are frequently deployed and utilized for applications in areas including aerial imaging [Wyp+14] and manipulation [RLO18]. Among all multi-rotors, although the vehicles with the configuration of 4 rotors, namely quadcopters, are advantageous due to its simplicity of design and manufacturing, the 6-rotor configuration drones, namely hexacopters, perform better in reliability due to the tolerance to the failure of one or two motors [GSG16], and generally with greater agility due to more thrust provided by 2 extra motors. Moreover, the standard quadcopters have to rely on rolling and pitching to generate lateral forces, result in an under-actuated system which cannot track an arbitrary flight path in its 6 degrees of freedom (DoF). Although some researchers have proposed canted-rotor quadcopter designs [NK14a][Bin+18], their involvement of extra actuators for extra DoFs can increase the battery consumption and control latency and complexity.

While hexacopters, on the other hand, can be rendered as a fully-actuated system if it is configured with canted rotor-axes[Ras+20a]. A standard hexacopter design align all its six propellers in parallel axes, thus maximizing the vertical thrust force and improving its redundancy and capability of carrying payload. However, this configuration can only produce an input force parallel to the vertical axis in its body frame. In order to exert lateral forces on the vehicle, the spinning speed of each propeller is controlled to generate roll and pitch torques to rotate the vehicle around X- and Y-axis in the body frame, respectively. The re-oriented vehicle body obtains lateral force components in the XY-plane of the inertial frame as discussed in [Her+15]. The standard multi-rotor configurations result in the vehicle dynamics coupling the pitch and role angles with both lateral and vertical force components in the inertial frame. In this case, the hexacopter's pitch and roll angles

cannot be controlled at will without affecting its translational motions, such that the lateral translations of the hexacopter are not directly controlled. In order to decouple hexacopter's lateral translations from pitching and rolling, some of the canted-rotor hexacopters are installed with extra actuators to realize fully actuated maneuvers[Kam+18], while others obtain full 6-DoF control authorities via specific rotor layout targeting omni-directional flight[BD16], or via deviating the rotor axes from the vertical axes by optimized canted and dihedral angles[Raj+15]. Moreover, although the coupled multicopter dynamics is proven to be deferentially flat thus making it capable of tracking smooth trajectories [MK11a], the hexacopter configuration with the ability to directly control its 6 DoFs is shown in Sec. §2.5 to perform better in path tracking than do the standard multicopters.

In this paper, the canted-rotor configuration of fixed-tilt hexacopters [Ras+20a] is defined and illustrated in Sec. §2.4.1, and each rotor axis defines the rotor's position and orientation in the body frame via a position vector and a rotation matrix, respectively. Theses rotor-to-body position vectors and rotation matrices are later involved in the vehicle's equations of motion in Sec. §2.4.4.

This paper presents the mathematical model of the canted-rotor hexacopters. The control designs are also formulated to realize path tracking control and unique tilt-hovering control maneuvers for canted-rotor hexacopter UAVs using the linear quadratic regulator (LQR) design technique. In Sec. §2.4, the canted rotor-axes and forces and control input are defined, and the UAV dynamics are derived using the Newton-Euler formalism [Ard06]. In Sec. §2.5, the hexacopter dynamics is linearized, and the tracking errors of translational and rotational DoFs are defined in order to design linear controllers using LQR method. In Sec. §2.6, the simulation results of both path tracking and tilt-hovering controllers are presented and evaluated. And In Sec. §2.7, the conclusions are reached that the simulated control designs live up to the authors' expection, and future work for this project is discussed.

2.4 System Modeling

In order to describe the dynamics of canted-rotor hexacopters, we define the Earth-fixed inertial frame $\mathcal{F}_E$: $\{\mathrm{O}_E\text{–}\mathrm{X}_E\mathrm{Y}_E\mathrm{Z}_E\}$, the body frame $\mathcal{F}_B$: $\{\mathrm{O}_B\text{–}\mathrm{X}_B\mathrm{Y}_B\mathrm{Z}_B\}$, and the local frame of the i-th rotor $\mathcal{F}_{m_i}$: $\{\mathrm{O}_{m_i}\text{–}\mathrm{X}_{m_i}\mathrm{Y}_{m_i}\mathrm{Z}_{m_i}\}$ with $i = 1 \ldots 6$ as Cartesian coordinate systems. The Earth-fixed inertial frame $\mathcal{F}_E$ denotes the inertial frame of reference fixed on the earth and its origin and three axes align with those of the body frame respectively before the vehicle takes off. The body frame $\mathcal{F}_B$ is fixed on the vehicle body and its origin coincides with the center of mass(CoM) of the vehicle. The three axes of body frame conform to the NED convention [CCL11b], with X-axis pointing out the nose of the vehicle, Z-axis pointing out the bottom of the vehicle, and Y-axis pointing to the right. The origin of each rotor's local frame is located at the center of this motor-propeller group, while its three axes are determined by its tilting angles as defined in Sec.§2.4.1. Once the coordinate systems are defined, the equations of motions of the hexacopter UAV are derived in Sec.§2.4.4.

2.4.1 Canted Rotor-Axes Definitions

The definition of the i-th rotor-axis orientation vector with $i = 1 \ldots 6$ is schematically demonstrated in Fig. 2.1.The body frame $\mathcal{F}_B$ is centered at point O_B, which is the center of mass of the vehicle. And the X_B-axis of $\mathcal{F}_B$ points out the hexacopter's nose, the Z_B-axis points out the bottom, and Y_B points out the right side. Point O_{m_i} is the center of mass of the motor-propeller group, which also defines the origin of the i-th local frame $\mathcal{F}_{m_i}$. Since O_{m_i} is defined to be on the XY-plane of $\mathcal{F}_B$, the length of the i-the rotor arm L_{arm} and the i-th rotor arm's yaw angle ψ_i are sufficient to locate the position of O_{m_i} in $\mathcal{F}_B$ as in Eq. 2.2. And the i-th rotor-axis vector before tilting V_i is defined by a unit vector of the coordinates $[0, 0, -1]^T$ in the body frame. At first, the thrust vector is rotated by an angle

ϕ_i around the first imaginary pivot axis which passes through point O_{m_i} and is parallel to X-axis of the body frame, which gives a vector $V_i' = [0, \sin\phi_i, -\cos\phi_i]^T$. Secondly, the rotor-axis vector is rotated by an angle θ_1 around the second imaginary pivot axis passing through point O_{m_i} and parallel to Y-axis of the body frame. The final orientation vector of i-th motor-propeller group is $V_i'' = [-\cos\phi_i\sin\theta_i, -\sin\phi_i, -\cos\phi_i\cos\theta_i]^T$.

The rotated rotor-axis orientation vector also helps define the three axes of the i-th local frame, with V_i'' pointing out the direction $[0, 0, -1]^T$ in the local frame, thus defining the Z_{m_i}-axis. The X_{m_i}-axis of i-th local frame is perpendicular to both V_i'' and the rotation pivot of θ_i. Since the θ_i pivot is parallel to Y_B-axis of the body frame, we can calculate the cross product between V_i'' and Y-axis unit vector to find the unit vector indicating the X-axis of i-th local frame to be $[\cos\theta_i, 0, -\sin\theta_i]^T$ in the body frame. And the Y-axis unit vector in i-th local frame is calculated by taking the cross product between Z_{m_i}- and Y_{m_i}-axis unit vector, rendered as $[-\sin\phi_i\sin\theta_i, \cos\phi_i, -\sin\phi_i\cos\theta_i]^T$ in the body frame.

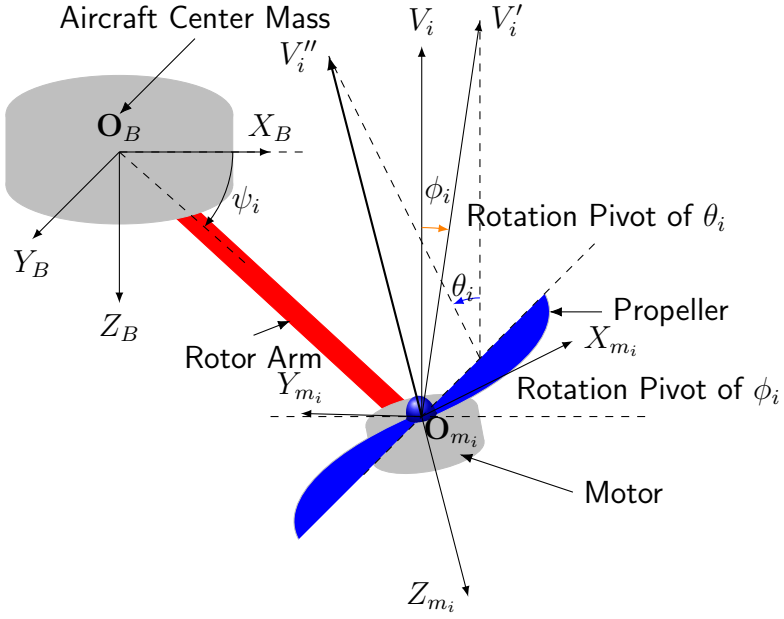


Figure 2.1. The illustration of canted rotor-axes.

After defining the rotor-axis orientation vectors, each rotor-axis can be characterized by a pair of rotation angles (ϕ_i, θ_i). An illustration of a rotor-axis layout with defined orientation vectors V_i in the body frame is presented in Fig. 2.2. The rotor-axis notation numbers 1 through 6 are defined along the counter-clockwise sequence when looking from above (negative Z-direction in the body frame).

In the equations of motion that are introduced in Sec.§2.4.4, the characterizations of the canted-rotor axes are needed in matrix form to describe the directions of force and moment generated by each motor-propeller group. Thus, the rotation matrix $\mathbf{R}_{Bm_i}^{[3\times3]}$ with $i = 1 \dots 6$ is formulated such that the force and moment in the local frame of i-th rotor can be transformed into the body frame:

$$\mathbf{R}_{Bm_i} = \begin{bmatrix} \cos\theta_i & \sin\phi_i \sin\theta_i & \cos\phi_i \sin\theta_i \\ 0 & -\cos\phi_i & \sin\phi_i \\ -\sin\theta_i & \cos\theta_i \sin\phi_i & \cos\phi_i \cos\theta_i \end{bmatrix} \tag{2.1}$$

Due to the fact that each rotor-axis has 2 rotational degrees of freedom based on the definition above, all 6 rotor axes can introduce a total of 12 degrees of freedom which increases the complexity of the hexacopter layout definition. For simplification, the symmetry about the X-Z plane of the body frame is adopted such that all the rotor-axis orientations can be described by two 3-tuples $\mathbf{\Phi} = (\phi_1, \phi_2, \phi_3)$ and $\mathbf{\Theta} = (\theta_1, \theta_2, \theta_3)$ of the rotation angles around their rotation pivots:

$$\phi_1 = -\phi_6, \phi_2 = -\phi_5, \phi_3 = -\phi_4;$$
$$\theta_1 = \theta_6, \theta_2 = \theta_5, \theta_3 = \theta_4.$$

In addition, we also need to locate the positions of i-th rotor center O_{m_i} with $i = 1 \dots 6$ in the body frame $\mathcal{F}_B$ for dynamic modelling. The position of each O_{m_i} in $\mathcal{F}_B$

is described by a vector $\mathbf{p}_{Bm_i}$. For simplicity, we set all rotor arms to be equally long and all O_{m_i} to be coplanar with the origin of the body frame O_B, thus leaving 2 degrees of freedom in $\mathbf{p}_{Bm_i}$. In order to determine $\mathbf{p}_{Bm_i}$, we need the rotor arm length L_{arm} and the i-th rotor arm's yaw angle ψ_i in X-Y plane of $\mathcal{F}_B$ as illustrated in Fig. 2.1. For a layout of normal hexagonal rotor positions, the i-th rotor arm's yaw angle is $\psi_i = \pi/6 + (i-1) \cdot \pi/3$ with respect to the body X-axis. And the i-th rotor's position vector $\mathbf{p}_{Bm_i}$ in $\mathcal{F}_B$ can be calculated via:

$$\mathbf{p}_{Bm_i} = \begin{bmatrix} \cos\psi_i & -\sin\psi_i & 0 \\ \sin\psi_i & \cos\psi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} L_{arm} \\ 0 \\ 0 \end{bmatrix} \tag{2.2}$$

2.4.2 Transformation Matrices

The transformation matrix for linear positions, velocities, and accelerations components from the hexacopter's body frame to the Earth-fixed inertial frame needs to be derived based on the rotation angles about the North, East, and Down directions of the inertial frame. Given that the NED convention of the inertial frame corresponds to the real-world North, East, and Down directions, and the body frame is defined by pointing out the nose, the right side, and the bottom of the vehicle body, the body-to-inertial transformation matrix $\mathbf{R}_{EB}$ can be represented by:

$$\mathbf{R}_{EB} = \begin{bmatrix} c\theta c\psi & s\theta s\phi c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ s\psi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\phi c\theta \end{bmatrix} \tag{2.3}$$

where "c" and "s" represent "cos()" and "sin()" in Eq. (2.3), respectively, and θ, ϕ, ψ are the pitch, roll, and yaw body rotation angles between the body frame's North, East, Down axes and those of the frame, respectively. Similarly, the rotation angles, angular rates and accelerations measured in the body frame need to be transformed to the inertial frame by another transformation matrix:

$$\mathbf{T}_{EB} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix} \tag{2.4}$$

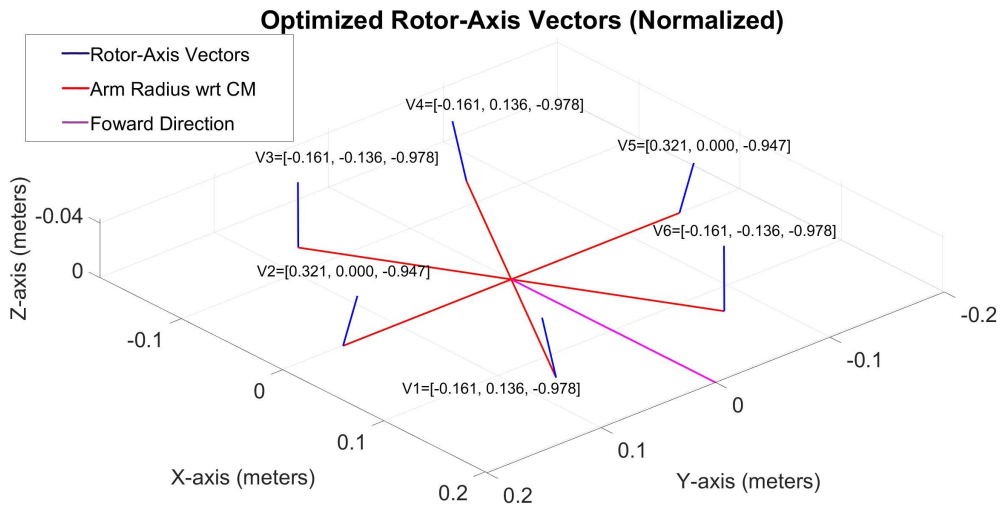


Figure 2.2. An example of canted rotor-axes layout in the body frame.

2.4.3 Forces and Control Input

Next, the thrust forces generated by i-th motor-propeller group, where $i = 1 \ldots 6$, is derived. For each rotor, the propulsive force produced by the spinning of its propeller is

approximated by a quadratic equation of the angular speed of the propeller:

$$T_i = k_f \cdot \omega_i^2 \tag{2.5}$$

where k_f is a coefficient related to the area of the rotor disk and air density, which is regarded as a constant in our scenario. ω_i is the angular speed of the i-th propeller. In a thrust testing experiment of the selected motor-propeller group in this paper, the measured thrusts of a rotor fit well with a second-order equation the angular speed where k_f estimated to be $9.23 \times 10^{-7}\ N \cdot s^2/rad^2$, thus verifying Eq. (2.5).

Each rotor also generates a moment due to the propeller's drag, which acts in the opposite direction of the propeller angular speed. This moment of i-th motor-propeller group can be approximated by:

$$M_i = -k_m \cdot \omega_i |\omega_i| \tag{2.6}$$

where k_m is the drag moment coefficient, which calculated via an equation based on the blade element theory incorporating air density, angle of resultant flow, propeller chord length, total blade area, and propeller drag coefficient[GD17]. In case of the selected motor-propeller assembly in this paper, k_m is calculated to be $2.73 \times 10^{-8}\ Nm \cdot s^2/rad^2$. In order to determine the signs of ω_i with $i = 1 \dots 6$, we use a typical hexacopter rotation pattern, with 1 and -1 indicating the counter-clockwise and clockwise rotation direction with respect to the Z-axis of the i-th local frame $\mathcal{F}_{m_i}$:

$$\mathbf{sgn}(\omega_i) = (-1)^i,\ i = 1 \dots 6 \tag{2.7}$$

In the i-th local frame $\mathcal{F}_{m_i}$, both vectors of $\mathbf{F}_{m_i} \in$

$$\mathbf{F}_{m_i} = \begin{bmatrix} 0 \\ 0 \\ -k_f \omega_i^2 \end{bmatrix}, \mathbf{M}_{m_i} = \begin{bmatrix} 0 \\ 0 \\ (-1)^{i+1} k_m \omega_i^2 \end{bmatrix} \tag{2.8}$$

We notice that both $\mathbf{F}_{m_i}$ and $\mathbf{M}_{m_i}$ are proportional to the square of the spinning speed i-th propeller ω_i, thus the intermediate force vector $\Omega \in \mathbb{R}^6$ of the system is defined as:

$$\Omega = [\omega_1^2 \ \omega_2^2 \ \omega_3^2 \ \omega_4^2 \ \omega_5^2 \ \omega_6^2]^T \tag{2.9}$$

The brushless DC motor dynamics is set to be controlled by a electronic speed controller (ESC) via pulse width modulation (PWM) duty cycles. The ESC receives PWM signals from 1000 μs to 2000 μs in each 20 ms cycle to control the voltage input to the motor, thus controlling the angular speed. In another thrust testing experiment conducted by authors, the selected rotor group is investigated to collect data point of PWM signals and ω^2 at various steady states. We found for the selected motor-propeller assembly, ω^2 fits well in a second-order polynomial of the PWM signals as:

$$\omega^2 = a \cdot PWM^2 + b \cdot PWM + c \tag{2.10}$$

where ω is in the unit rad/s and PWM is in μs, a, b, and c are polynomial coefficients, while in the 2 case studies in this paper, the motor and propeller are selected to be Lumenier RX2206-11 2350kv brushless motor and Lumenier 5x4.5" 2-blade propeller, respectively, and their coefficients are found to be 3.5191, -4.2585×10^3, and 8.0159×10^5, respectively. However, the brushless motors are identified to be first-order filter in its response to PWM signals[Hon+07], such that we assume this motor-propeller group is

also a first-order dynamic system when responding to the PWM inputs which adjusts the input voltage. The corresponding first-order transfer function is modelled as:

$$G(s) = \frac{\omega(s)}{PWM(s)} = \frac{K}{\tau_0 s + 1} \tag{2.11}$$

Setting $K = 1$ and estimating $\tau_0 = 0.2\ s$ from experimental data, we map the intermediate force vector Ω to the PWM input of each motor by:

$$\Omega(PWM_i, t) = \begin{bmatrix} \omega^2_{1,t_{0_1}} + (1 - e^{-0.2(t-t_{0_1})}) \cdot (\omega^2_{1,t} - \omega^2_{1,t_{0_1}}) \\ \omega^2_{2,t_{0_2}} + (1 - e^{-0.2(t-t_{0_2})}) \cdot (\omega^2_{2,t} - \omega^2_{2,t_{0_2}}) \\ \vdots \\ \omega^2_{6,t_{0_6}} + (1 - e^{-0.2(t-t_{0_6})}) \cdot (\omega^2_{6,t} - \omega^2_{6,t_{0_6}}) \end{bmatrix} \tag{2.12}$$

where t_{0_i} is the moment of receiving the latest PWM command in i-th channel.

2.4.4 Newton-Euler Equations of Motion

Translational Dynamics

We express the translational dynamics of the canted-rotor hexacopter model in the inertial frame $\mathcal{F}_E$, using the standard Newton-Euler equation:

$$m\ddot{\xi} = m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \mathbf{R}_{EB} \sum_{i=1}^{6} \mathbf{R}_{Bm_i} \mathbf{F}_{m_i} + \mathbf{F}_{ext} \tag{2.13}$$

where m is the mass of the aircraft, $\xi \in \mathbb{R}^3$ is the hexacopter's linear position $[x_E, y_E, Z_E]^T$ in the inertial frame $\mathcal{F}_E$, g is the gravitational acceleration, and $\mathbf{F}_{ext} \in \mathbb{R}^3$ is the external disturbance forces. In our case, $\mathbf{F}_{ext}$ consists of the in-flight drag force acting in the opposite direction of the linear velocities, and unmodelled aerodynamic and structural

effects as:

$$\mathbf{F}_{ext} = \mathbf{R}_{EB} \begin{bmatrix} -C_{D_x} \cdot \dot{x}_B |\dot{x}_B| \\ -C_{D_y} \cdot \dot{y}_B |\dot{y}_B| \\ -C_{D_z} \cdot \dot{z}_B |\dot{z}_B| \end{bmatrix} + \text{unmodelled effects} \tag{2.14}$$

where C_{D_x}, C_{D_y}, C_{D_z} are the coefficients to calculate drag forces in the body frame $\mathcal{F}_B$ as modelled in [SK17], $\dot{x_B}$, $\dot{y_B}$, $\dot{z_B}$ are the velocity components in $\mathcal{F}_B$.

Rotational Dynamics

Adding up the inertial term, the torques produced by rotor thrusts and drag moments, and the external disturbances on the right-hand side, we can formulate the rotational dynamics of the hexacopter in the body frame $\mathcal{F}_B$ using the Newton-Euler equation of motion:

$$\mathbf{I_B}\dot{\eta}_B = -\eta_B \times \mathbf{I}_B\eta_B + \sum_{i=1}^{6} \mathbf{p}_{Bm_i} \times \mathbf{R}_{Bm_i}\mathbf{F}_{m_i} + \sum_{i=1}^{6} \mathbf{R}_{Bm_i}\mathbf{M}_{m_i} + \tau_{ext} \tag{2.15}$$

where $\mathbf{I}_B \in \mathbb{R}^{3\times3}$ is the hexacopter inertia matrix in $\mathcal{F}_B$, $\eta_B \in \mathbb{R}^3$ is the the angular rates $[p, q, r]^T$ in $\mathcal{F}_B$, and $\tau_{ext} \in \mathbb{R}^3$ accounts for the external disturbances torques. The rotational rates and accelerations in $\mathcal{F}_B$ can be transformed to those in $\mathcal{F}_E$ by left multiplying $\mathbf{T}_{EB}$ in Eq. (2.4):

$$\dot{\eta}_E = \mathbf{T}_{EB} \cdot \dot{\eta}_B,\ \eta_E = \mathbf{T}_{EB} \cdot \eta_B \tag{2.16}$$

This transformation equation is applied primarily in the tilt-hovering control as discussed in Sec.§2.5.5, but can be neglected when applying path tracking control since the roll and pitch angles remain close to zero making $\mathbf{T}_{EB}$ nearly identity matrix.

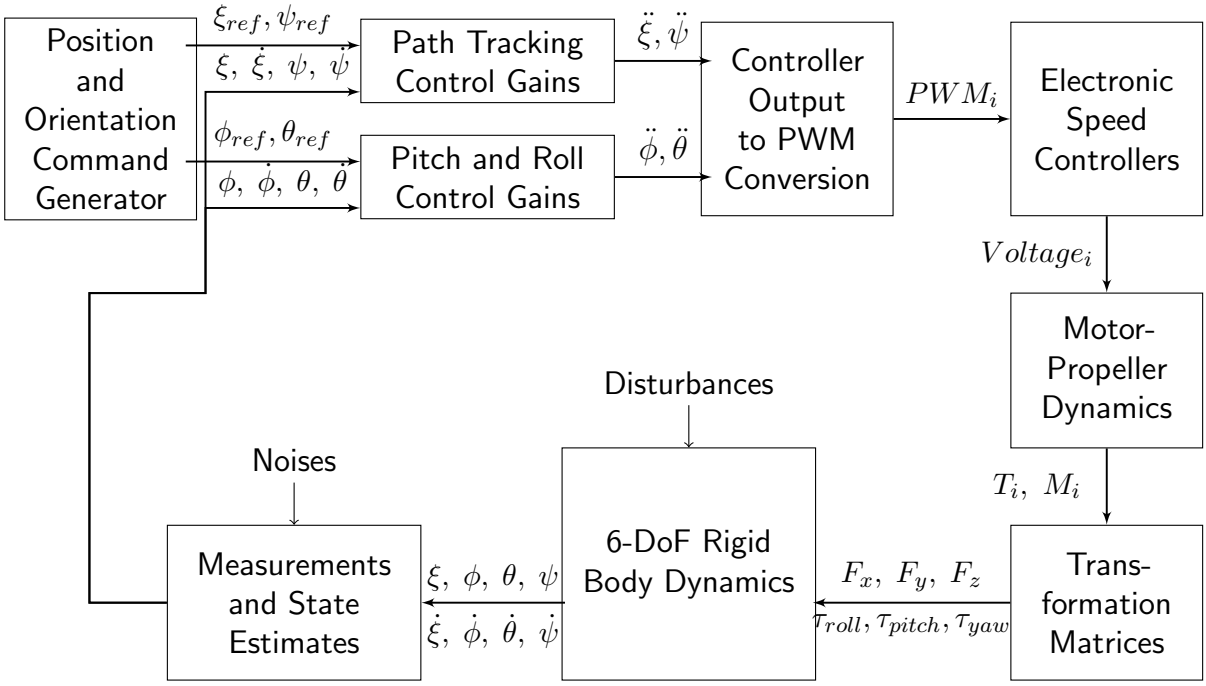


Figure 2.3. Hexacopter control scheme architecture.

2.5 Control Design

In this section, the control strategies of the both path tracking controller and tilt-hovering controller are described. The overall block diagram of the feedback control system is shown in Fig. 2.3. The purpose of control designs here is to decouple the translational and rotational dynamics of the hexacopter, and to ensure that the hexacopter can follow a desired path of its translational states without performing pitch and roll rotations in the "path tracking" mode, as well as tracking a controlled purely rotational trajectory of vehicles' states while keeping its center-of-mass stationary in the inertial frame when switched to the "tilt-hovering" mode.

2.5.1 Model Linearization

In order to rapidly compute the desired control input, the system model is linearized between the output states $\ddot{\xi}$, $\dot{\eta}_B$ and intermediate force vector Ω which further maps to the

PWM commands as system input. First, since both $\mathbf{F}_{m_i}$ and $\mathbf{M}_{m_i}$ are linearly dependent on ω_i, we can formulate the linear terms in Eq. (4.9) and (4.10) as:

$$\begin{aligned}\mathbf{B}_F\Omega(PWM_i,t) &= \frac{1}{m}\mathbf{R}_{EB}\sum_{i=1}^{6}\mathbf{R}_{Bm_i}\mathbf{F}_{m_i} \\ \mathbf{B}_\tau\Omega(PWM_i,t) &= \mathbf{I}_B^{-1}(\sum_{i=1}^{6}\mathbf{p}_{Bm_i}\times\mathbf{R}_{Bm_i}\mathbf{F}_{m_i}+\sum_{i=1}^{6}\mathbf{R}_{Bm_i}\mathbf{M}_{m_i})\end{aligned} \tag{2.17}$$

where $\mathbf{B}_F \in \mathbb{R}^{3\times6}$ and $\mathbf{B}_\tau \in \mathbb{R}^{3\times6}$ are linear mapping matrices for $\mathbf{\Omega}$. In this case, we ignore the non-linear disturbances terms in (4.9) and (4.10), combine overall linear and rotational accelartions to vector $\mathbf{a} \in \mathbb{R}^6$, and view gravitational and inertial-related angular acceleration terms as generalized internal forces and formulate them as vector $\mathbf{q} \in \mathbb{R}^6$:

$$\mathbf{a} = \begin{bmatrix}\ddot{\xi}_{[3\times1]} \\ \dot{\eta}_{B[3\times1]}\end{bmatrix}, \mathbf{q} = \begin{bmatrix}0 \\ 0 \\ g \\ -\mathbf{I}_B^{-1}(\eta_B\times\mathbf{I}_B\eta_B)_{[3\times1]}\end{bmatrix} \tag{2.18}$$

thus the dynamic equations in (4.9) and (4.10) can be linearized as:

$$\mathbf{a}-\mathbf{q} = \mathbf{B}\Omega \tag{2.19}$$

where $\mathbf{B} = [\mathbf{B}_F\ \mathbf{B}_\tau]^T \in \mathbb{R}^{6\times6}$. When $\mathbf{B}$ is of full rank, Ω can be calculate by left multiplying $\mathbf{B}^{-1}$ on both sides. Even if $\mathbf{B}$ is rank-deficient, Ω could still be solved using Moore-Penrose pseudo-inverse. Once the desired Ω is found, the desired PWM command for each motor can be mapped using (2.10), and the saturation limits of each motor shall be met as well.

2.5.2 Tracking Error Dynamics

The proportional-integral-derivative (PID) feedback control strategy is implemented to track the reference signals and stabilize the hexacopter states. Since the lateral body force components in $\sum_{i=1}^{6} \mathbf{R}_{Bm_i}\mathbf{F}_{m_i}$ can be decoupled from the pitch and roll angles, we can discard the traditional multicopter control architecture in which the attitude control serves as lower-level controller. Instead, the attitude controller and position controller can track uncorrelated reference signals. In this case, tracking errors in all 6 DoFs are needed in order to design the state feedback controllers. The position, linear velocity and acceleration errors of the hexacopter in the inertial frame $\mathcal{F}_E$ are:

$$\begin{aligned} &e_x = x - x_{ref}, \dot{e}_x = \dot{x} - \dot{x}_{ref}, \ddot{e}_x = \ddot{x} - \ddot{x}_{ref} \\ &e_y = y - y_{ref}, \dot{e}_y = \dot{y} - \dot{y}_{ref}, \ddot{e}_y = \ddot{y} - \ddot{y}_{ref} \\ &e_z = z - z_{ref},\ \dot{e}_z = \dot{z} - \dot{z}_{ref}, \ddot{e}_z = \ddot{z} - \ddot{z}_{ref} \end{aligned} \tag{2.20}$$

thus we can formulate a position error state vector $\mathbf{e}_\xi$ and introduce the error input vector $\mathbf{u}_\xi$ in terms of the second-order error as:

$$\mathbf{e}_\xi = \begin{bmatrix} e_x & \dot{e}_x & \int_{t_0}^{t} e_x dt & e_y & \dot{e}_y & \int_{t_0}^{t} e_y dt & e_z & \dot{e}_z & \int_{t_0}^{t} e_z dt \end{bmatrix}^T,\ \mathbf{u}_\xi = \begin{bmatrix} \ddot{e}_x & \ddot{e}_y & \ddot{e}_z \end{bmatrix}^T, \tag{2.21}$$

such that the dynamics of position tracking error in $\mathcal{F}_E$ is formulated in the similar pattern of state-space equation as in [Her+15]:

$$\dot{\mathbf{e}}_\xi = A_\xi \mathbf{e}_\xi + B_\xi \mathbf{u}_\xi, \tag{2.22}$$

where

$$A_\xi = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, B_\xi = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.23}$$

In this scenario, the controllability matrix $\mathcal{C}(A_\xi, B_\xi)$ is of full rank, such that (A_ξ, B_ξ) is controllable. Similarly, an orientation error state vector can be formulated as $\mathbf{e}_R = [e_\phi,\ \dot{e}_\phi,\ \int_{t_0}^{t} e_\phi dt,$
$e_\theta,\ \dot{e}_\theta,\ \int_{t_0}^{t} e_\theta dt,\ e_\psi,\ \dot{e}_\psi,\ \int_{t_0}^{t} e_\psi dt]^T$, the rotational error vector can be given as $\mathbf{u}_R = [\ddot{e}_\phi,\ \ddot{e}_\theta,\ \ddot{e}_\psi]^T$, and the dynamics of orientation tracking error is rendered as:

$$\dot{\mathbf{e}}_R = A_R \mathbf{e}_R + B_R \mathbf{u}_R, \tag{2.24}$$

with $A_R = A_\xi$, $B_R = B_\xi$.

2.5.3 Linear Quadratic Control Design

In order to design a state feedback controller $\mathbf{u} = -K\mathbf{e}$ where $K \in \mathbb{R}^{3\times 9}$ to stabilize the error dynamics and converge fast, we implement the optimal control design approach of Linear Quadratic Regulator [Bem+02]. In order to minimize a cost function

$$J = \int_0^\infty \mathbf{e}^T Q \mathbf{e} + \mathbf{u}^T R \mathbf{u} dt, \tag{2.25}$$

with Q and R selected based on control requirements, the gain matrix K is determined by $K = R^{-1}B^T P$, where P matrix is the solution to the continuous-time algebraic Ricatti equation

$$A^T P + PA - PBR^{-1}B^T P + Q = 0. \tag{2.26}$$

Applying LQR to our PID control scheme, the obtained gain matrix K includes the gain for each of the proportional, integral, and derivative tracking errors in each of the 3 DoFs. Take the translational 3 DoFs as an example, the gain matrix can be written as:

$$K_\xi = \begin{bmatrix} k_{p,x} & k_{d,x} & k_{i,x} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{p,y} & k_{d,y} & k_{i,y} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_{p,z} & k_{d,z} & k_{i,z} \end{bmatrix} \tag{2.27}$$

such that the second-order state vector to be controlled is given as:

$$\ddot{\xi} = \begin{bmatrix} \ddot{x}_{ref} - k_{p,x}e_x - k_{d,x}\dot{e}_x - k_{i,x}\int_{t_0}^{t} e_x dt \\ \ddot{y}_{ref} - k_{p,y}e_y - k_{d,y}\dot{e}_y - k_{i,y}\int_{t_0}^{t} e_y dt \\ \ddot{z}_{ref} - k_{p,z}e_z - k_{d,z}\dot{e}_z - k_{i,z}\int_{t_0}^{t} e_z dt \end{bmatrix} \tag{2.28}$$

Similarly, $\dot{\eta}_B$ can also be controlled using gains obtained from LQR design approach as well, and these control signals can be used to solve for Ω vector as in Eq. (2.19), and Ω can be mapped to discretized PWM commands to control 6 rotors.

2.5.4 Path Tracking Controller

In order to track the 3-dimensional position trajectories, the controllers need to possess enough authority and control bandwith to generate flat output states tracking closely reference signals in X, Y, Z and yaw directions in $\mathcal{F}_E$ [MK11a], while the pitch

and yaw controller only serve to track zero setpoints and resist disturbance torque. For the path tracking case study in this paper, whose Φ and Θ-tuple are $(0.1364, 0, -0.1364)$ and $(0.1627, -0.3268, -0.1627)$, respectively, and $L_{arm} = 0.207\ m$, the Q_ξ and R_ξ matrices are selected to be

$$Q_\xi = diag([200,\ 2000,\ 10,\ 200,\ 2000,\ 10,\ 4000,\ 200,\ 5]^T),\ R_\xi = 2 \cdot I_3, \tag{2.29}$$

which renders the gain matrix K_ξ for translational 3 degrees of freedom to be

$$K_\xi = \begin{bmatrix} 15.61 & 32.11 & 2.24 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15.61 & 32.11 & 2.24 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 45.21 & 13.80 & 1.58 \end{bmatrix}. \tag{2.30}$$

while the Q_R and R_R matrices for the orientation control are

$$Q_R = diag([100,\ 2000,\ 10,\ 100,\ 2000,\ 10,\ 1000,\ 2 \times 10^5,\ 100]^T),\ R_R = 2 \cdot I_3, \tag{2.31}$$

which renders the gain matrix K_R for rotational 3 degrees of freedom to be

$$K_R = \begin{bmatrix} 13.91 & 32.06 & 2.24 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 13.91 & 32.06 & 2.24 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 30.80 & 401.23 & 7.07 \end{bmatrix} \tag{2.32}$$

2.5.5 Tilt-Hover Controller

The concept of tilt-hover controller is based on the expectation to hover the hexacopter at a certain altitude and rotate it to a pose with certain pitch or roll angle, while the its center of mass can remain stationary because of the special configuration of the canted-rotor hexacopter. This objective requires the hexacopter to have special

configuration of rotor-axes canted to adequate amount according to the desired rotation angle, and its controller to have enough control authority and bandwith in pitch and roll rotational direction. For the tilt-hover control case study in this paper, whose Φ and Θ-tuple are $(0.4115, 0, -0.4115)$ and $(0.3338, -0.6971, -0.3338)$, respectively, and $L_{arm} = 0.207\ m$, the Q_ξ and R_ξ matrices are selected to be

$$Q_\xi = diag([50,\ 200,\ 1,\ 50,\ 200,\ 1,\ 4000,\ 200,\ 5]^T),\ R_\xi = 2 \cdot I_3, \tag{2.33}$$

which renders the translational gain matrix K_ξ:

$$K_\xi = \begin{bmatrix} 6.33 & 10.61 & 0.71 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6.3253 & 10.61 & 0.71 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 45.21 & 13.80 & 1.58 \end{bmatrix} \tag{2.34}$$

While the Q_R and R_R matrices for the orientation control in tilt-hovering mode are selected as

$$Q_R = diag([1 \times 10^5,\ 2 \times 10^6,\ 100,\ 1 \times 10^5,\ 2 \times 10^6,\ 100,\ 1000,\ 2 \times 10^4,\ 10]^T),\ R_R = 2 \cdot I_3, \tag{2.35}$$

which produce the rotational gain matrix K_R for tilt-hovering:

$$K_R = \begin{bmatrix} 253.27 & 1000.25 & 7.07 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 253.27 & 1000.25 & 7.07 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 30.80 & 100.31 & 2.24 \end{bmatrix} \tag{2.36}$$

2.6 Simulation Results

2.6.1 Path Tracking Simulations

The validation of the path tracking controller designed in Sec. §2.5.4 is conducted with a simulation tracking a $50m \times 50m$ square path and another simulation tracking

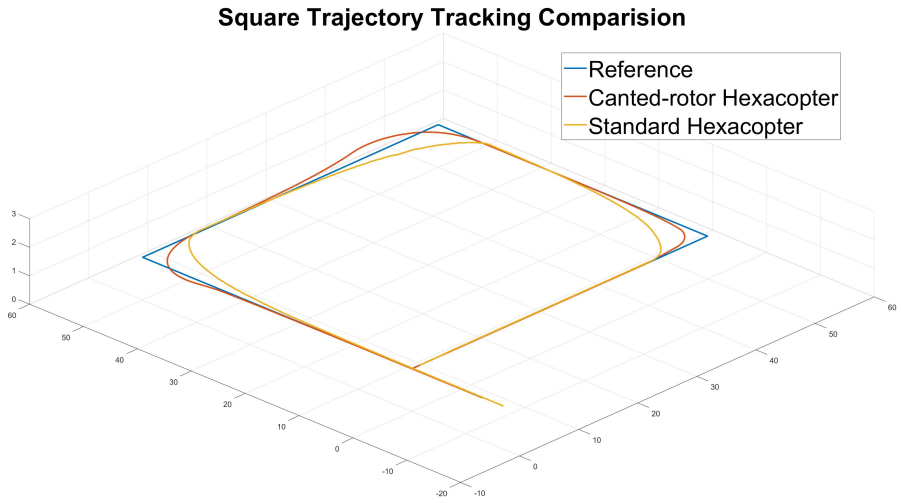


Figure 2.4. Tracking of square path.

a circular path of $1.5m$ radius at $1.5m$ altitude above ground ($z_E = -1.5m$ in NED coordinates). As shown in Fig. 2.4, the square path consists of waypoints in a sequence of $[0, 0, -1.5]^T$, $[50, 0, -1.5]^T$, $[50, 50, -1.5]^T$, $[0, 50, -1.5]^T$, $[0, 0, -1.5]^T$. And the circular path with its center located at $[-0.25, 0, -1.5]^T$ is shown in Fig. 2.5. For the path tracking case study in this paper, the canted-rotor hexacopter to be simulated with its Φ and Θ-tuple to be $(0.1364, 0, -0.1364)$ and $(0.1627, -0.3268, -0.1627)$, respectively, and $L_{arm} = 0.207\ m$. The mass of the canted-rotor hexacopter is $m = 0.754kg$, and the moments of inertia $\mathbf{I}_B = diag([5.5 \times 10^{-3}, 5.8 \times 10^{-3}, 1.09 \times 10^{-2}]^T)kg \cdot m^2$. The standard hexacopter is of the same size, mass, moments of inertia, and thrust range of the canted-rotor one.

In the square path simulation, the desired path is marked by $4,000$ look-ahead points [OR05] with a look-ahead distance of $10m$ to avoid significant overshoots. For both canted-rotor and standard hexacopters, the maximum velocity is set to be $5m/s$ and the time scope of both simulations is 80 seconds. And the $\mathbf{F}_{ext}$ is modelled according to Eq. 2.14 with each C_{D_i} selected as 0.1. As shown in Fig. 2.4, the canted-rotor hexacopter

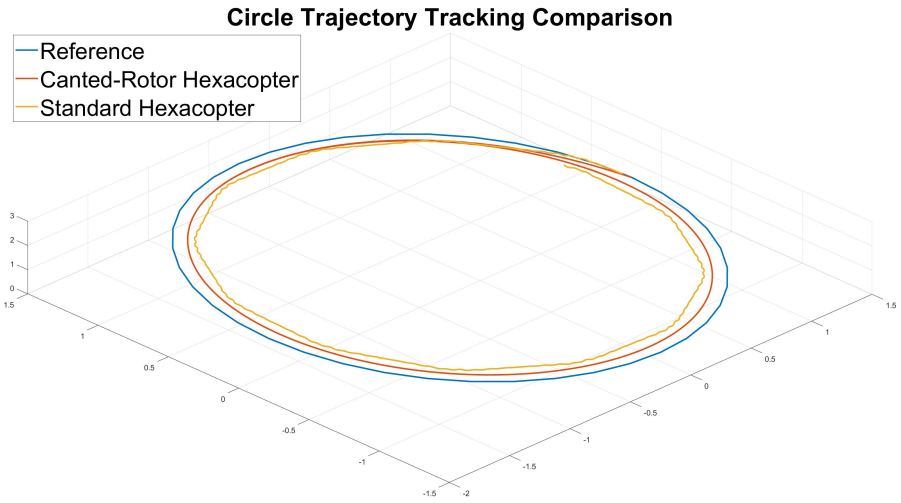


Figure 2.5. Tracking of circular path.

outperforms the standard one which has the identical mass, moments of inertia, and thrust range of the canted-rotor one, and also incorporates carefully tuned PID control gains. The trajectory of canted-rotor hexcopter has an average cross-track error of $0.266m$ with respect to the entire path, while the standard hexacopter has an average cross-track error of $2.238m$.

In the circular path simulation, the desired path is marked by 100 look-ahead points with a look-ahead distance of $0.5m$. For both canted-rotor and standard hexacopters, the maximum velocity is set to be $1m/s$ and the time scope of both simulations is 40 seconds. As shown in Fig. 2.5, the canted-rotor hexacopter also outperforms the standard one with smoother trajectory and lower cross-track errors. The canted-rotor hexcopter's average cross-track error is $4.2 \times 10^{-2}m$ along the path, and the standard one has an average cross-track error of $7.1 \times 10^{-2}m$. The oscillations on the trajectory of the standard hexacopter might be caused by the fact that it needs to frequently alternate the pitch and roll angles to output relatively small linear accelerations along the path.

2.6.2 Tilt-Hover Simulations

In order to test the tilt-hovering controller, the case study hexacopter is set to have $\Phi = (0.4115, 0, -0.4115)$ and $\Theta = (0.3338, -0.6971, -0.3338)$, $m = 0.754kg$, $\mathbf{I}_B = diag([5.5 \times 10^{-3}, 5.8 \times 10^{-3}, 1.09 \times 10^{-2}]^T)kg \cdot m^2$, and $L_{arm} = 0.207\ m$. The simulation rotates the drone by a certain pitch or roll angle while hovering its center of mass stationary at $[0, 0, -1.5]^T$ in $\mathcal{F}_E$. Fig. 2.6 shows the step response plot of $\phi_{ref} = -0.3rad$ with respect to $\mathcal{F}_E$, and Fig. 2.7 illustrates the simulation results of $\theta_{ref} = 0.3rad$, both are shown to reach steady state within 5 seconds. Although the hexacopter successfully hovers at the origin of $\mathcal{F}_E$ with tilted orientations, the oscillations appear to be considerable. In Fig.2.7, the hexacopter first overshoots beyond $-0.2rad$ then oscillates till stabilizing at $0.3rad$. This might be caused by relatively large proportional gains compared to derivative gains in the pitch direction.

2.7 Chapter Conclusion

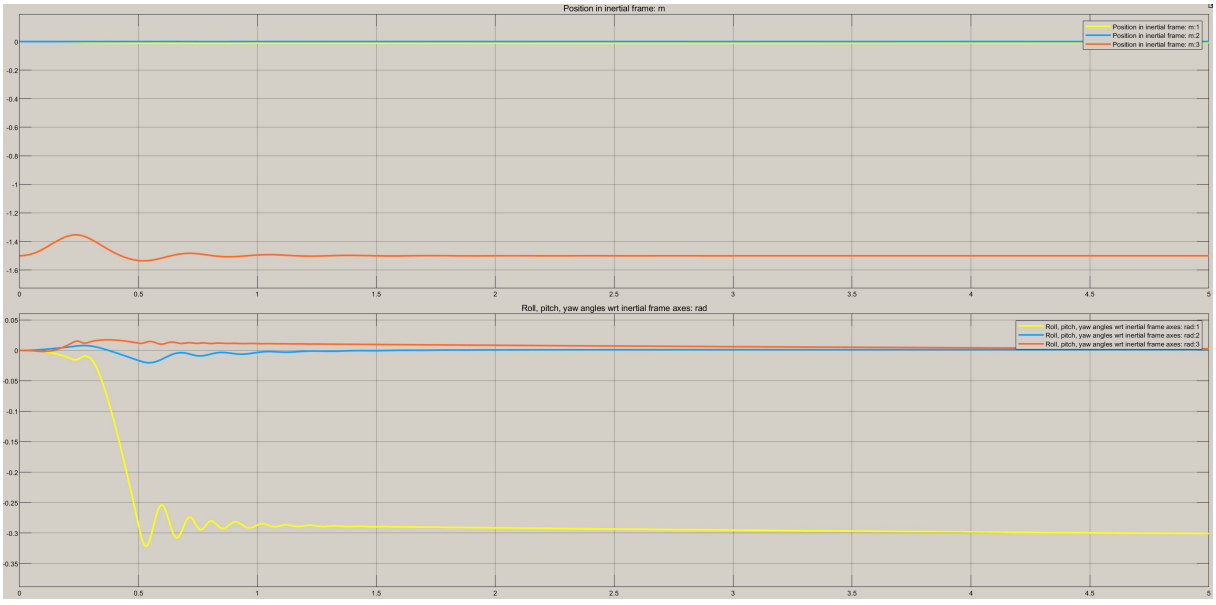


Figure 2.6. Tilt-hover simulation with $\phi_{ref} = -0.3rad$.

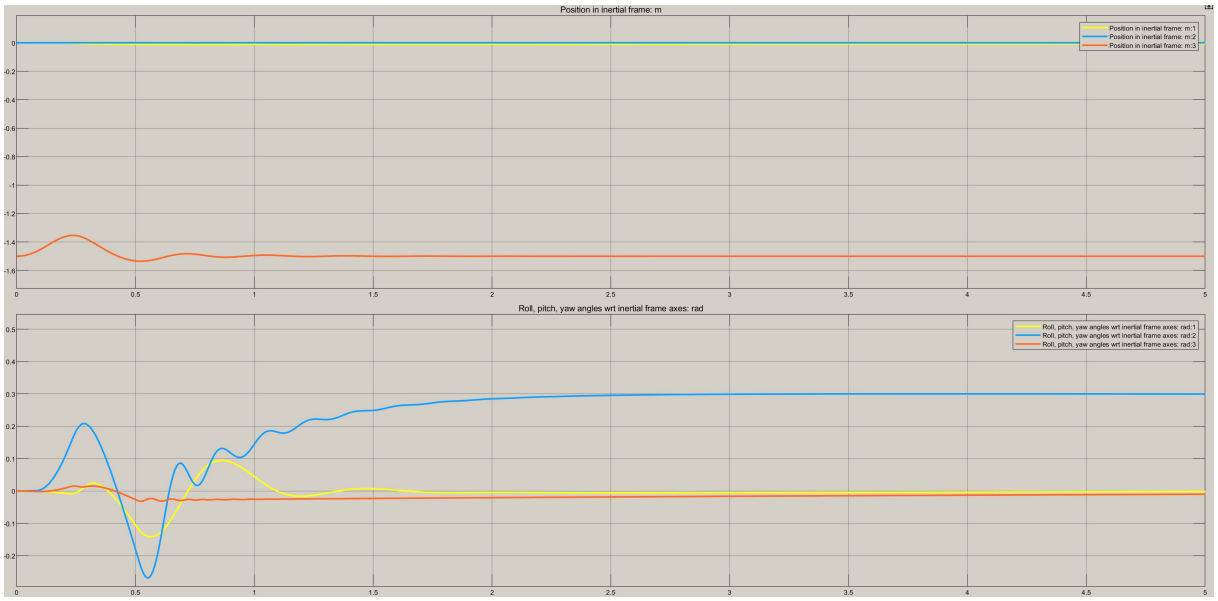


Figure 2.7. Tilt-hover simulation with $\theta_{ref} = 0.3rad$.

This paper demonstrates the mathematical model of canted-rotor hexacopter in order to enhance the control maneuverability of conventional multi-rotor systems. The system dynamics is then linearized and the tracking error dynamics is formulated to implement the linear quadratic control design technique, based on which the path tracking and tilt-hover controllers are designed and fine-tuned. The path tracking controller decouples the lateral movements from the pitch and roll rotations of the vehicle, and the tilt-hover controller stabilizes the position of the center of mass of the vehicle as well as its heading while it is rotated by a certain pitch or roll angle, decoupling pitch and roll from lateral translations as well. Numerical simulations are then performed to corroborate the mathematical model and control designs. The path tracking controller of the canted-rotor hexacopter performs better than do the linear controller of standard hexacopter as presented in Sec. §2.6.1. And the tilt-hover controller lives up to the expected performance as shown in Sec. §2.6.2.

Despite the satisfactory simulation results, the proposed model and controllers

will need to be implemented on real-world hexacopter UAV systems with canted-rotors to be further validated in the future work. The sim-to-real gap between simulation and real-world implementation remains challenging to be resolved. The future work will also involve the path tracking and tilt-hovering experiments using data from onboard sensor and motion capture system, as well as a series of maneuvers to explore the functionality of canted-rotor hexacopters.

2.8 Acknowledgements

This chapter, in full, is a reprint of the material as it appears, titled "Decoupled translational and rotational flight control designs of canted-rotor hexacopters", in *Proceedings of AIAA Scitech 2021 Forum*. Cao, Pengcheng; Strawson, James; Bewley, Thomas; Kuester, Falko. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Mission-oriented Path Planning and Trajectory Optimization for Multi-rotor UAVs

3.1 Chapter Abstract

Planning Search-and-Rescue (SAR) missions for UAVs in cluttered and GPS-impaired environments remains a challenging topic in both robotics and aerospace related research areas. Small multirotors, especially quadrotors, are deployed by many in indoor or confined spaces due to their structural simplicity, ability to hover and take off and land vertically, and good maneuverability. However, most of small multirotors suffer from less-than-30-min flight time on one charge imposing constraints on missions including searching for survivors and payload delivery. In this paper, we introduce a mission-oriented trajectory generation approach in order to improve the quantitative performance of multirotors by reducing single-flight energy and time consumption. In our methodology, each flight mission is initially planned using computationally cheap path planning algorithms to generate a set of waypoints as an initial guess of the trajectory. Next, a multi-phase optimal control problem is formulated with the mission-specific objective function. Apart from constraints of initial and final conditions, this multi-phase optimal control problem

is also subject to dynamics constraints from both quadrotor equations of motion and rotor aerodynamics models as well as path constraints. The optimal control problem is then solved in the frame of multi-disciplinary design optimization (MDO) by setting the energy consumption, times, control inputs, and states as design variables, and it has the potential to be converted into a simultaneous design and control process when including quadrotor design parameters as new design variables. After that, simulations are performed to validate the effectiveness of proposed methodology and to compare its results with those from peer methods. Three mission scenarios are set up to validate the proposed methodology. The simulation results indicate that the proposed method can decrease the flight time by up to 6.69% or mechanical energy consumption by 3.00%, and it outperforms two other trajectory generation methods for specific mission performance.

3.2 Nomenclature

$\mathbf{x}(t)$	=	quadrotor state variable, $\mathbf{x}(t) \in \mathbb{R}^{12\times 1}$.
$\mathbf{u}'(t)$	=	quadrotor control input, $\mathbf{u}'(t) = [n_1, n_2, n_3.n_4]^T \in \mathbb{R}^{4\times 1}$.
t	=	flight time in seconds.
$E(t)$	=	mechanical energy comsumption over time in Joules .
u_x	=	induced airflow velocity in m/s.
$\Omega,\ \omega$	=	angular velocity of propeller in rad/s.
$D,\ R$	=	diameter and radius of a propeller and rotor disk in m.
ρ	=	density of air, in this paper $\rho = 1.225kg/m^3$
$c(\bar{r}),\ \theta(\bar{r})$	=	radial chord and twist angle distribution of a propeller.
$V_\infty,\ \overline{V}_\infty$	=	free-stream velocity, $\overline{V}_\infty = V_\infty/\Omega R$.
u'_x	=	wake air velocity in m/s.
$\lambda(\bar{r}, \psi, t)$	=	local dimensionless induced airflow velocity at $(\bar{r}, \psi)$ and time t

λ_{0,r_i}	=	dimensionless uniform axial induced velocity component of i-th rotor.
λ_{s,r_i}	=	lateral variation component of induced velocity of i-th rotor.
λ_{c,r_i}	=	fore-to-aft variation component of induced velocity of i-th rotor.
$r, \bar{r}$	=	radial coordinate in rotor disk, $\bar{r} = r/R$.
ψ_r	=	angular coordinate in rotor disk in radians.
α	=	angle of incidence on a rotor disk in radians.
α_e	=	angle of attack at a blade element in radians.
θ_e	=	blade twist angle at a blade element in radians.
ϕ_e	=	inflow angle at a blade element in radians.
J	=	propeller advance ratio, $J = V_\infty \sin\alpha/\Omega R = V_\infty \sin\alpha/nD$.
$n,\ n_i$	=	propeller spin speed in RPM or rev/s.
$C_T,\ T$	=	thrust coefficient and thrust of a propeller.
$C_Q,\ Q$	=	torque coefficient and drag torque of a propeller.
$C_{roll},\ M_r$	=	roll moment coefficient and local pitch moment of a propeller.
$C_{pitch},\ M_p$	=	pitch moment coefficient and local pitch moment of a propeller.
C_{P,r_i}	=	power coefficient of i-th propeller.
N_b	=	number of blades, $N_b = 2$ in this chapter.
C_{dT}	=	local thrust coefficient at the blade element.
C_{dQ}	=	local torque coefficient at the blade element.
C_{dL}	=	local lift coefficient at the blade element.
C_{dD}	=	local drag coefficient at thr blade element.
$\psi,\ \theta,\ \phi$	=	yaw, pitch, and roll angles by Euler ZYX convention wrt inertial frame.
$p,\ q,\ r$	=	roll, pitch, and yaw rotation rates wrt quadrotor body-fixed frame.
F_z	=	total thrust wrt the quadrotor body-fixed frame.
τ_x	=	total roll torque wrt the body-fixed frame.

τ_y = total pitch torque wrt the body-fixed frame.

τ_z = total yaw torque wrt the body-fixed frame.

f_{obj} = the objective function.

t_f = final time in seconds when a mission is completed.

t_{phase_i} = time of finishing i-th phase in seconds.

3.3 Introduction

The search-and-rescue (SAR) operations with robotic assistance have become an emerging topic in both academia and industry over the last several decades. The development of more robust and versatile field robots has been spurred by more frequent occurrences of tragic events and risks and challenges faced by first responders during SAR operations. Among these robots, the unmanned aerial vehicles (UAVs) have demonstrated their advantages of being agile and fast with low operating costs and capable of detecting hazards to be avoided by task force members [WT10]. For example, two rotary-wing UAVs were deployed in a total of 14 missions and 38 flights by a team of 3 personnel to conduct search-and-rescue tasks in cluttered and destroyed urban areas in Florida and Mississippi in the aftermath of Hurricane Katrina [MPB08]. In addition, a multipurpose UAV was developed and deployed by Silvagni et al. [Sil+17] for mountain rescue operations in the event of avalanche. Other applications of SAR UAVs include the deployment of a unmanned helicopter to rapidly detect collapsed buildings to provide guidance for ground rescue teams after the 2013 Lushan earthquake in China [Qi+16].

As discussed above, UAVs are primarily implemented for SAR missions in outdoor urban or rural areas where GPS or other GNSS signals are available. However, more recent research efforts are focused on navigation, localization, and mission planning in cluttered and GPS-denied environments. Tomic et al. [Tom+12] developed a UAV system which is able to process an extended Kalman filter (EKF) for localization and navigate

through waypoints via recognition of known objects onboard using a distributed multi-board computation platform. This configuration enables the UAV to execute both indoor and outdoor urban SAR missions with no GPS access. Also, Kulkarni et al. [Kul+20] performed the simulation where the UAV searches for and locates the victim in two indoor scenarios and then navigates with the Q-learning framework based on the received signal strength (RSS) and rewards of the reinforcement learning algorithm. Other work includes search-and-rescue missions in a forest of no GPS-access by Tian et al. [Tia+20] using collaborative simultaneous localization and mapping (CSLAM) and UAV target searching and tracking using partially observable Markov decision process (POMDP) by Vanegas et al [Van+16].

However, most electrical rotary-wing or multirotor UAVs are known to suffer from relatively short flight time upon a single charge [JJK19]. In addition, the payload capacity of these UAVs is known to be constrained [VBS20] due to size and actuator limitations. The bottleneck of implementing quadrotor UAVs for SAR purposes is to achieve a balance between payload capacity, battery capacity, and flight time [VBS20]. In an attempt to overcome this bottleneck, researchers have been formulating and solving optimal control and path planning problems to minimize certain mission objectives. Andersen [And14] described five search patterns [CCH03] in outdoor SAR missions and compared their path coverages and travelled distances given the same search area. Cabreira et al. [Cab+19] [Cab+18] introduced an energy-aware grid-based coverage path planning algorithm (EG-CPP) in order to minimize energy consumption of UAV missions when covering irregular-shaped areas. On the other hand, Hayat et al. [Hay+17] developed a multi-objective optimization algorithm to plan paths for the UAV to locate a target and navigate in a bounded area and used a genetic algorithm to minimize the mission completion time. These approaches, however, either use a indirect method to estimate mechanical energy consumption like in [Cab+19] or linearize or over-simplify the UAV

equations of motion and aerodynamics, which results in significant deviation between the computed UAV paths, time, and energy consumption and those of real experiments thus undermining the reliability of these methods in real missions.

On the other hand, multi-disciplinary design optimization (MDO) has revealed its potential in optimizing mission trajectories and modeling nonlinear dynamical systems including multirotors [FG19]. Furthermore, OpenMDAO [Gra+19] is developed to provide a framework for assembling complex optimization problems from many components as well as an efficient architecture for computing gradients in large-scale systems. Relying on OpenMDAO and other MDO frameworks, a number of researchers have applied MDO to solving complex aircraft or robot design and trajectory optimization problems, including Hwang et al. 's work [Hwa+14] on satellite's design and operation, Kao et al. 's work [Kao+15] on aircraft mission analysis and trajectory optimization, and Yan et al. [Yan+19] 's work on optimal control and design of a ball-pitching robotic arm. In authors' previous work, we have studied the multi-objective design optimization (MOO) [Str+21b] [Str+21a] and optimal control [Cao+21] for a fully-actuated hexrotor UAV. However, the application of MDO to multirotor optimal control problems is not found by authors although MDO does have the capability of solving trajectory optimization for nonlinear dynamical systems by viewing time, controls, and states as design variables.

In this work, the authors formulate the trajectory optimization as a multi-phase optimal control problem for quadrotors in SAR scenarios. In our problem formulation, we include nonlinear equations of motion and aerodynamics of each rotor as dynamic constraints, and all phases are constrained by initial condition, waypoints, and final condition. By viewing energy consumption, times, control inputs, and states as design variables, we are able to solve the trajectory optimization using MDO techniques [ML13]. The proposed method can output optimal trajectories with respect to mission objectives for SAR UAVs in GPS-denied environments. The ordinary differential equations (ODE)

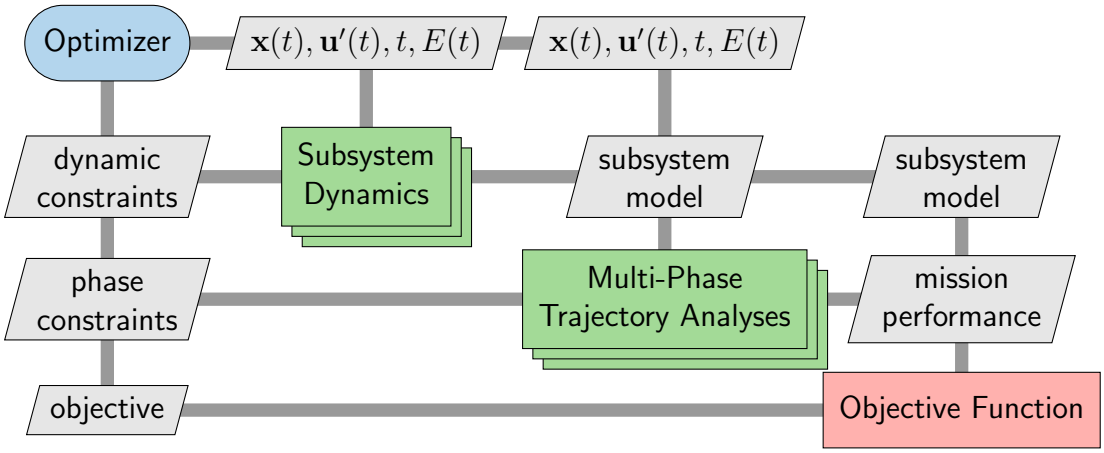


Figure 3.1. The extended design structure matrix (XDSM) of a multi-phase trajectory optimization problem.

system structure of the proposed method is illustrated in Fig. 3.1. In future work, this approach has the potential to be converted into a simultaneous design and control process when including the quadrotor design parameters.

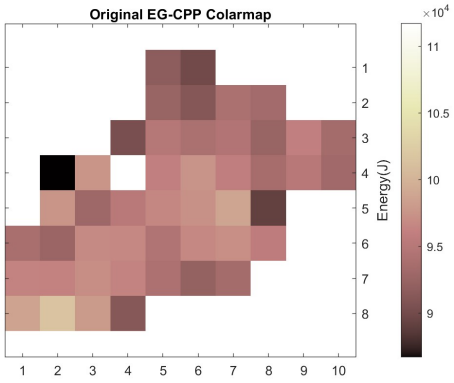
This paper is organized as follows: Section §3.4 discusses the set of methods and models implemented by proposed trajectory optimization approach, with Section §3.4.1 discussing the path planning methods providing waypoints and initial guesses and Section §3.4.2 discussing the two subsystems of ODEs that builds up the dynamic model and constraints. Section §3.4.3 and section §3.11c describes this paper's approach of formulating optimization problem, solving methods, and collision checks. Section §3.5 discusses the simulation environment and provides comparisons of the results from different trajectory generation methods. Last but not least, Section §3.6 discuss the conclusions from this research and goals of future work.

3.4 Methodology

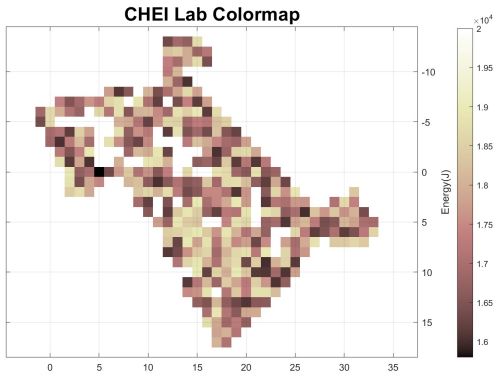
3.4.1 Flight Path Planning

Path planning is the process of generating collision-free paths from an initial state to a goal state with optimal or sub-optimal cost [NKH+16]. In this work, the initial planning of the mission path is realized using computationally inexpensive algorithms available in the literature. These path planning algorithms are expected to render a set of waypoint and initial guess for each mission flight. In this work, we implement Rapidly Exploring Random Tree Star (RRT*) algorithm [KF11] for minimal-time missions, and an energy-aware grid-based coverage path planning (CPP) algorithm [Cab+19] for the missions minimizing mechanical energy consumption.

Coverage Path Planning



(a) Original Colormap from [Cab+19].



(b) Energy-Aware Colormap of CHEI Lab of UCSD.

Figure 3.2. Energy Colormaps of Path Start Nodes.

We define the searching-for-survivors mission as the following: In this scenario, the UAV needs to visit every collision-free square cell in the work space, for example, a grid map, via a path of minimum energy consumption. This is a simplified scenario where the UAV carries on the task of searching for victims in a known indoor space and return

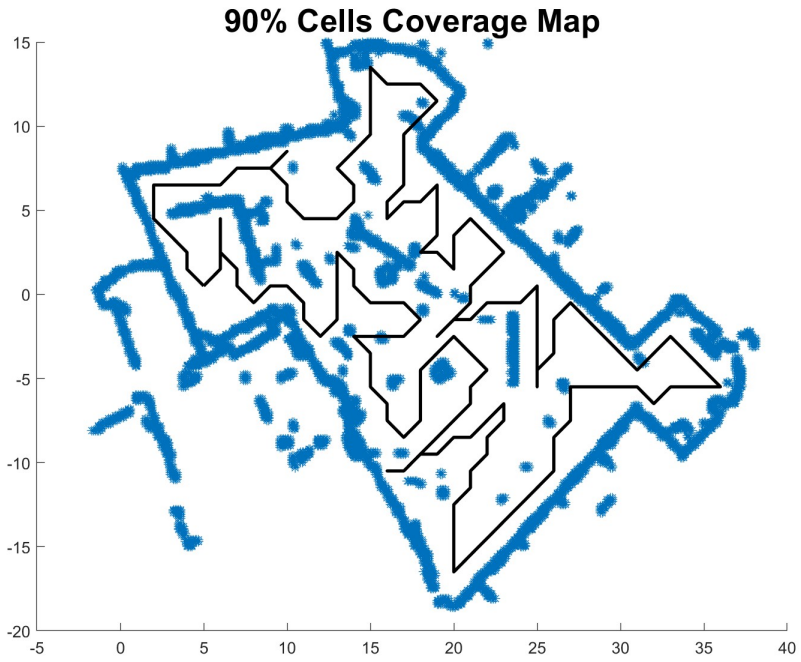


Figure 3.3. EG-CPP coverage path for CHEI Lab.

their locations. In order to perform fast search for coverage path, we assume the UAV flies at the altitude of $2m$ and simplify the 3D map to be a 2D binary occupancy grid. In this grid, each cell is a $1m \times 1m$ square which the UAV needs to cover on the path. Here we assume the UAV flies at the altitude of $2m$ above ground level (AGL) and set the Z-coordinates of every waypoint to be $2m$.

In the energy-aware grid-based coverage path planning (EG-CPP) method proposed by Cabreira et al. [Cab+19], the irregular-shaped search area is discretized into a regular grid, and each cell in the grid is assigned with a coverage path starting from this cell marked with its associated energy cost. This step is also visualized as a colormap where each cell in the free configuration space (C-space) is filled with a color representing the energy consumption of the path starting from this cell as illustated in Fig. 3.2a. Similarly, when planning the path for our missions, we also discretize the free C-space into grids and

obtain its colormap as shown in Fig. 3.2b. We assume the UAV can cover a $2m \times 2m$ area surrounding it when visiting a cell. Given the colormap information, one can find that the optimal start node is $[5.0, 0.0, 2.0]$ with the minimum cost of $15,863$ J. Its associated coverage path is shown in Fig 3.3, with a guarantee of covering 90% of the cells in free C-space. This path include in total 83 waypoints and has a length of $205.95m$.

RRT* Path Planning

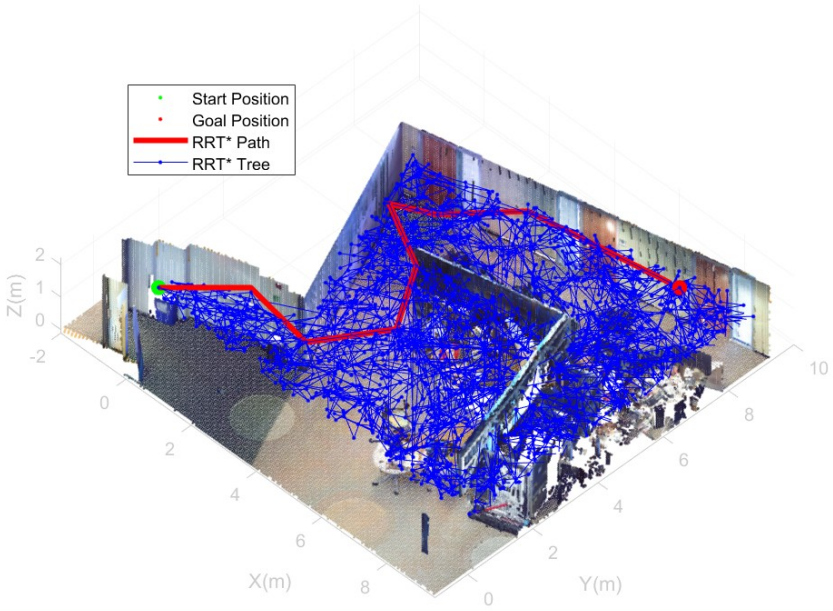


Figure 3.4. RRT* path planned on CHEI Lab point cloud map.

In some scenarios, the UAV will need to visit several rooms in the building to locate survivors and report their positions. In some other cases, the UAV needs to deliver important payloads like medications and life-aid devices to victims on time. These missions are typically in the interest of time such that in these missions time is set to be the objective. In order to plan the paths fast and robustly, we select RRT* [KF11], a renowned sampling-based path planning algorithm, to find feasible paths for our missions.

RRT* builds a randomly exploring tree incrementally, but only connects the neighboring nodes along a minimum-cost path, and rewires the tree by shifting neighboring nodes to another lower-cost random neighbor at each iteration. RRT* is proven to be asymptotically optimal and computationally efficient [NKH+16].

Given a fixed maximum speed, the time consumption of a path is positively associated with its total length. Therefore, we attempt to find the optimal or sub-optimal path with minimum path length. A minimum-length path is illustrated in Fig. 3.4 where a UAV travels from the origin to a waypoint location ($[8.0, 8.0, 2.0]$) on the CHEI Lab point cloud map. By setting the maximum connection distance to be $2.0m$, RRT* gives a path of 8 waypoints and a total length of $17.41m$.

3.4.2 Quadrotor Dynamics Modeling

We now need to study the quadrotor dynamics which describes the input-to-output relationship from the rotary speeds of 4 propellers to the x-, y-, z-coordinates, and yaw angles of the quadrotor UAV in the inertial frame. We divide the dynamics into two major subsystems which are rotor aerodynamics and equations of motion. Both subsystems are characterized as systems of ordinary differential equations (ODEs). In the rotor aerodynamics modeling, we combine the variations of dynamic inflow components of 4 rotor disks and the resulting wrench applied on quadrotor center of mass as a $[4 \times 1]$ vector (including total thrust and total moments around x-, y-, and z-axes of body-fixed frame). While in the equations of motion of quadrotor flight dynamics, we implement the most commonly used $[12 \times 1]$ state vector including x-, y-, z-position and velocity components and yaw, pitch, roll attitude components in the inertial frame plus the yaw, pitch, roll rates in the body-fixed frame.

In the next three sub-sections, we will introduce both the dynamic inflow model and blade element momentum theory (BET) to help characterize the aerodynamics of

a single rotor, and then combine aerodynamic equations of 4 rotors. And In the last sub-section, we present the quadrotor equations of motion as its flight dynamics.

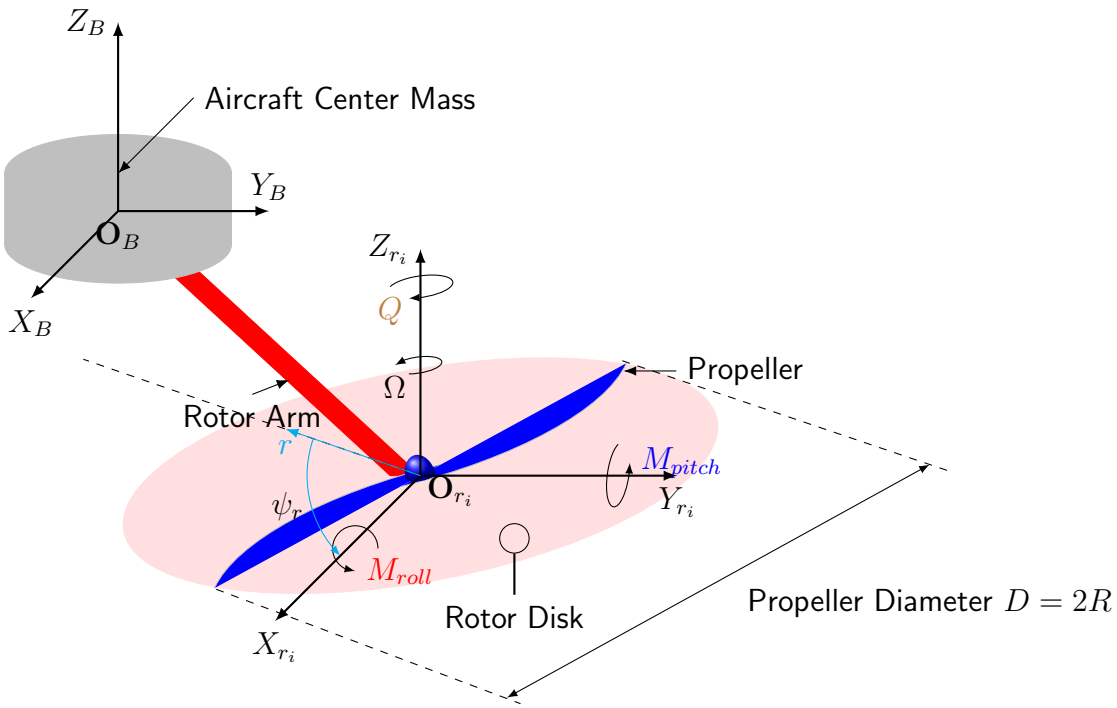


Figure 3.5. The illustration of body-fixed frame and rotor disk coordinate systems.

Pitt-Peters Rotor Inflow model

Pitt and Peters [Pit80] further developed a system of linear ODEs to describe the dynamic inflow model of a rotor disk which has become one of the most popular theories in this area during the last 4 decades. The Pitt-Peters model elegantly relates the dimensionless aerodynamic loading to dimensionless induced flow distribution. Knowing that the induced velocities distribution in a rotor disk is given by a function of polar coordinates $(\bar{r}, \psi)$ and time t:

$$\lambda(\bar{r}, \psi_r, t) = \lambda_0(t) + \lambda_s(t)\bar{r}\sin\psi_r + \lambda_c(t)\bar{r}\cos\psi_r, \tag{3.1}$$

where λ is the local dimensionless induced airflow velocity $u_x/\Omega R$, λ_0, λ_s, and λ_c are the uniform, side-to-side, and fore-to-aft variations in induced airflow, respectively. And the dimensionless radial coordinate $\bar{r}$ is r/R where R is the propeller radius. The inflow dynamics of this rotor can be described by the following state-space model where only first-harmonic terms are kept to render the linear first-order state-space representation:

$$\begin{bmatrix} \dot{\lambda}_0 \\ \dot{\lambda}_s \\ \dot{\lambda}_c \end{bmatrix} = -\mathbf{M}^{-1}\mathbf{L}^{-1} \begin{bmatrix} \lambda_0 \\ \lambda_s \\ \lambda_c \end{bmatrix} + \mathbf{M}^{-1} \begin{bmatrix} C_T \\ C_{roll} \\ C_{pitch} \end{bmatrix},$$

$$\text{where } \mathbf{L} = \frac{1}{\overline{V}_\infty} \begin{bmatrix} \frac{1}{2} & 0 & \frac{15\pi}{64}\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}} \\ 0 & -\frac{4}{1+\sin\alpha} & 0 \\ \frac{15\pi}{64}\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}} & 0 & -\frac{4\sin\alpha}{1+\sin\alpha} \end{bmatrix}, \mathbf{M} = \begin{bmatrix} \frac{128}{75\pi} & 0 & 0 \\ 0 & \frac{-16}{45\pi} & 0 \\ 0 & 0 & \frac{-16}{45\pi} \end{bmatrix}.$$

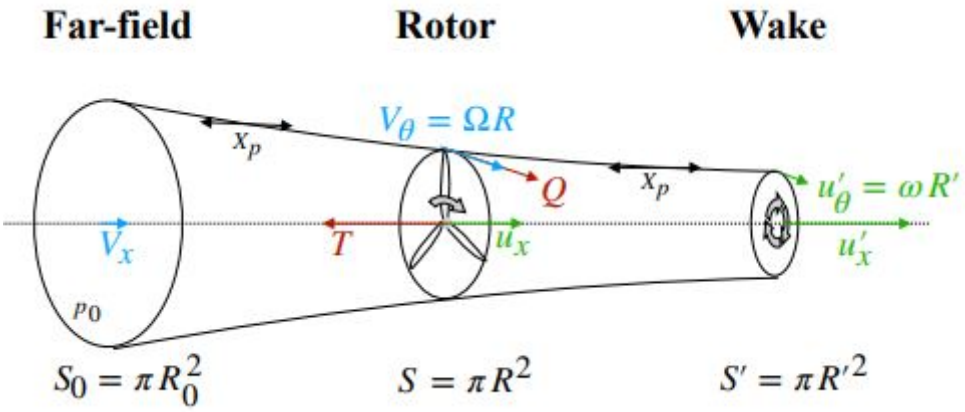
Here $\overline{V}_\infty$ is the dimensionless free-stream velocity $V_\infty/\Omega R$ and α is the angle of incidence. In the scope of this work, we compute the magnitude of free-stream velocity to be the norm of current flight velocity vector $[\dot{x}, \dot{y}, \dot{z}]$, and angle of incidence α to be complimentary angle of the angle between the free-stream velocity and axial inflow velocity, i.e. the normal vector of the rotor disk. The rotor disk coordinates is shown in Fig. 3.5, and the equations to compute $\overline{V}_\infty$ and α is given as follows:

$$\overline{V}_\infty = \frac{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}}{2\pi n R},\ \alpha = \pi/2 - \text{acos}(\frac{[\dot{x}, \dot{y}, \dot{z}] \cdot \mathbf{n}_{rotordisk}}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \cdot 1}), \tag{3.2}$$

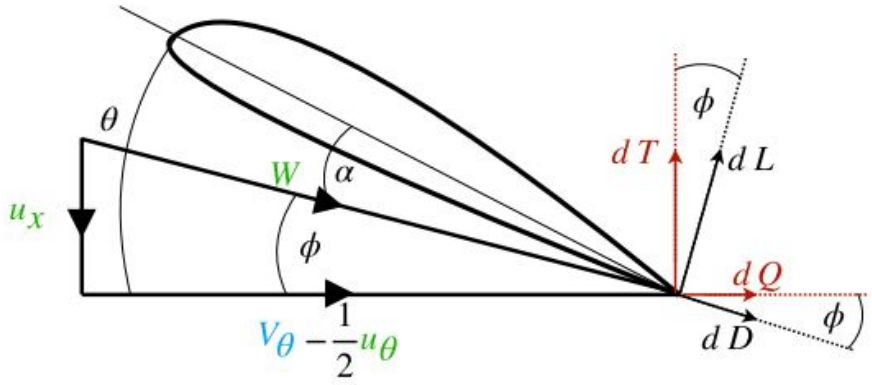
such that we will need the input of quadrotor flight states here to evaluate the inflow

states of each propeller at different times.

Blade Element Theory



(a) Diagram of a streamtube with rotor disk and purely axial inflow.



(b) Diagram of a blade element.

Figure 3.6. Illustration of BET parameters [RH21].

In order to provide the non-dimensional aerodynamic loads, we need a fast and robust approach to estimate the thrust, moment, and power coefficients. These aerodynamic loads are usually computed using Blade Element Theory with the angle of attack on the blade element estimated by either induced velocity from either momentum theory or other inflow models. As illustrated in Fig. 3.6b, the angle of attack α_e exerted on this element

Algorithm 1. BET Estimation

Input $c(\bar{r})$, $\theta(\bar{r})$, D (propeller geometry)

for $n \in [n_{min}, n_{max}]$ rev/s **do**
  for $V_\infty \in [0.01, V_{max}]$ m/s **do**

    $J = \dfrac{V_\infty \sin\alpha}{nD}$, store to J-array

    if $0.01 \leq J \leq 1.00$ **and** $J \notin J$-array **then**
      Solve for ϕ iteratively via $\tan\phi = \dfrac{J}{\pi}\dfrac{1+a_x}{1-a_r}$ where $a_x = \dfrac{1}{\frac{4\sin^2\phi}{\sigma C_{dT}} - 1}$, $a_r = \dfrac{1}{\frac{4\sin\phi\cos\phi}{\sigma C_{dQ}} + 1}$
      Compute a_x and a_r with current ϕ
      Compute T, Q, M_r, M_p using (3.4 - 3.7), respectively
      $C_T = \dfrac{T}{\rho n^2 D^4}$, store to C_T-array
      $C_Q = \dfrac{Q}{\rho n^2 D^5}$, store to C_Q-array
      $C_{roll} = \dfrac{M_r}{\rho n^2 D^5}$, store to C_{roll}-array
      $C_{pitch} = \dfrac{M_p}{\rho n^2 D^5}$, store to C_{pitch}-array
    end if
  end for
end for

is computed by

$$\alpha_e = \theta_e - \phi_e, \tag{3.3}$$

where θ_e is the blade twist angle at the element location and ϕ_e is called inflow angle computed by the axial and in-plane components of induced velocity at this blade element. The approach to compute the inflow angle with momentum theory only considering uniform and steady axial and in-plane inflows is known as Blade Element Theory (BET) [MV17].

Some literature points out that this approach results in non-negligible errors in predicting aerodynamics loads when advanced ratio is smaller or equal to than 0.3 ($J < 0.3$)

[GR05] or at lower Reynolds numbers ($Re < 3\times10^5$). While for our trajectory optimization problem setting, the range of $J < 0.3$ is replaced by the experimental data measured with thrust testing apparatus one can find in [Str+21b], and low speed assumption holds for us to neglect complicated aerodynamic effects [Zuo10]. However, the implementation of BET still provides a good estimate for $J > 0.3$, and thus we create the direct mapping from the advance ratio of a propeller to its corresponding thrust, pitch, roll, and yaw moment coefficients.

In the BET method, the propeller is simplified as a rotor disk model fixed within a stream tube of the airflow as shown in Fig. 3.6a. The stream tubes are constructed differently for a UAV rotor during hovering and forward flight [DD19]. Thus we introduce the angle of incidence α to help describe the non-axial inflow such that the advance ratio $J = V_\infty \sin\alpha / nD$. Therefore, we estimate the rotor thrust T, drag torque Q, pitch moment M_p, and roll moment M_r using the following 4 equations:

$$T = \frac{N_b}{2\pi}\int_0^{2\pi}\int_{r_{hub}}^{R}\frac{1}{2}\rho V_\infty^2 \sin^2\alpha \; c\frac{(1+a_x)^2}{\sin^2\phi}C_{dT}\; dr d\psi_e, \tag{3.4}$$

$$Q = -\frac{N_b}{2\pi}\int_0^{2\pi}\int_{r_{hub}}^{R}\frac{1}{2}\rho V_\infty \sin\alpha \; c\omega r^2\frac{(1+a_x)(1-a_r)}{\sin\phi\cos\phi}C_{dT}\; dr d\psi_e, \tag{3.5}$$

$$\begin{aligned} M_p = &\frac{N_b}{2\pi}\frac{(1+a_x)^2}{\sin^2\phi}\Big(\int_0^{\pi}\int_{r_{hub}}^{R}\frac{1}{2}\rho V_\infty^2\sin^2\alpha \; cr\sin\psi_e C_{dT}\; dr d\psi_e \\ &- \int_{\pi}^{2\pi}\int_{r_{hub}}^{R}\frac{1}{2}\rho V_\infty^2\sin^2\alpha \; cr\sin\psi_e C_{dT}\; dr d\psi_e\Big), \end{aligned} \tag{3.6}$$

$$
\begin{aligned}
M_r = & \frac{N_b}{2\pi}\frac{(1+a_x)^2}{\sin^2\phi}\Big(\int_{\pi/2}^{3\pi/2}\int_{r_{hub}}^{R}\frac{1}{2}\rho V_\infty^2 \sin^2\alpha \; cr \; \cos\psi_e C_{dT} \; dr d\psi_e \\
& - \int_{-\pi/2}^{\pi/2}\int_{r_{hub}}^{R}\frac{1}{2}\rho V_\infty^2 \sin^2\alpha \; cr \; \cos\psi_e C_{dT} \; dr d\psi_e\Big),
\end{aligned}
\tag{3.7}
$$

where V_∞ is the free stream velocity, c is the chord length of the blade element, N_b is the number of blades, a_x and a_r are the axial and radial induction factor, respectively [Gla35], and ϕ is the inflow angle as indicated in Fig. 3.6b. And C_{dT} and C_{dQ} are the element thrust and torque coefficients obtained by projecting the element lift and drag coefficients via

$$
\begin{bmatrix} C_{dT} \\ C_{dQ} \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} C_{dL} \\ C_{dD} \end{bmatrix} \tag{3.8}
$$

Also the power coefficient of each rotor need to be recomputed and stored at each time step with the aerodynamic loads and inflow states with the following analytical equation [Kim99][Dav+20]:

$$
C_{P,r_i} = \frac{1.15 C_T}{2\sqrt{\lambda_0^2 + J^2}} + \frac{\sigma C_{d0}}{8}(1 + 4.6\, J^2) + \frac{1}{8}\frac{f}{A} J^3 \tag{3.9}
$$

where σ is the blade solidity computed by (blade area b)/(rotor area A), C_{d0} is the profile drag coefficient with a typical value of 0.008. f/A is the equivalent flat plate area f correlated with rotor area A which can be approximated as 1 for a quadrotor [Dav+20]. Since the power coefficient C_P is equivalent to C_Q, and we learned from experimental data that C_Q computed by BET predictions can possibly deviate from the real-world data in higher range of propeller RPM, we replace the C_Q with the C_P value computed here if BET C_Q predictions differs from C_P by over 5%.

State-Space Model of Rotor Aerodynamics

After computing thrust and torque coefficients using Algorithm 1, we are able to sum up the thrust and torques of the 4 rotors and also predicts the dynamic inflow states of each rotor which will help us recompute the power coefficient and efficiency they deviate significantly from the BET predictions [MV17]. The output vector in rotor aerodynamics is $\mathbf{y}_{aero} = [F_z\,, \tau_x\,, \tau_y\,, \tau_z]^T$. F_z is the sum of vertical thrusts of 4 motors, τ_x is the sum of moments with respect to x-axis of body-fixed frame. Similarly, τ_y and τ_z are sum of moments with respect to its y- and z axes, respectively. These four output states are the inputs to the equations of motion flight dynamics. Since each rotor has its individual inflow dynamics, we formulate the state vector $\mathbf{x_0} \in \mathbb{R}^{16\times 1}$ to be

$$\mathbf{x}_0 = [\lambda_{0,r_1}\,, \lambda_{s,r_1}\,, \lambda_{c,r_1}\,, \ldots, \lambda_{0,r_4}\,, \lambda_{s,r_4}\,, \lambda_{c,r_4}\,, F_z\,, \tau_x\,, \tau_y\,, \tau_z]^T, \tag{3.10}$$

where $\lambda_{0,r_i}\,, \lambda_{s,r_i}\,, \lambda_{c,r_i}$ are the inflow states of i-the rotor. Also we consider the square of each motor's rotary speed as the control $\mathbf{u}_0 \in \mathbb{R}^{4\times 1}$, and the vector of aerodynamic loads as disturbance vector $\mathbf{v}_0 \in \mathbb{R}^{12\times 1}$, with

$$\begin{aligned} \mathbf{u}_0 &= [n_1^2, n_2^2, n_3^2, n_4^2]^T \\ \mathbf{v}_0 &= [C_{T,r_1}, C_{roll,r_1}, C_{pitch,r_1}, \ldots, C_{T,r_4}, C_{roll,r_4}, C_{pitch,r_4}]^T. \end{aligned} \tag{3.11}$$

Therefore, we can now conclude the derivation of rotor aerodynamics subsystem. The state equations of the rotor aerodynamics can be written as follows:

$$\dot{\mathbf{x}}_0 = \mathbf{A}_0 \cdot \mathbf{x}_0 + \mathbf{D}_0 \cdot \mathbf{v}_0 + \mathbf{B}_0 \cdot \mathbf{u}_0 \tag{3.12}$$

$$\text{with } \mathbf{A}_0 = \begin{bmatrix} -\mathbf{M}^{-1}\mathbf{L}^{-1} & & & & & \\ & -\mathbf{M}^{-1}\mathbf{L}^{-1} & & & & \\ & & -\mathbf{M}^{-1}\mathbf{L}^{-1} & & & \\ & & & -\mathbf{M}^{-1}\mathbf{L}^{-1} & & \\ & & & & -\mathbf{I}^{4\times 4} \end{bmatrix},$$

$$\mathbf{B}_0 = \begin{bmatrix} 0^{12\times 1} & 0^{12\times 1} & 0^{12\times 1} & 0^{12\times 1} \\ b_1 & b_1 & b_1 & b_1 \\ b_2 + k_{roll} & -(b_2 + k_{roll}) & -(b_2 + k_{roll}) & b_2 + k_{roll} \\ b_2 + k_{pitch} & b_2 + k_{pitch} & -(b_2 + k_{pitch}) & -(b_2 + k_{pitch}) \\ k_{yaw} & -k_{yaw} & k_{yaw} & -k_{yaw} \end{bmatrix},$$

$$\mathbf{D}_0 = \begin{bmatrix} -\mathbf{M}^{-1} & & & \\ & -\mathbf{M}^{-1} & & \\ & & -\mathbf{M}^{-1} & \\ & & & -\mathbf{M}^{-1} \\ 0^{4\times 1} & 0^{4\times 1} & 0^{4\times 1} & 0^{4\times 1} \end{bmatrix}.$$

where $b_1 = C_T \rho D^4$, $b_2 = b_1 \cdot \frac{\sqrt{2}}{2} l_{arm}$, $k_{roll} = C_{roll} \rho D^5$, $k_{pitch} = C_{pitch} \rho D^5$, $k_{yaw} = C_Q \rho D^5$.

Quadrotor Flight Dynamics

The other sub-system of the quadrotor ODE model is its equations of motion (EoM) in 6 degrees of freedom. We set the inertial frame of the indoor space with north-west-up convention where x-axis points north, y-axis points west, and z-axis points upwards. The UAV body-fixed frame uses the same convention and has its three axes coincident with those of the inertial frame if starting from the origin. Therefore, we can organize the state vector of a quadrotor to be

$$\mathbf{x} = [x\ y\ z\ \psi\ \theta\ \phi\ \dot{x}\ \dot{y}\ \dot{z}\ p\ q\ r]^T \tag{3.13}$$

which represents x, y, z displacement components, ψ, θ, ϕ rotational displacement components, x, y, z velocity components in the inertial frame and p, q, r rotational rates with respect to x, y, z-axes of the body-fixed frame, respectively. The control input of quadrotor EoMs is

$$\mathbf{u} = [F_z \ \tau_x \ \tau_y \ \tau_z]^T. \tag{3.14}$$

The 6-DoF EoMs of quadrotor have been well studied and documented in the literature [Sab15], such that we only give the final State-Space form of the quadrotor EoMs as the second sub-system. We use $s(\phi)$, $c(\phi)$, $s(\theta)$, $c(\theta)$, $t(\theta)$ to represent $\sin(\phi)$, $\cos(\phi)$, $\sin(\theta)$, $\cos(\theta)$, and $\tan(\theta)$, respectively, such that:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \tag{3.15}$$

$$\text{where } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ q\frac{s(\phi)}{c(\theta)} + r\frac{c(\phi)}{c(\theta)} \\ q \cdot c(\phi) - r \cdot s(\phi) \\ p + q \cdot s(\phi)t(\theta) + r \cdot c(\phi)t(\theta) \\ 0 \\ 0 \\ -g \\ \frac{I_y - I_z}{I_x} qr \\ \frac{I_z - I_x}{I_y} pr \\ \frac{I_x - I_y}{I_z} pq \end{bmatrix}, \mathbf{g}(\mathbf{x}) = \begin{bmatrix} 0^{6\times1} & 0^{6\times1} & 0^{6\times1} & 0^{6\times1} \\ g_1 & 0 & 0 & 0 \\ g_2 & 0 & 0 & 0 \\ g_3 & 0 & 0 & 0 \\ 0 & 1/I_x & 0 & 0 \\ 0 & 0 & 1/I_y & 0 \\ 0 & 0 & 0 & 1/I_z \end{bmatrix} \text{ and} \tag{3.16}$$

$$\begin{aligned} g_1 &= -\frac{1}{m}[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] \\ g_2 &= \frac{1}{m}[c(\psi)s(\phi) - c(\phi)s(\psi)s(\theta)] \\ g_3 &= -\frac{1}{m}[c(\phi)c(\theta)] \end{aligned} \tag{3.17}$$

3.4.3 Mission-Oriented Trajectory Optimization

Finding optimal solutions for dynamic system inputs given a task constitutes an optimal control problem. As can be seen from Equations (3.15) - (3.17), the EoM of the quadrotor is not a linear model. Both $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are coupled with components of the state vector. Therefore when we attempt to find a solution trajectory minimizing a mission-specific objective function, the ODE system provides non-linear constraints converting the optimal control problem in to a non-linear programming problem which

can be rapidly and robustly solved using multidisciplinary design optimization (MDO) methods. In the mean time, this optimal control problem is also expected to have the potential to become a simultaneous control and design optimization given we have modern computational tools like SNOPT [GMS05] to run large-scale constrained optimization. Therefore, it is possible to run optimization over the free configuration and control space over the entire mission by viewing numerous discrete control inputs, trajectory segments, and travelling time as design variables in an MDO problem.

Design Variables

Optimal control problem can be solved in the context of the multidisciplinary design, analysis, and optimization (MDAO) [Gra+19]. As an initial step to formulate a simultaneous design and optimization problem and The UAV design and trajectory optimization can be coupled in a monolithic approach with the state variable $\mathbf{x}(t)$ and control variable $\mathbf{u}'(t) = [n_1, n_2, n_2, n_4]^T$ as part of the design variables [FG19]. In addition, we add the mechanical power consumption $E(t)$ and time consumption through all phases t as other two design variables due to the need for fulfilling missions objectives.

The state vector $\mathbf{x}(t)$ and control input vector $\mathbf{u}(t)$ are given as

$$\mathbf{x}(\mathbf{t}) = [x\ y\ z\ \psi\ \theta\ \phi\ \dot{x}\ \dot{y}\ \dot{z}\ p\ q\ r]^T, \tag{3.18a}$$

$$\mathbf{u}'(t) = [n_1, n_2, n_2, n_4]^T, \tag{3.18b}$$

respectively, and the mechanical power consumption $E(t)$ is computed by

$$E(t) = 2\pi\rho D^5 \int_0^t C_{P,r_1} \cdot n_1^3 + C_{P,r_2} \cdot n_2^3 + C_{P,r_3} \cdot n_3^3 + C_{P,r_4} \cdot n_4^3 \, dt. \tag{3.19}$$

Problem Formulation

In a multi-phase mission of SAR quadrotor, the objective function is to be minimized with respect to the entire mission although it is evaluated at the final time t_f in the MDO method we use. Objective functions are considered as one of design variables and is kept record of and simulated throughout the entire mission. They can be arbitrarily constructed while the only two mission objectives we investigate in this paper are time and energy consumption of the mission. During each phase, the UAv is expected to travel from $i-1$-th waypoint to i-th waypoint, and this phase terminates once the UAV arrives within the proximity of the i-th waypoint. And We denote the terminal time of this phase as t_{phase_i}. The design variables here are state variable $\mathbf{x}(t)$, control variable $\mathbf{u}'(t)$, mechanical energy $E(t)$, and phase time t. In Section 3.4.3 and 3.4.3, we will propose an approach to solve the multi-phase optimal control problem using a nonlinear programming approach within the MDAO context. The formulation of the optimal control problem to optimize the trajectories of an SAR quadrotor UAV within the MDAO context is given as follows:

$$
\begin{aligned}
\text{minimize:} \quad & J = f_{obj}(\mathbf{x}(t), \mathbf{u}'(t), E(t), t)|_{t=t_f}; \\
\text{with respect to:} \quad & \mathbf{x}(t), \mathbf{u}'(t), E(t), t; \\
\text{subject to dynamics:} \quad & \dot{\mathbf{x}}_0 = \mathbf{A_0} \cdot \mathbf{x}_0 + \mathbf{D_0} \cdot \mathbf{v}_0 + \mathbf{B}_0 \cdot \mathbf{u}_0 \text{ (Eq. 3.12)} \\
& \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}) \cdot \mathbf{u} \text{ (Eq. 3.15)} \\
\text{and initial conditions:} \quad & x(t=0) = x_0 \\
& x(t=0) = y_0 \\
& z(t=0) = z_0 \\
& \psi(t=0) = \psi_0 \\
& \text{all linear and angular velocity components} = 0 \\
\text{and final conditions:} \quad & x(t=t_f) = x_f \\
& y(t=t_f) = y_f \\
& z(t=t_f) = z_f \\
& \psi(t=t_f) = \psi_f \\
& \text{all linear and angular velocity components} = 0 \\
\text{and path constraints:} \quad & x(t=t_{phase_i}) = x_{phase_i} \\
& x(t=t_{phase_i}) = y_{phase_i} \\
& z(t=t_{phase_i}) = z_{phase_i} \\
& \psi(t=t_{phase_i}) = \psi_{phase_i} \\
& 0 \leq n_i \leq 350 \text{ rev/s} \\
& x, y, z, \psi \text{ within the free C-space.}
\end{aligned}
\tag{3.20}
$$

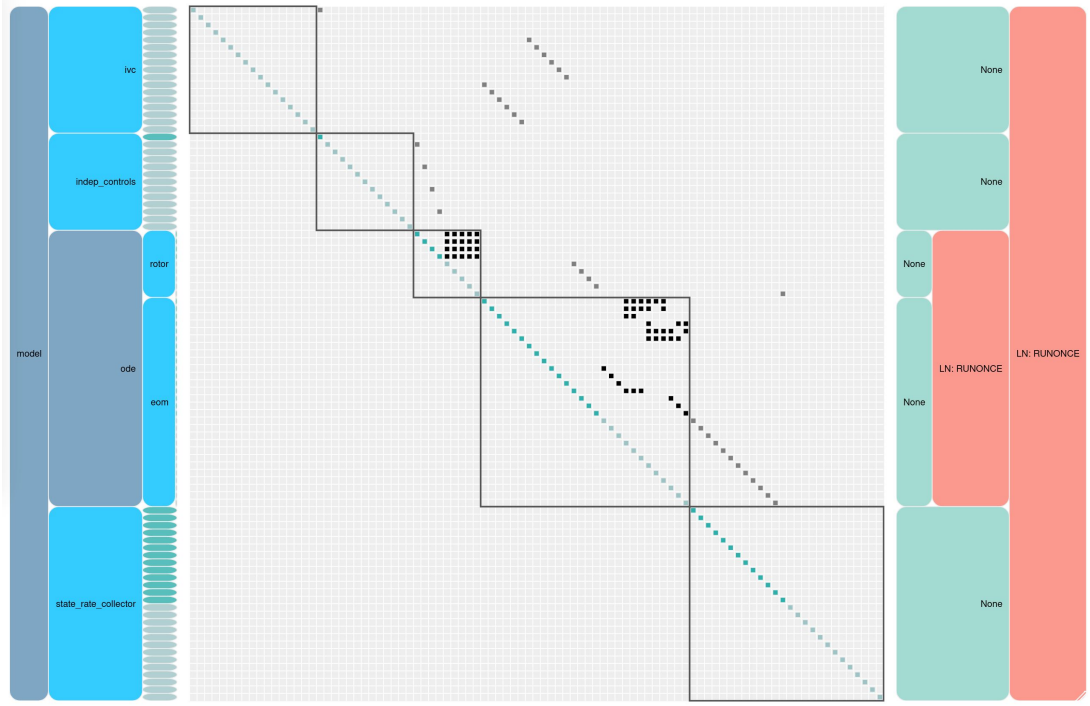


Figure 3.7. N^2 diagram of the ODE system with partial derivative components structure in OpenMDAO.

Pseudo-Spectral Methods

In order to minimize the computational time when computing the partial derivatives and changes of the design variables over time, we select to use implicit collocation techniques which are well-suited to gradient-based optimization with analytically computed derivatives[FG19]. Compared to time-marching, implicit collocation schemes can achieve same level of accuracy while utilize fewer grid points needed for analysis since the number of grid points is fixed during the optimization.

One implicit collocation used in this paper is high-order Legendre-Gauss-Lobatto (LGL) transcription which helps solve multi-phase optimal control problem. It discretizes each phase into segments at the LGL nodes such that the total number of nodes within each segment is odd. Nodes with an even index are the state discretization nodes. And controls are provided at both collocation nodes and state discretization nodes, both nodes are

called as control discretization nodes. The states and state-rates are then interpolated to the collocation nodes using Hermite-interpolation [DHS87]. We then obtain the collocation defects by subtracting interpolated values from ODE evaluated values. In the transcribed nonlinear programming problem, we constrain the collocations defects to be zero.

Another implicit collocation method used in this paper is called Radau Pseudo-Spectral Method (RPM). The RPM uses the Legendre-Gauss-Radau (LGR) nodes to discretize the problem within each segment. Discretized states and controls are provided at each LGR nodes. Different from LGL phases, the ODEs are evaluated at all nodes simultaneously, and the state rates are approximated using a Lagrange differentiation matrix [RF03].

Adjoint Method to Evaluate Derivatives

Analytic-derivative methods to evaluate model derivatives can be more efficient than numerical approaches since they solve for the total derivatives leveraging only inexpensive partial derivatives of internal model calculations [FG19]. Adjoint methods are one type of analytic-derivative methods and prove to be more efficient when number of input variables n_x is greater than that of output variables n_f [MH13]. We first start with the total derivative Jacobian of the objective fucntion as

$$\frac{df_{obj}}{d\mathbf{X}} = \frac{\partial f_{obj}}{\partial \mathbf{X}} + \frac{\partial f_{obj}}{\partial \mathbf{Y}} \frac{d\mathbf{Y}}{d\mathbf{X}}. \tag{3.21}$$

The two partial derivative terms in Eqn. (3.21) are relatively cheap to compute, but the total derivative term $d\mathbf{Y}/d\mathbf{X}$ is expensive to compute directly. Instead, the governing equations $R(\mathbf{X}, \mathbf{Y}) = 0$ can be implemented to compute $d\mathbf{Y}/d\mathbf{X}$ indirectly with much lower computational cost.

$$R(\mathbf{X}, \mathbf{Y}) = 0, \tag{3.22}$$

$$\frac{dR}{d\mathbf{X}} = \frac{\partial R}{\partial \mathbf{X}} + \frac{\partial R}{\partial \mathbf{Y}}\frac{d\mathbf{Y}}{d\mathbf{X}} = 0 \quad \Rightarrow \quad \frac{d\mathbf{Y}}{d\mathbf{X}} = -\Big[\frac{\partial R}{\partial \mathbf{Y}}\Big]^{-1}\frac{\partial R}{\partial \mathbf{Y}} \tag{3.23}$$

With Eqns. (3.22) and (3.23), one can derive the adjoint method in the form of

$$\frac{df_{obj}}{d\mathbf{X}} = \frac{\partial f_{obj}}{\partial \mathbf{X}} + \Psi^T \frac{\partial R}{\partial \mathbf{X}} \tag{3.24}$$

$$\text{where } \Psi = -\Big[\frac{\partial R}{\partial \mathbf{Y}}^T\Big]^{-1}\Big[\frac{\partial f_{obj}}{\partial \mathbf{Y}}^T\Big]. \tag{3.25}$$

Solver Software

In order to solve this optimal control problem with proposed methodology, we first implement the quadrotor ODE system in OpenMDAO [Gra+19] as shown in Fig. 3.7. OpenMDAO is a handy tool in building systems of ordinary differential equations and differential algebraic equationsm, and is also a widely-used open-source python framework for multi-disciplinary design, analysis, and optimization which is also great at solving nonlinear programming problems. Recently it has also been used for simultaneous design and control optimization of mechanical systems [Yan+19]. In addition, we install the Dymos [Fal+21] library on the basis of OpenMDAO to solve for multi-phase trajectory optimization problem with aforementioned set of methods. Moreover, Dymos implements single unified derivative equation (UDE) approach proposed by Hwang and Martins [HM18] to evaluate derivatives. This outperforms sole adjoint method by using a pair of direct and adjoint methods to solve for derivatives and generalizing them in an UDE. Another N2 diagram showing the structure of phase 24 of one solution trajectory is illustrated in Fig. 3.8. The ODE system and trajectory optimizer code can be found on the author's github page https://github.com/p5cao/quadTrajOpt.

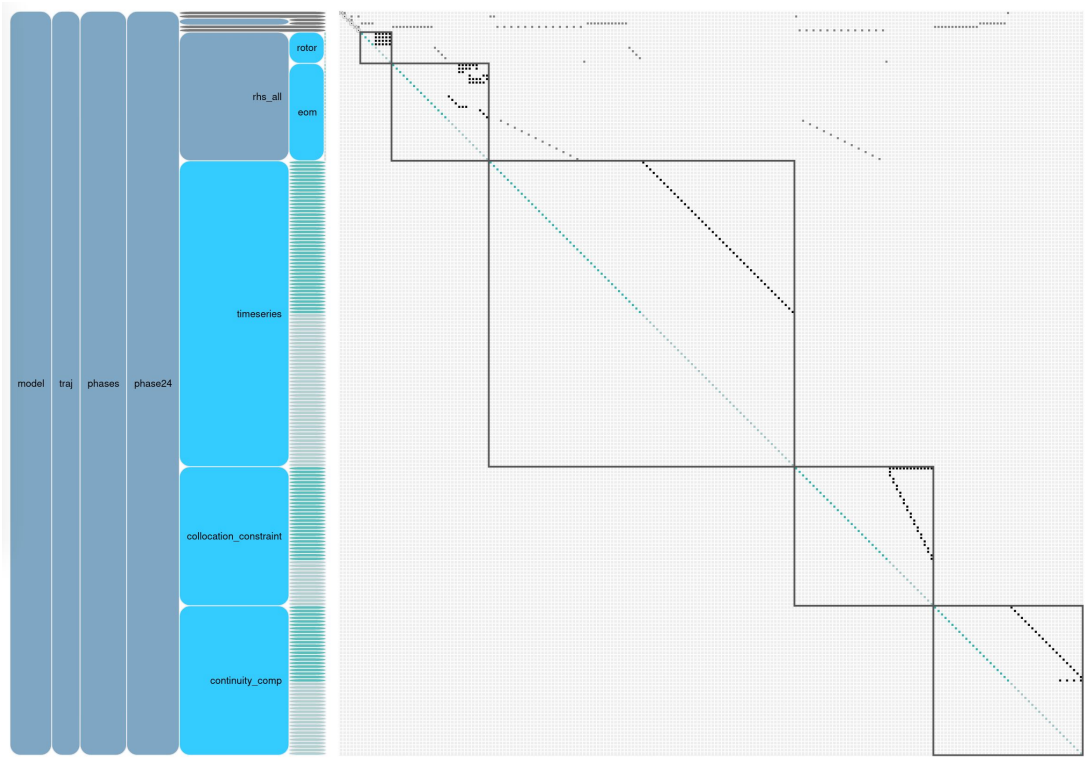


Figure 3.8. N^2 diagram of structure of the 24th phase's timeseries and collocation constraints of a solution trajectory.

3.4.4 Collision Check and Trajectory Modification

Once a solution trajectory is obtained our OpenMDAO program, we will need to guarantee it does not collide with the walls or obstacles. Also, any of the points in this trajectories should not exceed the indoor space boundary. The reason we need to revisit this step is that we do not add collision-free constraints to this multi-phase trajectory problem. Although the path planning algorithm does guarantee collision-free paths, the optimized trajectory needs to be checked and modified if necessary.

We first obtain the occupancy map of our indoor space as shown in Fig. 3.9, and create the state space based on this occupancy map. Then we run a fixCollsion() algorithm to sequentially recompute the segments with invalid states by adding intermediate waypoints and attempting to find minimal-time or minimal energy sub-trajectory between these waypoints. This fixCollsion() function also recompute the adjacent segments by varying their initial or final conditions. We iterate through every segment with invalid states until all the states on the trajectory become valid. After finishing the trajectory modifications, we recompute the mechanical energy consumption and time consumption of the modified trajectory and compare it with that the original trajectory result. We found that the variations are usually less than 5 % of the original objective function values.

3.5 Results and Simulations

In this section, we present the results for three mission scenarios and compare the effectiveness of our mission-oriented optimal control solutions with those from other methods. At first, We show the optimized result of a quadrotor to fly from origin to the alcove to display its ability of computing the rotor RPM profile (i.e. rotory speed profile of each rotor). Next, we simulate the scenario where the UAV needs to travel across each of the waypoints obtained by the coverage path planning algorithm. Last but not least,

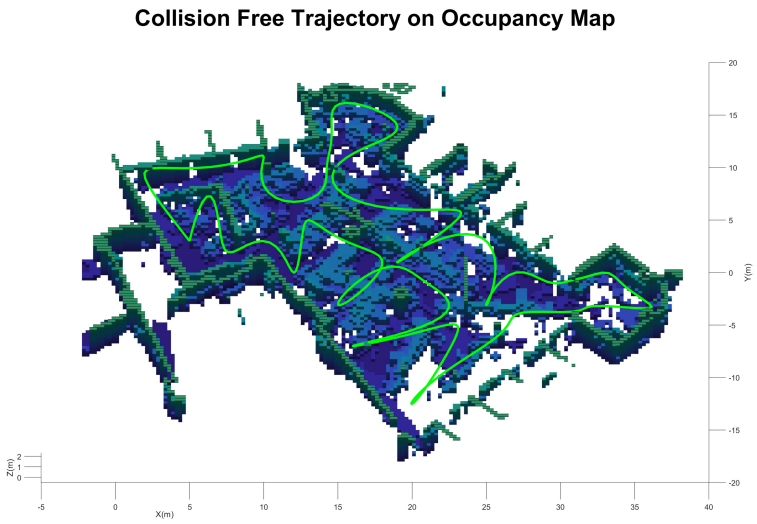


Figure 3.9. Illustration of the collision check function.

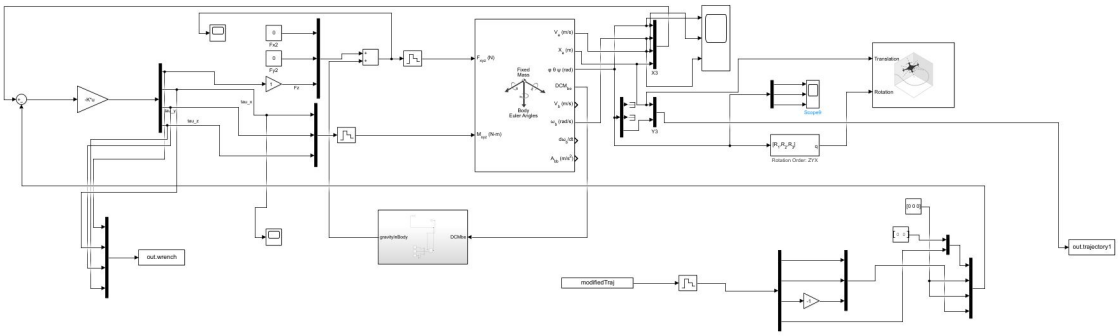
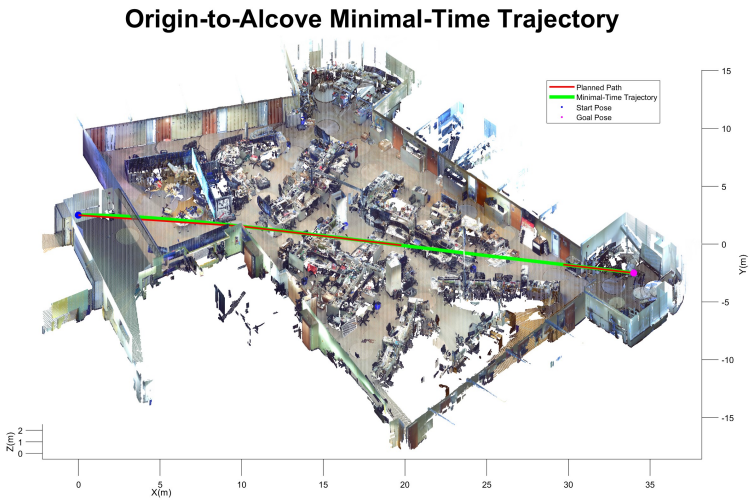


Figure 3.10. Simulink setup to simulate various trajectories for a quadrotor UAV.

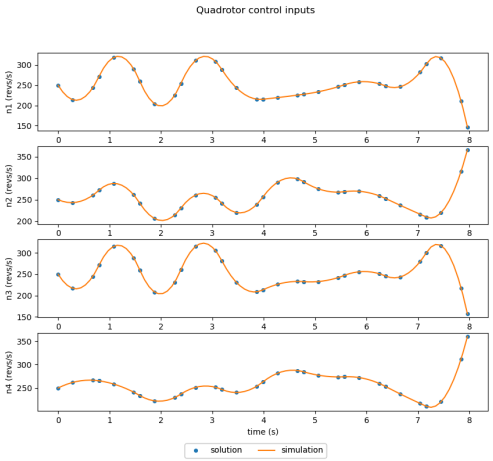
we execute the room visiting paths and compare the time consumption between proposed methods and results of other trajectory generation methodology.

3.5.1 Quadrotor Parameters and Simulation Settings

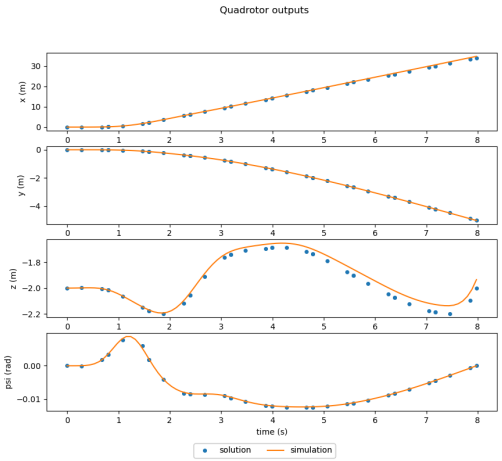
The model of the quadrotor UAV we use in this case study is an ModalAI M500 drone. It has a propeller diameter of 6 inches (0.1524m), an arm distance (body center to propeller center) of 0.225m, and gross take-off weight without payload to be 0.69kg. In



(a) Minimal-time trajectory from origin to alcove.



(b) The rotor speeds profile of the trajectory.



(c) Quadrotor output of the trajectory.

Figure 3.11. Origin-to-alcove minimal-time trajectory.

addition, we manage to measure its moments of inertia around its three body-fixed frame axes to be $I_x = 4.69 \times 10^{-2}\, kg \cdot m^{-2}$, $I_y = 3.58 \times 10^{-2}\, kg \cdot m^{-2}$, $I_z = 6.73 \times 10^{-2}\, kg \cdot m^{-2}$, respectively. And the moment of inertia of each rotor is found to be $I_r = 3.357 \times 10^{-5} kg \cdot m^{-2}$. In addition, we assume the air density to be uniformly $\rho = 1.225 kg/m^3$, and gravitational acceleration to be $g = 9.80 m \cdot s^{-2}$. With these parameters, we are able to set up the rigid

body dynamics in the simulink program in Fig. 5.9, and have the optimizer to compute the UAV trajectory accordingly.

In order to validate and compare various quadrotor trajectories in a virtual setting, we set up the simulation environment in Simulink with the help of its UAV toolbox [Mat21]. This setup helps execute the UAV missions with our control solutions in various virtual 3D maps while computing power consumption and realistic UAV paths to compare with other trajectories.

3.5.2 Origin-to-Alcove Path

The first solution trajectory of the quadrotor flying from origin to the alcove is illustrated in Fig. 3.11a. The maximum air speed is set to be $5m/s$, and this mission takes the UAV 7.97 seconds to fly from start position ($[0.0, 0.0, 2.0]$) to the goal position ($[34.0, -5.0, 2.0]$) in the alcove with the minimal time trajectory. In comparison, the minimum-snap trajectory generation algorithm [MK11b] renders a flight time of 8.43 seconds, and a linear-quadratic-regulator based path follower will take 8.53 seconds to arrive at the goal position. For this flight mission, the proposed optimizer decreases the flight time by 5.45% from the minimum-snap trajectory. In addition, the fixCollision() function detects no collision on this trajectory.

3.5.3 Coverage Path – Open Space Search Mission

The search mission with coverage path of the lab space case study is based on the 90% cells coverage path from Fig. 3.3. The scenario of this mission is to search for victims within the space or cells covered by the path. In this mission, we set the objective to be mechanical or rotor energy consumption. In order to make the trajectory optimizer converge faster, we smooth the path in Fig. 3.3 to get a set of 30 waypoints. Thus, our mission-oriented trajectory optimizer partition the mission into 30 phases and manages to

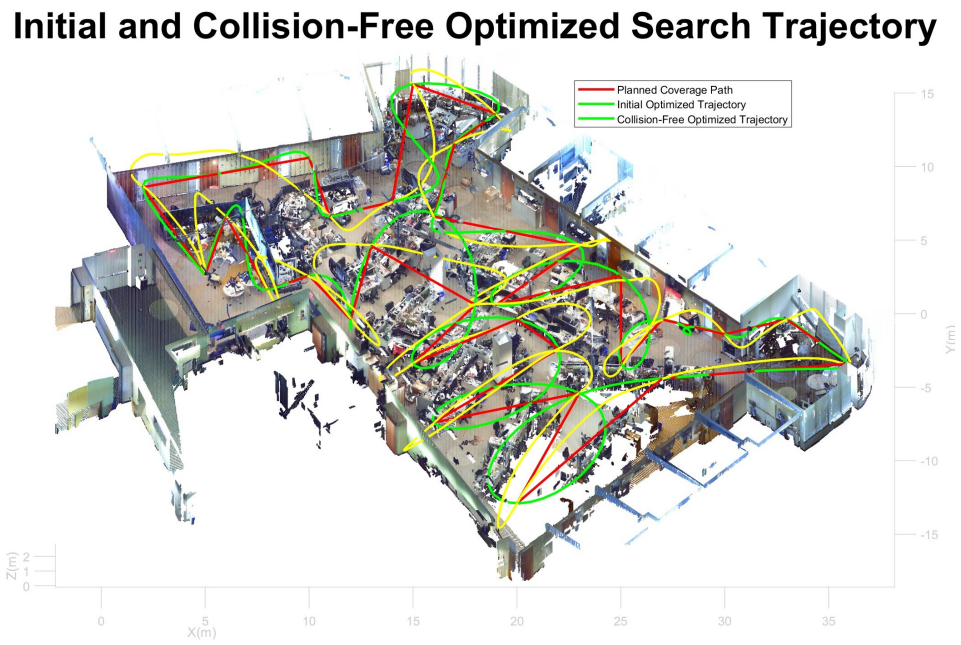


Figure 3.12. Initial and collsion-free open space search trajectory obtained by proposed optimizer.

find a trajectory with associated control inputs through out the path. The initial solution is shown in Fig. 3.11c with invalid state points. Thus fixCollsion() function is run to modify the path and eliminate the invalid states. The original invalid trajectory has a total mechanical energy consumption of $12,392$ J, and the energy consumption increases to $13,074$ J after running fixCollsion(). For this flight mission, the proposed optimizer decreases the mechanical energy consumption by 3.00% from the minimum-snap trajectory. The illustration of the invalid and modified trajectory is shown in Fig. 3.12.

We compare the total flight time, mission energy consumption, trajectory length, and computational time of three trajectory solvers in Table 3.1. Although the effectiveness of the proposed approach is validated, its computational time is considerably longer than the other two, and the energy saving is not significant compared to the minimum snap path generator. The future goal of this optimizer can be to make the software more light-weight and eliminate the need to apply fixCollision() algorithm to save computational time.

Table 3.1. Comparison of proposed optimizer, Minimum snap and linear quadratic path tracker performance

	t_f (s)	$E(t_f)$ (J)	Trajectory length (m)	Computational time (s)
Proposed	77.40	**13,074**	**183.74**	55
Minimum snap	**76.30**	13,479	188.32	41
LQR	92.35	15,967	203.34	**real-time**

3.5.4 Multi-Room Search Path – Rooms Visiting Mission

The last mission trajectory interests the authors is the task where the quadrotor needs to visit 3 rooms connected to the open lab space to confirm the presence of victims in any of the three rooms. The objective of this mission is time consumption since saving time is paramount in search-and-rescue tasks and it is desirable to be aware of numbers and locations of victims within a shortest possible time. In this mission simulation we remove all door obstacles in the free configuration space such that the quadrotor is able to enter rooms through the doors. The UAV is required to start from $[0.0, 0.0, 2.0]$ and visit room 1 at $[8.0, 12.0, 2.0]$, room 3 at $[31.0, -1.25, 2.0]$, and at last arrive at room 5 ($[13.0, 10.0, 2.0]$).

The trajectory generated by proposed optimizer is shown in Fig. 3.13. Both the length and time consumption of the proposed minimal-time trajectory and standard minimum-snap output do not deviate significantly from each other. With $V_{max} = 3m/s$, the optimizer-rendered trajectory has a total length of $68.82m$ and time consumption of $25.22s$, and minimum-snap trajectory has a total length of $67.34m$ and time consumption of $27.03s$. For this flight mission, the proposed optimizer decreases the flight time by 6.69%. In addition, the fixCollision() function detects no collision on this trajectory.

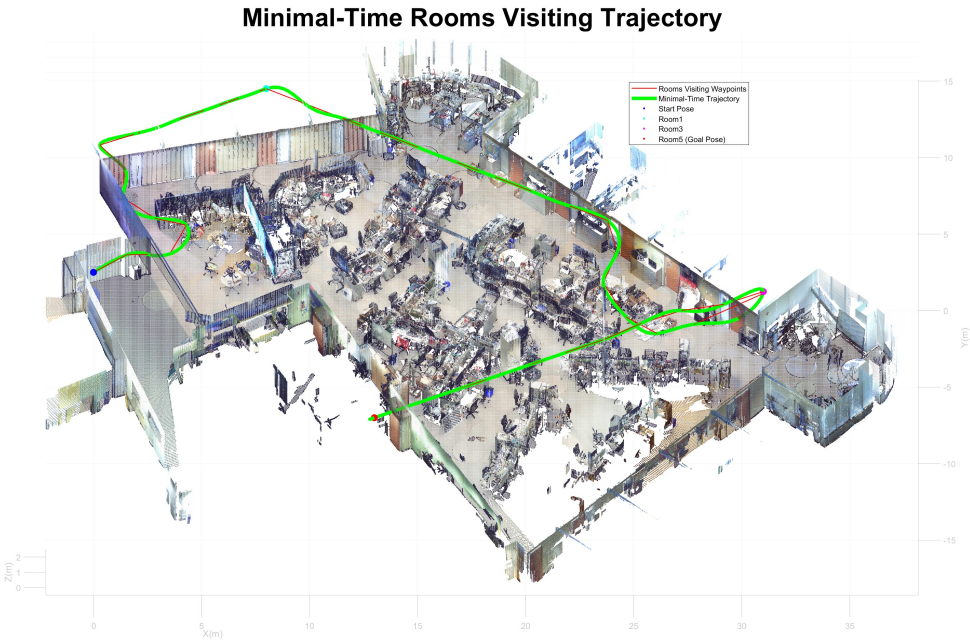


Figure 3.13. Minimal-time trajectory in rooms visiting mission.

3.6 Chapter Conclusion

This paper proposes a novel approach to optimize trajectories with respect to mission specific objective functions. The optimal control problem is solved using nonlinear programming approach which is widely used for solving MDO problems. The equivalence of optimal control and MDO is discused where the control input, objective functions, and trajectory output of the entire mission can be viewed as design variables to obtain an optimum for a quantitative performance. As expected, the MDO approach is able to minimize time or energy consumption of pre-designed missions to greater extent than do the common control approaches but with higher computational time.

The up-to-date progress only allow for minimizing mechanical energy and time consumption throughout the flight mission, but more types of objective functions can will be explored in the future work. For example, one can minimize the cumulative errors

between flight trajectory and waypoints to improve path tracking, or average magnitude of UAV body rotation rates to better stabilize the camera pose for aerial imaging.

This trajectory optimization methodology is planned as stage 1 of a simultaneous design and control optimization for quadrotor UAVs serving search and rescue purposes. Future work will involve the formulation of a large-scale design optimization problem to optimize both the design parameters and mission trajectories for SAR-deployable quadrotors. In addition, the authors would want to improve the current algorithm to lower its computational cost. Last but not least, both simulation and experimental validation of the proposed trajectory optimizer are to be performed to evaluate the future results and be compared with peers' solutions.

3.7 Acknowledgements

This chapter, in full, is a reprint of the material as it appears, titled "Mission-Oriented Trajectory Optimization for Search-and-Rescue Multi-rotor UAVs in Cluttered and GPS- Denied Environment", in *Proceedings of AIAA Aviation 2022 Forum*. Cao, Pengcheng; Hwang, John T; Bewley, Thomas; Kuester, Falko. The dissertation author was the primary investigator and author of this paper.

This chapter is based on work supported by the US Army Corps of Engineers under research Cooperative Agreement W912HZ-17-2-0024 and under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and by the LLNL-LDRD Program under Project No. 20-SI-005. The authors thank all collaborators at the Qualcomm Institute and the Contextual Robotics Institute of UC San Diego, as well as those at Flow Control & Coordinated Robotics Labs (FCCR) and Large-Scale Design Optimization Lab (LSDO) of UC San Diego to ideas, suggestions and comments. Opinions, findings, and conclusions from this study are those

of the authors and do not necessarily reflect the opinions of the research sponsors.

Chapter 4

Design of a Bio-Inspired Multi-Modal UAV with Compliant Wings

4.1 Chapter Abstract

Multirotors have become the most popular UAV or aerial robot category due to their structural simplicity and ability to take off and land vertically. However, most multirotors suffer from short airborne time and range due to limited battery capacity. Thus we propose a novel hybrid multirotor design, QuadGlider, to increase its capability of traveling long distances with minimal battery consumption. QuadGlider is inspired by the body structure and gliding mechanism of gliding animals. The airframe design of QuadGlider imitates flying squirrels' skeletal and muscular structure with implementation of servo motor to actuate its compliant membrane wings. Therefore, it can transition from quadrotor flight mode to forward gliding mode via morphing and lowering motor speeds to save power. In this work, we first present the conceptual design of QuadGlider. Next, we model its flight dynamics in different gliding scenarios. Then, the design is verified in computational fluid dynamics simulations of gliding scenarios with angles of attack of $0° - 30°$. At last, preliminary gliding experiments are conducted at low Reynolds numbers of around $Re = 4.7 \times 10^5$. The equilibrium glide simulation gives a maximum glide ratio of $4.27 : 1$ at a takeoff velocity of 14.0 m/s, while the experimental result indicates a glide

ratio of 2.97 : 1 at a takeoff velocity of 4.17 m/s.

4.2 Introduction

In recent years various micro and small unmanned aerial vehicles (UAV) are developed and deployed to perform aerial tasks where compact sizes are required (of $\leq$ 2-meter dimensions) or close flight range ($\leq$ 50 km) can meet such needs [FGS22]. The micro to small-sized UAVs, depending on the ways they produce lift, can be further categorized into fixed-wing, rotary-wing, and flapping-wing [UYB21]. However, the needs of realizing multiple flight modes on a single UAV have triggered the emergence of a number of multi-modal and hybrid designs. Some of the multi-modal UAVs featuring propulsion of tiltable propellers include fully-actuated multirotors [BD18][Zhe+20][Str+21b][Str+21a][Cao+21] and tilt-rotor quad-plane UAVs [YH09][KFL06][Flo+12]. Apart from tilt-rotors, fixed-wing vertical take-off and landing (FW-VTOL) UAVs [DBU20][Car14][Ozd+14][Tya+17] are developed to take off, land, and hover as a rotary-wing drone but perform fixed-wing cruise flight during most of missions. The other FW-VTOL hybrid designs include tail-sitter UAVs [Sto04][KS08][KWC18] which take off and land vertically on their tails and pitch forward to transition to cruise flight after taking off. The aforementioned design ideas, however, incorporated rigid wings for lift generation which generally required more mechanical power to lift off the aircraft compared to their compliant or flexible counterparts [ETM10][KA17]. Moreover, actuable compliant wing surfaces were proved to be advantageous over conventional rigid ailerons in terms of increasing roll control authority [Vos+07][Di +17].

In fact, various novel UAV designs with compliant membrane wings were proposed and studied thanks to bio-inspirations. Gerdes et al. [Ger+14] developed the Robo Raven which is a bird-inspired flapping-wing UAV which is able to actuate each of the two highly

compliant wings independently. Abdulrahim et al. [AGL05] presented a micro aerial vehicle design with morphing membrane wings to validate the roll control scheme of twisting the wing tips instead of using ailerons. These two designs indicate that membrane wings can be successfully applied on micro to small UAVs to generate lift but each incorporates only one single flight or locomotion mode. Shin et al. [SPP18][SPP19] presented a multi-modal robot inspired by *Pteromyini* or flying squirrels which is able to achieve both locomotion modes of walking and gliding while can also control its angle of attack using robotic legs and tail during the flight. This design, however, cannot take off and land vertically or hover in its airborne mode. Karasek et al. [Kar14] [Ros+17] presented the designs of several robotic hummingbirds capable of performing near-hover and forward flapping flight. Ma et al. [MFW12][Ma+13] presented the design and control of Havard Robobee which is a coin-sized micro aerial robot which can perform vertical take-off, hovering, and maneuvering of its flight directions. However, although flapping wings could have higher efficiency than did propellers in hover propellers [Chi+20], flapping-wing UAVs consumed significantly higher power than fixed membrane-wing UAVs during cruise or glide since flapping would increase the induced drag considerably [Sac16].

This paper presents the initial design work of the QuadGlider which is a multi-modal UAV with four rotors and retractable membrane wings. One of the key innovations of this design is the hybrid of a bio-inpired glider UAV with compliant wings and a quadrotor which can take off and land vertically and hover in the air. In its glider mode, QuadGlider is able to perform stable high-angle-of-attack (HAoA) gliding with low aspect ratio (LAR) of wings and at low Reynolds number ($< 10^6$). The primary contributions of this work are listed as follows: first, we review and compare the flight mechanisms of flying squirrels and flying lizards both capable of gliding flights as the main sources of bio-inspiration. Second, we propose the conceptual design of the quadrotor-glider with membrane wings and morphing mechanism. Third, we presents the modeling and simulation of its flight

dynamics in two gliding scenarios. Last but not least, we validate the gliding ability of this design via computational fluid dynamics (CFD) simulations and preliminary experimental tests.

The structure of this paper is therefore: Section 4.3 introduces the gliding animals inspiring the proposed design; Section 5.3 presents the conceptual design of the wings, the fuselage, and the morphing mechanism; Section 4.5 derives the equations of motion and flight dynamics of QuadGlider in different flight scenarios; Section 4.6 presents the CFD simulations of gliding QuadGlider at various AoAs; Section 4.7 presents the results of a preliminary gliding test to validate the design; Section 4.8 delivers the conclusion and discussion of future work.

4.3 Background of Bio-inspiration

In this section we will introduce the animals which provide bio-inspiration for this research project. On one hand, we will introduce *Pteromyini* tribe or flying squirrels in terms of their body structure and gliding kinematics. On the other hand, we will review *Draco* genus or flying lizards for their gliding ability and retractable wings.

4.3.1 Flying Squirrels

Pteromyini is a tribe of around 50 species of squirrels under the *Sciuridae* family which can utilize their muscles and membrane wings to perform gliding in unstructured environment like forests. Despite their misleading commonly known name as flying squirrels, *Pteromyini* cannot achieve full flight like bats or birds. However, their body structure can help them produce lift for them to transport from higher tree branches or leaves to lower ones by gaining lateral speed component from aerodynamics and gravity. It is learned from previous research [Bah+13] that some *Pteromyini* don't perform gliding in force equilibrium during most of their flight as shown in Fig. 4.1. The sum of lift, drag,

and gravity will produce a non-zero net force for the flying squirrel to change its gliding trajectory, and such net force is modulated by fast re-deployment of membrane wing using its muscles. For QuadGlider, we will utilize servo motors to actuate morphable membrane wings in order to produce similar gliding ability as simulated in Section 4.5.4.

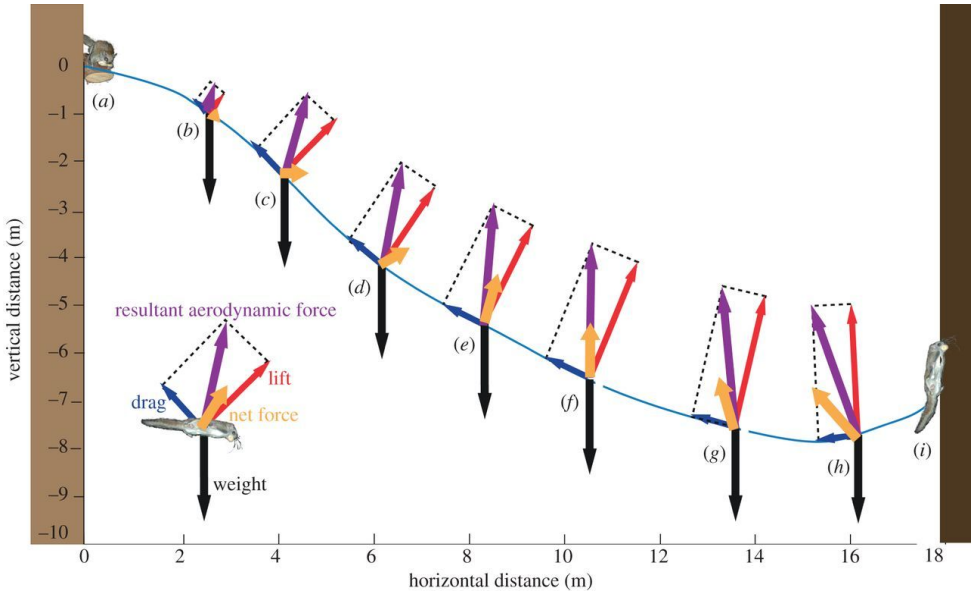


Figure 4.1. Simulation of a flying squirrel's non-equilibrium gliding flight to land on a tree (figure reused from [Bah+13] with permission of Royal Society Publishing).

In addition, *Pteromyini* have evolved several body features for better maneuvering the posture for gliding. Patagia are the membrane wing structures attached to the limbs of a gliding mammal. For flying squirrels, two plagiopatagia stretch between each pair of fore limb and hind limb as shown in Fig. 4.2 and they are the main source of lift in the flight. Apart form two plagiopatagia, a flying squirrel also has two pieces of propatagia between each fore limb and its head, and two pieces of uropatagia between each hind limb and its tail. These extra membranous muscles can improve its agility as well as decrease the aspect ratio (to around 1.0) which allows flying squirrel to glide at higher angle of attack and produce more drag to decelerate for gentler landing. Moreover, the platysma, tibiocarpallis, and semitendinosus are a group of muscle structures located at the edges of the membrane

wings to help hold their shapes and reduce undesired membrane fluctuations and edge fluttering during gliding. And the existence of wingtips, on the other hand, will produce wingtip vortices in gliding to reduce the induced drag exerted on the body [TDA98].

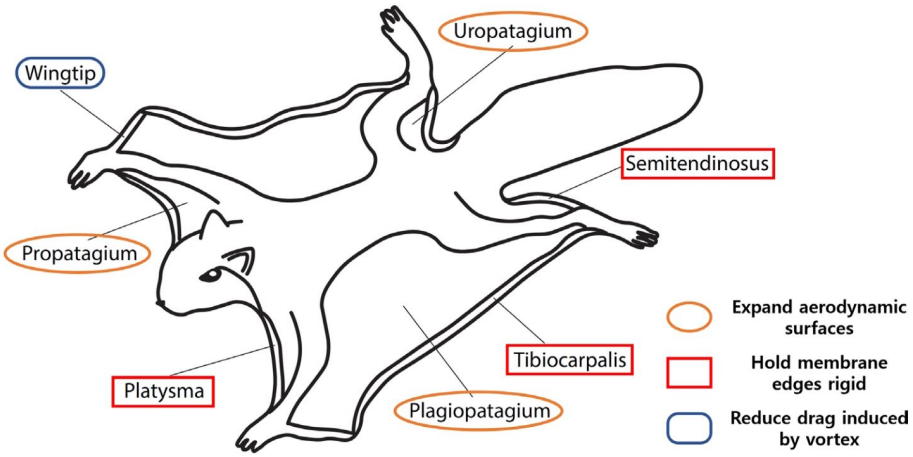


Figure 4.2. Body structure schematic of a flying squirrel (figure reused from [SPP18] with permission of IEEE Explore).

4.3.2 Flying Lizards

Draco or flying lizards are a genus of lizards under *Agamidae* family which is capable of gliding and landing on trees with retractable membrane wings. Again, flying lizards cannot achieve full flights but can utilize their patagia supported by elongated ribs to generate lift for gliding [Her58]. Different from flying squirrels, *Draco* fold their membrane wings using their muscles after landing and unfold them when they leap away to take off. In addition, some research indicates that flying lizards can achieve equilibrium or quasi-equilibrium glide [MD05] by observing a plateau in measured velocity profiles. A newer work, however, claims that such equilibrium glide was not observed in their gliding experiments with flying lizards [KH22]. And in the same paper Khandel and Hendrick proposed a novel approach for measuring the real-time body posture and airfoil orientation and camber in gliding by utilizing only 5 motion capture landmarks. In this work we adopt

the ideas of reaching equilibrium glide and simplified model of aircraft configurations from aforementioned studies of flying lizards' gliding behavior.

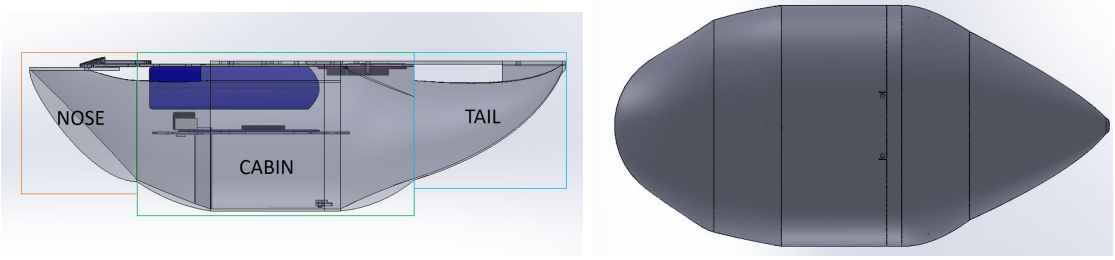
4.4 Conceptual Design

In this section we will introduce the design concept of multiple components of the QuadGlider prototype. The basis of QuadGlider is a quadrotor with 4 DC motors and 10-inch propellers as thrusters, and each of the rotors is positioned at one of the four corners of a $540mm \times 540mm$ square. However, in order to achieve better aerodynamic performance as a quadrotor-glider, a number of original quadrotor components need to be altered. We will first present the fuselage design and explain the design principles for it. Next, we will discuss the design of flexible membrane wings and the bio-inspiration behind this idea. Last but not least, we will present the morphing mechanism serving the UAV's multi-modes and explain the four-bar linkage mechanism for wings expansion and retraction.

4.4.1 Fuselage Design

The fuselage design of QuadGlider is generally attributed to the following 3 principles. First, the geometry of the fuselage is designed to be spacious enough to house the avionics and Li-Po battery while being inspired by fuselages of VTOL and fixed-wing aircrafts instead of adopting the common square-shaped quadrotor fuselage. Second, the fuselage cross-sections can serve as part of overall airfoils to modulate airflow for stable gliding. Last but not least, the fuselage plays a great role in weight distribution of the aircraft such that the CoM is set to be ahead of the neutral point (NP) for pitch stablility concerns [SPP18] and beneath the wing chord lines to enhance roll stability which is inspired by the anatomy of flying squirrels [Zha+19].

The preliminary fuselage design of QuadGlider is shown in Fig. 4.3. The critical



(a) Right view. **(b)** Bottom view.

Figure 4.3. QuadGlider fuselage design.

dimension constraints of the fuselage are the width and length of the Holybro S500 power distribution board, which are 117.9 and 196.8 mm, respectively. Therefore, the maximum width of the cabin is set to be 160.4 mm, and maximum height 89.1 mm making this widest section of cabin half of a elliptic cylinder. And the fore-to-aft longitudinal length of the fuselage is 329.5 mm from nose's start to tail's end. The reason to have a wide cabin is on one hand to house a 14.8 V 4S Li-Po battery, a BeagleBone Blue flight controller board, and an electronical speed controller (ESC) board as shown in 4.3a and on the other hand to include another support board from Holybro S500 kit (119.8×119.8 mm). With its shell manufactured with glycol modified polyethylene terephthalate (PETG) via 3D-printing, the mass of fuselage is 633 g, taking 66.9% of the total mass of the vehicle (946 g). And the CoM position is measured to be 38.3 mm ahead of and 15.8 mm below the neutral point of the membrane wings which conforms to the roll stability margin [SPP19] and makes the CoM stay inside the fuselage.

4.4.2 Membrane Wings Design

The wings design of QuadRotor is featuring bio-inspired compliance and inherent roll stability for gliding. On one hand, the bio-inspired compliant membrane wings, mimicking the flight behavior of flying mammals and flying lizards, can achieve a higher lift slope and stall at much higher angle of attack compared to their rigid counterparts [Son+08]. On

the other hand, some birds' flying behavior indicates that a positive dihedral angle can largely enhance the roll stability when gliding at low Reynolds number ($10^3 \sim 10^5$) [TT01]. As shown in Fig. 4.5b, the sideslip or banking will introduce a lateral airflow velocity component U_∞. Due to positive wing dihedral, the effective angle of attack α_{eff} of the lower wing with respect to U_∞ will be smaller than that of the upper wing. Therefore, having the same airfoil geometry, the lower wing can produce a larger lift than do the upper wing, resulting in a roll moment in the opposite direction of the roll angle therefore hindering the increase of roll rate.

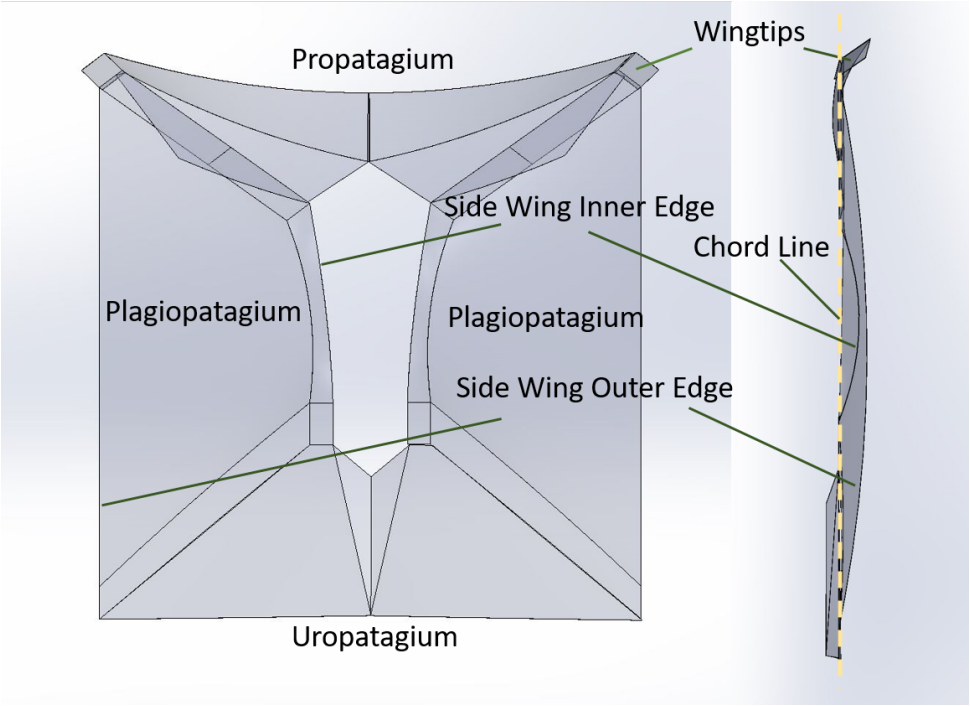
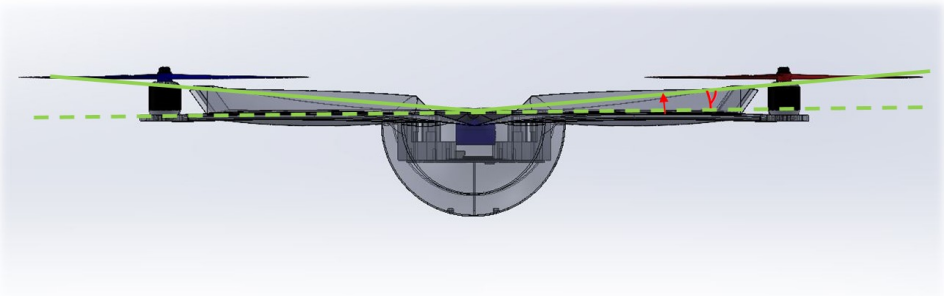
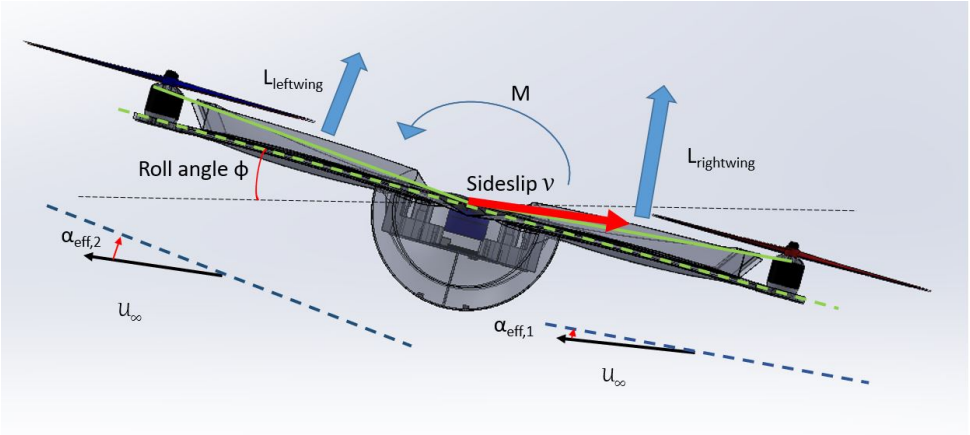


Figure 4.4. Membrane wing components.

The composition of the membrane wing is illustrated in Fig. 4.4 with both top view and left view. Analogous to the wing components of a flying squirrel, we have one Propatagium or fore membrane, two Plagiopatagia or side wings, one Uropatagium or aft membrane, and two wingtips located at the end of two front arms which is useful for reducing induced drag by generating wingtip vortices [SPP19]. We manufactured each



(a) Positive wing dihedral.



(b) Illustration of inherent roll stability. Here the right wing will generate more lift than does the left wing in a sideslip v therefore producing a roll moment countering the increase of roll rate.

Figure 4.5. QuadGlider wing dihedral.

membrane using ripstop nylon which is a lightweight woven fabric resistant to tearing and ripping and often used for hot air baloons and wingsuits. In addition, each wing edge is embedded with aluminum wire to hold the wing shape and reduce membrane fluctuations. The total area of membrane wings is measured as 270,515 mm^2, while the total mass is merely 103 g including the aluminum wires. As shown in the left view of Fig. 4.4, we can approximately find the position of the chord line and therefore simplify various membrane wing airfoils to be a non-symmetrical cambered airfoil for further analyses in pitching and gliding in XZ-plane.

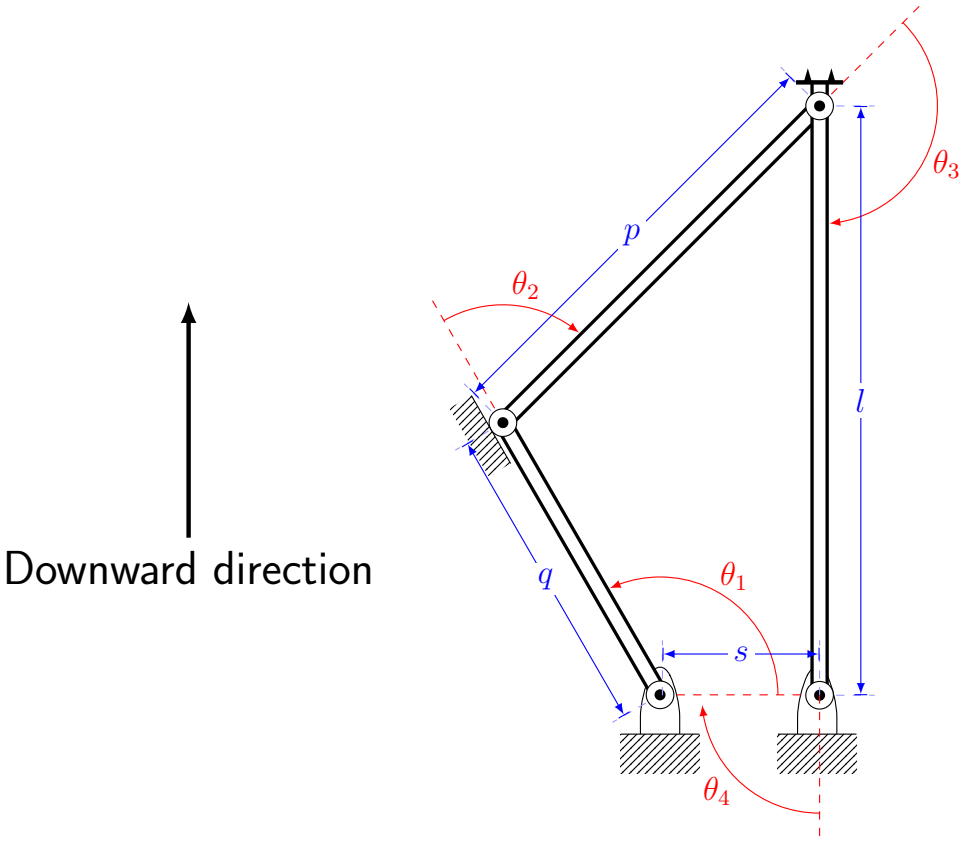
In addition, the shape of fortified membrane wings can form a positive wing dihedral

angle of $\gamma = 5.1\,\text{deg}$. This helps the QuadGlider resist roll moment disturbance and restore the roll angle to zero during low Reynolds number gliding. However, the roll stability cannot be assured merely by the wing dihedral, and instead both the rotors and morphable wings can serve as robust source of roll moments making the QuadGlider design advantageous in maintaining roll stability.

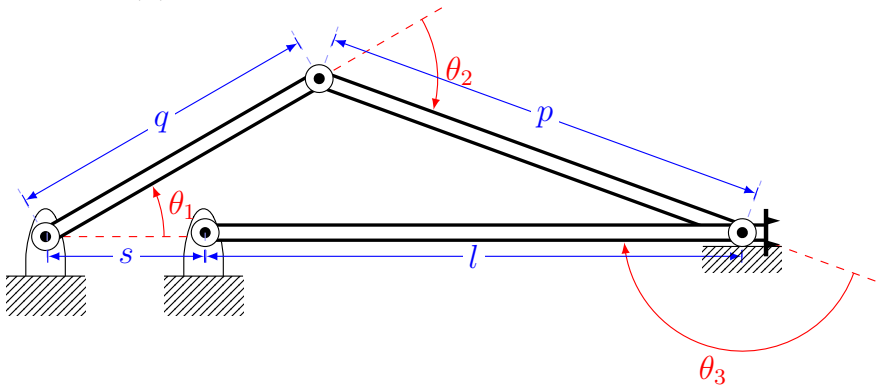
4.4.3 Morphing Mechanism

Morphing between QuadGlider's quadrotor mode and glider mode is realized via four 4-bar linkages attached to each of the rotor arms. Beneath each rotor arm, a drag-link 4-bar linkage mechanism is installed to expand and retract wings to change the effective membrane wing areas under different flight modes. The illustration of the 4-bar linkage mechanism used here is shown in Fig. 4.6. We design the frame or fixed link to be of $s = 40mm$ and the output link or wing expansion bar to be of $l = 150mm$. The input link or crank is set to be $q = 80mm$ and the coupler link is $p = 122mm$. Therefore this mechanism satisfy $s + l < p + q$ ensuring continuous circular motions for both crank and output link. A servo motor is used to drive q-bar or input link at the left fixed hinge in Fig. 4.6a. And the actuable part of membrane wings is attached to all four l-bars or wings expansion bars pointing towards each of the rotors. The linkage will form a quadrilateral in quadrotor mode to face the bottom of landing gear towards downward direction as shown in Fig. 4.6a. And it will also form a triangle in glider mode as shown in Fig. 4.6b. With these two scenarios as two extreme positions, QuadGlider has the ability to vary the effective lift-generating wing area continuously between these two positions to resist disturbance or perform maneuvers during gliding.

With aforementioned design elements and morphing mechanism introduced in this section, we can now present the assembly of QuadGlider prototype in both full-wing glider mode and quadrotor mode as in Fig. 4.7.



(a) Quadrotor and landing mode 4-bar position.



(b) Glider mode 4-bar position.

Figure 4.6. 4-bar linkage configurations.

(a) Demo of 4-bar linkage on a rotor arm.

(b) fully expanded glider mode.

(c) Quadrotor mode.

Figure 4.7. QuadGlider design assembly.

4.5 Flight dynamics

In this section, we want to investigate the flight dynamics of QuadGlider's both modes of flight, namely quadrotor mode and glider mode. The commonly-used quadrotor

equations of motion (EoM) is first presented and followed by variations for transitioning into glider mode. While in terms of QuadGlider's glider mode dynamics, its gliding motions are categorized by whether the vehicle is at force equilibrium or not and also by its different configurations. As inspired by both flying lizards [MD05] and flying squirrels [Bah+13], QuadGlider is designed to develop a constant velocity with fixed gliding angles as well as break the force equilibrium by morphing the wing shape to land on and perch to inclined surfaces.

For the analyses in gliding flights, we make the following 2 assumptions to simplify the dynamical model: (1) we formulate all the analyses on the 2-dimensional sagittal or longitudinal plane of the UAV. By doing this we focus only on the pitch dynamics and stability which primarily affects the lift (C_L) and drag (C_D) coefficients in its flight. However, although QuadGlider's left-right symmetry ensures inherent lateral force balance, we will still discuss its lateral static stability to withstand sideslip disturbance. (2) We assume all the wing surfaces to be planar and only count their projections on the XY-plane of QuadGlider's body frame as wing surface areas. And we assume the chord lines of thin airfoils of QuadGlider's membrane wings are parallel to the XY-plane.

4.5.1 Flight with Mode Transition

QuadGlider is designed to be a multi-modal UAV which can transition between quadrotor and glider flight modes. In order to smoothly perform mode transitions, we incorporate the morphing configurations as well as dynamics-based flight controls. On one hand, the morphing mechanisms has been presented and discussed in Section 4.4.3. On the other hand, the transition-related flight dynamics is investigated and discussed in this section.

We start the derivation of QuadGlider's flight dynamics with defining the coordinate systems. First, we define the Earth-fixed inertial frame $\mathcal{F}_E : \{\mathrm{O}_E\text{–}\mathrm{X}_E\mathrm{Y}_E\mathrm{Z}_E\}$. The Earth-

fixed inertial frame $\mathcal{F}_E$ denotes the inertial frame of reference fixed on the Earth and its origin and X, Y, and Z-axes align with those of the pre-flight body frame, respectively. Second, the body frame $\mathcal{F}_B : \{\mathrm{O}_B\text{–}\mathrm{X}_B\mathrm{Y}_B\mathrm{Z}_B\}$ is fixed on the vehicle body and its origin coincides with the CoM. The three axes of body frame conform to the NED convention [CCL11a], with X-axis pointing out the nose of the vehicle, Y-axis pointing out right, and Z-axis pointing out the bottom. Last, the gliding frame $\mathcal{F}_G : \{\mathrm{O}_G\text{–}\mathrm{X}_G\mathrm{Y}_G\mathrm{Z}_G\}$ also has its origin coinciding with the CoM, while its X-axis points out the direction of gliding velocity, Y-axis coincides with Y_B and Z-axis points out the opposite direction of its lift which is perpendicular to gliding velocity. The presence of gliding frame $\mathcal{F}_G$ helps compute the lift and drag being exerted on the UAV CoM with respect to its gliding direction. The simplified right-view illustration of coordinate systems is showed in Fig. 4.8 with a focus of studying XZ and pitch motions.

The flight dynamics of a quadrotor UAV has been thoroughly studied in the literature [Cao+22a] [Sab15], such that we only adopt the final nonlinear form of the quadrotor EoM as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g_1}(\mathbf{x})\mathbf{u_1} + \mathbf{W}_{ext}, \tag{4.1}$$

where the 12-dimensional state vector and 4-by-1 quadrotor input wrench vector are

$$\begin{aligned} \mathbf{x} &= [\mathrm{x\ y\ z}\ \psi\ \theta\ \phi\ \dot{\mathrm{x}}\ \dot{\mathrm{y}}\ \dot{\mathrm{z}}\ \mathrm{p\ q\ r}]^T, \\ \mathbf{u}_1 &= [\mathrm{F_z}\ \tau_{\mathrm{x}}\ \tau_{\mathrm{y}}\ \tau_{\mathrm{z}}]^T, \end{aligned} \tag{4.2}$$

respectively. x, y, z, and $\dot{x}$, $\dot{y}$, $\dot{z}$ are the displacement and velocity components in the $\mathcal{F}_E$, respectively. And ψ, θ, ϕ are the rotational displacement components in $\mathcal{F}_E$.p, q, r are rotational rates with respect to x, y, z-axes of the $\mathcal{F}_E$. And $\mathbf{W}_{ext}$ represents the

disturbance of external aerodynamic wrenches caused by wind. However, for QuadGlider in mode transition, we need also incorporate its aerodynamic wrenches exerted by its membrane wings. These forces are considered as the second control input $\mathbf{u_2}$.

Accordingly, we model the aerodynamic forces exerted on the CoM of the vehicle during its glide. Similarly to the force expressions in [SPP19], we express the force vector in $\mathcal{F}_G$ as

$$\mathbf{u}_2 = \begin{bmatrix} -D_G \\ 0 \\ -L_G \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\rho C_D V_E^2 S_w \\ 0 \\ -\frac{1}{2}\rho C_L V_E^2 S_w \end{bmatrix}, \tag{4.3}$$

where D_G and L_G are the magnitude of drag and lift in $\mathcal{F}_G$, and their minus signs indicate both are pointing towards the negative X_G and Z_G directions, respectively. ρ is the density of air which we consider as 1.204 kg/m^3 throughout this paper. V_E indicates the magnitude of gliding velocity in $\mathcal{F}_E$ and is equivalent to $(\dot{x}^2 + \dot{z}^2)^{1/2}$ as the UAV is assumed to only travel in XZ-plane. S_w is the total wing surface area and in equilibrium glide, the wings are fully expanded and their projected areas sum up to 0.1785 m^2. C_D and C_L are drag and lift coefficients, respectively, and according to CFD simulation results in Section 4.6, at a fixed Reynolds number of 4.7×10^5 both coefficients vary with angle of attack α (AoA) and can be approximated with following two polynomials:

$$\begin{aligned} C_L(\alpha) &= 0.7830 \cdot \alpha^3 - 3.8915 \cdot \alpha^2 + 3.9464 \cdot \alpha + 0.2660 \\ C_D(\alpha) &= 0.9854 \cdot \alpha^2 - 0.2190 \cdot \alpha + 0.1935, \end{aligned} \tag{4.4}$$

with AoA α in radians. With α and the UAV pitch angle θ, one can compute the gliding angle θ_G by

$$\theta_G = \alpha - \theta \tag{4.5}$$

where θ_G is the gliding angle, i.e. the direction of gliding velocity and angle between X-axes of $\mathcal{F}_E$ and $\mathcal{F}_G$. During equilibrium glide, θ_G can also by computed by taking the arc-tangent of the ratio between the vertical and horizontal velocity component in the XZ-plane, i.e. $\theta_G = \text{atan}(\dot{z}/\dot{x})$.

Therefore we can derive the rotation matrices between 3 different coordinate systems. $\mathbf{R}_{EG}$ and $\mathbf{R}_{BG}$ are rotation matrices from gliding frame $\mathcal{F}_G$ to $\mathcal{F}_E$ and $\mathcal{F}_B$, respectively, defined as

$$\mathbf{R}_{EG} = \begin{bmatrix} \cos\theta_G & 0 & \sin\theta_G \\ 0 & 1 & 0 \\ -\sin\theta_G & 0 & \cos\theta_G \end{bmatrix}, \mathbf{R}_{BG} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}. \tag{4.6}$$

All in all, the Newton-Euler equation of motion of QuadGlider can be written as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g_1}(\mathbf{x})\mathbf{u_1} + \mathbf{g_2}(\mathbf{x})\mathbf{u_2} + \mathbf{W}_{ext}, \tag{4.7}$$

and the details of EoM terms $\mathbf{f}(\mathbf{x})$, $\mathbf{g_1}(\mathbf{x})$, and $\mathbf{g_2}(\mathbf{x})$ are introduced in Eqn. 4.21.

4.5.2 Equilibrium Glide

Next we can obtain the Newton-Euler equations of motion in the earth-fixed frame $\mathcal{F}_E$ to depict the equilibrium glide dynamics. According to the definition of equilibrium glide [Ste22], the desired glide angle can be calculated by lift-to-drag ratio at force balance:

$$\cot(\theta_G) = \frac{L_G}{D_G} = \frac{C_L(\alpha)}{C_D(\alpha)}. \tag{4.8}$$

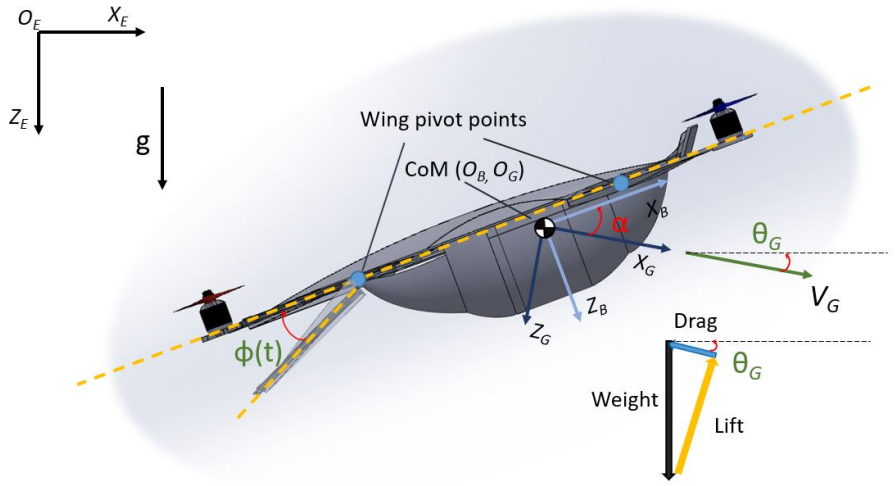


Figure 4.8. Definitions for QuadGlider flight dynamics and an example of equilibrium glide forces balance.

Therefore to describe both the translational motion in XZ-plane and rotational motion of body pitch, we have

$$m\ddot{\xi} = m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \mathbf{R}_{EG}\mathbf{u}_2 + \mathbf{F}_{ext}, \tag{4.9}$$

$$I_{yy}\ddot{\theta} = \{\mathbf{X}_{AC} \times \mathbf{R}_{BG}\mathbf{u}_2 + \tau_{ext}\}_y, \tag{4.10}$$

where m is the mass of the UAV and I_{yy} is the $y-y$ component of the UAV moments of inertia matrix. ξ is the UAV's position vector in $\mathcal{F}_E$ and θ_E is the body pitch angle with respect to the Y-axis of $\mathcal{F}_E$. Similar to the simplification of rigid body rotation of quadrotors [Cao+22a] [Sab15], the rotational acceleration of pitch angle $\ddot{\theta}$ is made equal to the time derivative of pitch rate q in body frame, and with yaw rate set to be zero, pitch moment $I_{yy}\ddot{\theta}_E$ is therefore identical to sum of torques around Y-axis. g is the gravitational

acceleration (9.807 m/s^2 in this paper). $\mathbf{X}_{AC}$ is the position of aerodynamic center for equilibrium glide located at the quarter-chord position between the leading and trailing edges of the wings ($[0.055m, 0, -0.016m]^T$ in the body frame). And $\mathbf{F}_{ext}$ and τ_{ext} are sum of external forces and torques caused by wind.

4.5.3 Non-equilibrium Glide

Non-equilibrium glide, mimicking the flight of a flying squirrel or *Draco*'s, is the gliding motion performed when there is non-zero total aerodynamic force or torque (not from wind disturbance) exerted at QuadGlider's CoM. This means either the lift, drag, and gravity will not sum up to zero or the airflow around the current configuration will generate a torque for the body angles to change over time. To track a specific gliding trajectory in non-equilibrium, *Draco*, as indicated in [KH22], change their body shape and posture precisely to control and modulate the aerodynamic forces. In this section, the dynamics and body posture changes are studied for QuadGlider to follow a gliding trajectory analogous to *Draco*'s gliding behavior.

The coordinate systems used for non-equilibrium glide are identical to those used in Section 4.5.2. We keep use of the inertial frame $\mathcal{F}_E : \{\mathrm{O}_E\text{–}\mathrm{X}_E\mathrm{Y}_E\mathrm{Z}_E\}$, body frame $\mathcal{F}_B : \{\mathrm{O}_B\text{–}\mathrm{X}_B\mathrm{Y}_B\mathrm{Z}_B\}$, and gliding frame $\mathcal{F}_G : \{\mathrm{O}_G\text{–}\mathrm{X}_G\mathrm{Y}_G\mathrm{Z}_G\}$. However, in order to control QuadGlider's aerodynamic forces by changing its body shape, the UAV is able to change its effective wing areas attached to the rear half of the body by actuating the wing expansion bar as shown in Section 5.3. With the presence of a wing angle $\phi(t)$ between the front group and rear group of the wings, we model the aerodynamic forces of front and rear wing groups as

$$\mathbf{F}_{G,fw} = \begin{bmatrix} -D_{G,fw} \\ 0 \\ -L_{G,fw} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\rho C_D^{fw} V_E^2 S_{fw} \\ 0 \\ \frac{1}{2}\rho C_L^{fw} V_E^2 S_{fw} \end{bmatrix}, \tag{4.11}$$

$$\mathbf{F}_{G,rw} = \begin{bmatrix} -D_{G,rw} \\ 0 \\ -L_{G,rw} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\rho C_D^{rw} V_E^2 S_{rw} \\ 0 \\ \frac{1}{2}\rho C_L^{rw} V_E^2 S_{rw} \end{bmatrix}, \tag{4.12}$$

where fw and rw indicate the front and rear wings. We use Eqn. 4.4 to compute C_D^{fw} and C_L^{fw} as we now consider only the front wing groups remain fixed with the body frame. And due to the morphed wing shape, the drag and lift coefficients of rear wing groups C_D^{rw} and C_L^{rw} are therefore calculated by

$$C_L^{rw}(\alpha) = C_L(\alpha + \phi_{rw}), C_D^{rw}(\alpha) = C_D(\alpha + \phi_{rw}), \tag{4.13}$$

as wing angle increases the AoA of rear wing groups.

In addition, instead of having one aerodynamic center (AC) of wing groups, we now have 2 different aerodynamic centers which located at quarter-chord positions of front and rear wing groups. The front AC is at a fixed position $X_{AC,fw} = [0.110m, 0, -0.016m]^T$ in the body frame. The rear AC, however, will rotate around the CoM as wing angle $\phi_r w(t)$ changes such that

$$\mathbf{X}_{AC,rw} = \begin{bmatrix} \cos\phi_{rw} & 0 & \sin\phi_{rw} \\ 0 & 1 & 0 \\ -\sin\phi_{rw} & 0 & \cos\phi_{rw} \end{bmatrix} \begin{bmatrix} -0.140 \\ 0 \\ -0.016 \end{bmatrix}. \tag{4.14}$$

With aforementioned quantities, we can derive the equations of motion for non-

equilibrium glide simulation as

$$m\ddot{\xi} = m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \mathbf{R}_{EG}(\mathbf{F}_{G,f} + \mathbf{F}_{G,r}) + \mathbf{F}_{ext} \tag{4.15}$$

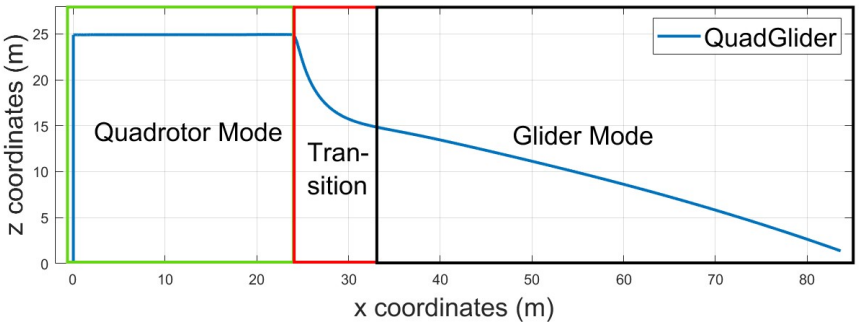
$$I_{yy}\ddot{\theta}_E = \{\sum_{fw}^{rw} \mathbf{X}_{AC,i} \times \mathbf{R}_{BG}\mathbf{F}_{G,i} + \tau_{ext}\}_y \tag{4.16}$$

In bio-inspired non-equilibrium glide, the QuadGlider is also designed to track a desired gliding trajectory as do *Draco* with a variable wing angle. During the non-equilibrium glide when the UAV is landing on a steep slope analogous to Draco or flying squirrels' tree perching, it needs to decrease its speed by shape changing and pitching up. Therefore a closed-loop control law is designed for it to track the desired trajectory in the landing phase and stabilize its pose as

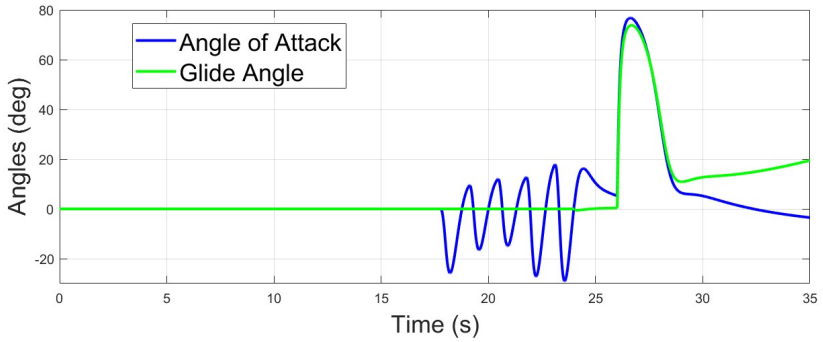
$$\begin{aligned} \dot{\phi}(t) = -k_1(\theta(t) - \theta_{des}) - k_2\dot{\theta}(t) - k_3(x(t) - x_{des}) \\ -k_4\dot{x}(t) - k_5(z(t) - z_{des}) - k_6\dot{z}(t). \end{aligned} \tag{4.17}$$

4.5.4 Flight Simulations

In this section, we simulate the equations of motion of transition flight, equilibrium, and non-equilibrium glides numerically using MatLab. First, we set up the scenario of the flight featuring transition from quadrotor flight mode to glider. The QuadGlider takes off from the ground and stops at a height of 25 m above ground level (AGL). Then it flies forward till it arrives at $[24.0, 0, 25.0]^T$ m and then starts transitioning to glider mode. The transition phase takes 2.81 seconds and the QuadGlider lost altitude of 10.1 m during this period. At last the quasi-equilibrium gliding phase takes 6.29 seconds and ends at



(a) Positions.

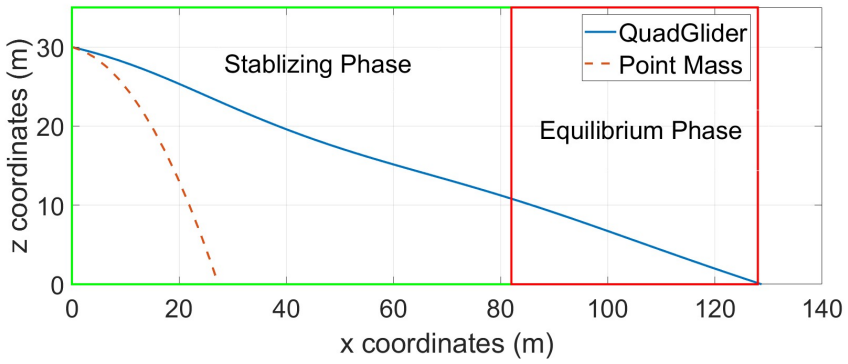


(b) Angles.

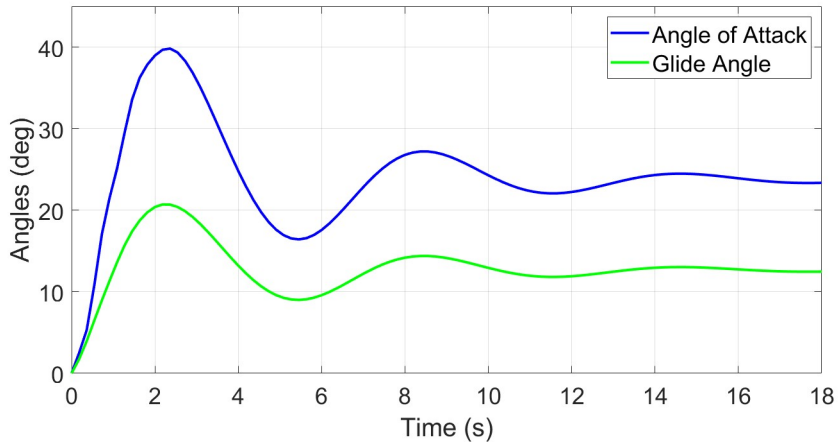
Figure 4.9. Flight with Transition Simulation.

$[83.65, 0, 1.38]^T$ m. The simulation result of this setup is shown in Fig. 4.9a and 4.9b.

Second, we simulate the equilibrium glide by first setting up the scenario where the QuadGlider is released at a height of 30 m AGL. Both the initial angle of attack and glide angles are zero, and the initial horizontal and vertical speed components are 10 m/s and 0, respectively. The effects of external wind are omitted in this simulation and therefore both F_{ext} and τ_{ext} are set to zero. The simulation result of this setup is shown in Fig. 4.10a and 4.10b. This glide can be divided into 2 phases, namely stabilizing phase and equilibrium phase. In the stabilizing phase ($t = 0 \sim 14s$, x-distance $= 0 \sim 82$m), the velocity vector increases from $[10, 0, 0]^T$ m/s to around $[12.28, 0, 2.61]^T$ m/s which is the equilibrium velocity. And AoA first overshoots the equilibrium AoA of 23.7 °to 40 °then stabilizes to the equilibrium AoA, and similarly the glide angle stabilizes to the equilibrium



(a) Positions.

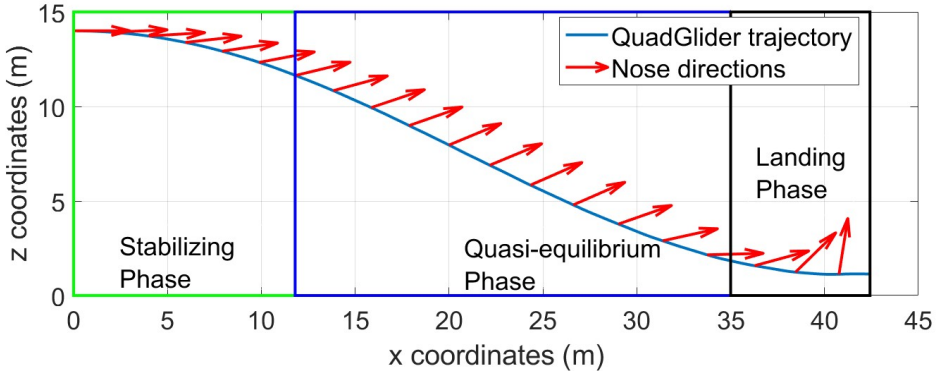


(b) Angles.

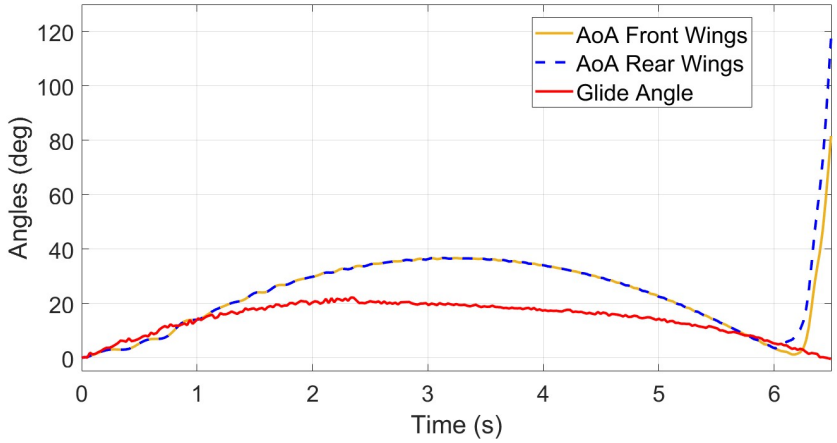
Figure 4.10. Equilibrium Glide Simulation.

glide angle of 12.5 °. And during the equilibrium phase ($t = 14 \sim 18s$, x-distance = $82 \sim 128$ m), the velocity vector remains within $\pm 0.5\%$ range of $[12.28, 0, 2.61]^T$ and both AoA and glide angle remain close to equilibrium as well. This simulation result meets the computed equilibrium AoA and glide angle θ_G from Eqn. 4.8. And the overall glide ratio of this simulation is 4.27 and the equilibrium-phase glide ratio is 4.71.

Last but not least, we run the simulation of non-equilibrium glide of QuadGlider. We set the initial altitude to be 14 m AGL, and again the initial AoA and glide angle to be zero. The initial horizontal and vertical speed components are 5 m/s and 0, respectively. The simulation result of this setup is shown in Fig. 4.11a and 4.11b. This simulation



(a) Positions.



(b) Angles.

Figure 4.11. Non-equilibrium Glide Simulation.

can be divided into stabilizing phase, quasi-equilibrium phase, and landing phase. In the stabilizing phase ($t = 0 \sim 2s$, x-distance $= 0 \sim 11.8$m), the velocity vector changes from $[5, 0, 0]^T$ m/s to around $[7.73, 0, 2.21]^T$ m/s which is a quasi-equilibrium velocity. And AoA of both front and rear wings increase from 0 to 30°and glide angle increases to 20°. Next in the quasi-equilibrium phase ($t = 2 \sim 5s$, x-distance $= 11.8 \sim 35$m) the AoA of both front and rear wings first increases to 38°and then decreases to 15°similarly to the angle change in the equlibrium glide simulation. And the glide angle increases to 20°and then decreases to 10°. Last, in the landing phase($t = 5 \sim 6.5s$, x-distance $= 35 \sim 42.4$m), the wing shape is kept changing according to the control law depicted in Eqn. 4.17. The AoA

of rear wings first decreases to 5°and then increases drastically to 118°at $t = 6.5s$, and similarly AoA of front wings decreases to 0 and then increases to 81°at $t = 6.5s$. Therefore we can see the wing angle ϕ becomes 37°at $t = 6.5s$. The velocity vector changes from $[8.66, 0, 2.01]^T$ at $t = 5s$ to $[2.18, 0, 0.87]^T$ at $t = 6.5s$ due to fast increased drag caused by surged AoA and pitching angle.

4.6 Aerodynamic Validation

4.6.1 OpenFOAM Software Package

OpenFOAM is a complex open-source software consisting of multiple CFD simulation tools, including Finite Volume Method (FVM) discretization schemes [JJT+07]. The package contains both Reynolds-Averaged Naiver-Stokes (RANS) and large eddy simulation (LES) model solvers, allowing for the simulation of a wide range of Reynolds and Rayleigh numbers accompanied with either turbulent or laminar flow. These solvers have numerous different discretization methods and schemes to calculate the gradient, divergence, Laplacian, and interpolation from the discretized mesh.

All of the simulations were run using OpenFOAM with the SimpleFoam tool, which is based on Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm; this solves the continuity equation, momentum equation, and corrected pressure equation [MMD16]. Since we used turbulence flow, the method we used was the $k - \omega$ model for prediction without gravity. The run time for the simulations was one second of simulation time with a write interval of .1 second. Finally, the method we choose to calculate the gradients, divergence, Laplacian, and interpolation were all Gauss linear.

4.6.2 Geometric Model Preparation and Mesh Generation

In Section 5.3, we described the development of the bio-inspired QuadGlider CAD model, which was prepared for OpenFOAM using MeshMixer. The process we used in

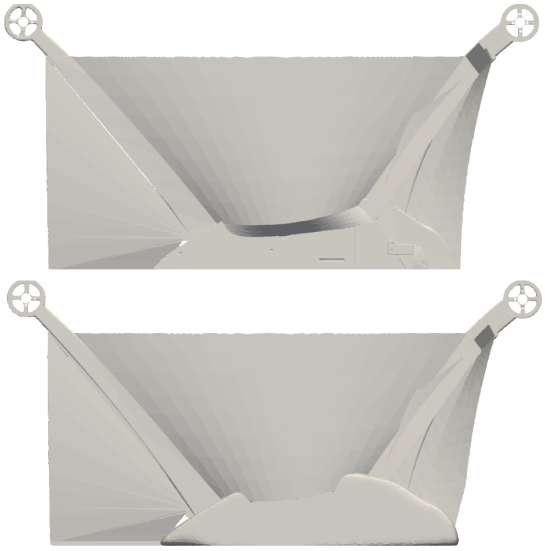


Figure 4.12. The complex mesh body is on top, while the simplified mesh is on the bottom.

MeshMixer was to separate individual components and repair issues when exporting the SOLIDWORKS file types to mesh file types. In order to preserve high-fidelity meshing results, each of the major components was selected to be in a separate group. This methodology was performed on two meshes: a complex and simplified mesh, demonstrated by Fig. 4.12. Since the CFD simulations diverge depending on the Courant number and intersecting cells, two different meshes were used to gather additional data about the convergence and refinement levels in the mesh refinement study. When running the mesh refinement study, the computational domain's resolution was tested, requiring multiple BlockMesh variants. Other than the resolution, no further computational domain testing was done; the two computational domains that were discretized into different cell resolutions are: (36 24 6) and (18 12 1) cell amounts, producing (33.33mm 18.75mm 48mm) (66.66mm 37.5mm 288mm) resolutions from the total rectangular domain of (1200mm 450mm 288mm), respectively.

After discretizing the rectangular computational domain, we paired each face with a specific boundary-condition type; for our simulations, all the boundary conditions had type wall except for the inlet and outlet, which are given the type patch with a velocity of 15m/s and 0m/s respectively. The inlet and outlet specifications dictated the Reynolds Number, from the equation:

$$\mathrm{Re} = \frac{uc}{v} \tag{4.18}$$

where v is the kinematic viscosity of air at 10°C, u is the inlet velocity, and c is the airfoil chord length. The airfoil's chord length is 447mm, yielding a Reynolds Number of 4.7×10^5.

4.6.3 Mesh Refinement Study

In order to choose an adaptive mesh refinement level converging under multiple angles of attack, we ran a mesh refinement study with various cell resolutions, cell refinement levels, and number of cells in each level. All of the mesh refinements were run with a 0 degrees angle of attack. As previously stated, the two initial discretization resolutions were (36 24 6) and (18 12 1); the refinement levels ranged from 5-8, and the number of cells in each level were either 2, 5, or 8. The entire computational domain was chosen to be in the refinement region, with an upper bound of ten million cells. After running each of the SnappyHexMesh options with both the complex and simplified QuadGilder model, the meshes were run with SimpleFoam using the same settings described in the previous subsection, checking the pressure convergence and meshing results. See Fig. 4.13 for an example of the meshed computational domain.

After running all the mesh refinement variations, the CheckMesh tool was used to validate the meshes and collect the cell amounts. For the initial higher resolution computational domain, the meshing levels achieved the global maximal cell count at

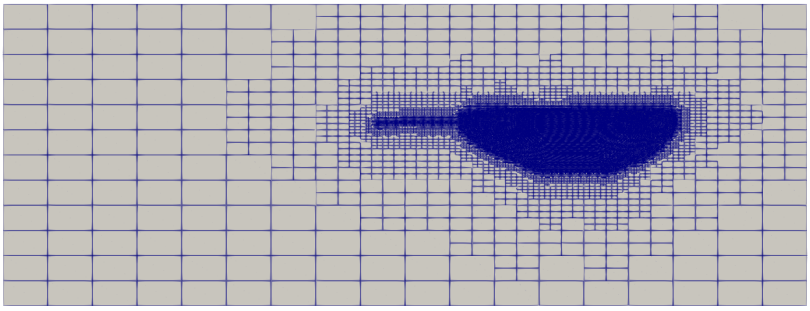


Figure 4.13. Mesh generated with SnappyHexMesh with a resolution of (18 12 1), a refinement level of 7, and 5 cells per level.

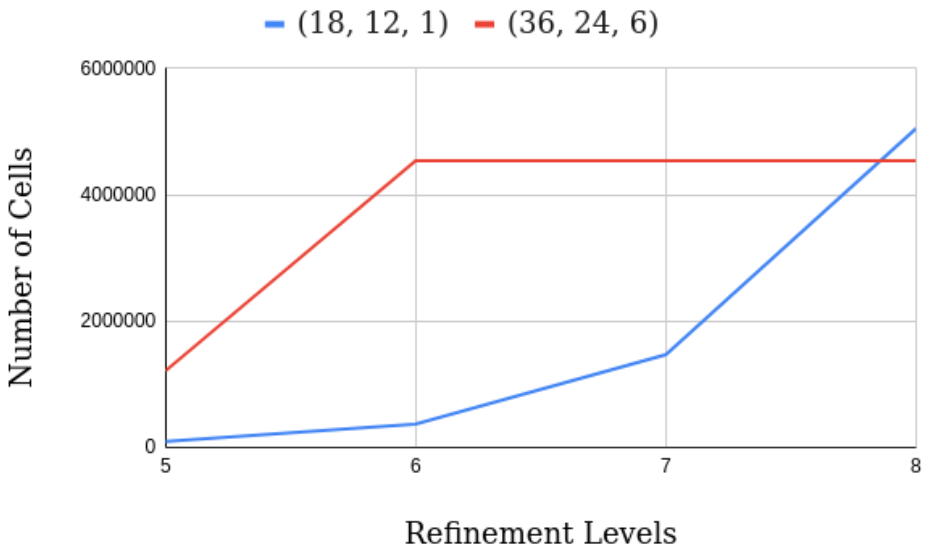


Figure 4.14. Graph of the refinement levels vs. number of cells with different initial resolutions.

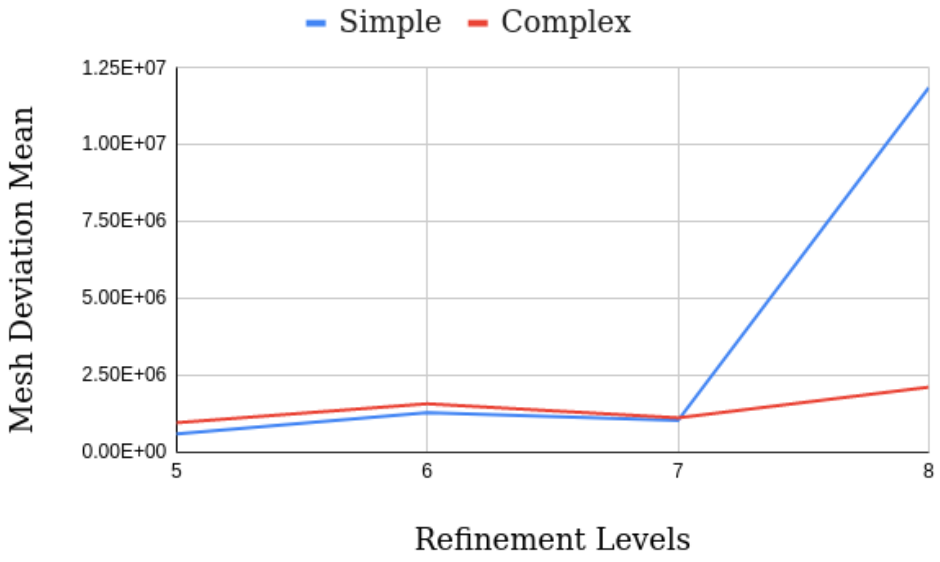


Figure 4.15. Graph of the refinement levels vs. the average deviation of all unique data.

refinement level 6, meaning that all additional refinements after level 6 were ignored; any setting tested similarly plateaued, in both the simple or complex mesh. Fig. 4.14 shows a graph of the number of cells compared to the refinement levels; all other cases followed similar trajectories. Although variations between the simplified and complex mesh were minimal, there was an average of 2.8% more cells in the complex meshes compared to the simplified ones. Therefore, there was little variation loss in the overall mesh topology between using the simplified or complex initial meshes.

After running the SimpleFoam simulations on all the different mesh settings, we used Paraview to extract the drag and lift forces acting on the airfoil. In order to compare these forces' convergence between different meshes, a minimal function was run on the standard deviation of subsets of the initial data of a single force, either lift or drag, with a cardinality 8 ; if there were less than 8 data points, then the function was taken to be 0. The equation is:

$$\min\{\sigma(X)|X \subset I, |X| = 8\}$$

where I is the initial data and σ is the method of calculating the standard deviation of a set. This was done for each setting on both the complex and simple mesh. We utilized and additional comparisons which entailed calculating the percent difference between the median force of the entire data with the medium of each simulation. In both comparison methods, the average of the two forces was then calculated and recorded. The table of the SnappyHexMesh study can be found in Table 4.1.

With all the data collected and calculated, there was a large difference between the refinement levels. The consistently worse performing refinement level was 8. In the complex mesh, there were three different settings where the refinement diverged before 80% of the expected data was collected. As apparent in Fig. 4.15, the data converged the best in refinement levels 5 and 7. From this data, along with the previous cell count data in Fig. 4.14, we decided to go with a refinement level of 7 instead of 5. Even though the convergence was better with level 5 than 7, there were significantly fewer cells present in the meshing, yielding coarser results. The specific settings we went with for our angle of attack simulation were: a low resolution (18 12 1) BlockMesh, 7 refinement levels, and 5 cells per level. This produced the best complex mesh results and better than median simplified mesh results, while the median percent change was small.

4.6.4 Simulation Results

During the mesh analysis, a range of settings were analyzed; where a low resolution BlockMesh with 7 refinement levels and 5 cells per level produced the best results. Also, we created the simplified mesh to produce data that converged over multiple different settings, which would yield higher quality data when run with different angles of attack. The angle of attack simulation kept the same the same model (simplified), mesh discretization scheme, and SnappyHexMesh settings. We ran a simulation in SimpleFoam with angles of attack varying 5 degrees from 0 to 30. These airfoil simulations allowed us to analyze the

pressure flow, coefficient of lift, and coefficient of drag at the different angles of attack.

From the airfoil simulations, we extracted streamline and contour plots to understand the behavior of QuadGlider while gliding at 15m/s wind speeds. Fig. 4.18b and 4.18a are the streamline plots with an angle of attack of 0. In the front and side streamline views at an angle of attack of 0, QuadGlider has relatively uniform laminar flow about the airfoil with a low pressure area on the upper surface near the wing tip. There were no present vortexes in the plot. Now compared with the plots for an angle of attack of 25 degrees in Fig. 4.19b and 4.19a, the underside of QuadGlider has less uniform flow, which ends up flowing out from the side of the membrane; this is different compared with 0 angle of attack. Also, the higher angle of attack streamline has a small vortex on the membrane's upper surface. Hence the higher angles of attack will lose more speed and glide less smoothly then the lower angles of attack.

While comparing only the streamline plots is useful, it is important to also look at the contour plots, highlighting pressure flow on a specific slice of the simulation. The contour plots in Fig. 4.18c and 4.19c have a difference with the uniformity and distribution of the pressure flow. During the 0 degree angle of attack contour, the pressure has a small pressure deviation throughout the computational domain. On the other hand, the 25 degree angle of attack contour has larger variations on the leading edge of QuadGlider: the bottom of the membrane is high pressure, while the low is high pressure. Another difference is the regions of high and low pressure. While the 0 degree angle of attack contour has a high pressure region at the leading edge with no low pressure region, the 25 degree angle of attack contour has a high and low pressure region at the leading edge with the whole upper surface of the airfoil being low pressure. Hence, a lower angle of attack produces more uniform airflow.

After collecting the lift and drag force from the data, we calculated the coefficient of lift (C_L) and the coefficient of drag (C_D) for each angle of attack. The equations we

used were:

$$C_L = \frac{2L}{\rho V_{air}^2 S_w}, \tag{4.19}$$

$$C_D = \frac{2D}{\rho V_{air}^2 S_w}. \tag{4.20}$$

With L and D as the force of lift and drag, respectively. Also, ρ is the air density of 1.204 kg/m^3, V_{air} is the velocity of air 15 m/s, and S_w is the sum of the QuadGlider's effective wing areas.

Now with the equations described above and the force data, we found the coefficient of lift and drag for each angle of attack. This data is present in Fig. 4.16. In Fig. 4.16c, any angle of attack between 0 and 30 imply that there will be a greater than 1 C_L/C_D, meaning the lift will be greater than the drag on the QuadGlider. These data points, along with measured AoA and equilibrium gliding angle in Section 4.7, are further fitted to polynomials and used to define the coefficients of Eqn. 4.4 to build a prediction model of C_L and C_D for various AoA.

4.7 Experimental Result

In this section we will introduce the preliminary experimental results of QuadGlider in its gliding mode. Although the primary focus of this work is the design concept and aerodynamic validation of QuadGlider, we are able to test its gliding ability by hand launching in an indoor testing setup. As expected, the QuadGlider is capable of entering equilibrium glide phase as predicted in Section 4.5 with fully expanded wing surface area.

Fig. 4.17 shows the time sequential trajectory extracted from the video footage of one of the experiments at every 0.20 s after the moment of launching till the UAV lands on the floor at 1.24s. The CoM position of each frame was estimated by mapping the UAV rotor positions in each frame to the ones of the original model as the rotor arms are

considered rigid. We use linear interpolations here to estimate the UAV positions between 2 frames and connect these positions using light blue line segments. The launch velocity is computed as 4.17 m/s by measuring the frame positions. If we assume the QuadGlider enters equilibrium gliding phase in the last 0.6 seconds before landing, we calculate the X velocity component to be 3.68 m/s and Z velocity component 2.67 m/s at equilibrium. Therefore the gliding angle is 36.0 °and the ratio of C_L/C_D is 1.3745 according to the definition of equilibrium gliding. Using Eqn. 4.4 we solve for the AoA $\alpha = 54$ °of this equilibrium phase which corresponds well with our observation in this experiment. The overall glide ratio from launching to landing is $2.97 : 1$ but the glide ratio in equilibrium phase is only $1.38 : 1$. The other gliding experiments with QuadGlider we conducted with various but roughly controlled initial velocities also indicate this correspondence. In this case, we validate the effectiveness of our modeling for gliding.

4.8 Chapter Conclusion

We hereby reach the conclusion of this chapter. First, we reviewed the body features and gliding behavior of two types of gliding animals and acquired bio-inspirations from the relevant studies. Second, design features including morphing wings, inherent roll stability, and 4-bar linkages were presented and discussed as components of conceptual design. Third, the computational fluid dynamics studies using OpenFOAM indicated the airfoil of QuadGlider would be able to modulate airflow and produce sufficient lift at $Re = 4.7 \times 10^5$ at relatively low angles of attack $(0 - 30°)$. Last but not least, the preliminary experiment validated QuadGlider's ability of developing a equilibrium glide after being launched at 1.6m AGL, and the estimated lift and drag coefficients model corresponded well with the experimental results.

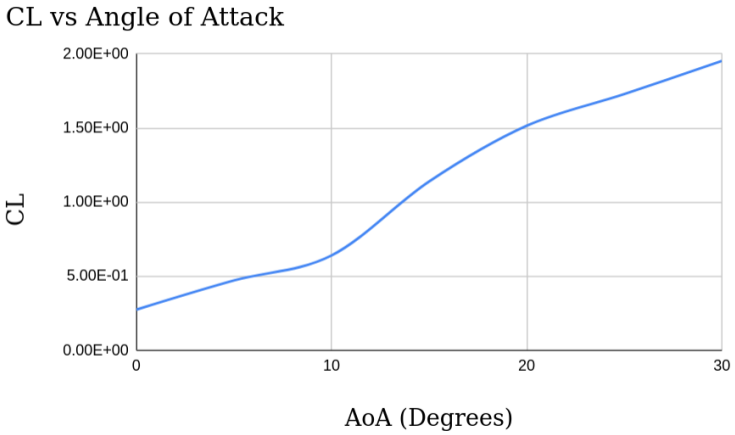
For future work, the authors would want to first investigate the potential existence of

multiple equilibrium types including stable and non-stable equillbria at which Quadglider can glide to compute and achieve control authority over the desired horizontal speed in the hope of robust and reliable transition control design. In addition, the input-to-output dynamical system needs to be modeled from the servo and DC motor inputs to QuadGlider's kinematic potentially using system identification methods or analytical dynamics modeling. Last but not least, the author would also want to achieve robust gliding control with active disturbance resistance.

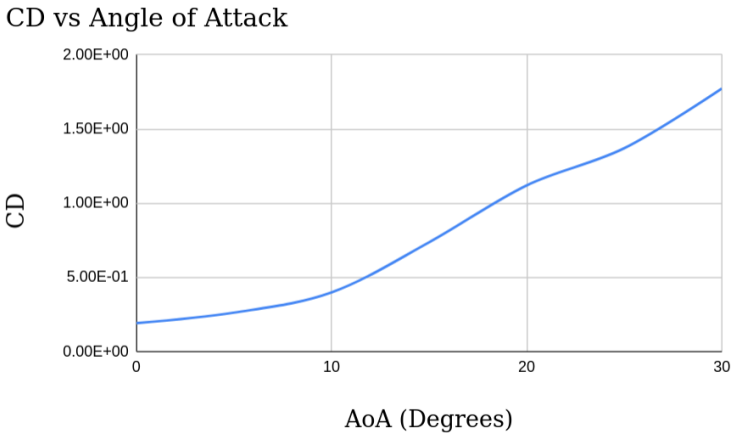
4.9 Acknowledgments

This chapter, in full, is a reprint of the material as it appears, titled "QuadGlider: Towards the Design and Control of a Bio-Inspired Multi-Modal UAV with Compliant Wings", in *Proceedings of 2023 IEEE Aerospace Conference*. Cao, Pengcheng; Phillips, Joseph; Bewley, Thomas; Kuester, Falko. The dissertation author was the primary investigator and author of this paper.

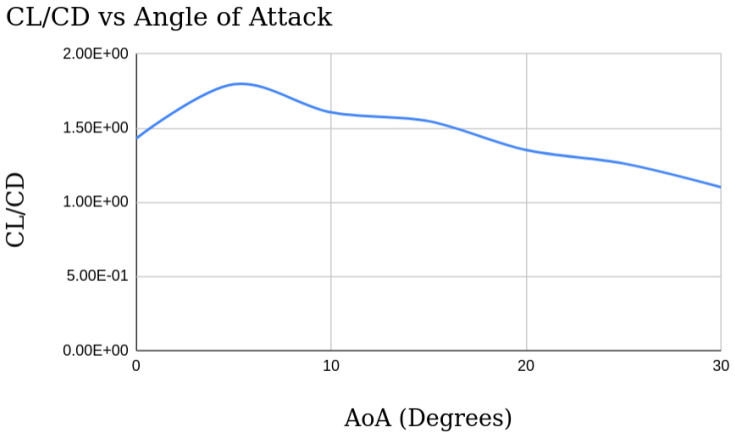
This chapter is based on work supported by the US Army Corps of Engineers under research Cooperative Agreement W912HZ-17-2-0024 and under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and by the LLNL-LDRD Program under Project No. 20-SI-005. Additional support was provided by the Kinsella Expedition Fund. We thank all collaborators at the Qualcomm Institute, and the Contextual Robotics Institute, UC San Diego, as well as all other contributors to ideas, suggestions and comments. Opinions, findings, and conclusions from this study are those of the authors and do not necessarily reflect the opinions of the research sponsors.



(a) Coefficient of lift vs. angle of attack graph.



(b) Coefficient of drag vs. angle of attack graph.



(c) Coefficient of lift over coefficient of drag vs. angle of attack graph.

Figure 4.16. Variation of lift and drag coefficient for $0 - 30$°angle of attack.

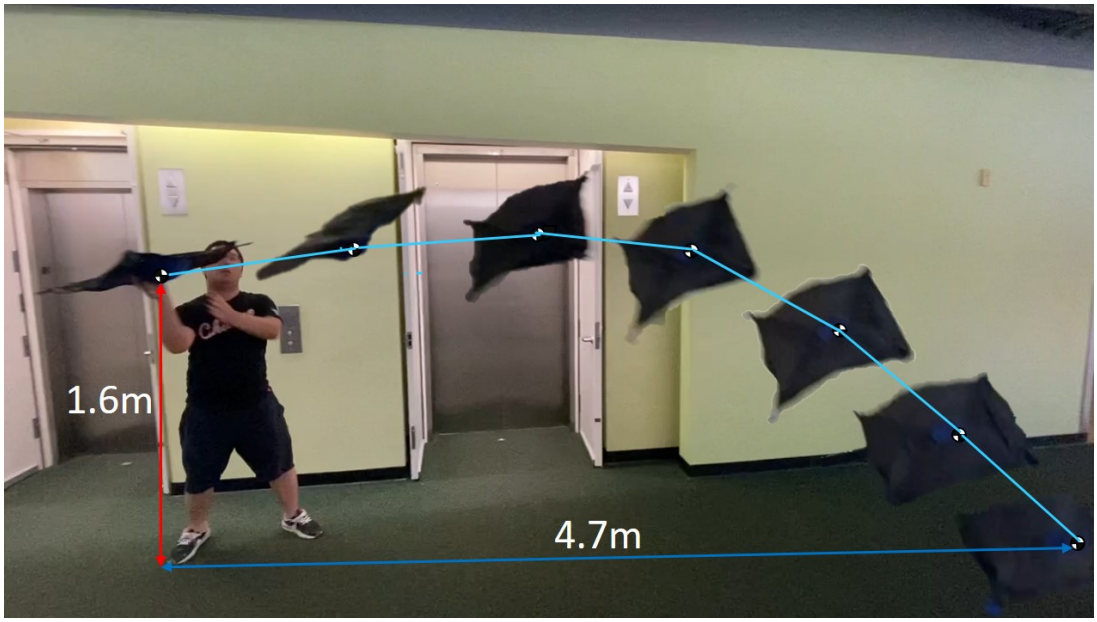


Figure 4.17. Time sequential trajectory of a hand-launched gliding experiment.

4.10 Appendix

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g_1}(\mathbf{x})\mathbf{u_1} + \mathbf{g_2}(\mathbf{x})\mathbf{u_2} + \mathbf{W_{ext}},$$

$$\text{where } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ q\frac{s(\phi)}{c(\theta)} + r\frac{c(\phi)}{c(\theta)} \\ q \cdot c(\phi) - r \cdot s(\phi) \\ p + q \cdot s(\phi)t(\theta) + r \cdot c(\phi)t(\theta) \\ 0 \\ 0 \\ -g \\ \frac{I_y - I_z}{I_x} qr \\ \frac{I_z - I_x}{I_y} pr \\ \frac{I_x - I_y}{I_z} pq \end{bmatrix}, \mathbf{g_1}(\mathbf{x}) = \begin{bmatrix} 0^{6\times1} & 0^{6\times1} & 0^{6\times1} & 0^{6\times1} \\ g_1 & 0 & 0 & 0 \\ g_2 & 0 & 0 & 0 \\ g_3 & 0 & 0 & 0 \\ 0 & 1/I_{xx} & 0 & 0 \\ 0 & 0 & 1/I_{yy} & 0 \\ 0 & 0 & 0 & 1/I_{zz} \end{bmatrix} \quad (4.21)$$

$$\text{and } \mathbf{g_2}(\mathbf{x}) = \begin{bmatrix} 0^{6\times1} & 0^{6\times1} & 0^{6\times1} \\ c(\theta_G) & 0 & s(\theta_G) \\ 0 & 1 & 0 \\ -s(\theta_G) & 0 & c(\theta_G) \\ 0 & 0 & 0 \\ z_{AC} \cdot c(\alpha) + x_{AC} \cdot s(\alpha) & 0 & z_{AC} \cdot s(\alpha) - x_{AC} \cdot c(\alpha) \\ 0 & 0 & 0 \end{bmatrix}, \text{ additionally}$$

$$
\begin{aligned}
g_1 &= -\frac{1}{m}[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] \\
g_2 &= \frac{1}{m}[c(\psi)s(\phi) - c(\phi)s(\psi)s(\theta)] \\
g_3 &= -\frac{1}{m}[c(\phi)c(\theta)].
\end{aligned}
$$

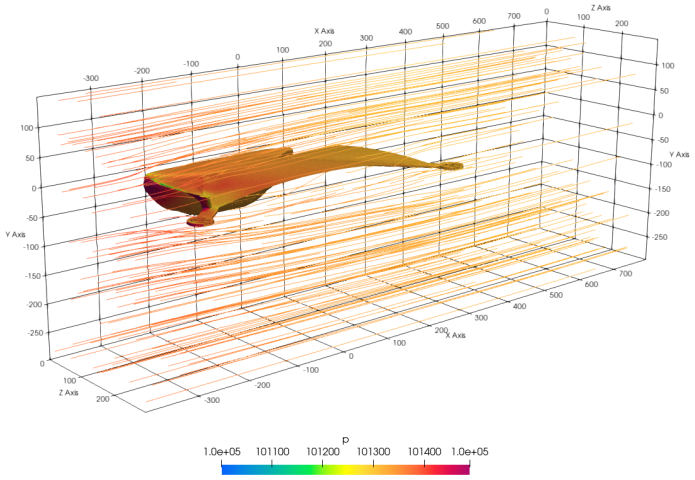
In Eqn. 4.21, $c(\cdot)$ and $s(\cdot)$ represent $\cos(\cdot)$ and $\sin(\cdot)$ functions, respectively. I_{xx}, I_{yy}, and I_{zz} are the diagonal terms of QuadGlider's moments of inertia matrix. In our case, they are 2.621×10^{-2}, 5.326×10^{-2}, and 3.021×10^{-2} $kg \cdot m^2$, respectively. θ_G and α are the gliding angle and AoA, respectively. With the equilibrium assumption, they are computed by

$$
\theta_G = \arctan(\frac{\dot{z}}{\dot{x}}),\ \alpha = \theta_G + \theta.
$$

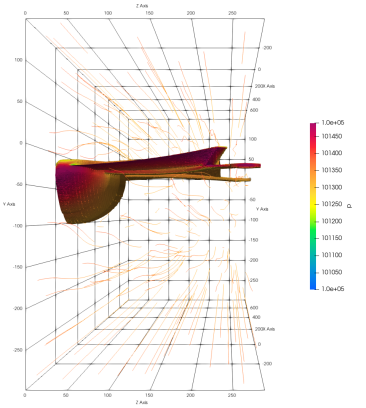
Last but not least, x_{AC} and z_{AC} are the X and Z component of the position of aerodynamic center in $\mathcal{F}_B$. For QuadGlider, the aerodynamic center $\mathbf{X}_{AC}$ is measured to be $[0.055, 0, -0.016]^T$ m.

Table 4.1. Refinement and convergence studies for all unique simulations.

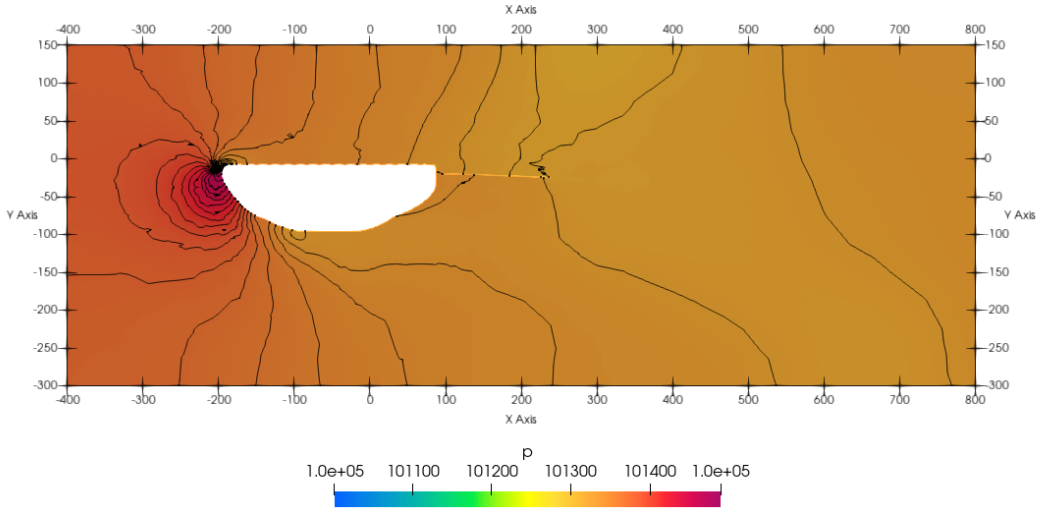
Blocks	Refinement Levels	Cells Per Level	Simplified Mesh σ	Complex Mesh σ	Median %-Change
(18 12 1)	8	8	7.39E+05	2.12E+06	0.91
(18 12 1)	7	8	2.11E+06	1.24E+06	0.77
(18 12 1)	6	8	2.02E+06	8.00E+05	6.28
(18 12 1)	5	8	1.00E+06	7.65E+05	6.74
(18 12 1)	8	5	7.32E+05	0	2.76
(18 12 1)	7	5	7.33E+05	2.09E+05	6.03
(18 12 1)	6	5	9.31E+05	2.19E+06	4.66
(18 12 1)	5	5	4.77E+05	2.68E+05	10.71
(18 12 1)	8	2	3.41E+07	0	0.45
(18 12 1)	7	2	2.67E+05	1.92E+06	667.82
(18 12 1)	6	2	8.38E+05	2.83E+06	3.09
(18 12 1)	5	2	2.21E+05	5.35E+05	10.89
(36 24 6)	6	8	2.37E+06	0	1.26
(36 24 6)	5	8	1.12E+06	3.15E+06	0.45
(36 24 6)	6	5	1.02E+06	1.12E+06	1.25
(36 24 6)	5	5	3.99E+05	6.46E+05	0.57
(36 24 6)	6	2	5.45E+05	9.55E+05	1.25
(36 24 6)	5	2	3.28E+05	4.08E+05	0.47



(a) Pressure streamline plot side view.

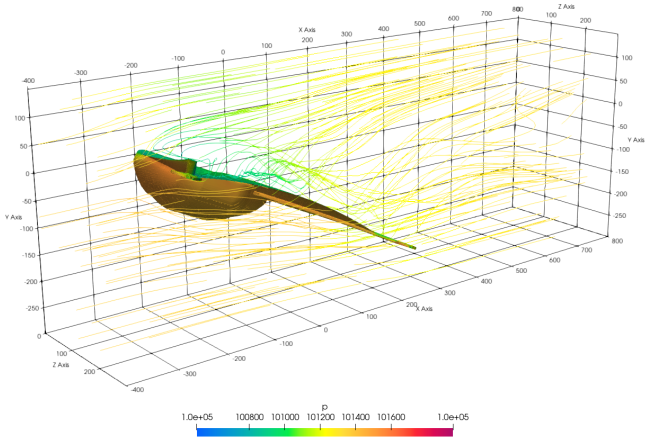


(b) Pressure streamline plot front view.

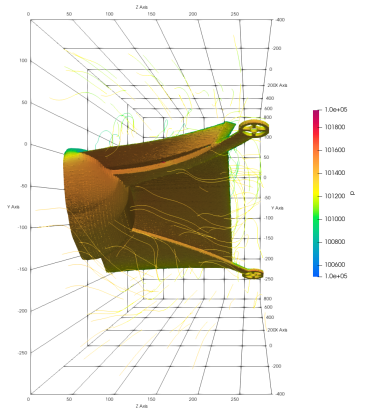


(c) Contour plot of a slice at 30mm.

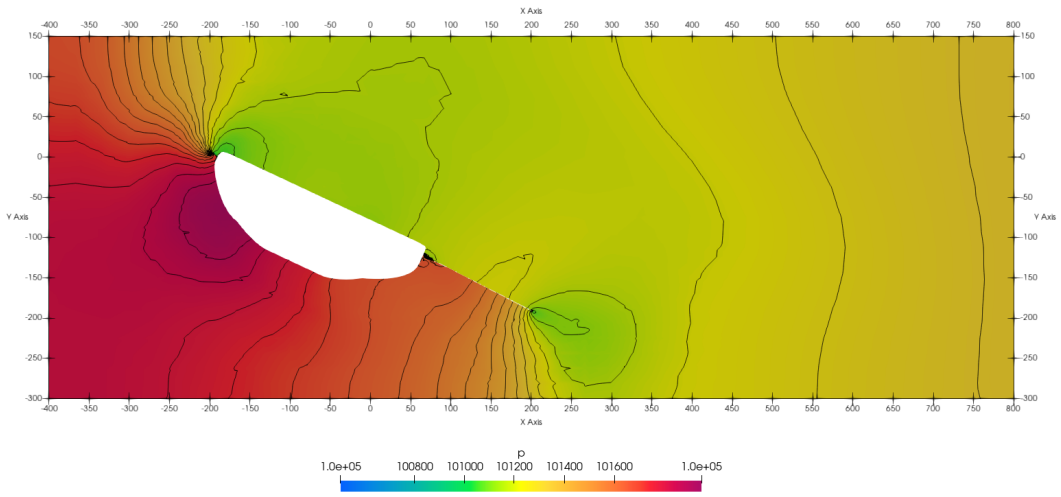
Figure 4.18. Pressure plots at an AoA of 0 degrees.



(a) Pressure streamline plot side view.



(b) Pressure streamline plot side view.



(c) Contour plot of a slice at 30mm.

Figure 4.19. Pressure plots at an AoA of 25 degrees.

Chapter 5

Design of a 3D-printable Double Ackerman Steering UGV

5.1 Chapter Abstract

Recent years have witnessed the rapid development of robotic platforms specialized in engineering education. And more robotic courses are being designed and taught in the curricula of different school levels and college programs. Among a number of the educational mobile robotic platforms, however, one can rarely find a small-sized, 4-wheel-driven mobile robot which can be considered as a chassis design for extraterrestrial or off-road rovers. In addition, it could difficult for the learners to understand the circuits configuration and electrical actuation of embedded systems from an off-the-shelf robotic platform. In this paper, we propose a low-cost open-source robotic platform named BeagleRover. BeagleRover is a 4-wheel-driven mobile robot of the size $236 \times 223 \times 85mm$ $(L \times W \times H)$. It includes 4 DC servo motors and 4 DC gear motors to over-actuate its motion in order to realize more precise path-tracking on uneven terrains. Most of its components are 3D-printable using hobbyist printers or can be procured at relatively low costs from hobbyist websites. The controller board which BeagleRover uses is BeagleBone® Blue by default but can be replaced by some other SoC computers due to BeagleRover's multiple sets of mounting holes. In addition, We utilized BeagleRovers during the MAE

40 and MAE 144 classes at UC San Diego to help instruct electrical circuit fundamentals and classical control theories, respectively. Next, code-based and Simulink-based software approaches are discussed for users to choose and start with for their own projects. Last but not least, we explore the feasibility of using BeagleRover as a research platform for indoor navigation and SLAM and object detection studies.

5.2 Introduction

Micro robot platforms have been involved in pre-college and college STEM educations since 1960s. In 1969, Seymour Papert first developed the idea of using Logo programming language and Turtle robots to teach mathematics [RO+90] to pupils. During the last 50 years, the development of educational robots has been boosted by both the advancements in technologies and increasing market demands from both parents and educational institutes. Some of the most successful educational robots include Lego Mindstorms EV3 [RL14] to teach robotic motions and programming, humanoid robot NAO [SR13] to interact with children, and MakeBlock mBot [mak18] as one of the low-cost STEM robots. For students during their college education in engineering, the Renaissance Robotics and UCSD Coordinates Robotics Lab have developed the EduMIP robot [Str18b] which is a 2-wheel mobile inverted pendulum driven by a BeagleBone Blue SoC computer[bea21].

Apart from educational robotics, 3D printing has also become an emerging area with both fast progressing research and applications. For robotics, 3D printing is implemented mainly for novel design and fast prototyping. In educational robotics, de Souza and Elisiário [SE19] designed multiple 3D-printable robotic projects to educate teenagers in STEM disciplines. In addition, Gonzalez et al. [Gon+12] and Lapeyre et al. [Lap+14] studied 3D-printable mobile and humanoid robots for educational uses, respectively. In soft robotics, Gul el al. [Gul+17] implemented fused deposition modeling

(FDM) based 3D printing and post-processing to manufacture a cylindrical actuator as a steerable catheter. In aerial robots, Strawson et al. [Str+21b] [Str+21a] and De Vivo et al. [DTK19] attempted on implementing large-scale 3D printers to manufacture aerodynamically functional components with structural and mechanical optimization for multirotor and fixed-wing unmanned aerial vehicles (UAVs), respectively.

Among a number of the educational mobile robots or 3-D printable robots, however, one can rarely find a small-sized, 4-wheel-driven mobile robot which can be considered as a base design for extraterrestrial or off-road rovers. And one can hardly find the literature deploying mobile robots with complex kinematics design fro educational and research uses. In this case, this paper introduces the robotic platform BeagleRover designed and tested through years of collaborations between researchers of different levels (undergraduates, graduates, post-docs and faculties) in CHEI Lab and Coordinated Robotics Lab at University of California, San Diego. This platform is mainly aimed at serving educational purposes, while it also plays a role in several research projects. This project is currently open-sourced and published on GitHub at https://github.com/p5cao/BeagleRover.

The structure of this chapter is as follows: in Section 5.3 we discuss the design motivations and workflow of BeagleRover; in Section 5.4 we derive the kinematics equations for both 4-wheel steering and 2-wheel-driving with self-balancing of the robot; in Section 5.5 we talk about two primary software approaches for users to drive the robot and start their own projects; in Section 5.6 we discuss the educational uses of BeagleRover in college curricula at UC San Diego; and in Section 5.7, we introduce the research deployment of BeagleRover.

5.3 Design Workflow

5.3.1 Design Motivation

The concept of BeagleRover was first inspired by the double Ackermann steering configuration which enables 4-wheel steering and proves to provide better path-tracking ability on rough terrains compared to front-wheels-steering [Muh+12], and a similar design of an active suspension and independent steering can also be found on the extraterrestrial rover VIPER by NASA [Col+19]. As illustrated in Fig. 5.1, the double Ackermann steering significantly reduces the minimum turn radius compared to the front wheel steering on the same vehicle. Second, another consideration for this design is to enable the BeagleRover to balance itself in the upright pose with 2 wheels and drive on both top and bottom sides with 4 wheels in the event of rollover like a RC stunt car [Wal23]. Third, from educational perspective, it is intended to become a practical learning tool for UCSD engineering students to better understand embedded systems and control of a robotic vehicle. Last but not least, there has also been a need in the UCSD CHEI Lab to develop a small-sized robotic rover providing high agility and maneuverability to travel across the uneven surfaces on excavation sites. The aforementioned motivations have been driving our team of student researchers to improve the design of BeagleRover to the current version. In this section, we introduce and discuss the design phases, double Ackermann steering, two-wheeled inverted pendulum, chassis design, and layered manufacturing of BeagleRover.

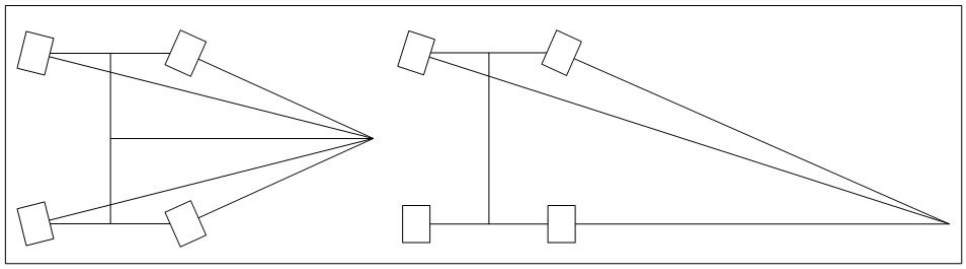


Figure 5.1. 4-wheel and front-wheel steerings.

5.3.2 Design Phases

Table 5.1. BeagleRover dimensions and parameters

Parameter	Value	Unit	Explanation
g	9.810	m/s^2	Gravitational acceleration
m_w	0.089	kg	Wheel assembly mass
D_w	0.085	m	Wheel diameter
w_w	0.040	m	Wheel width
J_w	$mD^2/8 =$		
	3.133×10^{-5}	kgm^2	Wheel moment of inertial
d_{cw}	0.089	m	Distance between vehicle center of mass and wheel base center
d_{pw}	0.007	m	Distance between steering pin (shoulder screw) and wheel base center
m_B	0.250	kg	Vehicle body block mass (chassis total + battery)
w_B	0.142	m	Vehicle body block width
l_B	0.183	m	Vehicle body block length
h_B	0.048	m	Vehicle body block height
M	0.607	kg	Vehicle total mass including battery (without other payload)
W	0.223	m	Vehicle total width
L	0.236	m	Vehicle total length
H	0.085	m	Vehicle total height
J_ψ	$Md_{cw}^2/3$		
	$= 1.592 \times 10^{-3}$	kgm^2	Vehicle pitch moment of inertia
J_ϕ	$M(w_B^2 + h_B^2)/12$		
	$= 1.137 \times 10^{-3}$	kgm^2	Vehicle yaw moment of inertia

The entire design process starts from determining the dimensions and parameters of the BeagleRover, including wheel and body masses and dimensions, total mass and dimension, and moments of inertia of interest. The list of dimensions and parameters are given in Table 5.3.2. After that, we select the electrical actuators for the BeagleRover. Table 5.2 enlists the parameters of four motors to drive the wheels and four servo motors to actuate the steering. We also select the 7.4V 2-cell LiPo battery as power supply of the robot and BeagleBone Blue as its controller board [bea21]. Meanwhile, we build the each component of 3D-printable wheels and steering mechanism in SolidWorks according to the components and actuators we select. Next, we design the chassis which mounts and supports all of above components. This chassis will also have mounting holes to fit with some other off-the-shelf controller boards. Both wheels and steering design and chassis design will be discussed in the following subsections.

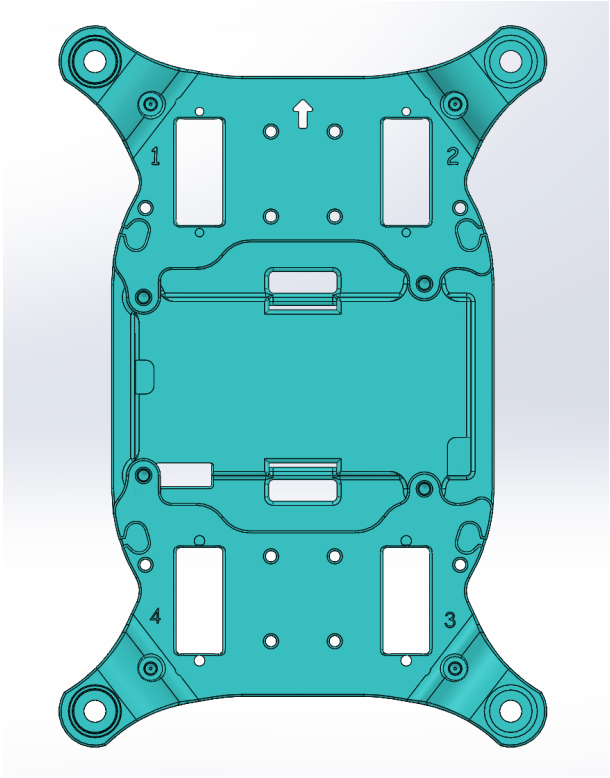


Figure 5.2. Top view of the whole chassis.

Table 5.2. Actuators parameters.

Parameter	Value	Unit	Explanation
m_s	1.060×10^{-2}	kg	Servo motor mass
m_m	9.410×10^{-3}	kg	Wheel motor mass
T_m	0.176	Nm	Wheel motor maximum torque
J_m	2.767×10^{-7}	kgm^2	Wheel motor moment of inertia
I_m	1.6	A	Wheel motor working current
V_m	7.4	V	Wheel motor nominal voltage
R_m	2.7	Ω	Wheel motor resistance
K_b	7.407×10^{-4}	$V \cdot s/rad$	Wheel motor back EMF constant
K_t	$T_m/I_m = 0.110$	Nm/A	Wheel motor torque constant
K_g	$100:1$		Wheel motor gear ratio

After finishing chassis design, all the components either come from additive manufacturing or from procurement. We then assemble and test the BeagleRover, and design and implement the BeagleBone or Jetson Nano-based software for different uses. The whole design workflow is illustrated as in Fig. 5.3.

5.3.3 Wheels and Steering Design

At each corner, the corner sub-assembly mounted on the chassis consists of a servo motor and a wheel motor with encoder, as shown in Fig. 5.4. The in-total 8 actuators on a BeagleRover vehicle control its 3 Degrees of Freedom when traveling on a 2-D plane, which results in over-actuation. With proper control allocation, over-actuation can help realize more precise path-tracking on uneven terrains [BZ17]. For each servo, the black plastic part called servo saver is attached and fixed on to the servo output gear as can be seen at the lower left corner of Fig. 5.4. The use of servo saver here is to prevent the

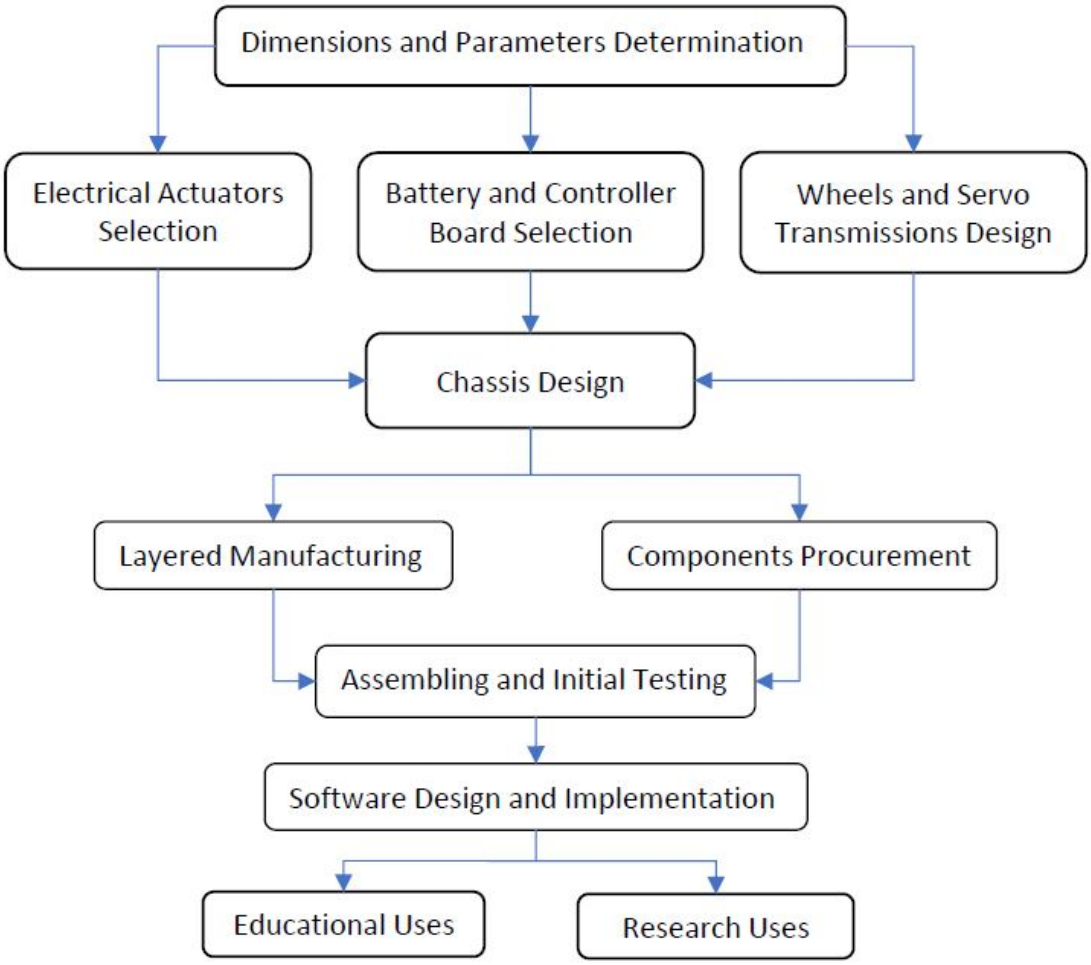
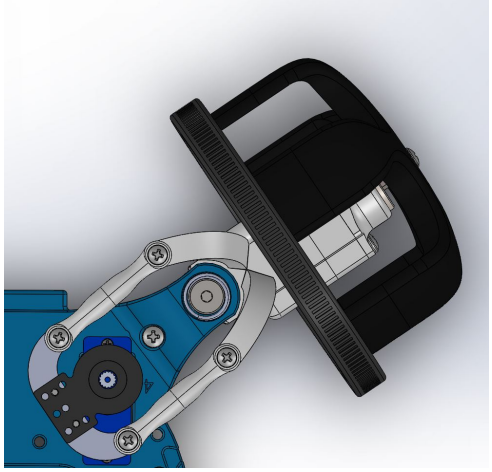


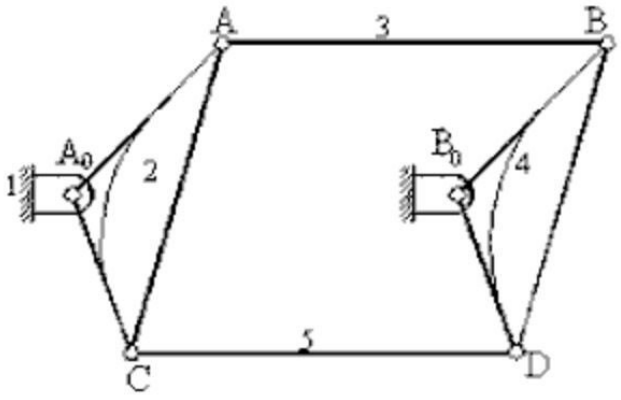
Figure 5.3. The flowchart illustrating the design phases of BeagleRover.

servo motor from being damaged by vibrations or impacts.

The wheel itself is of a quasi-cylindrical shape with $85mm$ diameter and $45mm$ width or height. The base of the wheel has an extra $8mm$ in radius and is wrapped by a rubber tire. The reason for this quasi-cylindrical design is to allow enough space to accommodate different motors with or without gearbox and rotary encoder attached to one side of the motor, and offer protections for these parts as well. The diameter of this wheel is greater than the body block height $h_B = 48mm$ on both top and bottom sides so



(a) Corner sub-assembly in Solidworks.



(b) Kinetics diagram of the steering mechanism [Uni20].

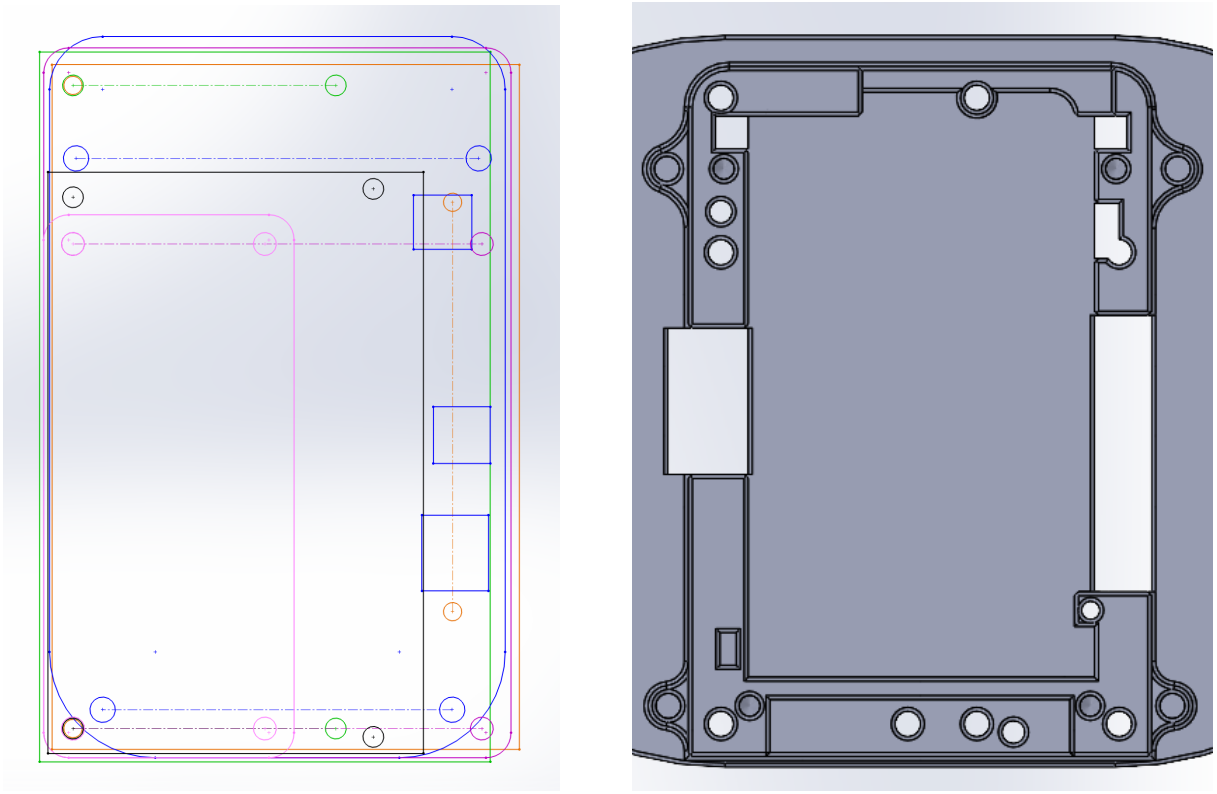
Figure 5.4. Corner sub-assembly and its steering mechanism.

that it can prevent damage to the electronics or servos.

The steering of each wheel is realized via a 4-bar parallelogram mechanism as shown in Fig. 5.4. The 2 fixed pivot joints of this 4-bar linkage are the center of servo shaft and the center of stainless steel pin (shoulder screw), respectively. With the servo motor driving the triangular rigid body ΔB_0BD, the first triangle ΔA_0AC rotates around A_0, and $\overline{AC}$ bar and $\overline{BD}$ bar are parallel to each other.

5.3.4 Chassis Design

Once we selected electrical actuators, battery, controller board, and hardware, we design the chassis to connect and support these components after designing the corner sub-assembly. Fig. 5.2 illustrates the top view of the combined bottom, front, and back pieces of chassis. Considering the low-profile tool-path slicer and 3-D printers used in this study, we round off most of the sharp edges for better layered manufacturing effect and to avoid cutting the users in future deployment.



(a) Mounting holes sketch. **(b)** Bottom view of mounting holes.

Figure 5.5. Chassis design with mounting holes design.

In addition, we design 6 sets of mounting holes on the chassis bottom, as illustrated in Fig. 5.5. We implement this design in order to adapt to a number of System-on-a-Chip (SoC) boards for users to carry out their own projects on top of the uses of BeagleRover in this study. These SoC systems include but not limited to Raspberry Pi, Raspberry Pi Zero, Jetson Nano, and BeagleBone Blue.

5.3.5 Additive Manufacturing

As discussed above, the majority of the BeagleRover components are produced using additive manufacturing. In fact, excluding cables and hardware, 37 out of 54 parts are manufactured in PETG filament by a hobbyist-level Ultimaker 3 desktop printer and they weigh only 267.36 grams in total, accounting for 44% of the total weight. Printing

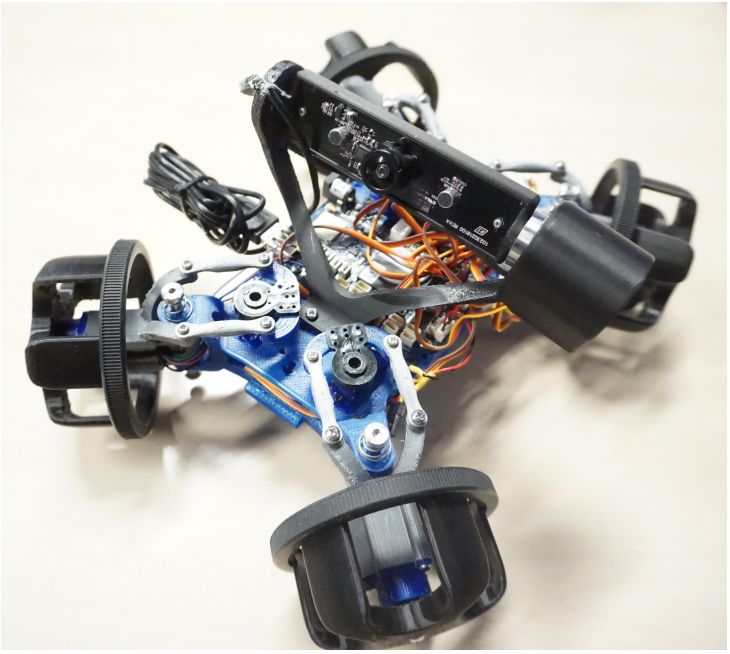


Figure 5.6. A 3D-printed BeagleRover mounted with 1-DoF camera gimbal.

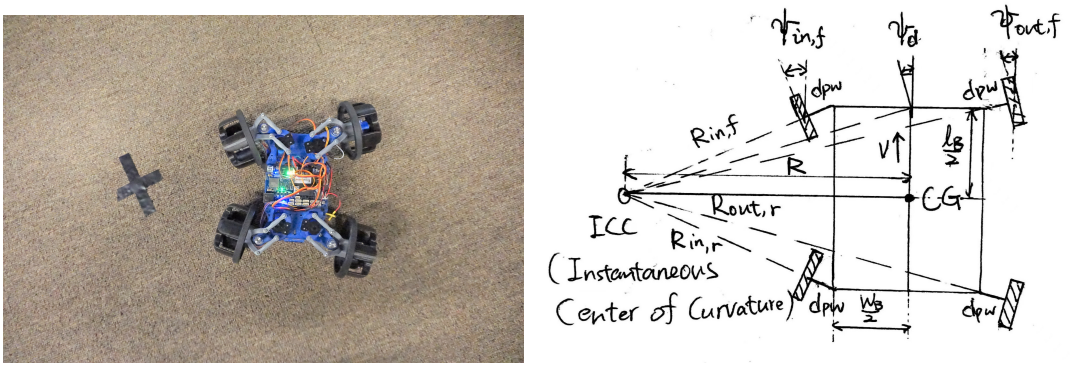
these parts takes 44.3 hours in total. An assembled BeagleRover prototype with camera is displayed in Fig. 5.6.

5.4 Kinematics and Control

5.4.1 Double Ackermann Steering

The BeagleRover mobile platform is a non-holonimic or over-actuated car-like vehicle with four-wheel steering (4WS). This type of steering mechanism can also be categorized as double Ackermann steering [HM20] if both front and rear wheels are actuated but steered to different angles. In general, 4WS improves the maneuverability of the vehicle by allowing sharper turns due to requiring a smaller admissible turning radius than does 2-wheel steering. In addition, since the BeagleRover does not have linkage mechanism for steering but utilize servo motors to steer each wheel individually, we can produce a versatile set of steering modes as shown in Fig. 5.10 providing the BeagleRover

with higher agility.



(a) BeagleRover steering. **(b)** Schematics of steering.

Figure 5.7. Double Ackermann steering of BeagleRover.

Next, we perform the inverse kinematics to compute the desired steering angle ψ_i^d and motor rotary speed n_i^d of i-th wheel. Our implementation of double Ackermann steering is design for a symmetric 4WS configuration, meaning the steer angles of front and rear wheel pairs are mirrored as shown in Fig. 5.7b [Cho+08]. Therefore, We first approximate this kinematics using a bicycle model where both front and rear pairs of wheels are reduced to single imaginary front middle wheel and rear middle wheel [Thu+02], respectively. Next, according to the path it is following, we compute the turn radius R from the length of the vehicle body and desired steer angle δ_d if imaginary middle wheel by

$$R = \frac{l_B}{2\tan\psi_d} \tag{5.1}$$

and find its instant center of rotation (ICR). Next, from the desired velocity, we compute the instant angular velocity ω of imaginary front and rear middle wheels with respect to ICR by

$$\omega = \frac{V_d}{R}. \tag{5.2}$$

Next, we compute the turn radii of inner wheels R_{in} and outer wheels R_{out}. Since

front and rear wheels are mirrored to each other by the vertical bisector of the body, we have

$$\begin{aligned} R_{in,f} = R_{in,r} &= \sqrt{(R - w_B/2)^2 + (l_B/2)^2} \\ R_{out,f} = R_{out,r} &= \sqrt{(R + w_B/2)^2 + (l_B/2)^2} \end{aligned} \tag{5.3}$$

to compute the turn radius for each wheel. Due to the symmetry, we now only need to compute the wheel velocities and steer angles of the front 2 wheels. The equations are

$$V_{in,f} = \omega \cdot (R_{in,f} - d_{pw}),\ V_{out,f} = \omega \cdot (R_{out,f} - d_{pw}). \tag{5.4}$$

where $V_{in,f}, V_{out,f}$ are the wheel velocities of inner and outer front wheels, respectively. And the front steering angles are

$$\psi_{in,f} = \tan^{-1}(\frac{\frac{l_B}{2}}{R - \frac{w_B}{2}}),\ \psi_{out,f} = \tan^{-1}(\frac{\frac{l_B}{2}}{R + \frac{w_B}{2}}). \tag{5.5}$$

for inner and outer front wheels, respectively. In addition of the steering angles, we can also compute the rotary speed of the motor shaft from the wheel speed knowing that we have a gearbox to transfer the motor speed to gearbox shaft speed (wheel speed) by

$$n_{w,i} = \frac{2 * V_i / D_w}{2\pi},\ n_{m,i} = K_g n_{w,i} \tag{5.6}$$

where $n_{w,i}$ is the wheel rotary speed driven by i-th motor, V_i is linear speed of this wheel, and $n_{m,i}$ is the motor shaft speed.

5.4.2 Inverted Pendulum and 2-Wheel Driving

In order to protect the BeagleRover from damaging in the event of rollover, we design the mobile inverted pendulum mode where BeagleRover can be driven on two rear wheels. In addition, the balancing task for a mobile inverted pendulum requires detailed studies of its equations of motion, which is often considered as a good project or assignment for students learning dynamic systems and controls. Indeed, we assigned this task as the final project of a course as mentioned in Section 5.6.2.

For a BeagleRover in its inverted pendulum mode, we derive its equations of motion based on Lagrangian methods as follows:

$$\begin{aligned}((2m+M)R^2+2J_w+2n^2J_m)\ddot{\theta}+(MLR\cos\psi-2n^2J_m)\ddot{\psi}-MLR\sin\psi\dot{\psi}^2 \\ -((2m+M)R^2\theta+MLR\sin\psi)\dot{\phi}^2=F_\theta,\end{aligned} \tag{5.7a}$$

$$\begin{aligned}(\frac{mW^2}{2}+\frac{W^2}{2R^2}(J_w+n^2J_m)+J_\phi+2MLR\theta\sin\psi+(2m+M)R^2\theta^2)\ddot{\phi} \\ +2(2m+M)R^2\theta\dot{\theta}\dot{\phi}+2ML^2\sin\psi\cos\psi\dot{\psi}\dot{\phi}+2MLR\dot{\phi}(\dot{\theta}\sin\psi+\cos\psi\theta\dot{\psi})=F_\phi,\end{aligned} \tag{5.7b}$$

$$\begin{aligned}(mL^2+J_\psi+2n^2J_m)\ddot{\psi}+(-2n^2J_m+MLR\cos\psi)\ddot{\theta}-MgL\sin\psi \\ -(MLR\theta+ML^2\sin\psi)\dot{\phi}^2\cos\psi=F_\psi,\end{aligned} \tag{5.7c}$$

where $R=D_w/2$ is the wheel radius, $m=m_w$ is the mass of the wheel, θ is the average rotational angle of left and right wheel, ψ is the body pitch angle, and ϕ is the body yaw angle. The equations of motion are then linearized around the equilibrium point, when the body angle ψ changes at about $\pm5^\circ$. We linearize Eqn. 5.7 as follows, with ψ approximately being 0, meaning that $\sin\psi\simeq0$, $\cos\psi\simeq1$, and $\dot{\psi}^2\simeq0$.

$$((2m+M)R^2+2J_w+2n^2J_m)\ddot{\theta}+(MLR\cos\psi-2n^2J_m)\ddot{\psi}-MLR\sin\psi\dot{\psi}^2=F_\theta, \quad (5.8a)$$

$$(\frac{mW^2}{2}+\frac{W^2}{2R^2}(J_w+n^2J_m)+J_\phi)\ddot{\phi}=F_\phi, \quad (5.8b)$$

$$(mL^2+J_\psi+2n^2J_m)\ddot{\psi}+(-2n^2J_m+MLR)\ddot{\theta}-MgL\psi=F_\psi. \quad (5.8c)$$

Then we design the state vector $x_1=[\theta\ \ \psi\ \ \dot{\theta}\ \ \dot{\psi}]^T$, $x_1=[\phi\ \ \dot{\phi}]^T$, and $u=[v_l\ \ v_r]^T$ as control input, where $v_l,\ v_r$ are the input voltage to motors of the rear left and right wheels, respectively. Thus, we can re-write Eqn. 5.8 as state space equations form:

$$\begin{aligned}\dot{x}_1&=A_1x_1+B_1u,\\ \dot{x}_2&=A_2x_2+B_2u,\end{aligned} \quad (5.9)$$

Then, an optimal controller is designed to stabilize the BeagleRover angles around equilibrium points using the linear-quadratic-regulator (LQR) method [Bem+02]. With the 2-wheel balancing realize, we perform the steering and driving with a simple differential driving control based on Fig. 5.8.

5.5 Software Approaches

The software implementation for BeagleRover is designed according to the needs for its application for educational and research purposes. The two main approaches at this moment are code-based and Simulink based, and both code and Simulink interfaces allow users to develop purely on-board applications or control from a personal computer.

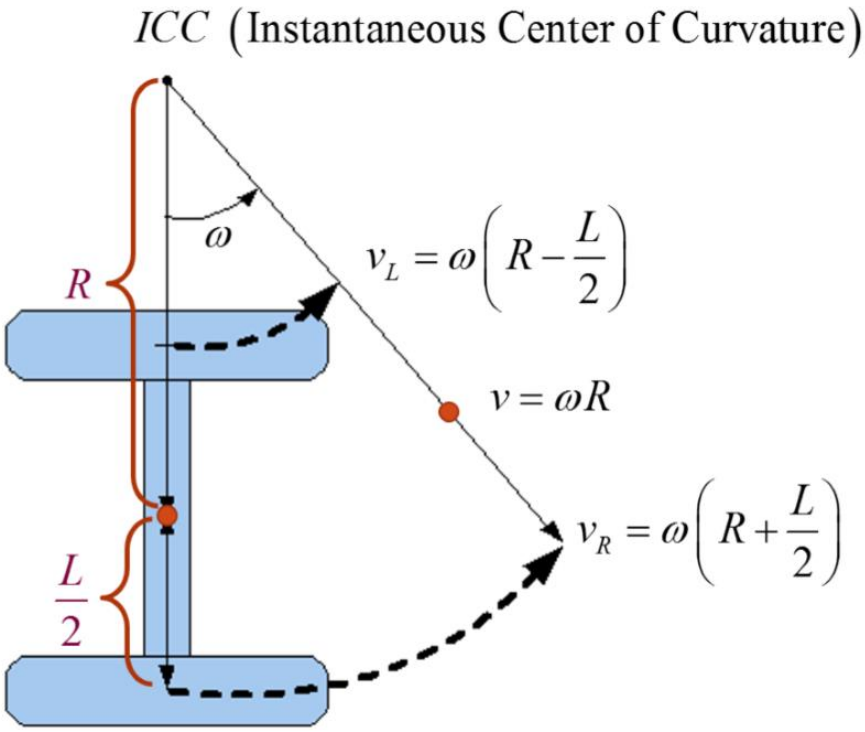


Figure 5.8. 2-wheel differential driving.

5.5.1 Code-Based Implementation

Table 5.3 lists the current available software libraries which are known to the authors. BoneScript is the JavaScript library developed and maintained by the beagleboard organization [bea21]. In addition, we know that the implementation of Robot Operating System (ROS) has been successful by researchers at the John Hopkins University for teaching a class using EduMIP [SL17]. The Robot Control Library by StrawsonDesign [Str18b] is a C library with the most comprehensive APIs and great documentations for the users wishing to writing high-performance programs then compile directly it into the machine code. And the rcpy [Oli17] is the python library developed by Dr. Mauricio C. de Oliveira at UC San Diego and is currently the most populer BeagleBone APIs. The BlackLib is a C++ library mainly design for BeagleBone Blue's previous version

BeagleBone Black, but some C++ users still prefer to build software based on this library.

With basic programming skills and some knowledge of the BeagleBone embedded systems, users can choose their own code-based implementation for BeagleRover. From the authors' perspectives, the most reliable libraries are Robot Control Library and rcpy, with which the authors developed most of the applications in this paper.

Table 5.3. BeagleBone Blue libraries.

Name	**Language**	**Community Support**	**Discontinued**
BoneScript	JavaScript	Yes	Unknown
ROS	python or C++	Yes	No
Robot Control Library	C	Yes	No
rcpy	python	No	No
BlackLib	C++	No	No

5.5.2 Simulink Implementation

The authors also worked on the control interface software via MATLAB Simulink. These software implementations are made possible by the Simulink support package for BeagleBone Blue. As illustrated in Fig. 5.9, The user can select from 4 driving modes which are spinning mode (for the BeagleRover to start self-spinning around its CoM), quarter-turn turn, sidling mode, and normal car (driving) mode. These commands generated from Simulink are sent to BeagleBone Blue controller board to drive the servo motors and wheel motors using the pre-programmed blocks from Simulink Coder Support Package for BeagleBone Blue Hardware. Note that the authors have only achieved open-loop control for both servo motors and wheel motors at this moment.

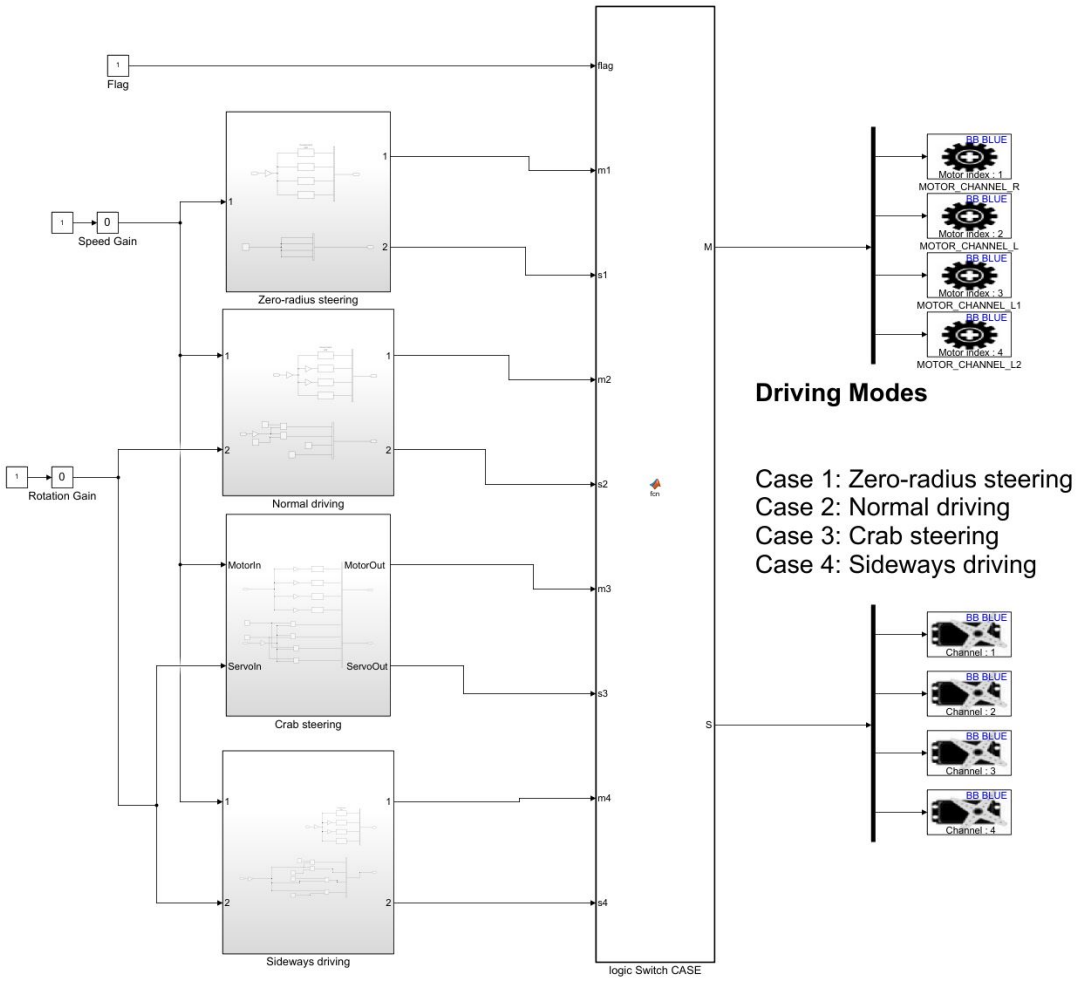
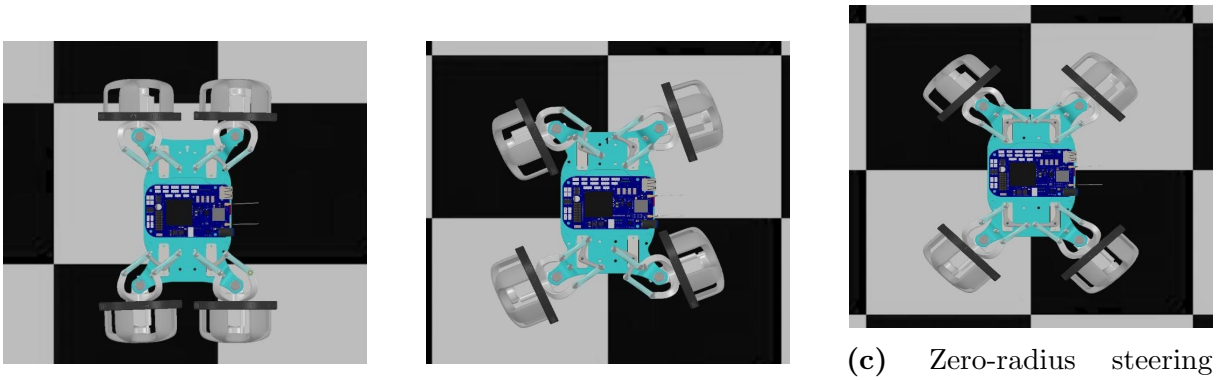


Figure 5.9. Driving control interface for BeagleRover via Simulink.

Due to the existence of 4 servo motors, BeagleRover is able to perform at least 3 more driving modes in addition to the normal forward driving and Ackermann steering modes, which are spinning mode, sideways mode, and sidling mode, as illustrated in Fig. 5.10.



(a) Sideways mode. **(b)** Crab steering mode. **(c)** Zero-radius steering mode.

Figure 5.10. 3 additional driving modes due to over-actuation.

5.6 Educational Deployment

In this section, we introduce two course-based case studies with the students in the Department of Mechanical and Aerospace Engineering (MAE) at UC San Diego where we use the BeagleRover as practical learning tool. These two courses are MAE 40: Linear Circuits and MAE 144: Embedded Control & Robotics. Next, we discuss the approaches and results of both courses in detail.

5.6.1 MAE 40: Linear Circuits

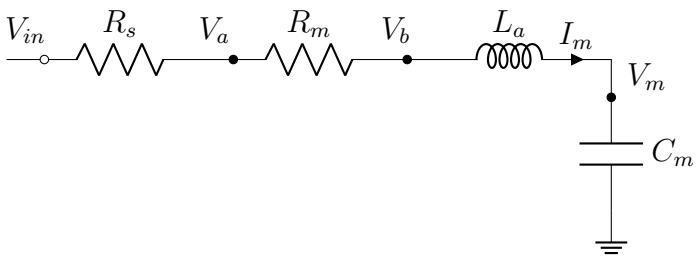


Figure 5.11. A simplified linear circuit schematic of BeagleRover motor.

MAE 40 Linear Circuits is a lower-division undergraduate course offered by the Department of Mechanical and Aerospace Engineering (MAE) at UC San Diego. It aims at teaching the fundamentals of linear electrical circuits to MAE undergraduate students.

The contents of this course include but do not limit to steady-state and dynamic behavior of linear, lumped-parameter electrical circuits, Kirchhoff's laws, RLC circuits, node and mesh analysis, operational amplifiers, signal acquisition and conditioning, electric motors, design applications in engineering and data sheet reading. During the last assignment (Homework 5) for both Summer 2020 and 2021 cohorts, we required students to work on 3 major tasks which involve transfer function modeling of a DC motor (Fig. 5.12) and analysis of a circuit according to Fig. 5.11 and the datesheets. The details of this assignment are enlisted in Table 5.4.

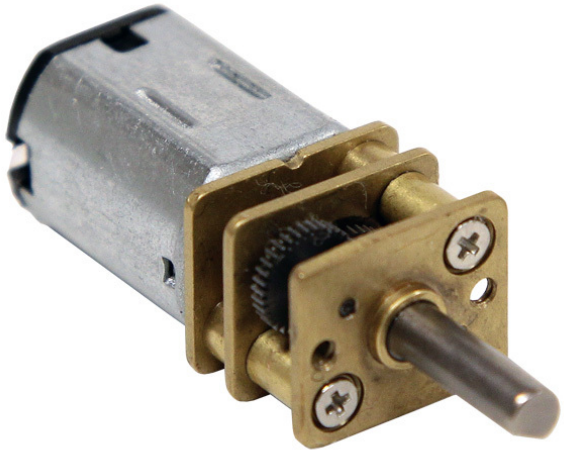


Figure 5.12. Photo of ServoCity Premium N20 high-torque gear motor used in BeagleRover (100:1 gear ratio, 270 max RPM).

For the 2021 cohort, we organized and delivered a lecture using BeagleRover as an instrument to validate the results computed from example problems similar to the problems in Homework 5, and exemplified the uses of these computations in the design of robots and electromechanical systems, and then assigned this homework. While for students in Summer 2020 cohort, we had just given them instructions on how to solve similar linear circuit problems without using a practical instrument or giving examples of real-world systems following the old syllabus. In addition, we attempted to balance the

difficulty level between the assignments from previous years and this year.

Table 5.4. Details of MAE 40 Homework 5 for Summer 20&21.

Task	Contents	Goal of the Task
T1	Find the general solution of equivalent motor capacitor voltage $V_m(t)$.	Recap the use of Kirchhoff's Voltage Law and general solution of ODEs.
T2	Find the transfer function from motor input voltage to current shaft position $\Theta_m(s)/V_{in}(s)$.	Understand Laplace Transform, transfer functions, and system damping.
T3	Compute the bootstrap (BST) pin voltage, NTC pin voltage, and currents from the schematic of battery management circuit.	Learn to read IC datasheet, and recap Kirchhoff's Laws.

After grading all the assignments, we can compare the Summer 20& 21 students' performance on each of the tasks. In both years the course was taught in a remote format so we do not need to consider the gap between physical and web-based lectures. We present the students scores on all three tasks as in Fig. 5.13. Each score of the tasks is scaled to 10 points in maximum.

MAE 144 Embedded Control & Robotics is a upper-division undergraduate course offered by the MAE department at UC San Diego. During this course, the students

are taught about classical control theories, digital circuits, fundamental dynamics, and how this knowledge set is applied in robotics. It is also very practical since students are expected to work on hand-on projects with embedded systems.

Table 5.5. Fall 2020 MAE 144 Assignments.

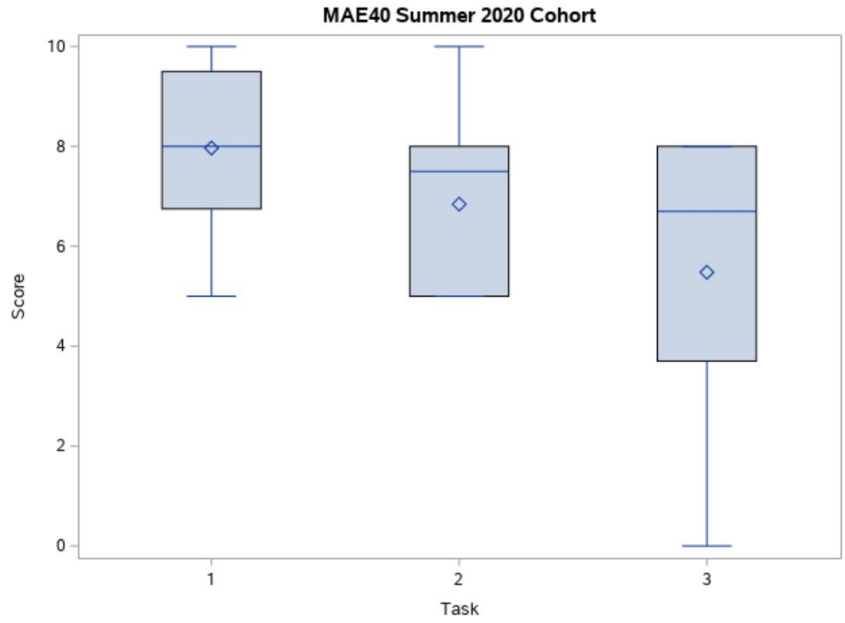
Assignments	Main Tasks	Goal of Assignments
Homework 1	Assemble the BeagleRover, flash the BeagleBone Blue, and write the first program to drive motors and servos.	Getting started with embedded system setup and programming.
Homework 2	Program the microprocessor tasked with counting a wheel encoder. Write the pseudo-code that increments the global variable 'position' up or down depending upon the direction of rotation.	Understand the logic and mechanism of rotary incremental encoders.
Homework 3	Rotate the BeagleRover about each of these axis and observe the gyroscope measurements, and write a simple program printing the accelerometer data to the console at 10Hz.	Learn to read and log IMU readings.
Homework 4	Combine the accelerometer and gyroscope readings for better body angles and linear position estimates. Design a high-pass filter for gyroscope and a low-pass filter for acceleromter to provide smooth and precise enough state estimates.	Learn to filter the noisy IMU readings to provide reliable state estimates of the robot.
Final Project	Balance the robot on 2 wheels. Write a program where BeagleRover climbs up against a wall and balance itself with upright pitch angle when standing on 2 rear wheels. Do the feedback control design in MatLab and implement the controller at 100Hz.	Practice the control design workflow using classical design tools to stabilize the pose of an unstable system.

However, during the fall quarter of year 2020, due to the supply chain problems amid the COVID-19 pandemic, the shipping of EduMIP kits [edu18] from overseas were expected to have a long delay. In light of the situation, we decided to mail the 4 full sets of BeagleRover components in to students who were close to campus (it was during remote-teaching period). Each of these kits can be assembled as one BeagleRover and has 37 3D-printed parts out of the total 54 as discussed in Section 5.3.5. Based on BeagleRover, we assigned 5 assignments for students to work on off-line as listed in Table 5.6.2.

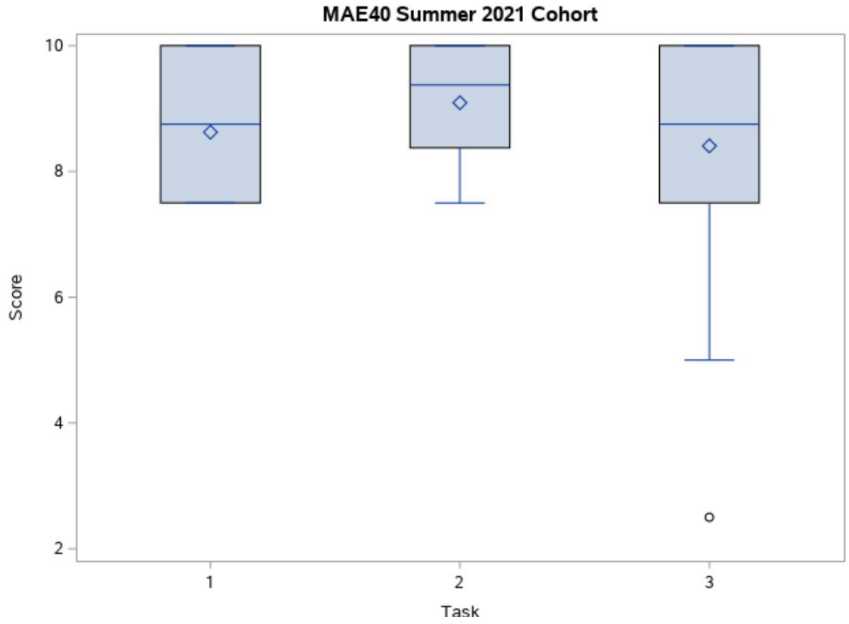
As can be seen from the grade distribution, the mean of Task 1 scores increases from 7.96 in 2020 to 8.62 in 2021, while Task 2 and Task 3 means increase by 2.25 and 2.59, respectively. The improvement of the students' performance on these tasks can be considered significant especially for Task 2&3 after the introduction of BeagleRover as teaching instrument.

5.6.2 MAE 144: Embedded Control & Robotics

On top of the requirement of the final project, one team of the students became the overachievers who managed to balance the robot on both rear wheels pair and left wheels pair as in Fig. 5.14. They were also able to steer and drive around on 2 wheels such that we gave them extra credits on their final project. Moreover, we collect the students' comments from the course evaluations at the end of the quarter, where the words "fun" and "interesting" were mentioned 11 times in total, and the words "hard", "difficult", and "challenging" appeared 8 times. This actually meets quite well with us instructors' predictions on students' feeling about hands-on classes. However, an unexpected merit of learning with BeagleRover was that it actually helped students stay focused on course even as off-line learning tool, and with adequate community support students could always taught themselves and eventually acquired strong problem solving skills.



(a) Scores distribution of Summer 2020 Cohort.



(b) Score distribution of Summer 2021 Cohort.

Figure 5.13. Comparison of score distributions of Homework 5 answers from Summer 2020 and 2021 MAE 40 cohorts.

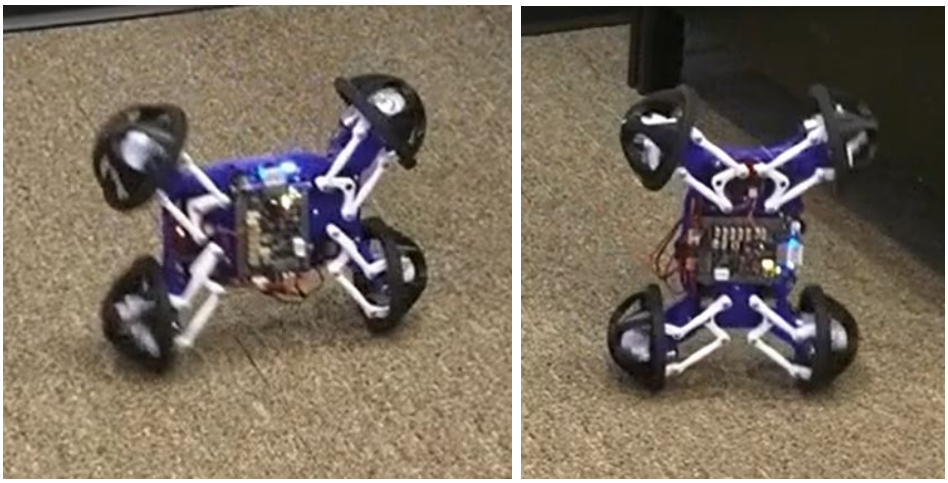


Figure 5.14. Demo video screenshots of the MAE 144 final project.

5.7 Research Deployment

5.7.1 Indoor Navigation and SLAM

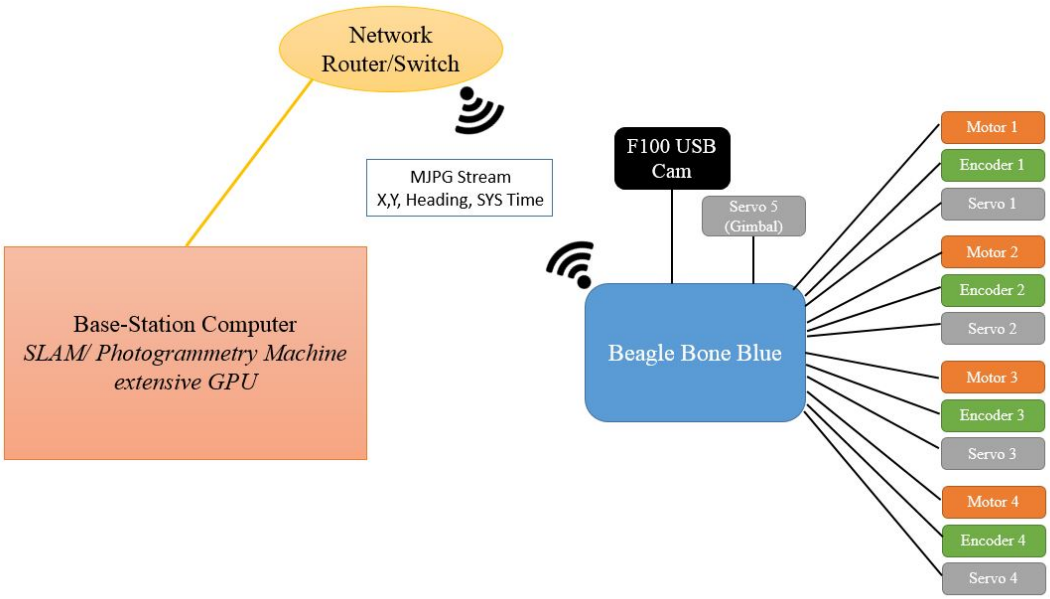


Figure 5.15. System architecture for BeagleRover SLAM tasks.

Indoor navigation of robotic platforms have played a key role in the history of robotics research. Specifically, understanding how to navigate a previously unseen envi-

ronment, where the robot needs to understand what is around it, and how to navigate it. This problem is most commonly referred to as Simultaneous Localization and Mapping (SLAM).

Monocular Visual-Inertial SLAM represents a significant advancement in the field of autonomous navigation and spatial mapping, blending the capabilities of a single-lens camera with inertial sensors. In this innovative approach, the visual information from a monocular camera is combined with data from inertial measurement units (IMUs), such as gyroscopes and accelerometers. This synergy allows the system to accurately track its position and orientation in space, compensating for the limitations inherent in using the camera or inertial sensors alone. Monocular visual-inertial SLAM is particularly effective in environments where GPS signals are weak or unavailable, making it ideal for a wide range of applications, from robotics and augmented reality to autonomous vehicles and drones. The fusion of visual and inertial data results in a more robust and reliable system, capable of handling rapid movements and various lighting conditions, while also providing a continuous estimation of the device's trajectory and the surrounding environment in 3D.

In indoor mapping scenarios, monocular visual-inertial SLAM offers unique advantages. Indoor environments often present challenges such as limited space, varied lighting conditions, and the absence of GPS signals. Here, the technology excels by utilizing the camera to recognize and track visual features in the environment, while the inertial sensors provide continuous motion tracking, even in the absence of visual cues. This combination enables precise navigation and detailed 3D mapping of complex indoor spaces, such as office buildings, shopping malls, or industrial sites. On the other hand, in outdoor exploration scenarios, the system adapts to the vastly different conditions, such as changing lighting, weather, and varied terrains. The robustness of the visual-inertial fusion becomes crucial here, as it ensures consistent performance in diverse environments, from urban landscapes to remote natural areas. The technology's ability to operate independently of external

infrastructure like GPS makes it particularly useful in remote or uncharted areas, enabling detailed mapping and exploration tasks that were previously challenging or impossible.

The BeagleRover's hyper-articulation and 4-wheeled double Ackermann steering capability place this design at an interesting advantage for indoor exploration of constrained environments. It can tightly navigate in the areas where traditional 2-wheeled steering systems could not maneuver within. With the highly agile double Ackermann steering as discussed in 5.4.1, BeagleRover is able to navigate in the mapping environment with smaller turning radii and over actuation in 3 DoFs on a planar floor with 8 actuators.

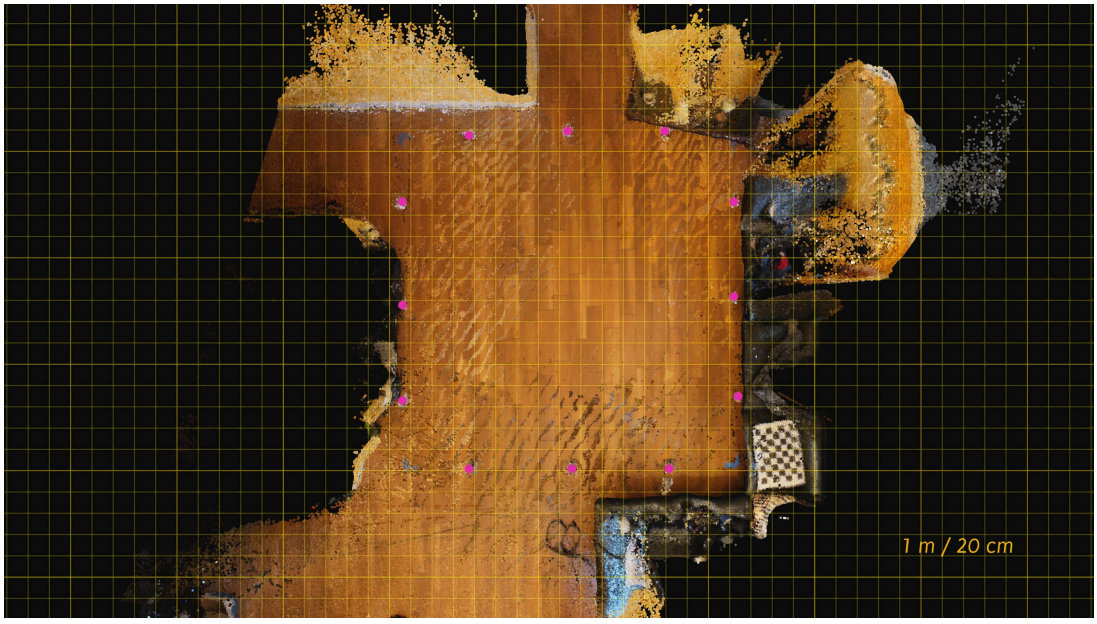


Figure 5.16. Map of the room with landmark locations marked in magenta.

The addition of a camera onto the platform enables the expansion into Visual-Inertial SLAM. A set of sparse image features can be tracked over time, in image space. With the use of camera models which are either previously calibrated, or auto-calibrated during operation, such image features correspondences can be triangulated into 3D space over time to provide global pose and motion estimates. During one experiment in a home living room, we set up 12 QR codes as landmarks as illustrated in Fig. 5.16 and set up

a path for the BeagleRover to perform room coverage while gives localizations of itself simultaneously. With the image data collected, we are able to reconstruct the 3D-map of the room in the form of point cloud data sets as illustrated in Fig 5.17. As the onboard computational power of BeagleBone Blue is limited, we utilize the off-board computational power of a remote computer to construct the 3D point cloud map in real time. Fig. **??**

Figure 5.17. Point cloud-based 3D map generated using SLAM on BeagleRover.

The indoor navigation project has been conducted by the authors on the current BeagleRover platform. Waypoints tracking algorithms are still being developed and increased to obtain the minimum cross-track errors and power consumption. The Vicon mocap readings are used as ground truth, and the fusion of camera, IMU, and encoder data gives the state estimates throughout the experiments.

5.7.2 Object Detection via Deep Learning

Apart from indoor navigation, we also attempt on realizing the navigation and control of BeagleRover based on object detection in outdoor or fieldwork scenario. For

this, we design 2 object detection based experiments for avoiding obstacles and structure from motion (SfM), respectively. To train the correponding weights for different objects, we implement YOLOv5 [Joc+20] which is the object detection algorithm based on grid scanning of the image frame. Implementing YOLOv5-based software package on Jetson Nano, the BeagleRover is able to avoid the obstacle, a rock, in the first experiment as illustrated in Fig. 5.18. And in the second experiment (Fig. 5.19) we perform SfM around the target object, a cup of orange juice, by driving the BeagleRover around the object and collects image data automatically.

(a) Detected obstacle. **(b)** Steer to avoid obstacle.

Figure 5.18. Obstacle avoidance experiment.

This model stands out for its speed and accuracy in identifying and classifying objects in images or video streams. YOLOv5 operates by dividing the input image into a grid and predicting bounding boxes and class probabilities for these boxes in a single pass, hence the name "You Only Look Once." This approach enables the rapid identification of various objects within a scene, a crucial feature for real-time applications such as autonomous driving, where quick decision-making is vital. In the context of obstacle avoidance, YOLOv5 can continuously analyze visual data from cameras mounted on a

Figure 5.19. Structure from motion experiment.

vehicle or robot, accurately detecting and classifying objects like pedestrians, vehicles, or other obstacles. The system can then calculate the relative position and trajectory of these objects, enabling it to anticipate potential collisions. By integrating this information with the vehicle's control systems, YOLOv5 facilitates real-time navigational adjustments, steering the vehicle away from obstacles and ensuring a safer path. This combination of high-speed detection and accuracy makes YOLOv5 a powerful tool for enhancing the safety and efficiency of autonomous navigation systems.

5.8 Chapter Conclusion

This paper introduces the design and implementation of the robotic platform BeagleRover developed for multiple educational and research purposes. We can draw the conclusion that low-cost 3D-printable mobile robots like BeagleRover can serve various purposes such as practical learning tool as well as research platform for robotic control and vision. Besides, with proper kinematics and control design, BeagleRover can realize complicated motions which are usually not possible for traditional 2-wheel or 4-wheel

low-cost mobile robots, which allows for better maneuverability.

However, there are also some unexpected challenges occurring from both teaching-and-learning side and robotic research side. During MAE 144 we found that code-based approach to start with robotic programming could still be difficult even for students in engineering disciplines. In some of the "traditional" engineering departments like aerospace engineering, mechanical engineering, and material science, teaching robotics is a difficult task since students can feel unfamiliar with basic understanding of embedded systems and programming logic. On the other hand, the limited on-board computational resources prevent many tasks to be fully run within the BeagleRover, making outdoor SLAM and navigation deployment currently non-achievable.

The authors' future work with BeagleRover will feature design of better graphical or modular programming interface to lower the learning barrier with BeagleRover. In addition, the authors would like to thoroughly study the correlation between teaching instrument robots and students' performance. Last but not least, the authors would like to develop robust SfM algorithms which can run on the fly for field deployment of BeagleRover.

5.9 Acknowledgements

This chapter, in full, is a reprint of the material as it appears, titled "BeagleRover: An Open-Source 3D-Printable Robotic Platform for Engineering Education and Research", in *Proceedings of AIAA Scitech 2022 Forum.* Cao, Pengcheng; Strawson, James; Zhu, Xuebin; Zhou, Everbrook; Lazar, Chase; Meyer, Dominique; Zheng, Zhaoliang; Bewley, Thomas; Kuester, Falko. The dissertation author was the primary investigator and author of this paper.

This chapter was supported by the National Institute for Standards and Tech-

nology (NIST) under award #70NANB17H21, "Improving Disaster Resilience Through Scientific Data Collection with UAV Swarms," and the National Science Foundation (NSF) under award #DGE-0966375, "Training, Research and Education in Engineering for Cultural Heritage Diagnostics." Additional support was provided by the NSF under award #CNS-1338192, "MRI: Development of Advanced Visualization Instrumentation for the Collaborative Exploration of Big Data," as well as the Kinsella Heritage Engineering Fund and the Qualcomm Institute at UC San Diego. Opinions, findings, and conclusions from this study are those of the authors and do not necessarily reflect the opinions of the research sponsors . The authors would also like to thank the students of MAE 40 and MAE 144 in year 2020 and 2021 at UC San Diego for their participation in course-related research.

Chapter 6

UGV Point Cloud Mapping for Infrastructure Inspection

6.1 Chapter Abstract

Recent advances in 3D point cloud mapping algorithms have significantly promoted the applications of light detection and ranging sensors or LiDARs in this area. However, in each of the accumulated sequential scans obtained by LiDAR sensors, it is very likely that the traces left by dynamic objects will negatively affect the mapping agent's ability of by occlusion, misalignment or impeding the odometry from working normally. In addition, dynamic objects will cause the ghost trail effect leaving undesired residuals of dynamic objects in static maps. Given this, a number of map cleaning methods were proposed to filter out these dynamic objects. However, most of these algorithms can only run offline or choose to post-process the point cloud data after a prior map is built and generally have a relatively long runtime. In this work, we propose a lightweight dynamic object filtering algorithm for building LiDAR-based static point cloud maps in real-time. On one hand, we propose an egocentric motion detection method of using improved ICP to register 3D clusters and extract their poses and twists to identify dynamic objects. One the other hand, we connect the proposed dynamic object filter with LiDAR-based SLAM algorithms to build point cloud maps and validate the effectiveness of the proposed methodology

on both our custom dataset and SemanticKITTI. We also compare the performance of the proposed method against state-of-the-art methods in terms of both filtering accuracy and processing time. As experimentally verified on SemanticKITTI, our method yields promising performance with relatively small time costs and therefore has great potential to be used as point cloud data source for a number of LiDAR-inertial-visual fusion mapping methods.

***Keywords*— Mapping, perception, LiDAR, SLAM.**

6.2 Introduction

In recent years, the use of mobile robotic vehicles has witnessed significant growth in perception and mapping. From autonomous ground vehicles navigating complex urban [Xia+23; KJD18; Fay+20] or unstructured environments [Alb+08; Cao+22b; SKI05] to drones charting unexplored terrains [BS07] [Zho+21], the fusion of perception and mapping capabilities has unleashed a new era of possibilities.

On the sensor side, 3D LiDAR technology plays an essential role in the field of mapping and perception in autonomous systems. Its ability to reconstruct dense 3D point clouds of the environment has paved the way for various applications. The popularity of LiDAR has also foster the research in mapping algorithms, including the Iterative Closest Point (ICP) [Taz+18] [LS11], Simultaneous Localization and Mapping (SLAM) with LiDAR alone [ZS14] [Che+22] and fusing LiDAR with other sensors [Sha+20; XZ21; Xu+22]. Among these, LiDAR-inertial odometry-based SLAM has the advantages of being able to mitigate accumulated drifts [Xia+23], relatively low computational cost [XZ21; Xu+22; Che+22], and maintain accuracy over long-distance travel [Sha+21] [LLH21]. Therefore, LiDAR-inertial odometry is often used in the scenarios with impaired visual features or where long-term real-time stable motion estimation is required such as autonomous driving.

However, one of the significant challenges in utilizing LiDAR data for mapping and navigation is the presence of dynamic objects within the environment [LT10]. On one hand,

during the point cloud registration, dynamic objects may introduce inconsistencies in static maps by occluding the static part of environment [Hah+03] and drifting away the odometry to cause misalignment when registering multiple scans into a single map [Yan+18]. On the other hand, regarding map construction, dynamic objects can cause ghost trail effect and therefore leave residual or noise points to become part of the final generated static map [LHM21]. To overcome this challenge, researchers have proposed a number of state-of-the-art algorithms, including ERASOR [LHM21], Removert [KK20], and Peopleremover [SN18]. All of these methods, however, require a prior map to be created before filtering out dynamic objects points in the post-processing. There has been no outstanding findings or methods in removing dynamic objects on the fly when a prior map is generated known to the authors.

To fill this research gap, we develop a fast real-time dynamic object detection and filtering algorithm. The contributions of this work are three-fold. First, we give the problem definition of cleaning dynamic objects points while building the "prior" map and consider it as a component of LiDAR-Inertial-Visual fused mapping algorithm. Second, we propose an egocentric motion detection method of for point cloud clusters and estimate their poses and twists to identify dynamic objects. Last but not least, we connect the proposed filter with LiDAR-based SLAM algorithms to build point cloud maps and validate the effectiveness of the proposed methodology on both our custom dataset and SemanticKITTI [Beh+19].

The structure of this chapter is as follows: Section 6.2 presents an introduction to this chapter; Section 6.3 discusses the problem formulation, object detection, and motion detection methodology used in this chapter; Section 6.4 gives a brief introduction of the experimental setup of the robot used in the chapter; Section 6.5 gives the results of the case study on both custom dataset and publich benchmark; Section 6.6 discusses the conclusion and potential future directions of this work.

6.3 Methodology

In this section, we are going to introduce the problem definition and workflow of the proposed cluster-based dynamic object filtering method. Note that the methods proposed in this work most likely apply to LiDAR sensors mounted on the ground vehicles or near-ground flying vehicles perceiving indoor or outdoor environments. Thus noise filtering for aerial-to-ground imaging or air-to-air perception is beyond the scope of this work.

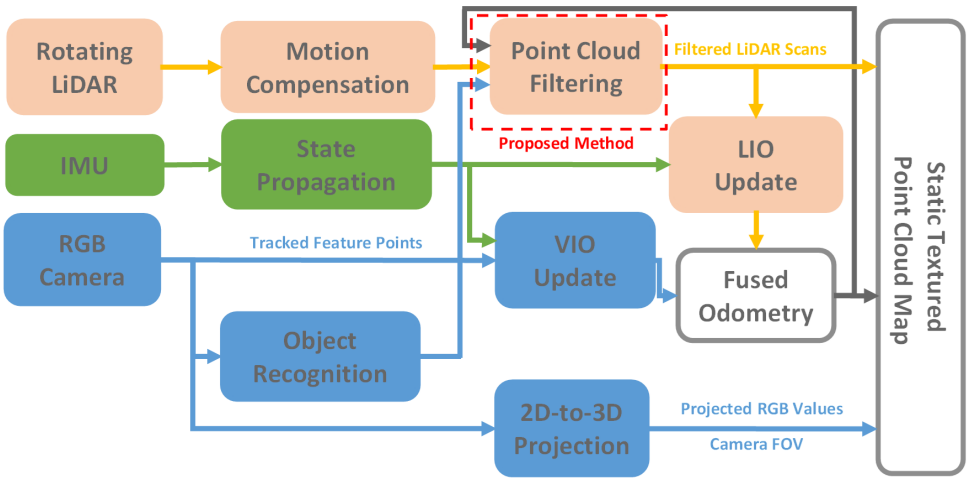


Figure 6.1. LiDAR-inertial-visual fused SLAM system overview.

6.3.1 Problem Definition

First we start with the definition of a generic point cloud mapping algorithm. Let $\mathcal{S}^t_{scan} = \{\mathbf{p}^t_1,\ \mathbf{p}^t_2,\ \mathbf{p}^t_3, \ldots,\ \mathbf{p}^t_n\}$ be the set of all points in LiDAR's current frame of scan, where each point with timestamp t is a vector of $\mathbf{p}^t_k = [x_k,\ y_k,\ z_k]^T$ if we assume other information like intensity and RGB values is not available. And let $\mathcal{S}^t_Q$ be the set of points in the keyframe or being queried at timestamp t. If we assume the beginning pose of the LiDAR corresponds to the origin and axes definition of the inertail frame, we can compute the SE(3) world-to-local transformation for this query ${}^W\mathbf{T}^t_Q$ at timestamp t. Therefore, for a mapping algorithm with raw point cloud streaming, the map $\mathcal{M}_{raw}$ can be obtained by:

$$\mathcal{M}_{raw} = \bigcup_{t \in \tau_Q} \{ {}^{W}\mathbf{T}_{Q}^{t} \cdot \mathbf{p}_{k}^{t} \mid \mathbf{p}_{k}^{t} \in \mathcal{S}_{Q}^{t}, k = 1,\ 2,\ \ldots, n \} \tag{6.1}$$

where τ_Q represents the set of timestamps when a query scan is obtained. Note that $\mathcal{M}_{raw}$ may contain dynamic objects and is mounted on the cartesian coordinate systems origined at the LiDAR's initial pose.

In post-processing algorithms like ERASOR [LHM21], the problem formulation may be considered as removing the estimated dynamic object points after the $\mathcal{M}_{raw}$ is built, such as:

$$\hat{\mathcal{M}}_{static} = \mathcal{M}_{raw} - \bigcup_{t \in \tau_Q} {}^{W}\mathbf{T}_{Q}^{t} \odot \hat{\mathcal{S}}_{dyn}^{t}, \tag{6.2}$$

where $\hat{\mathcal{S}}_{dyn}^{t}$ represents the set of estimated dynamic object points at timestamp t, and $\odot$ here indicates point-wise multiplications similar to the operation in Eq. (6.1).

However, since we are attempting to filter out the dynamic points at the very time when the query scan is received, our problem is defined in a different way. We attempt to subtract the points directly from each query scan before unioning the query scans instead of subtracting the estimated dynamic objects after the raw map is built, such as:

$$\hat{\mathcal{M}}_{static} = \bigcup_{t \in \tau_Q} {}^{W}\mathbf{T}_{Q}^{t} \odot (\hat{\mathcal{S}}_{Q}^{t} - \hat{\mathcal{S}}_{dyn}^{t}). \tag{6.3}$$

Therefore, as the query or keyframe point clouds are filtered at the time they are achieved, it is made possible that we can build an estimated static point cloud map in real-time if the filters can have relative low computational time for each keyframe.

6.3.2 Potential Dynamic Objects

As we are proposing a dynamic object filtering algorithm mainly focus on near-ground objects, we observed a number of dynamic object clusters in various datasets and need to make reasonable assumptions before proposing the cluster-based filtering method. We discuss the

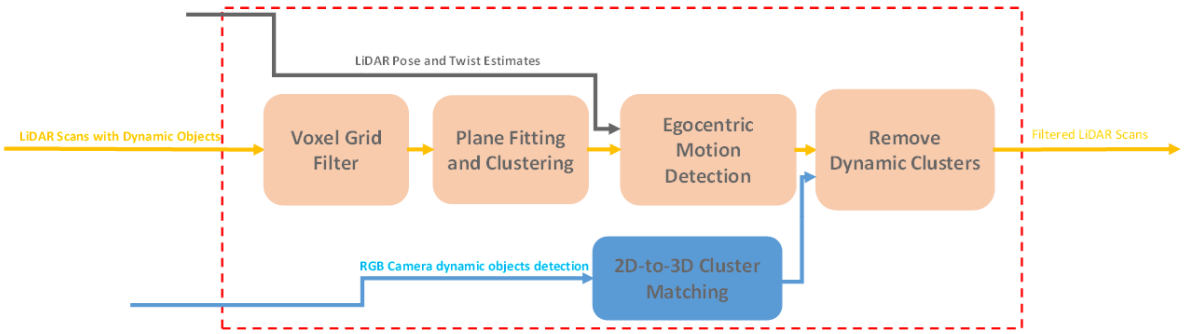


Figure 6.2. Dynamic object filter design.

outcome of observations and assumptions as follows.

Observation 1

The dynamic objects in both indoor and outdoor environments are most likely in contact with the ground plane. The categories of these objects include but are not limited to pedestrians, moving furniture, carts, bike and scooter riders, non-airborne animals, and vehicles etc.

Observation 2

The dynamic objects are most likely rigid bodies whose cross-section on XY-plane at each Z-level can be contained in convex hulls with the centroid of cross-section of this object inside its convex hull.

With the above mentioned statements, we can construct six possible cases for each of the detected object cluster:

- *Case 1:* A moving object cluster is in contact with the ground plane and is observed in both $\hat{\mathcal{S}}^t_{dyn}$ and $\mathcal{S}^t_Q$, and therefore identified as a moving object. This is considered true positive (TP).

- *Case 2:* A static object cluster is in contact with the ground plane and is observed in $\mathcal{S}^t_Q$ but not in $\hat{\mathcal{S}}^t_{dyn}$, and therefore identified as a static object. This is considered true negative (TN).

- *Case 3:* A static object cluster is in contact with the ground plane and is observed in $\mathcal{S}_Q^t$ and $\hat{\mathcal{S}}_{dyn}^t$, and therefore identified as a moving object. This is considered false positive (FP).

- *Case 4:* A moving object cluster is in contact with the ground plane and is observed in $\mathcal{S}_Q^t$ but not in $\hat{\mathcal{S}}_{dyn}^t$, and therefore identified as a static object. This is considered false negative (FN).

- *Case 5:* A static or moving object cluster is not in contact with the ground plane. This cluster is not considered in the scope of this work and therefore not admissible to either $\mathcal{S}_Q^t$ or $\hat{\mathcal{S}}_{dyn}^t$.

- *Case 6:* A static or moving object cluster is in contact with the ground plane but not observed in either $\mathcal{S}_Q^t$ or $\hat{\mathcal{S}}_{dyn}^t$. This may be caused by the flaw of clustering algorithm being used.

According to the derivation of error metrics in [LHM21], it is possible that the number of static points falsely predicted as moving objects (FP) in a map can be much larger than correctly predicted moving objects (TP). Therefore, we adopt two novel quantatative metrics named as Rejection Rate (RR) and Preservation Rate (PR) as follows:

$$\begin{aligned} \mathrm{PR} &= \frac{\text{\# of preserved static points}}{\text{ground truth of static points}} \\ &= \frac{\mathrm{TN} + \text{ground plane}}{\mathrm{TN} + \mathrm{FN} + \text{ground plane}}, \\ \mathrm{RR} &= \frac{\text{\# of removed dynamic points}}{\text{ground truth of dynamic points}} \\ &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \end{aligned} \tag{6.4}$$

6.3.3 Cluster-based Object Detection

Since we only consider objects connected to the ground plane, it is suitable to build a clustering pipeline for the environment once we can easily fit and extract the points on or close

to the ground plane. Mathematically, a distinct cluster can be defined as a set of points $\mathcal{C}_i^t$, and we also define another cluster $\mathcal{C}_j^t$ if all points $\mathbf{p}_j^t \in \mathcal{C}_j^t$:

$$\min \|\mathbf{p}_j^t - \mathbf{p}_i^t\|_2 \geq d_{ij}^{th} \text{ for all } \mathbf{p}_i^t \in \mathcal{C}_i^t, \tag{6.5}$$

where d_{ij}^{th} is the distance threshold between points in clusters $\mathcal{C}_i^t$ and $\mathcal{C}_j^t$. In this work, we use a kd-tree representation-based nearest neighbors queries to partition keyframe LiDAR scans into clusters. The details of the Euclidean distance based clustering method can be found in [Rus10].

6.3.4 Egocentric Motion Detection for Clusters

In LiDAR based SLAM algorithm, Since we have the estimated localization and self-motion streamed from odometry, we can compute the estimation of LiDAR keyframe pose ${}^W\hat{\mathbf{T}}_Q^t$ and corresponding twist $\hat{\zeta}_W^t$ at timestamp t. With the time sequence of ${}^W\hat{\mathbf{T}}_Q^t$ and $\hat{\zeta}_W^t$ characterizing LiDAR's ego-motion, one can estimate the motion of identified clusters in the environment as well.

In this work, we propose an adapted version of ICP-based algorithm for object motion detection as in Algorithm 2 depending on the time step length between two keyframes. By taking advantage of multiple processor cores onboard, it's possible to achieve near-real-time performance, especially with well-optimized implementations and capable hardware. However, for more complex environments or resource-constrained devices, real-time processing might be more challenging to achieve.

After obtaining the motion estimations of clusters at each time step, we can identify dynamic objects by inspecting the list of object to check their inconsistency with the inversed self-motion of the LiDAR query scans ${}^W\mathbf{T}_Q^{t\,-1}$. It is quite obvious that the static objects ought to have trivial twist estimation in the world frame, while dynamic object cluster can have varying twists estimation and changing poses the world frame. We use this principle to filter out the dynamic object clusters by setting the threshold for twist variations.

Algorithm 2. Egocentric Motion Detection of Environmental Object Clusters in One Time Step (**NEED WORK**)

Require:

Original query data $\mathcal{S}_Q^t$ at time $t-h$ and t.

LiDAR or fused odometry ${}^W\hat{\mathbf{T}}_Q^t$ and $\hat{\zeta}_W^t$ at time $t-h$ and t.

List of object clusters $\{\mathcal{C}_i^t \mid i = 1, \ldots, m\}$ at time $t-h$ and t, where m is the number of clusters.

Ensure: $\{{}^W\hat{\mathbf{T}}_i^t \mid i = 1, \ldots, m\}$, $\{\hat{\zeta}_{i,W}^t \mid i = 1, \ldots, m\}$

Initialize empty list of object poses $\{{}^W\hat{\mathbf{T}}_i^t \mid i = 1, \ldots, m\}$.

Initialize empty list of object twists $\{\hat{\zeta}_{i,W}^t \mid i = 1, \ldots, m\}$.

for each object cluster $\mathcal{C}_i^t$ **do**

    Initialize transformation ${}^W\hat{\mathbf{T}}_{i,\text{init}}^t$ based on ${}^W\hat{\mathbf{T}}_{i,refined}^{t-h}$.

    Apply ICP to align points in $\mathcal{C}_i^t$ using ${}^W\hat{\mathbf{T}}_{i,\text{init}}^t$.

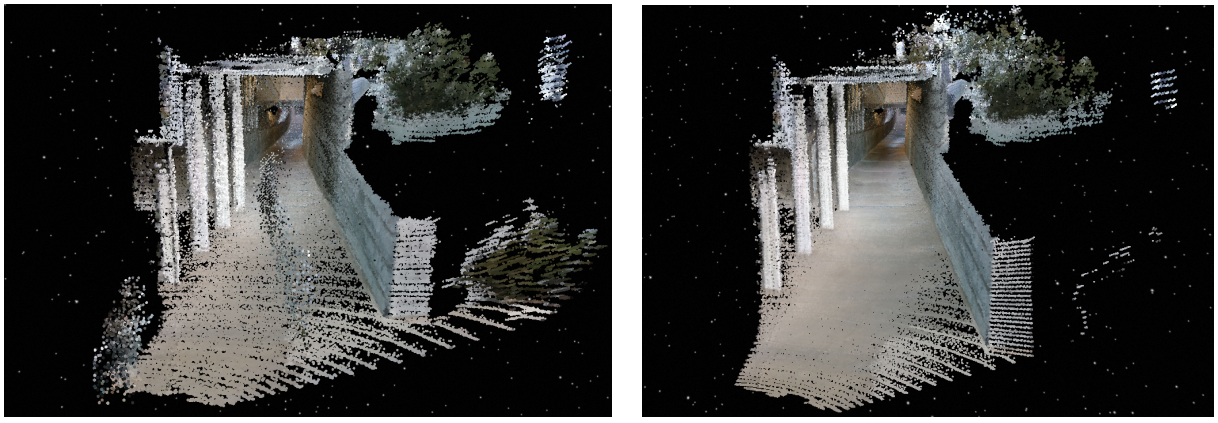
    Set ${}^W\hat{\mathbf{T}}_{i,\text{refined}}^t$ as the ICP-refined transformation.

    Replace ${}^W\hat{\mathbf{T}}_{i,\text{init}}^t$ with ${}^W\hat{\mathbf{T}}_{i,\text{refined}}^t$ in $\{{}^W\hat{\mathbf{T}}_i^t \mid i = 1, \ldots, m\}$.

    Compute using poses of $t-h$ and t .

    Add the twist $\hat{\zeta}_{i,W}^t = (\mathbf{v_i^t}, \mathbf{w_i^t})$ to $\{\hat{\zeta}_{i,W}^t \mid i = 1, \ldots, m\}$.

end for



(a) Before filtering. **(b)** After filtering.

Figure 6.3. Ghost trail removal on reconstructed UCSD Calit2 tunnel.

6.4 Experimental Setup

In this section, we present the experimental setup as featured in [CBK23]. The robotic platform implemented in this paper, namely Dambot Mini, is set to revolutionize the world of dam inspection and maintenance. Developed as a part of a sub-task dedicated to advancing

the capabilities of uncrewed ground vehicles (UGVs), the DamBot Mini project represents a significant leap forward in remote inspection technology and the creation of 3D digital twins for critical infrastructure. At the heart of DamBot Mini's capabilities lies a sensor stack equipped with an impressive array of modalities. These modalities include 3D LiDAR, Intel Realsense RGBD cameras, TunnelCAM, and Looq AI sensors, all carefully chosen to serve a multitude of purposes, including path planning, real-time navigation, and the creation of 3D digital twins for remote inspections. This innovative approach allows DamBot Mini to comprehensively map the interiors of dams, enabling the acquisition of critical data for both monitoring and maintenance purposes.

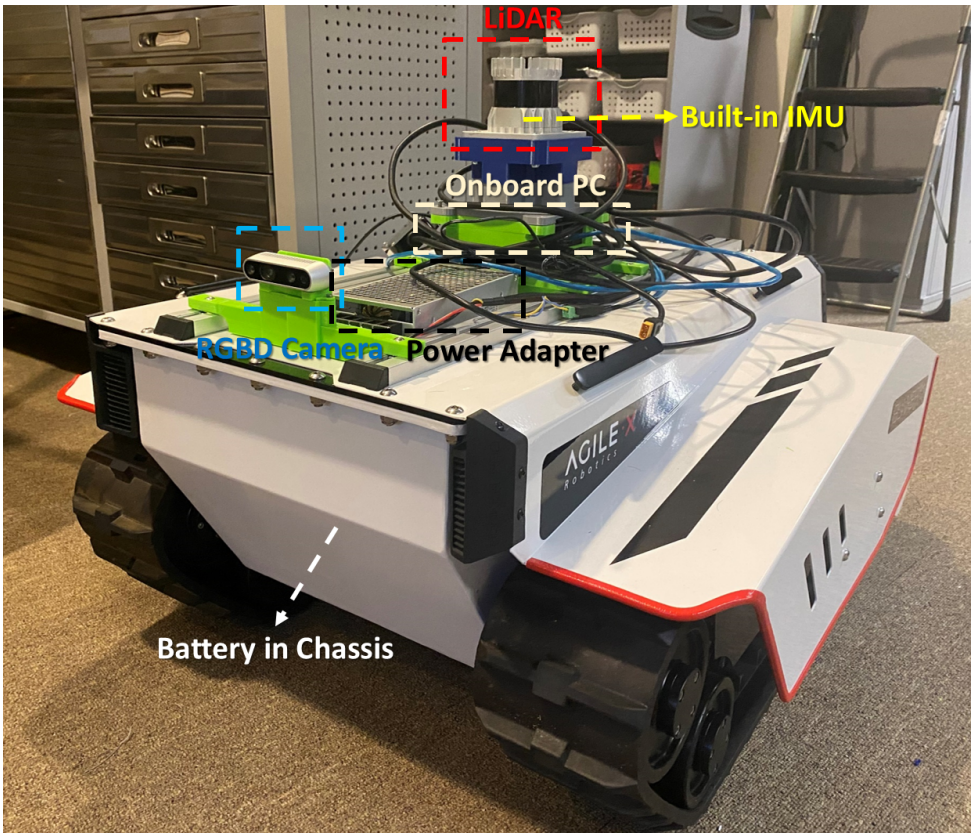


Figure 6.4. Experimental robot setup used in this chapter.

6.5 Results and Discussion

In this section, we present the results of evaluating our dynamic object filtering algorithm on both our custom rosbag SLAM dataset and the publicly available SemanticKITTI sequences [Beh+19]. We assess the effectiveness of the algorithm in identifying and filtering out dynamic objects from LiDAR scans.

6.5.1 Custom dataset

We conducted preliminary case study tests using our custom research robotic platform DamBot-Mini. The dataset is simulating a tunnel environment with a mix of dynamic and static objects. Therefore, we apply our dynamic object filtering algorithm to perform LiDAR-inertial-visual mapping from the LiDAR scans to remove the "ghost trails" left by pedestrians from the static environment. In Fig. 6.3, we can tell the ghost trails have been removed and the precision of the map is improved from a qualitative perspective.

6.5.2 SemanticKITTI

We extend our evaluation to the widely used SemanticKITTI dataset using a segment of Sequence 01 (frame no. 150 to 250). We convert the The algorithm's performance was assessed qualitatively by visually comparing the detected dynamic object regions with the point cloud sequences. Fig. 6.5 illustrate the effectiveness of our dynamic object filter applied to the sequence of selected frames of Sequence 01. We also apply the change detection method using CloudCompare's M3C2 algorithm [LBL13] and mark the removed points in red as shown in Fig. 6.5. We also perform comparsons of different dynamic objects removing methods on the same sequnce using OctoMap [Hor+13], Peopleremover [SN18], Removert [KK20], and ERASOR [LHM21] as shown in Table 6.1. Our proposed methodology has the shortest runtime compared to the other SOTA dynamic points cleaning methods.

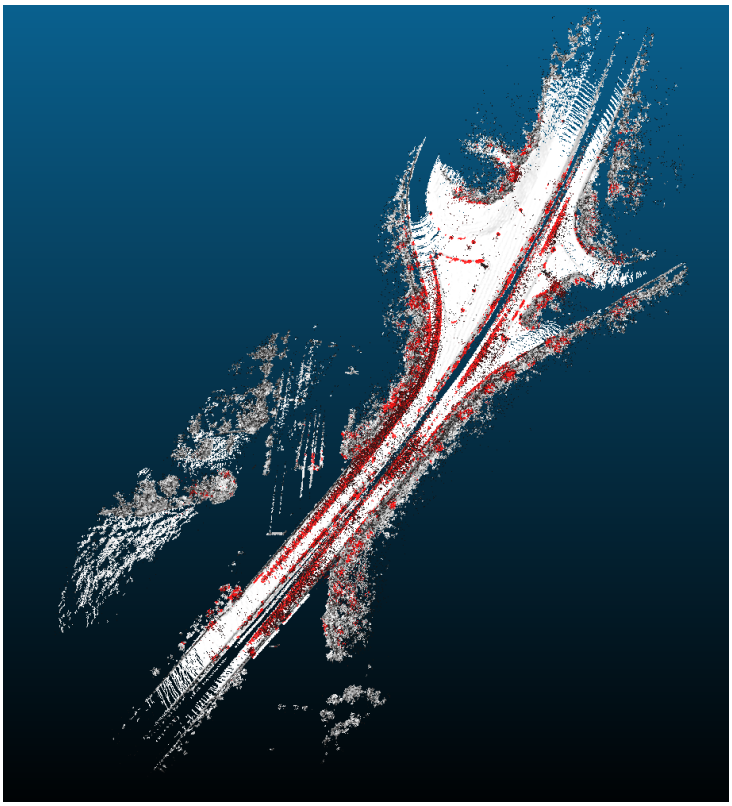


Figure 6.5. Dynamic objects removal (red) on SemanticKITTI Sequence 01.

Table 6.1. Quantitative Comparisons on SemanticKITTI Sequence 01 (102 frames, bag time 10.4s).

Method	RR [%]	PR [%]	Runtime [s]
OctoMap[4]	**99.863**	20.777	120.254
Peopleremover[5]	93.116	36.349	112.412
Removert[6]	57.077	**95.815**	95.131
ERASOR[1]	95.383	91.487	17.866
Proposed	94.231	93.427	**10.706**

6.6 Chapter Conclusion

In conclusion, this chapter attempts to address a critical challenge in point cloud mapping, namely removing dynamic objects points from query LiDAR scans. We take a step forward by proposing a novel and lightweight dynamic object filtering algorithm, designed to facilitate the

real-time generation of LiDAR-based static point cloud maps.

The method outlined in this chapter holds great potential as a reliable point cloud data source for diverse LiDAR-inertial-visual fusion mapping approaches. Future work may involve studying the pros and cons of deep learning based, transformer based, and distanced cluster based methods for identifying dynamic objects.

6.7 Acknowlegments

This chapter, in full, is a reprint of the material as it may appear, titled "Cluster- based Dynamic Object Filtering via Egocentric Motion Detection for Building Static 3D Point Cloud Maps", in *Proceedings of The Seventh IEEE International Conference on Robotic Computing (2023)*. Cao, Pengcheng; Bewley, Thomas; Kuester, Falko. The dissertation author was the primary investigator and author of this paper.

This chapter is based on work supported by the US Army Corps of Engineers under research Cooperative Agreement W912HZ-17-2-0024 and under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and by the LLNL-LDRD Program under Project No. 20-SI-005. We thank all collaborators at the Qualcomm Institute and the Contextual Robotics Institute of UC San Diego, as well as all other contributors to ideas, suggestions and comments. Opinions, findings, and conclusions from this study are those of the authors and do not necessarily reflect the opinions of the research sponsors.

Chapter 7

Multi-UAV Video Analysis for Seismic Studies

7.1 Chapter Abstract

Unmanned aerial vehicle or UAV imagery has become an emerging technology for structural health monitoring (SHM) and post-disaster damage assessment of civil structures (e.g., bridges, tunnels, and buildings). In the area of earthquake engineering, UAV-based video analysis has shown great potential to reliably track the dynamic response of earthquake-excited buildings. Such prior efforts, however, primarily focused on measuring 3-DoF planar displacements and rotations using plan-view video data, with limited interrogation of the full 6-DoF motion response. To address the research gaps mentioned, we first propose a novel video analysis methodology aiming to track the full 6-DoF dynamic response of an earthquake-excited building. We use off-the-shelf UAV platforms to capture three global views, i.e., two elevation views and one plan view of the full-scale building specimens. Both the camera pose drifts and camera intrinsic of the on-board image sensors are calibrated to allow accurate image-by-image feature detection and tracking. Proposed methodology is validated to extend the motion estimation to 6-DoF with desirable accuracy. Subsequently, in order to track the motion of the entire specimen in the image frame even without reference targets, we adopt a Vision Transformer-based video frame segmentation model. This pre-trained model is improved by trainings on collected video data to provide desired motion tracking performance. Two-dimensional bounding boxes and

masks are obtained to estimate the motions of the entire building specimen during earthquake tests. The proposed methodology is validated by monitoring the structural health and dynamic response of a 10-story full-scale building specimen during recent shake table test program at UC San Diego. Uniquely, we compare the outcome of the proposed UAV video-based method with the ground truth information retrieved and processed from the multiple analog sensors and validate its effectiveness towards dynamic response tracking for earthquake excited buildings. The segmentation-based tracking method is also evaluated to show great potential for future study directions.

7.2 Introduction

Recently in the field of structural health monitoring (SHM), many digital innovations have emerged to improve and supplement the conventional practices of data collection and assessment. Among these emerging technologies, unmanned aerial vehicle (UAV) visual sensing has proven to be effective to assist tracking and reconstructing the dynamic response and assess the post-earthquake structural health of earthquake excited buildings as illustrated in authors' previous works [HK04] [Wan+22]. However, previous methodologies were limited to viewing the assessed specimen or structure from a single perspective only or failed to combine the information from multiple views to obtain motion estimation for all 6 degrees of freedom (DoFs). To resolve this research deficit, we propose the novel UAV-imagery-based video analysis method to incorporate UAV collected video data from plan view, north, and east-elevation views to estimate the dynamic response or the building in all 6 DoFs. In addition, on top of the feature tracking-based video analyses which heavily depend on reference targets, we also propose a semantic segmentation-based motion tracking method utilizing a Transformer-based system to understand the motion of non-targeted regions of the structure.

The structure of this paper is as follows: Section 2 presents a literature review of peers' work related to this paper; Section 3 discusses the feature tracking-based video analysis incorporating both 3-DoF and 6-DoF motion estimation of the structure; Section 4 gives a brief

introduction of the proposed semantic segmentation based motion tracking method; Section 5 gives the results of the case study on a 10-story mass timber building test specimen; Section 6 discusses the conclusion and potential future directions of this work.

7.3 Related Work

Vision-based structural motion tracking has been widely used in structural health monitoring for the past few decades, including application towards monitoring the response of structures subject to earthquake motions [HK04]. A lot of efforts have been spent on the earthquake response tracking of buildings using image datasets in real time. The structural behavior can be captured as images or videos by a normal RGB camera or other vision-based sensors. Cheng and Kawaguchi (2015) [CK15] performed seismic response analysis of high-rise steel buildings using a video of TV broadcasting captured by a surveillance camera. For video-based structural displacement monitoring, reference targets are usually placed on structures to enhance the robustness of monitoring results. Target tracking digital image correlation (DIC) [Gao+15] was applied by Ngeljaratan and Moustafa (2020) [NM20] to monitor the earthquake response of a bridge structure using a stationary camera. Recently, UAV-based video analysis has attracted the interest of researchers due to its cost-effectiveness. Wang et al. (2022) [Wan+22] performed UAV-based video analysis to monitor the earthquake response of full-scale structures. Horizontal displacements of the structure in two directions were obtained by tracking the square reference targets on the rooftop from the plan-view video applying a sub-pixel edge detector [Tru+13]. The camera pose and orientation were recovered by world-to-image correspondences of stationary targets. Han et al. (2022) [HWF22] used circular reference targets and laser spot projected from a stationary highly penetrating laser lamp as reference points for the videos captured by the UAV to monitor the in-plane displacement of small-scale and full-scale structures. The motion of the UAV parallel and perpendicular to the plane of the structure was eliminated by the "physical to pixel" scaling relationship of the distance between the geometric centers of the dots on the marker. The structural displacement was obtained by the marker displacement relative to the

laser spot after scaling factor conversion.

In real structural health monitoring tasks, placing reference targets on periphery of structures may not be possible due to the consideration of integrity and aesthetics of structures. Also, target-based motion tracking methods make it difficult to estimate the motions of all other non-targeted regions of structures. Natural features on structures need to be extracted for a more general feature tracking. Wang et al. (2020) [Wan+20] applied speed-up robust features (SURF) algorithm [Bay+08] and Canny edge detector [Can86] to identify the region-based pixel features to monitor the earthquake response of a 6-story full-scale structure. Sun et al. (2022) [Sun+22] proposed an analysis approach for point cloud transformed from the UAV captured image, which was validated by the displacement monitoring of a small-scale building model. The principle of region growth was applied to point cloud segmentation to obtain the marking points. Weng et al. (2021) [Wen+21] proposed a homography-based method to monitor the structural displacement from video taken by a UAV-mounted camera. Feedforward neural network (FNN) was applied to obtain the image coordinates of selected features of the structure on its stationary position, which were further used as homography features. The current video-based displacement monitoring methods show good performance in the accuracy of measurement. However, prior efforts primarily focused on measuring 3-DoF in-plane displacements and rotations. There is still a research gap for measuring the full 6-DoF motion response using video-based displacement monitoring.

Neural network-based techniques, including semantic segmentation and depth estimation, have also been widely adopted by researchers in structural health monitoring, especially in damage detection and localization at structural component level. Narazaki et al. (2021) [Nar+21] developed a vision-based structural condition assessment system that applied convolutional neural network-based semantic segmentation and monocular depth estimation for identification, localization, and damage detection of critical structural components in a reinforced concrete railway bridge. Deep learning-based semantic segmentation was also applied by Sajedi et al. (2021) [SL21] and Liang (2019) [Lia19] for bridge component recognition and concrete crack

detection. Hoskere et al. (2020) [Hos+20] proposed a network that incorporated both material identification and damage detection where semantic segmentation was applied to assign a material and damage label for each pixel in the image. However, deep learning-based methods usually require extensive amount of data and computational power for training to provide reasonable performance. Therefore, Vision Transformer (ViT) models, for example Semantic Anything Model by Meta [Kir+23] and Track Anything by Yang et al. (2023) [Yan+23], are recent emerging inventions to perform visual tracking and segmentation tasks without needs of custom training on massive datasets. Recent progresses in this direction also incorporate Wang et al. (2023) [Wan+23] on damage segmentation and classification using copy-paste data augmentation in training and AdaLN in Guo et al. (2022) [Guo+22] on extracting pre-earthquake building information from images.

7.4 Feature tracking-based video analysis

As proposed in [Wan+22], to overcome the flaws caused by non-intrusive sensing techniques for dynamic response, a video analysis methodology was developed and validated mainly focusing on tracking the movements of checkerboard targets in the world frame coordinate system. This prior method, however, utilizes mainly the roof target tracking to estimate two-dimensional structural dynamic response in 3 degrees of freedom (DoFs). In this work, as the multi-view UAV footage is made available, we are able to reconstruct the three-dimensional structural displacements via three UAVs equipped with synchronized monocular cameras. In this section, we will first discuss using the prior method to track 3-DoF dynamic response of targets in subsection 3.1, and then in subsection 3.2 how we can estimate the 6-DoF dynamic response based on some assumptions.

7.4.1 Feature Tracking-based target detection

In the experimental setup of this study, the UAVs are used as sensor carrier platforms for monocular cameras. The first major use of on-board monocular cameras in this study is to track

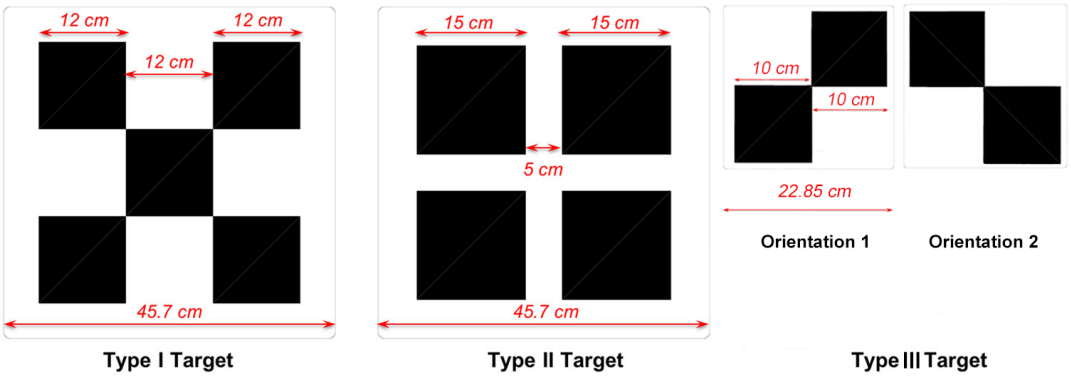


Figure 7.1. 3 types of reference targets used in the experimental setup of this paper. Type I and II are used as stationary targets and Type III as moving targets with two possible variants in orientation.

the displacements of checkerboard markers as indicated in [Wan+20; Wan+22] . Concretely, we utilize the correspondence of world frame points (XYZ-coordinates) and camera image frame points (uv-coordinates) as the basic principle of target-based motion tracking. Targets can partially represent the dynamic response of projections from 3D world points in the unit of meters to 2D image frame points in pixel values.

On one hand, strategically deployed reference targets are placed on the rooftop and facades of the structure as well as the stationary background regions. To begin with, it's essential to select key locations that provide comprehensive coverage of the structure. On the roof, targets should be placed near corners, edges, and any areas of interest. These locations should be easily visible from different angles, ensuring more precise tracking. On facades, it is more desirable for targets to be more evenly distributed with respect to each story and different height levels. Additionally, it's essential to use high-contrast colors for the checkerboard patterns, making them easily detectable by UAV-carried cameras. Therefore, black-and-white checkerboard targets of 3 types, as shown in Figure 7.1, are used to represent the displacements of the structure.

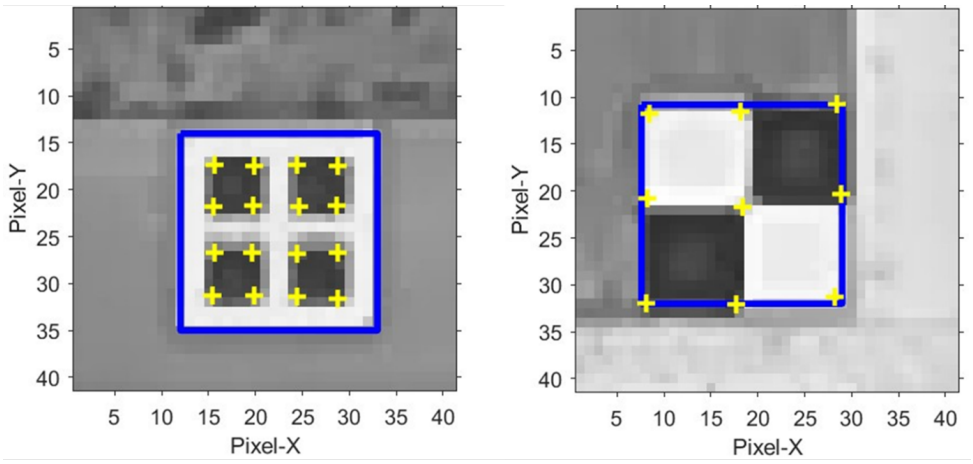
On the other hand, both moving targets and stationary targets installed are tracked using feature tracking algorithms. The way how the feature tracking code works is as follows.

First, we transfer the colored image into grayscale as black-to-white contrasts can be more easily classified using a single-value threshold as intensity variation rather than setting thresholds on all RGB channels, and grayscale images can compress the processed data size to make the algorithm computationally cheaper. Next, bounding boxes representing regions of interest (ROI) are manually drawn in the vicinity of the target. The reason we redefine ROIs every time restarting the code is that in different UAV camera recorded videos the pixel position of targets may be subject to significant shifts as UAVs could drift to different positions rendering unknown initial camera poses. Then, edge detection methods, including Sobel method [KVB88] and Canny method [Can86], are implemented to determine the refined bounding box area with manually adjustable intensity threshold.

The reference target position tracking algorithm used in this paper is as follows. First, we draw bounding boxes on grayscale images to indicate ROI for each target. Next, we implement Harris feature detection algorithm [HS+88] to find the positions of feature points inside the ROIs. At last, if this is a Type I or II target, we take the average of all feature points as the target centroid. If this is a Type 3 target, we retrieve the center point of the target shared by both black blocks as target centroids as shown in Figure 7.2. The centroid position of each target is tracked as shown in the image frame illustrated by both blue and red crosses in Figure 7.3.

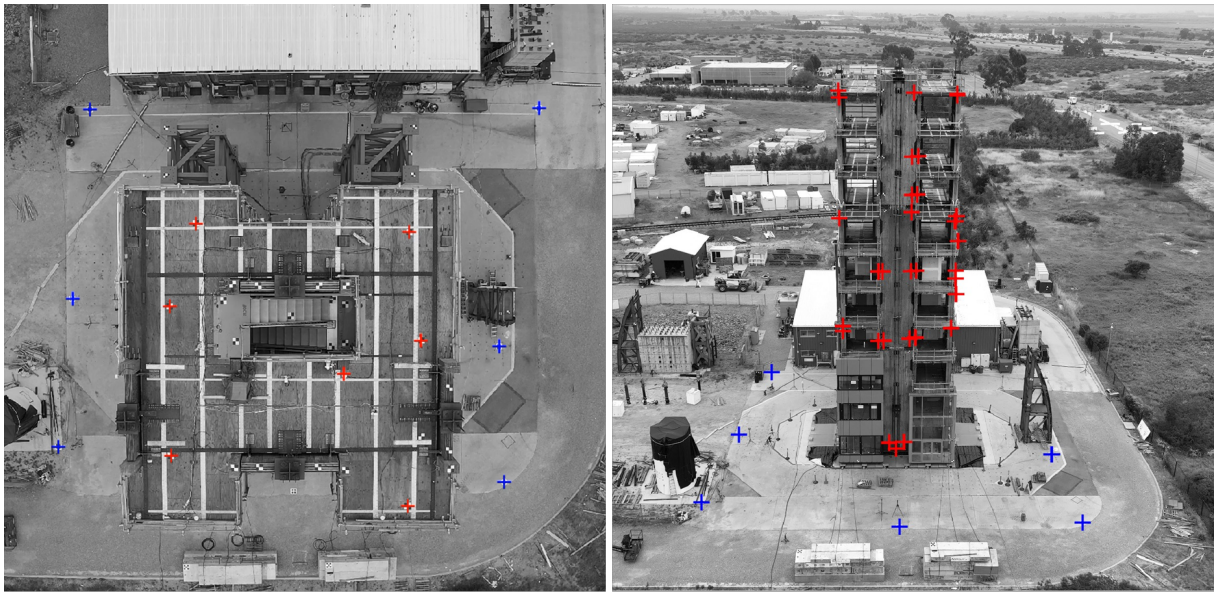
7.4.2 Camera pose recovery

The 6-DoF camera pose is characterized using a 4-by-4 pose matrix $\mathbf{T}(t)$ for each time step $t = h \cdot k$, where h is $1/f$ and f is the camera frame rate in frames per second (FPS). In the scope of this paper, all video data from different cameras and views are synchronized and sampled to have a unified frame rate of $f = 30$ FPS. Pose matrix $\mathbf{T}(t)$ can be decomposed into a 3-by-3 rotation matrix $\mathbf{R}$ and a 3-by-1 translation vector t to represent the camera orientation and position with respect to the world coordinates, respectively. We use homogeneous coordinates for both $\mathbf{R}$ and $\mathbf{t}$ by padding $[\mathbf{R}\ \mathbf{t}]^{[3\times4]}$ with $[0\ 0\ 0\ 1]^{[1\times4]}$ below to give a single 4-by-4 pose matrix $\mathbf{T}(t)$ to make computations faster. In the present study, the camera pose



(a) Type II target on the ground (b) Type III target on the rooftop

Figure 7.2. Examples of how feature tracking algorithms determine the centroids of targets. The blue bounding boxes give the perimeter of the target, and the yellow cross represents the detected feature points for frame-by-frame tracking.



(a) Targets being tracked in the plan-view (b) Targets being tracked in the north elevation-view

Figure 7.3. Image frame positions of the centroids of all 3 types of reference targets are tracked as marked by crosses. Blue crosses indicate stationary target centroids on the ground plane, and red crosses indicate moving target centroids on the building.

associated with an individual video frame, or sometimes referred to as viewpoint, is resolved using the Perspective-n-Point (PnP) technique [Gao+03] [LMF09] which estimates the pose of a

calibrated camera with known intrinsic parameters given a set of n world points in 3D space and the corresponding 2D image points. Although PnP methods require a minimum of only three non-colinear world-to-image point correspondences for recovering the camera pose, increasing the number of point correspondences enhances the robustness and accuracy of the estimated camera pose due to the noisy measurements associated with the image and world points. In the scope of this work, an open-source MatLab package called ASPnP [ULH16] is used to estimate the pose changes over time as it outperforms MatLab's built-in Computer Vision Toolbox. Camera pose rectifications based on this estimation are applied to all 3-DoF, 6-DoF and segmentation-based analyses. More details of pose estimation methods can be found in [Wan+22].

7.4.3 Translational displacements extraction

The 3-DoF dynamic response analysis using plan-view video only provides X-displacement, Y-displacement, and Z-rotation at the rooftop only. Endeavoring to capture a more complete understanding of the building specimen's response, a 6-DoF dynamic response analysis is undertaken. This requires one analysis of data collected in the plan-view (XY-view) and two elevation-views (YZ-view and XZ-view). Target detection and camera pose recovery are completed to obtain the target pixel coordinates in the image frame and camera extrinsic parameters using the same method described in subsection 3.1 and 3.2 for all the three videos. Since all the videos are captured by a monocular camera, scale ambiguity leads to four unknown variables (XYZ world coordinates and scale factor $\mathbf{s}$) in the camera projection equation for each view. In 3-DoF analysis based on the plan-view video, it is assumed that the Z-coordinates of all the reference targets remained constant during the test. However, this assumption is no longer feasible for 6-DoF dynamic response analysis incorporating the elevation-view videos. In order to solve the problem of scale ambiguity for elevation view analysis, a new method for the 6-DoF dynamic response analysis is presented in Figure 7.4. The world coordinate X (for YZ-view analysis) and Y (for XZ-view analysis) are determined before solving the camera projection equation by assuming a displacement map over the height of the structure.

The X-displacement and Y-displacement on the rooftop can be determined from the plan-view (XY-view) analysis. In addition, the X-displacement and Y-displacement of the shake table can be obtained from the analog measurements obtained directly at the platen. Since the foundation of the structure is connected to the shake table with post-tensioned (PT) rods, a rigid connection between the foundation and shake table is assured, i.e. it is thus assumed that the foundation has the same displacement as the shake table in the X and Y directions. If a displacement map in the X or Y directions is assumed over the height of the structure, the X-displacement or Y-displacement can be determined for a target at a known height of the structure. Combined with the initial world coordinate of the target which is obtained from the geo-referenced point cloud data of the test scene, the world coordinate X (for YZ-view analysis) and Y (for XZ-view analysis) can be updated in each frame. In this study, while simplified, a linear displacement map in the XY plane is assumed for the structure that the displacement at a certain height of the structure is calculated as the linear interpolation of the displacement at the top and bottom of the structure. By substituting the updated world coordinate X and Y into the camera projection equation for YZ view and XZ view, the 3D world coordinates and displacements for a target at a known height of the structure can be determined.

7.4.4 Rotational displacements extraction

The rotational displacements considered in this study include the in-plane rotation of the roof panel about the Z axis and the rocking-induced rotation (rocking response) of the mass timber rocking wall about the X and Y axis. For the in-plane rotation of the roof panel, the shape of the roof floor plan is assumed to be unchanged. For the rocking response of the rocking wall, the cross section of the rocking wall is assumed to remain plain during the test. Therefore, instantaneous rotations can be determined as the rigid body rotation of a certain area defined by the location of several reference targets attached to the structure. The roof panel rotation for example about the Z axis can be determined by analyzing the plan-view video. It is assumed that the roof panel with six reference targets (A-F) on its perimeter is a rigid body, which is

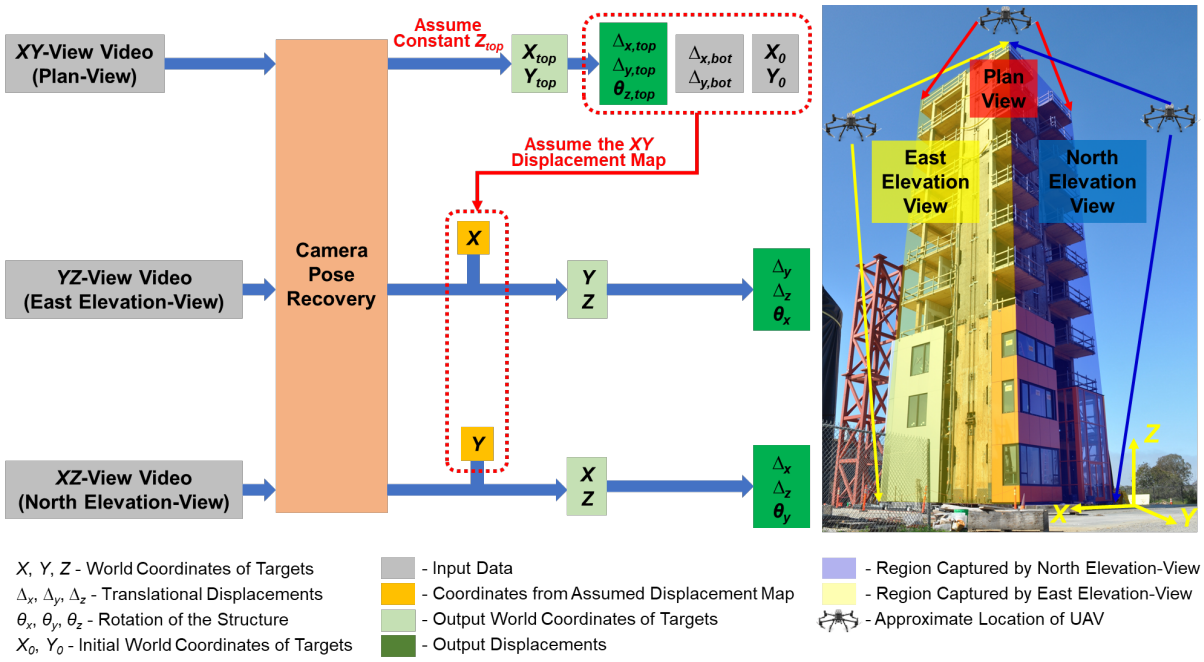


Figure 7.4. Flowchart of the proposed UAV vision-based 6-DoF dynamic response analysis.

shown as the area bounded by the green solid line in Figure 7.5(a). The roof panel has both translational and rotational displacements during the test and an exaggerated illustration of its location during the test is shown as the area bounded by the yellow dash line in Figure 7.5(a). The in-plane rotation of the roof panel about the Z axis can be calculated as the rigid body rotation of area ABCDEF using world coordinates.

The rocking response of the rocking wall about either the X or Y axis can be determined by analyzing either of the elevation-view videos. During the test, reference targets are placed on the rocking wall panel and the connecting line between these reference targets can be assumed as a rigid body (MN shown in Figure 7.5(c)). The rotational displacement can be extracted by tracking the orientation of MN in world coordinates.

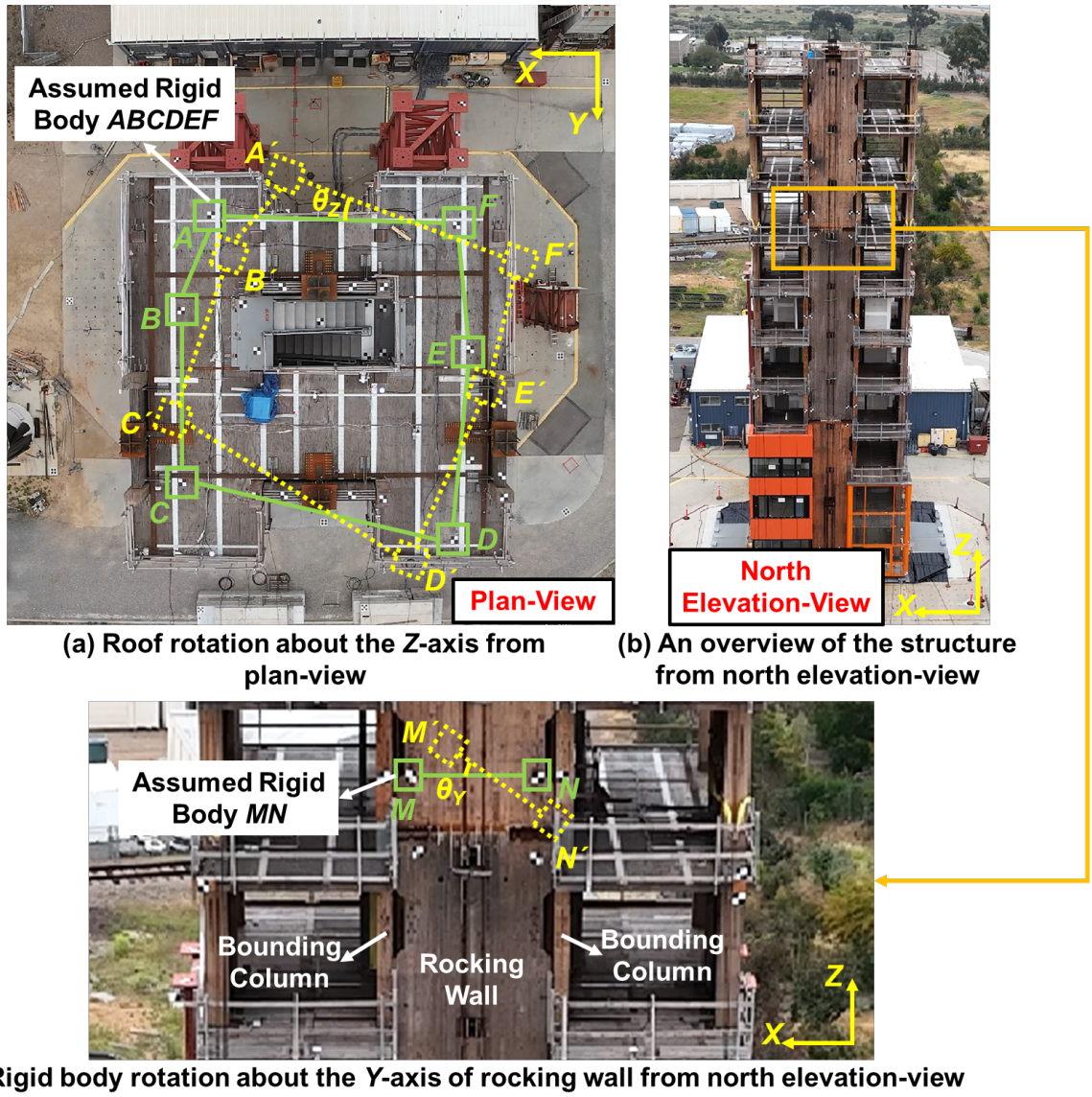


Figure 7.5. Illustration of rotational displacement characterization of the building.

7.5 Segmentation-based video analysis

This section proposes a preliminary method using a ViT model for semantic segmentation of the video data of earthquake excited buildings to create their motion-aware masks. We proposed a two-step approach to mask the moving building in videos taken from different perspectives. First, we utilize Scale-Invariant Feature Transform (SIFT) [Lin12] and Brute Force

Matcher (BFM) [JV18] to render bounding boxes of the entire building. Second, we feed these bounding boxes and initialized key points into the ViT model to perform stable and accurate image segmentation.

7.5.1 Bounding box generation

In this step, the SIFT and BFM algorithms are used to accurately locate and frame the target building in the videos. First, we prepare an image containing only the target building as a benchmark for feature matching. SIFT algorithm is invariant to image scale, rotation and brightness changes, which allows us to accurately match the target object under different conditions and angles. Next, we use the BFM algorithm to find the key points that best match the key points of the benchmark image in each video frame, to locate the position of the target object in the current frame. Then, to increase the matching accuracy, we adopt the random sample consensus (RANSAC) algorithm [FB81] to eliminate outliers. At last, we obtain a bounding box of the target object in the video frames through the calculation of the homography matrix. After successfully rendering bounding boxes for multiple video frames, we can proceed to the subsequent semantic segmentation task.

7.5.2 Video segmentation using SAM

Segment Anything Model

The Segment Anything Model (SAM) developed by Meta [Kir+23] is a phenomenal breakthrough in the domain of Vision Transformer studies. SAM is designed as a promptable segmentation framework capable of segmenting objects on images without necessitating additional training, generating high-fidelity masks from minimalistic input prompts such as points or bounding boxes, showcasing its robustness and adeptness in tackling various segmentation tasks.

Prompts for SAM

Due to the complex structure of the building area in video frames, solely using the bounding boxes as input will result in poor consistency of segmentation masks between the

consecutive video frames. To resolve this issue, we propose a method that only needs to fine-tune the masks for the first frame of the video but will apply changes to the rest of the video. We work with pre-trained SAM by providing both bounding boxes and key points as prompts to improve the segmentation. These key points are classified into positive and negative key points, which are used to indicate the part that is expected to be added to the mask and the part not expected to exist in the mask based on the current segmentation result.

7.6 Case study results

In this section, the proposed feature tracking-based and segmentation-based video analysis methods are applied to monitor the earthquake response of the 10-story mass timber building at the 6-DoF Large High-Performance Outdoor Shake Table (LHPOST6) at UC San Diego. The details of the building specimen and the shake table test program is introduced in subsection 7.6.1. The feature tracking-based video analysis results are subsequently validated by comparing them with the ground truth measurements from multiple analog sensors installed on the building specimen. And the segmentation-based video analysis results are compared against the ground truth markers to show its effectiveness in accurate masking and motion tracking.

7.6.1 NHERI Tallwood Project

To advance the application of a new seismic resilient lateral system using post-tensioned (PT) mass timber rocking walls, a series of shake table tests were performed on a 10-story mass timber building at the LHPOST6 at UC San Diego. The 10-story mass timber building (shown in Figure 7.6(a)) has a height of 112 ft (34.14 m) and applies an advanced seismic-resistant lateral system including cross-laminated timber (CLT) rocking walls in the east-west direction and mass plywood panel (MPP) rocking walls in the north-south direction. Besides, a 10-story operable prefabricated steel stair tower, exterior and interior cold-formed steel (CFS) wall subassemblies and a 2-story fire-rated curtain wall subassembly are also included in the building. For more details on the design of the 10-story mass timber building refer to [Bus23; Hua23; Pei+; SHR22].

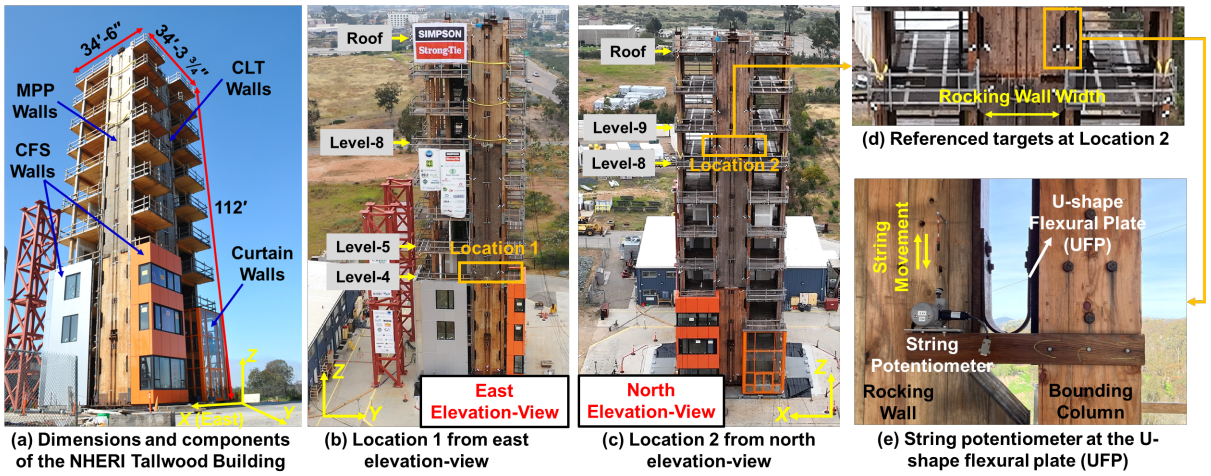


Figure 7.6. The components and overview of the 10-story mass timber building specimen (NHERI Tallwood).

During the shake table test program, multiple earthquake motions were selected for the validation of the proposed video analysis methods. Due to the limited length of this paper, this section only presents the analysis result from a particular earthquake motion tested on May 17, 2023. The motion presented here is the earthquake motion recorded at Fremont-Emerson Court station during the 1989 Loma Prieta earthquake in the United States. The selected motion has all 6 DoFs and is scaled to a maximum considered earthquake (MCER) hazard level. The achieved peak input acceleration (PIA) for the selected motion is 0.52g in X-direction, 0.72g in Y-direction and 0.18g in Z-direction.

The dynamic responses of the building specimen were captured by three commercial UAV platforms during the shake table test program in three different views: (1) the plan-view (XY-view) was recorded by the onboard camera of a DJI Phantom 4 RTK UAV with the frame rate of 60 FPS and 3840×2160 resolution; (2) the east elevation-view (YZ-view) was recorded by the onboard camera of a DJI Mavic 2 Pro with a frame rate of 30 FPS and 3840×2160 resolution; (3) the north elevation-view (XZ-view) was recorded by a DJI Zenmuse P1 camera mounted on a DJI M300 UAV with a rate of 30 FPS and 3840×2160 resolution. As has been pointed out in subsection 3.2, all video data are first sampled to 30 FPS and then synchronized with the test time in the Pacific Daylight Time Zone (PDT) as well as the Unix timestamps.

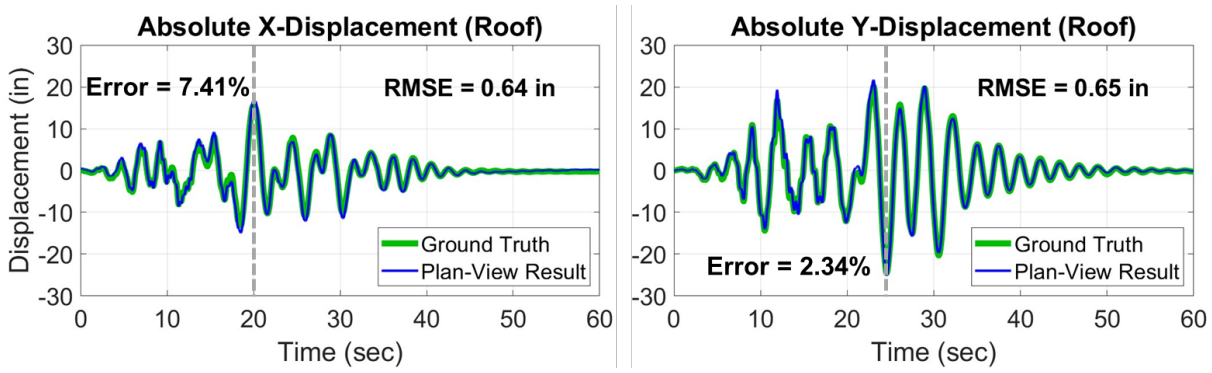


Figure 7.7. Comparisons between absolute XY translational displacement results of the roof and corresponding ground truth data with RMSEs from plan-view analysis.

7.6.2 Feature tracking-based video analysis results

3-DoF dynamic response analysis results

Since the proposed 6-DoF dynamic response analysis requires the XY-displacement at the rooftop for the displacement map formulation, a 3-DoF dynamic response analysis using plan-view video is conducted first to obtain the XY-displacement and Z-rotation at the rooftop. Based on the methodology discussed in subsection 7.4.1 and 7.4.2, target detection and camera pose recovery are completed first to obtain the target pixel coordinates in the image frame and the camera location and pose. In the 3-DoF analysis using the plan-view video only, the Z-coordinates of all the moving targets are assumed to remain constant, i.e. the Z-displacement of the building specimen is zero. Therefore, the X and Y coordinates of the moving targets on the rooftop can be solved using the camera projection equation. For the XY-displacement analysis, Figure 7.7 shows the absolute XY-displacements results at the rooftop of the building specimen from the 3-DoF analysis using plan-view video. The ground truth displacements are the results of double integration of accelerations from the accelerometers installed next to the stair landing on the rooftop (close to the center of mass of the roof panel). Reasonable agreement between the ground truth data and analysis results is observed. The root-mean-square errors (RMSEs) are 0.64 inch for X-displacement and 0.65 inch for Y-displacement. The vertical gray dash lines in the plot represent the peak amplitudes in displacements (maximum absolute values). The percentage

errors at peak amplitudes between the analysis results and ground truth measurements are 7.41 % in X-direction and 2.34 % in Y-direction. It is noted that the reference frame for the video analysis corresponds to a stationary frame several seconds before the earthquake starts, so small responses may be observed at zero time for analysis results, while the ground truth displacements are set to zero at zero time.

For the Z-rotation analysis, Figure 7.8 shows the absolute Z-rotation results of the roof. Since no sensor for direct rotation measurement was installed on the roof panel during the test, the ground truth rotation is calculated as the rigid body rotation of an area bounded by several accelerometers on the rooftop given the displacements from double integration of accelerations and the exact as-built coordinates of these sensors. In general, the video analysis results can match the ground truth data. The RMSE for the Z-rotation of roof panel is 0.0559 degree.

6-DoF dynamic response analysis results

Compared with the 3-DoF analysis using the plan-view video only, the 6-DoF analysis using the videos in three different views can provide a more comprehensive dynamic response at any floor that is captured in the videos. Figure 9 shows the absolute XYZ-displacement results at rooftop and level-8 of the building specimen from the 6-DoF analysis. In the plot, the XY-displacements from both north and east elevation-views analysis as well as the Z-displacements from east elevation-view analysis are presented and compared with the ground truth displacements obtained from double integration of accelerations at the stair landing locations at each floor. Reasonable agreement between the ground truth measurements and analysis results is observed except the low-amplitude response region in Z-displacement results. The root-mean-square errors (RMSEs) are less than 0.9 inch at rooftop and less than 1.0 inch at level-8. The percentage errors at peak amplitudes are in reasonable range. Some additional high-frequency vibrations not shown in the ground truth measurements are observed in the video analysis result for Z-displacements from east elevation-view. The similar difference is also found in Z-displacements from north elevation-view analysis (not shown in the paper). One of the possible factors lead to this difference is the different data source locations for ground truth measurements (next to stair landing) and

video analysis (building facades). Further investigation is needed to obtain the exact cause of this difference.

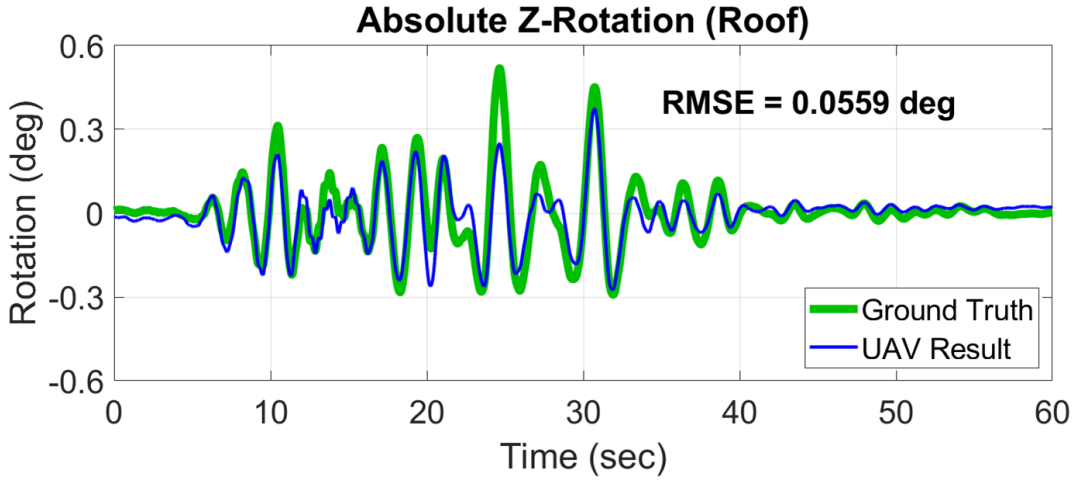


Figure 7.8. Comparisons between absolute Z-rotation results of the roof and corresponding ground truth data with RMSEs from plan-view analysis.

Besides, it is noted that the X-displacement in the east elevation-view (YZ-view) and the Y-displacement in the north elevation-view (XZ-view) are determined based on the assumed linear displacement map rather than the target tracking process. Therefore, the above two displacements should be ideally the same as the plan-view analysis results at the rooftop. However, slightly different error information is observed for these displacements due to the small differences between the elevations of reference targets used in plan-view analysis and elevation-view analysis. The assumed linear displacement map is feasible for extracting the XYZ-displacement since no huge difference is observed between the analysis results and ground truth data.

For the rocking-induced rotation of the mass timber rocking wall, two locations shown in Figure 7.6 (b) and Figure 7.6 (c) are selected for the video analysis. Location 1 is the east rocking wall panel between level-4 and level-5 from the east elevation-view and Location 2 is the north rocking wall panel between level-8 and level-9. Based on the method discussed in subsection 3.4, the rocking response of the rocking wall can be extracted by tracking the orientation of the assumed connecting lines between two reference targets on the rocking wall. To compare with the video analysis result, the ground truth measurement is determined as the relative slip

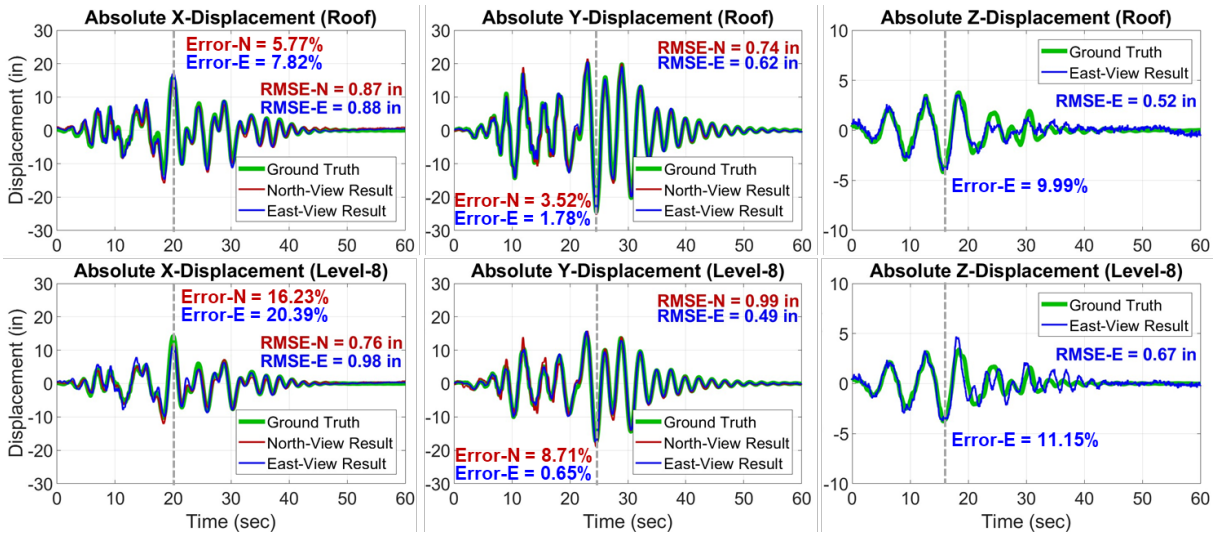


Figure 7.9. Comparisons between absolute XYZ translational displacement results and corresponding ground truth data with RMSEs from analysis incorporating videos in 3 different views.

between the rocking wall and bounding column obtained from the string potentiometer installed at the U-shape flexural plate (UFP) connection (Figure 7.6 (e)) divided by the width of the rocking wall. Figure 10 shows the absolute XY-rocking response results at Location 1 and 2 of the building specimen from the 6-DoF analysis and the error information of the response history. Although the rocking induced rotations are small for the rocking wall (less than 1 degree), the proposed UAV-based video analysis are still able to capture them with a reasonable agreement with the ground truth measurement. In this study, the RMSEs for the rocking-induced rotation are 0.0618 degree at Location 1 and 0.433 degree at Location 2.

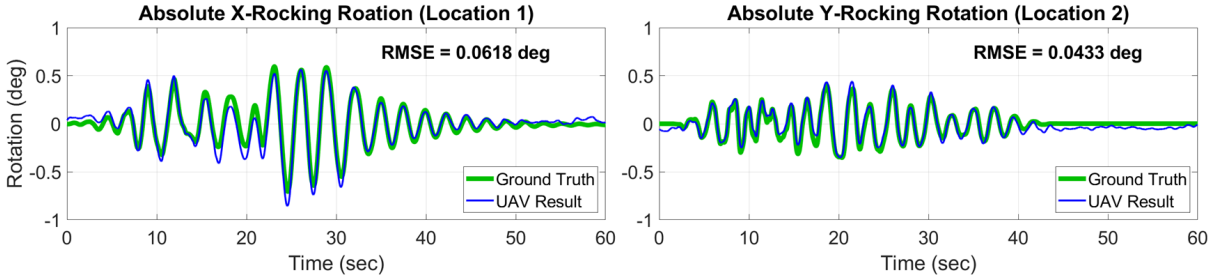


Figure 7.10. Comparisons between absolute XY-rocking-induced rotations from elevation-view analyses and corresponding ground truth data with RMSE.

7.6.3 Segmentation-based video analysis results

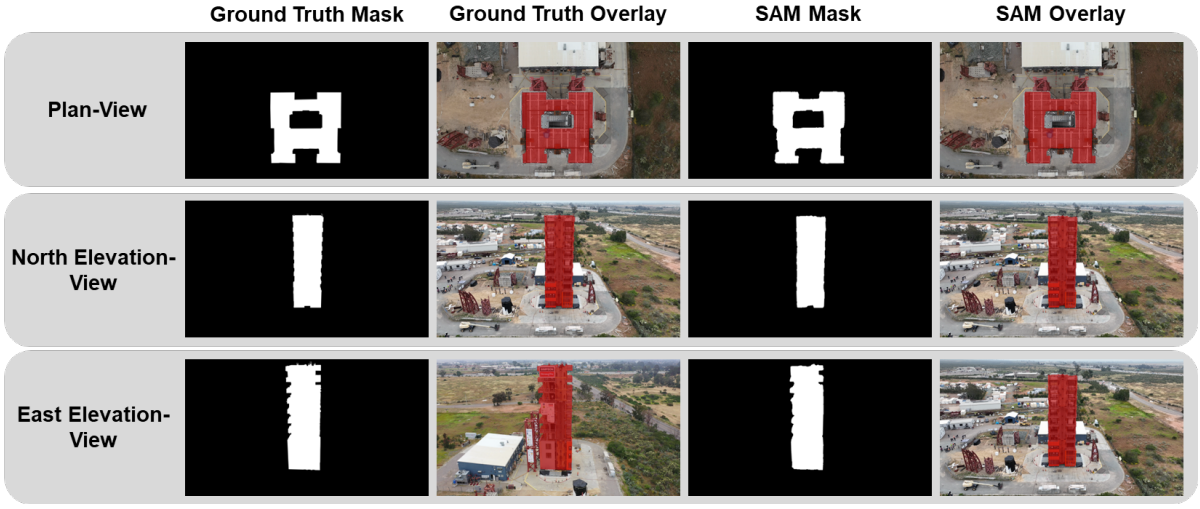


Figure 7.11. Qualitative comparisons of the ground truth and SAM segmented masks in video frames from three different views at t=5.03 s (Frame No.152).

The ground truth masks and semantic segmented masks are compared for each of the synchronized three-view videos. Figure 11 illustrates automatically generated masks of the building facades starting from the fine-tuned initial frame masks and their qualitative comparisons with the ground truth masks. The key performance metrics used for the assessment include accuracy, precision, recall, $F1$ score, and intersection over union (IoU). In this case, accuracy is measured by calculating the percentage of pixels that are correctly classified. And precision is computed using the number of true positive pixels divided by the sum of true positive and false positive pixels, while recall measures the number of true positive pixels divided by the sum of true positive and false negative pixels. F1 score in this segmentation task is computed by twice of the multiplication of precision and recall divided by the sum of precision and recall. And the IoU is calculated by dividing the area of overlap between the predicted segmentation and the ground truth by the area of their union. The quantitative comparison results are shown in Table 7.1.

Our proposed method demonstrates robustness and accuracy across video data of all three views. Specifically, the accuracy consistently exceeds 0.99, indicating an excellent classification

of the building against the background. The precision ranges from 0.955 to 0.995, thereby confirming the model's capability to correctly identify positive samples. Recall varies slightly with the lowest observed value being 0.966 in the east elevation view. Furthermore, $F1$ Scores and IoU values were also high across all perspectives, falling within the ranges of 0.968 to 0.983 and 0.938 to 0.966, respectively.

Table 7.1. Evaluation of the proposed segmentation method using SAM.

Views	**Accuracy**	**Precision**	**Recall**	$F1$ **Score**	IoU
Plan view	0.993	0.955	0.982	0.968	0.938
North elevation view	0.997	0.993	0.973	0.983	0.966
East elevation view	0.996	0.995	0.966	0.980	0.961

7.7 Chapter Conclusion

In conclusion, the presented study underscores the potential and advancements in utilizing UAVs for tracking dynamic building motions and in-earthquake real-time assessments of the dynamic response and validation of the conceptual design of earthquake resilient buildings. The authors notably address existing gaps in earthquake engineering applications, particularly in measuring the full 6-DoF motion response of buildings subjected to earthquake excitations. The novel video analysis methodology introduced in this chapter leverages off-the-shelf UAV platforms to capture comprehensive global views of building specimens. Both the feature tracking-based and semantic segmentation-based video analysis techniques are studied and explored, showing great potential of UAV imagery in earthquake engineering and SHM. For future work, the authors would like to further explore the segmentation-based methods to provide quantitative motion estimation of non-targeted regions of the test specimen. In addition, the potential of combining

two approaches for better understanding the complexity of building motions during an earthquake is yet to be studied. Last but not least, the research area of using UAV imagery as a tool for post-earthquake structural health assessments remains very intriguing to the authors.

7.8 Acknowledgements

This chapter, in full, is a reprint of the material as it may appear, titled "UAV-Based Video Analysis and Semantic Segmentation for SHM of Earthquake-excited Structures", in Proceedings of 2024 World Conference on Earthquake Engineering. Cao, Pengcheng; Ji, Ruipu; Ma, Zhaoyang; Sorosh, Shokrullah; Lo, Eric; Driscoll, John; Norton, Tanner; Wang, Xiang; Hutchinson, Tara; Pei, Shiling; Kuester, Falko. The dissertation author was the primary investigator and author of this paper.

The structural system scope of NHERI Tallwood Project is sponsored by NSF Grants No. 1635227, 1634628 and 1634204. The use and operation of the NHERI shake table facility is supported by NSF through Grant No.1520904. The nonstructural component scope of this project is sponsored by NSF Grant No. CMMI-1635363 and USFS Grant No. 19-DG-11046000-16, and industry sponsors. Numerous industry sponsors contributed to this effort (see http://nheritallwood.mines.edu/collaboration.html). The remote sensing and digital twin synthesis scope of this chapter is supported by the California Department of Forestry and Fire Protection (CALFIRE) under award #7CA04784, the USACE Engineer Research and Development Center Cooperative Agreement W9132T-22-2-0014 and the Kinsella Expedition Fund. The authors would like to thank the many contributors that are making this ongoing project possible. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of the sponsors.

Chapter 8

Dissertation Conclusion

8.1 Summary

This chapter serves as the epilogue of this PhD dissertation and also summarizes all the research efforts as composed in the format of seven chapters.

In Chapter 1, an introduction and a literature review are composed for different aspects involved in this dissertation. Next, the three critical problems that this dissertation strives to resolve are raised. At last, an overview of the skeleton and chapters is provided.

In Chapter 2, modeling methods and control schemes are proposed for fully-actuated hexacopters. This is aimed at circumventing the under-actuating nature of multi-rotor UAVs.

In Chapter 3, a multi-disciplinary design optimization based flight trajectory optimizer involving linear rotor inflow models is proposed to reduce flight time or energy consumption of specific missions.

In Chapter 4, a bio-inpired hybrid aerial robot design is proposed and validated to extend mutli-rotor flight time and fail-safe capability and provide fixed-wing glider with vertical take-off and landing or VTOL capability.

In Chapter 5, BeagleRover is invented and investigated as a low-cost 3D-printable double Ackermann steering chassis design with 2-wheel standing and balancing capability.

In Chapter 6, we propose the system design of a UGV capable of performing perception and mapping in a limited lighting, unstructured, and GPS-denied environment based on a

nevertheless nonholonomic chassis, where primary concern becomes the reliability in performing real-time LiDAR-based texture mapping and preservation of the solely static environment.

In Chapter 7, we combine the traditional UAV plan view perspective with north and east elevation view video data to provide motion estimation in all 6 degrees of freedom, as well as proposing Video Transformer for motion tracking.

8.2 Answers to Critical Problems

The critical problems this dissertation attempts to address are:

- For fully-actuated multirotor UAVs, can an navigation flight controller be designed to robustly and accurately track 3D trajectories while decoupling the attitude and linear velocity control?
- Can a mission-oriented multi-disciplinary design optimization (MDO) problem be formulated to optimize the multi-rotor flight trajectory considering advanced flight dynamics with rotor inflow model?
- Can we propose a bio-inspired UAV design with hybrid modality of both rotary-rotor drone and fixed-wing glider to provide fail-safe mechanism against motor failure?
- Can we design a low-cost 3D-printable UGV which can serve for both educational and field survey purposes?
- Can we develop a fast dynamic object filtering algorithm and make it a robust software package to filter out dynamic objects like moving pedestrians, bikers, scooters, and cars for real-time 3D point cloud mapping by UGVs or UAVs?
- And can we propose a novel UAV-based video analysis schemes utilizing all three-view perspectives to estimate 6-DoF structural motions? Additionally, can you apply the novel Visual Transformer model to our research?

Chapter 2 attempts to resolve the first critical problem. In Chapter 2, we formulate the mathematical model of fully-actuated hexacopters based on which the path tracking and tilt-hover controllers are designed and fine-tuned using the linear quadratic regulator. The path tracking controller decouples the lateral movements from the pitch and roll rotations of the vehicle, and the tilt-hover controller stabilizes the position of the center of mass of the vehicle as well as its heading while it is rotated by a certain pitch or roll angle, decoupling pitch and roll from lateral translations as well.

Chapter 3 attempts to resolve the second critical problem. In Chapter 3, we propose a novel approach to optimize trajectories with respect to mission specific objective functions. The optimal control problem is solved using nonlinear programming approach which is widely used for solving MDO problems. The equivalence of optimal control and MDO is discussed where the control input, objective functions, and trajectory output of the entire mission can be viewed as design variables to obtain an optimum for a quantitative performance. As expected, the MDO approach is able to minimize time or energy consumption of pre-designed missions to greater extent than do the common control approaches but with higher computational time.

Chapter 4 attempts to resolve the third critical problem. In Chapter 4, we first review the body features and gliding behavior of two types of gliding animals and acquired bio-inspirations from the relevant studies. Second, we discuss and present design features including morphing wings, inherent roll stability, and 4-bar linkages were presented and discussed as components of conceptual design. Third, the computational fluid dynamics studies indicate the airfoil of QuadGlider would be able to modulate airflow and produce sufficient lift at $Re = 4.7 \times 10^5$ at relatively low angles of attack $(0 - 30^\circ)$. Last but not least, the preliminary experiment validates QuadGlider's ability of developing a equilibrium glide after being launched at 1.6m AGL, and the estimated lift and drag coefficients model corresponded well with the experimental results. Overall, this is a successful attempt to design a hybrid UAV incorporating both multi-rotor UAV and fixed-wing glider modalities.

Chapter 5 attempts to resolve the fourth critical problem. In Chapter 5, we introduce

the design and implementation of the robotic platform BeagleRover developed for multiple educational and research purposes. We can draw the conclusion that low-cost 3D-printable mobile robots like BeagleRover can serve various purposes such as practical learning tool as well as research platform for robotic control and vision. Besides, with proper kinematics and control design, BeagleRover can realize complicated motions which are usually not possible for traditional 2-wheel or 4-wheel low-cost mobile robots, which allows for better maneuverability.

Chapter 6 attempts to resolve the fifth critical problem. In Chapter 6, we take a step forward by proposing a novel and lightweight dynamic object filtering algorithm, designed to facilitate the real-time generation of LiDAR-based static point cloud maps. This outlined method holds great potential as a reliable point cloud data source for diverse LiDAR-inertial-visual fusion mapping approaches.

Chapter 7 attempts to resolve the sixth critical problem. In Chapter 7, the presented study underscores the potential and advancements in utilizing UAVs for tracking dynamic building motions and in-earthquake real-time assessments of the dynamic response and validation of the conceptual design of earthquake resilient buildings. We notably address existing gaps by measuring the full 6-DoF motion response of buildings subjected to earthquake excitations. The novel video analysis methodology introduced in Chapter 7 leverages off-the-shelf UAV platforms to capture comprehensive global views of building specimens. Both the feature tracking-based and semantic segmentation-based video analysis techniques are studied and explored, showing great potential of UAV imagery in earthquake engineering and structural health monitoring.

8.3 Future Directions

Future directions in research on mobile robots, in particular UAVs and UGVs, are diverse and vibrant, reflecting the rapid advancements in technology and the expanding range of applications. Extending from the work in Chapter 2, future research directions can have more emphasis on payload handling and manipulation. Future work could explore how fully-actuated multi-rotors can interact with their environment, including manipulating objects, performing

repairs, or conducting scientific measurements in hard-to-reach places. In addition, swarming and formation control can be another interesting direction, where it could render valuable research by studying how multiple or a swarm of fully-actuated UAVs can work together efficiently, maintaining formation, coordinating tasks, or collectively manipulating objects.

Extending from the work in Chapter 3, the MDO frameworks can not only be used to solve trajectory optimization problem, but more importantly to perform simultaneous design and control or mission optimization for UAVs. Design parameters of UAVs, like payload weight, propeller diameter, and motor capacity, can be determined and optimized with the mission trajectory together.

Extending from the work in Chapter 4, the bio-inspired multi-modal UAV design can take inspirations from various control schemes and more control design attempts and tests need to be carried out.

Extending from the work in Chapter 5, the BeagleRover has been validated with preliminary SLAM tests but more data need to be collected through field work. The ROS based software package can have expanded functionality and also more control schemes can be implemented for 2-wheel balancing.

Extending from the work in Chapter 6, we would like to collect more field data from the tunnel sites to further validate the effectiveness of the methods. Future work may also involve studying the pros and cons of deep learning-based, transformer-based, and distanced cluster-based methods for identifying dynamic objects.

Extending from the work in Chapter 7, we plan to further explore the segmentation-based methods to provide quantitative motion estimation of non-targeted regions of the test specimen. In addition, the potential of combining two approaches for better understanding the complexity of building motions during an earthquake is yet to be studied. Last but not least, the research area of using UAV imagery as a tool for post-earthquake structural health assessments remains very intriguing to the dissertation author.

Bibliography

[AGL05] Mujahid Abdulrahim, Helen Garcia, and Rick Lind. "Flight characteristics of shaping the membrane wing of a micro air vehicle". In: *Journal of Aircraft* 42.1 (2005), pp. 131–137.

[Ahm+08] Ali Ahmadzadeh, James Keller, George Pappas, Ali Jadbabaie, and Vijay Kumar. "An optimization-based approach to time-critical cooperative surveillance and coverage with uavs". In: *Experimental robotics*. Springer. 2008, pp. 491–500.

[Alb+08] Joel Alberts, Dean Edwards, Terence Soule, Mike Anderson, and Michael O'Rourke. "Autonomous navigation of an unmanned ground vehicle in unstructured forest terrain". In: *2008 ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems (LAB-RS)*. IEEE. 2008, pp. 103–108.

[And14] Haavard Laegreid Andersen. "Path planning for search and rescue mission using multicopters". MA thesis. Institutt for teknisk kybernetikk, 2014.

[Ard06] Mark D Ardema. *Newton-Euler Dynamics*. Springer Science & Business Media, 2006.

[Bah+13] Joseph W Bahlman, Sharon M Swartz, Daniel K Riskin, and Kenneth S Breuer. "Glide performance and aerodynamics of non-equilibrium glides in northern flying squirrels (Glaucomys sabrinus)". In: *Journal of The Royal Society Interface* 10.80 (2013), p. 20120794.

[Bay+08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. "Speeded-up robust features (SURF)". In: *Computer vision and image understanding* 110.3 (2008), pp. 346–359.

[BD16] Dario Brescianini and Raffaello D'Andrea. "Design, modeling and control of an omni-directional aerial vehicle". In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2016, pp. 3261–3266.

[BD18] Dario Brescianini and Raffaello D'Andrea. "Computationally efficient trajectory generation for fully actuated multirotor vehicles". In: *IEEE Transactions on Robotics* 34.3 (2018), pp. 555–571.

[bea21] Beagle Board - beagleboard.org. *BeagleBone® Blue*. 2021. URL: https://beagleboard.org/blue.

[Beh+19] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. "Semantickitti: A dataset for semantic scene understanding of lidar sequences". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 9297–9307.

[Bem+02] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N Pistikopoulos. "The explicit linear quadratic regulator for constrained systems". In: *Automatica* 38.1 (2002), pp. 3–20.

[Bin+18] Ali Bin Junaid, Alejandro Diaz De Cerio Sanchez, Javier Betancor Bosch, Nikolaos Vitzilaios, and Yahya Zweiri. "Design and implementation of a dual-axis tilting quadcopter". In: *Robotics* 7.4 (2018), p. 65.

[BS07] Mitch Bryson and Salah Sukkarieh. "Co-operative localisation and mapping for multiple UAVs in unknown environments". In: *2007 IEEE aerospace conference*. IEEE. 2007, pp. 1–12.

[Bus23] Aleesha Busch. “Design and Construction of Tall Mass Timber Buildings With Resilient Post-Tensioned Mass Timber Rocking Walls”. PhD thesis. Colorado School of Mines, 2023.

[BZ17] Stefan Barthelmes and Sebastian Zehnter. “An all-terrain-controller for over-actuated wheeled mobile robots with feedforward and optimization-based control allocation”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 5215–5222.

[Cab+18] Taua M Cabreira, Carmelo Di Franco, Paulo R Ferreira, and Giorgio C Buttazzo. “Energy-aware spiral coverage path planning for uav photogrammetric applications”. In: *IEEE Robotics and automation letters* 3.4 (2018), pp. 3662–3668.

[Cab+19] Taua M Cabreira, Paulo R Ferreira, Carmelo Di Franco, and Giorgio C Buttazzo. “Grid-based coverage path planning with minimum energy over irregular-shaped areas with UAVs”. In: *2019 international conference on unmanned aircraft systems (ICUAS)*. IEEE. 2019, pp. 758–767.

[Can86] John Canny. “A computational approach to edge detection”. In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), pp. 679–698.

[Cao+21] Pengcheng Cao, James Strawson, Thomas Bewley, and Falko Kuester. “Decoupled translational and rotational flight control designs of canted-rotor hexacopters”. In: *AIAA Scitech 2021 Forum*. 2021, p. 1058.

[Cao+22a] Pengcheng Cao, John T Hwang, Thomas Bewley, and Falko Kuester. “Mission-Oriented Trajectory Optimization for Search-and-Rescue Multirotor UAVs in Cluttered and GPS-Denied Environments”. In: *AIAA AVIATION 2022 Forum*. 2022, p. 3999.

[Cao+22b] Pengcheng Cao, James Strawson, Xuebin Zhu, Everbrook Zhou, Chase Lazar, Dominique Meyer, Zhaoliang Zheng, Thomas Bewley, and Falko Kuester. "Beaglerover: An open-source 3D-printable robotic platform for engineering education and research". In: *AIAA SCITECH 2022 Forum*. 2022, p. 1914.

[Car14] Stephen Carlson. "A hybrid tricopter/flying-wing vtol uav". In: *52nd AIAA Aerospace Sciences Meeting*. 2014, p. 0016.

[CBK23] Pengcheng Cao, Thomas Bewley, and Falko Kuester. "Cluster-based Dynamic Object Filtering via Egocentric Motion Detection for Building Static 3D Point Cloud Maps". In: *Proceedings of the 2023 IEEE Conference on Robotic Computing*. IEEE, 2023.

[CCH03] Lance Champagne, R Greg Carl, and Raymond Hill. "Agent models II: Search theory, agent-based simulation, and U-boats in the Bay of Biscay". In: *Proceedings of the 35th conference on Winter simulation: driving innovation*. Citeseer. 2003, pp. 991–998.

[CCL11a] Guowei Cai, Ben M Chen, and Tong Heng Lee. "Coordinate systems and transformations". In: *Unmanned rotorcraft systems*. Springer, 2011, pp. 23–34.

[CCL11b] Guowei Cai, Ben M. Chen, and Tong Heng Lee. "Coordinate Systems and Transformations". In: *Unmanned Rotorcraft Systems*. London: Springer London, 2011, pp. 23–34. ISBN: 978-0-85729-635-1. DOI: 10.1007/978-0-85729-635-1_2.

[Che+22] Kenny Chen, Brett T Lopez, Ali-akbar Agha-mohammadi, and Ankur Mehta. "Direct lidar odometry: Fast localization with dense point clouds". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 2000–2007.

[Chi+20] Yao-Wei Chin, Jia Ming Kok, Yong-Qiang Zhu, Woei-Leong Chan, Javaan S Chahl, Boo Cheong Khoo, and Gih-Keong Lau. "Efficient flapping wing drone arrests high-speed flight using post-stall soaring". In: *Science Robotics* 5.44 (2020), eaba2386.

[Cho+08] Min Wan Choi, Jun Seok Park, Bong Soo Lee, and Man Hyung Lee. "The performance of independent wheels steering vehicle (4WS) applied Ackerman geometry". In: *2008 International Conference on Control, Automation and Systems*. IEEE. 2008, pp. 197–202.

[CK15] Chun Cheng and Ken'ichi Kawaguchi. "A preliminary study on the response of steel structures using surveillance camera image with vision-based method during the Great East Japan Earthquake". In: *Measurement* 62 (2015), pp. 142–148.

[Col+19] Anthony Colaprete, Daniel Andrews, William Bluethmann, Richard C Elphic, Ben Bussey, Jay Trimble, Kris Zacny, and Janine E Captain. "An overview of the volatiles investigating polar exploration rover (viper) mission". In: *AGU Fall Meeting Abstracts*. Vol. 2019. 2019, P34B–03.

[CWL09] Hai Chen, Xin-min Wang, and Yan Li. "A survey of autonomous control for UAV". In: *2009 International Conference on Artificial Intelligence and Computational Intelligence*. Vol. 2. IEEE. 2009, pp. 267–271.

[Dav+20] Behdad Davoudi, Ehsan Taheri, Karthik Duraisamy, Balaji Jayaraman, and Ilya Kolmanovsky. "Quad-rotor flight simulation in realistic atmospheric conditions". In: *AIAA Journal* 58.5 (2020), pp. 1992–2004.

[DBU20] Ozgur Dundar, Mesut Bilici, and Tarik Unler. "Design and performance analyses of a fixed wing battery VTOL UAV". In: *Engineering Science and Technology, an International Journal* 23.5 (2020), pp. 1182–1193.

[DD19] Behdad Davoudi and Karthikeyan Duraisamy. “A hybrid blade element momentum model for flight simulation of rotary wing unmanned aerial vehicles”. In: *AIAA Aviation 2019 Forum*. 2019, p. 2823.

[DHS87] Carl De Boor, Klaus Hollig, and Malcolm Sabin. “High accuracy geometric Hermite interpolation”. In: *Computer Aided Geometric Design* 4.4 (1987), pp. 269–278.

[Di +17] Matteo Di Luca, Stefano Mintchev, Gregoire Heitz, Flavio Noca, and Dario Floreano. “Bioinspired morphing wings for extended flight envelope and roll control of small drones”. In: *Interface focus* 7.1 (2017), p. 20160092.

[DTK19] Luca De Vivo, Danny Tran, and Falko Kuester. “Towards Design of a 3D Printable Prandtl Box-Wing Unmanned Aerial Vehicle”. In: *2019 IEEE Aerospace Conference*. IEEE. 2019, pp. 1–17.

[edu18] Team eduMIP. *EduMIP*. Feb. 2018. URL: https://beagleboard.org/p/edumip/edumip-13a29c.

[ETM10] Jeff D Eldredge, Jonathan Toomey, and Albert Medina. “On the roles of chord-wise flexibility in a flapping wing with hovering kinematics”. In: *Journal of Fluid Mechanics* 659 (2010), pp. 94–115.

[Fal+21] Robert Falck, Justin S Gray, Kaushik Ponnapalli, and Ted Wright. “dymos: A Python package for optimal control of multidisciplinary systems”. In: *Journal of Open Source Software* 6.59 (2021), p. 2809.

[Fay+20] Jamil Fayyad, Mohammad A Jaradat, Dominique Gruyer, and Homayoun Najjaran. “Deep learning sensor fusion for autonomous vehicle perception and localization: A review”. In: *Sensors* 20.15 (2020), p. 4220.

[FB81] Martin A Fischler and Robert C Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[FG19] Robert D Falck and Justin S Gray. "Optimal control within the context of multidisciplinary design, analysis, and optimization". In: *AIAA Scitech 2019 Forum*. 2019, p. 0976.

[FGS22] Paul G Fahlstrom, Thomas J Gleason, and Mohammad H Sadraey. *Introduction to UAV systems*. John Wiley & Sons, 2022.

[Flo+12] Gerardo Ramon Flores, Juan Escareno, Rogelio Lozano, and Sergio Salazar. "Quad-tilting rotor convertible mav: Modeling and real-time hover flight control". In: *Journal of intelligent & robotic systems* 65.1 (2012), pp. 457–471.

[FN17] Florentin von Frankenberg and Scott Nokleby. "Disturbance rejection in multi-rotor unmanned aerial vehicles using a novel rotor geometry". In: *Proc. 4th Int. Conf. Control Dynamic Systems and Robotics*. 2017.

[Gao+03] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. "Complete solution classification for the perspective-three-point problem". In: *IEEE transactions on pattern analysis and machine intelligence* 25.8 (2003), pp. 930–943.

[Gao+15] G Gao, S Huang, K Xia, and Z Li. "Application of digital image correlation (DIC) in dynamic notched semi-circular bend (NSCB) tests". In: *Experimental Mechanics* 55 (2015), pp. 95–104.

[GD17] Rajan Gill and Raffaello D'Andrea. "Propeller thrust and drag in forward flight". In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2017, pp. 73–79.

[Ger+14] John Gerdes, Alex Holness, Ariel Perez-Rosado, Luke Roberts, Eli Barnett, Johannes Kempny, Deepak Lingam, Chen-Haur Yeh, Hugh A Bruck, and Adrian Greisinger. "Robo Raven: a flapping-wing air vehicle with highly compliant and independently controlled wings". In: *Soft Robotics* 1.4 (2014), pp. 275–288.

[Gla35] Hermann Glauert. "Airplane propellers". In: *Aerodynamic theory*. Springer, 1935, pp. 169–360.

[GMS05] Philip E Gill, Walter Murray, and Michael A Saunders. "SNOPT: An SQP algorithm for large-scale constrained optimization". In: *SIAM review* 47.1 (2005), pp. 99–131.

[Gon+12] Juan Gonzalez-Gomez, Alberto Valero-Gomez, Andres Prieto-Moreno, and Mohamed Abderrahim. "A new open source 3d-printable mobile robotic platform for education". In: *Advances in autonomous mini robots*. Springer, 2012, pp. 49–62.

[GR05] Ohad Gur and Aviv Rosen. "Propeller performance at low advance ratio". In: *Journal of aircraft* 42.2 (2005), pp. 435–441.

[Gra+19] Justin S Gray, John T Hwang, Joaquim RRA Martins, Kenneth T Moore, and Bret A Naylor. "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization". In: *Structural and Multidisciplinary Optimization* 59.4 (2019), pp. 1075–1104.

[GSG16] J. I. Giribet, R. S. Sanchez-Pena, and A. S. Ghersin. "Analysis and Design of a Tilted Rotor Hexacopter for Fault Tolerance". In: *IEEE Transactions on Aerospace and Electronic Systems* 52.4 (Aug. 2016), pp. 1555–1567. ISSN: 2371-9877. DOI: 10.1109/TAES.2016.140885.

[Gul+17] Jahan Zeb Gul, Young Jin Yang, Kim Young Su, and Kyung Hyun Choi. "Omni directional multimaterial soft cylindrical actuator and its application as a steerable catheter". In: *Soft robotics* 4.3 (2017), pp. 224–240.

[Guo+22] Yunhui Guo, Chaofeng Wang, Stella X Yu, Frank McKenna, and Kincho H Law. "AdaLN: A Vision Transformer for Multidomain Learning and Predisaster Building Information Extraction from Images". In: *Journal of Computing in Civil Engineering* 36.5 (2022), p. 04022024.

[Hah+03] Dirk Hahnel, Rudolph Triebel, Wolfram Burgard, and Sebastian Thrun. "Map building with mobile robots in dynamic environments". In: *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*. Vol. 2. IEEE. 2003, pp. 1557–1563.

[Hay+17] Samira Hayat, Evsen Yanmaz, Timothy X Brown, and Christian Bettstetter. "Multi-objective UAV path planning for search and rescue". In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 5569–5574.

[Her+15] Eduardo Gamaliel Hernandez-Martinez, Guillermo Fernandez-Anaya, ED Ferreira, Jose Job Flores-Godoy, and Alexandro Lopez-Gonzalez. "Trajectory tracking of a quadcopter UAV with optimal translational control". In: *IFAC-PapersOnLine* 48.19 (2015), pp. 226–231.

[Her58] Albert W. Herre. "On the Gliding of Flying Lizards, Genus Draco". In: *Copeia* 1958.4 (1958), pp. 338–339. ISSN: 00458511, 19385110. URL: http://www.jstor.org/stable/1439979 (visited on 10/14/2022).

[HK04] Tara C Hutchinson and Falko Kuester. "Monitoring global earthquake-induced demands using vision-based sensors". In: *IEEE Transactions on Instrumentation and Measurement* 53.1 (2004), pp. 31–36.

[HM18] John T Hwang and Joaquim RRA Martins. "A computational architecture for coupling heterogeneous numerical models and computing coupled derivatives". In: *ACM Transactions on Mathematical Software (TOMS)* 44.4 (2018), pp. 1–39.

[HM20] Lionel Hulttinen and Jouni Mattila. "Flow-limited path-following control of a double Ackermann steered hydraulic mobile manipulator". In: *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE. 2020, pp. 625–630.

[Hon+07] Hongwei Fang, Changliang Xia, Zhengwei Chen, and Xile Wei. "Position servo control of brushless DC motor based on the second discrete filter". In: *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2007, pp. 1838–1842. DOI: 10.1109/ROBIO.2007.4522446.

[Hor+13] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees". In: *Autonomous robots* 34 (2013), pp. 189–206.

[Hos+20] Vedhus Hoskere, Yasutaka Narazaki, Tu A Hoang, and BF Spencer Jr. "MaDnet: multi-task semantic segmentation of multiple types of structural materials and damage in images of civil infrastructure". In: *Journal of Civil Structural Health Monitoring* 10 (2020), pp. 757–773.

[HS+88] Chris Harris, Mike Stephens, et al. "A combined corner and edge detector". In: *Alvey vision conference*. Vol. 15. 50. Citeseer. 1988, pp. 10–5244.

[Hua+05] Hui-Min Huang, Kerry Pavek, Brian Novak, James Albus, and E Messin. "A framework for autonomy levels for unmanned systems (ALFUS)". In: *Proceedings of the AUVSI's Unmanned Systems North America* (2005), pp. 849–863.

[Hua23] Da Huang. "Optimized Seismic Design and Dynamic Response Analysis of Mass Timber Rocking Wall Lateral System". PhD thesis. Colorado School of Mines, 2023.

[HW20] Bin Hu and Jiacun Wang. "Deep learning based hand gesture recognition and UAV flight controls". In: *International Journal of Automation and Computing* 17.1 (2020), pp. 17–29.

[Hwa+14] John T Hwang, Dae Young Lee, James W Cutler, and Joaquim RRA Martins. "Large-scale multidisciplinary optimization of a small satellite's design and operation". In: *Journal of Spacecraft and Rockets* 51.5 (2014), pp. 1648–1663.

[HWF22] Yitian Han, Gang Wu, and Dongming Feng. "Vision-based displacement measurement using an unmanned aerial vehicle". In: *Structural Control and Health Monitoring* 29.10 (2022), e3025.

[JJK19] Sunghun Jung, Yonghyeon Jo, and Young-Joon Kim. "Flight time estimation for continuous surveillance missions using a multirotor UAV". In: *Energies* 12.5 (2019), p. 867.

[JJT+07] Hrvoje Jasak, Aleksandar Jemcov, Zeljko Tukovic, et al. "OpenFOAM: A C++ library for complex physics simulations". In: *International workshop on coupled methods in numerical dynamics*. Vol. 1000. IUC Dubrovnik Croatia. 2007, pp. 1–20.

[Joc+20] Glenn Jocher, K Nishimura, T Mineeva, and R Vilarino. "Yolov5". In: *GitHub* (2020). URL: https://github.%20com/ultralytics/yolov5.

[JV13] Guangying Jiang and Richard Voyles. "Hexrotor uav platform enabling dextrous interaction with structures-flight test". In: *2013 IEEE international*

symposium on safety, security, and rescue robotics (SSRR). IEEE. 2013, pp. 1–6.

[JV18] Amila Jakubovic and Jasmin Velagic. "Image feature matching and object detection using brute-force matchers". In: *2018 International Symposium ELMAR*. IEEE. 2018, pp. 83–86.

[KA17] Cezary Kownacki and Leszek Ambroziak. "Local and asymmetrical potential field approach to leader tracking problem in rigid formations of fixed-wing UAVs". In: *Aerospace Science and Technology* 68 (2017), pp. 465–474.

[KAB16] Benjamin Kirsch, Alexander Alexopoulos, and Essameddin Badreddin. "Non-linear model based control and parameter identification of a hex-rotor UAV". In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2016, pp. 002609–002614.

[Kam+18] Mina Kamel, Sebastian Verling, Omar Elkhatib, Christian Sprecher, Paula Wulkop, Zachary Taylor, Roland Siegwart, and Igor Gilitschenski. "The Voliro Omniorientational Hexacopter: An Agile and Maneuverable Tiltable-Rotor Aerial Vehicle". In: *IEEE Robotics and Automation Magazine* 25.4 (Dec. 2018), pp. 34–44. ISSN: 1558-223X. DOI: 10.1109/mra.2018.2866758.

[Kao+15] Jason Kao, John Hwang, Joaquim RRA Martins, Justin S Gray, and Kenneth T Moore. "A modular adjoint approach to aircraft mission analysis and optimization". In: *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. 2015, p. 0136.

[Kar14] Matej Karasek. "Robotic hummingbird: Design of a control mechanism for a hovering flapping wing micro air vehicle". In: *Universite libre de Bruxelles: Bruxelles, Belgium* (2014).

[Ken12] Farid Kendoul. “Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems”. In: *Journal of Field Robotics* 29.2 (2012), pp. 315–378.

[KF11] Sertac Karaman and Emilio Frazzoli. “Sampling-based algorithms for optimal motion planning”. In: *The international journal of robotics research* 30.7 (2011), pp. 846–894.

[KFL06] Farid Kendoul, Isabelle Fantoni, and Rogelio Lozano. “Modeling and control of a small autonomous aircraft having two tilting rotors”. In: *IEEE Transactions on Robotics* 22.6 (2006), pp. 1297–1302.

[KH22] Pranav C Khandelwal and Tyson L Hedrick. “Combined effects of body posture and three-dimensional wing shape enable efficient gliding in flying lizards”. In: *Scientific reports* 12.1 (2022), pp. 1–11.

[Kim99] Ki C Kim. *Analytical Calculations of Helicopter Torque Coefficient (CQ) and Thrust Coefficient (CT) Values for the Helicopter Performance (HELPE) Model.* Tech. rep. ARMY RESEARCH LAB, 1999.

[Kir+23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. “Segment anything”. In: *arXiv preprint: 2304.02643* (2023).

[KJD18] Jelena Kocic, Nenad Jovic, and Vujo Drndarevic. “Sensors and sensor fusion in autonomous vehicles”. In: *2018 26th Telecommunications Forum (TELFOR).* IEEE. 2018, pp. 420–425.

[KK20] Giseop Kim and Ayoung Kim. “Remove, then revert: Static point cloud map construction using multiresolution range images”. In: *2020 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS). IEEE. 2020, pp. 10758–10765.

[KS08] Daisuke Kubo and Shinji Suzuki. "Tail-sitter vertical takeoff and landing unmanned aerial vehicle: transitional flight analysis". In: *Journal of Aircraft* 45.1 (2008), pp. 292–297.

[Kul+20] Shriyanti Kulkarni, Vedashree Chaphekar, Md Moin Uddin Chowdhury, Fatih Erden, and Ismail Guvenc. "Uav aided search and rescue operation using reinforcement learning". In: *2020 SoutheastCon*. Vol. 2. IEEE. 2020, pp. 1–8.

[KVB88] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. "Design of an image edge detection filter using the Sobel operator". In: *IEEE Journal of solid-state circuits* 23.2 (1988), pp. 358–367.

[KWC18] Yijie Ke, Kangli Wang, and Ben M Chen. "Design and implementation of a hybrid UAV with model-based flight capabilities". In: *IEEE/ASME Transactions on Mechatronics* 23.3 (2018), pp. 1114–1125.

[Lap+14] Matthieu Lapeyre, Pierre Rouanet, Jonathan Grizou, Steve Nguyen, Fabien Depraetre, Alexandre Le Falher, and Pierre-Yves Oudeyer. "Poppy project: open-source fabrication of 3D printed humanoid robot for science, education and art". In: *Digital Intelligence 2014*. 2014, p. 6.

[LBL13] Dimitri Lague, Nicolas Brodu, and Jerome Leroux. "Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (NZ)". In: *ISPRS journal of photogrammetry and remote sensing* 82 (2013), pp. 10–26.

[LF18] Wenjun Li and Zhaoyu Fu. “Unmanned aerial vehicle positioning based on multi-sensor information fusion”. In: *Geo-Spatial Information Science* 21.4 (2018), pp. 302–310.

[LHM21] Hyungtae Lim, Sungwon Hwang, and Hyun Myung. “ERASOR: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3D point cloud map building”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 2272–2279.

[Lia19] Xiao Liang. “Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization”. In: *Computer-Aided Civil and Infrastructure Engineering* 34.5 (2019), pp. 415–430.

[Lin12] Tony Lindeberg. “Scale invariant feature transform”. In: (2012).

[Liu+18] Chi Harold Liu, Zheyu Chen, Jian Tang, Jie Xu, and Chengzhe Piao. “Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach”. In: *IEEE Journal on Selected Areas in Communications* 36.9 (2018), pp. 2059–2070.

[LLH21] Kailai Li, Meng Li, and Uwe D Hanebeck. “Towards high-performance solid-state-lidar-inertial odometry and mapping”. In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 5167–5174.

[LMF09] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. “EP n P: An accurate O (n) solution to the P n P problem”. In: *International journal of computer vision* 81 (2009), pp. 155–166.

[Lop+16] Francisco Ronay Lopez-Estrada, Jean-Christophe Ponsart, Didier Theilliol, Youmin Zhang, and Carlos-Manuel Astorga-Zaragoza. “LPV model-based

tracking control and robust sensor fault diagnosis for a quadrotor UAV". In: *Journal of Intelligent & Robotic Systems* 84.1 (2016), pp. 163–177.

[LS11] Yong-Ju Lee and Jae-Bok Song. "Three-dimensional iterative closest point-based outdoor SLAM using terrain classification". In: *Intelligent Service Robotics* 4 (2011), pp. 147–158.

[LT10] Jesse Levinson and Sebastian Thrun. "Robust vehicle localization in urban environments using probabilistic maps". In: *2010 IEEE international conference on robotics and automation*. IEEE. 2010, pp. 4372–4378.

[Lyu+18] Hyeonsu Lyu et al. "Detect and avoid system based on multi sensor fusion for UAV". In: *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE. 2018, pp. 1107–1109.

[Ma+13] Kevin Y Ma, Pakpong Chirarattananon, Sawyer B Fuller, and Robert J Wood. "Controlled flight of a biologically inspired, insect-scale robot". In: *Science* 340.6132 (2013), pp. 603–607.

[mak18] makeblock. *Mbot*. 2018. URL: https://www.makeblock.com/mbot-3.

[Mas+18] Ruben Mascaro, Lucas Teixeira, Timo Hinzmann, Roland Siegwart, and Margarita Chli. "Gomsf: Graph-optimization based multi-sensor fusion for robust uav pose estimation". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1421–1428.

[Mat21] Mathworks. *UAV toolbox*. 2021. URL: https://www.mathworks.com/products/uav.html.

[MD05] Jimmy A McGuire and Robert Dudley. "The cost of living large: comparative gliding performance in flying lizards (Agamidae: Draco)". In: *The American Naturalist* 166.1 (2005), pp. 93–106.

[MFW12] Kevin Y Ma, Samuel M Felton, and Robert J Wood. "Design, fabrication, and modeling of the split actuator microrobotic bee". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 1133–1140.

[MH13] Joaquim RRA Martins and John T Hwang. "Review and unification of methods for computing derivatives of multidisciplinary computational models". In: *AIAA journal* 51.11 (2013), pp. 2582–2599.

[MK11a] D. Mellinger and V. Kumar. "Minimum snap trajectory generation and control for quadrotors". In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 2520–2525. DOI: 10.1109/ICRA.2011.5980409.

[MK11b] Daniel Mellinger and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors". In: *2011 IEEE international conference on robotics and automation*. IEEE. 2011, pp. 2520–2525.

[ML13] Joaquim RRA Martins and Andrew B Lambe. "Multidisciplinary design optimization: a survey of architectures". In: *AIAA journal* 51.9 (2013), pp. 2049–2075.

[MMD16] Fadl Moukalled, Luca Mangani, and Marwan Darwish. "The finite volume method". In: *The finite volume method in computational fluid dynamics*. Springer, 2016, pp. 103–135.

[MPB08] Robin R Murphy, Kevin S Pratt, and Jennifer L Burke. "Crew roles and operational protocols for rotary-wing micro-UAVs in close urban environments". In: *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*. 2008, pp. 73–80.

[Muh+12] Abubakr Muhammad, S Abbas, Talha Manzoor, Adnan Munawar, S Abbas, Mhequb Hayat, Ali Abbas, and M Awais. "Marwa: A rough terrain landmine detection robot for low budgets". In: *Proceedings of the 43rd International Symposium on Robotics*. 2012.

[MV17] Rens MacNeill and Dries Verstraete. "Blade element momentum theory extended to model low Reynolds number propeller performance". In: *The Aeronautical Journal* 121.1240 (2017), pp. 835–857.

[Nar+21] Yasutaka Narazaki, Vedhus Hoskere, Koji Yoshida, Billie F Spencer, and Yozo Fujino. "Synthetic environments for vision-based structural condition assessment of Japanese high-speed railway viaducts". In: *Mechanical Systems and Signal Processing* 160 (2021), p. 107850.

[NK14a] A. Nemati and M. Kumar. "Modeling and control of a single axis tilting quadcopter". In: *2014 American Control Conference*. 2014, pp. 3077–3082. DOI: 10.1109/ACC.2014.6859328.

[NK14b] Alireza Nemati and Manish Kumar. "Modeling and control of a single axis tilting quadcopter". In: *2014 American Control Conference*. IEEE. 2014, pp. 3077–3082.

[NKH+16] Iram Noreen, Amna Khan, Zulfiqar Habib, et al. "Optimal path planning using RRT* based approaches: a survey and future directions". In: *Int. J. Adv. Comput. Sci. Appl* 7.11 (2016), pp. 97–107.

[NM20] Luna Ngeljaratan and Mohamed A Moustafa. "Structural health monitoring and seismic response assessment of bridge structures using target-tracking digital image correlation". In: *Engineering Structures* 213 (2020), p. 110551.

[NR14] Francesco Nex and Fabio Remondino. “UAV for 3D mapping applications: a review”. In: *Applied geomatics* 6.1 (2014), pp. 1–15.

[Oli17] Mauricio C de Oliveira. “pyctrl: a Python Suite for Systems and Control”. In: (2017).

[OR05] John Osborne and Rolf Rysdyk. “Waypoint guidance for small UAVs in wind”. In: *Infotech@ Aerospace*. 2005, p. 6951.

[Ozd+14] Ugur Ozdemir, Yucel Orkut Aktas, Aslihan Vuruskan, Yasin Dereli, Ahmed Farabi Tarhan, Karaca Demirbag, Ahmet Erdem, Ganime Duygu Kalaycioglu, Ibrahim Ozkol, and Gokhan Inalhan. “Design of a commercial hybrid VTOL UAV system”. In: *Journal of Intelligent & Robotic Systems* 74.1 (2014), pp. 371–393.

[Pei+] Shiling Pei, John W van de Lindt, Jeffrey Berman, Keri Ryan, James D Dolan, Steve Pryor, Sarah Wichman, Aleesha Busch, and Reid Zimmerman. “FULL-SCALE 3-D SHAKE TABLE TEST OF A TEN-STORY MASS TIMBER BUILDING”. In: ().

[Pen+09] Kemao Peng, Guowei Cai, Ben M Chen, Miaobo Dong, Kai Yew Lum, and Tong H Lee. “Design and implementation of an autonomous flight control law for a UAV helicopter”. In: *Automatica* 45.10 (2009), pp. 2333–2338.

[Pen+20] Jing Peng, Ping Zhang, Lanxiang Zheng, and Jia Tan. “UAV positioning based on multi-sensor fusion”. In: *IEEE Access* 8 (2020), pp. 34455–34467.

[Pit80] Dale Marvin Pitt. *Rotor dynamic inflow derivatives and time constants from various inflow models*. Washington University in St. Louis, 1980.

[PPK17] Marzena Polka, Szymon Ptak, and Lukasz Kuziora. "The use of UAV's for search and rescue operations". In: *Procedia engineering* 192 (2017), pp. 748–752.

[QH14] Steven AP Quintero and Joao P Hespanha. "Vision-based target tracking with a small UAV: Optimization-based control strategies". In: *Control Engineering Practice* 32 (2014), pp. 28–42.

[Qi+16] Juntong Qi, Dalei Song, Hong Shang, Nianfa Wang, Chunsheng Hua, Chong Wu, Xin Qi, and Jianda Han. "Search and rescue rotary-wing uav and its application to the lushan ms 7.0 earthquake". In: *Journal of Field Robotics* 33.3 (2016), pp. 290–321.

[Raj+15] Sujit Rajappa, Markus Ryll, Heinrich H Bulthoff, and Antonio Franchi. "Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers". In: *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2015, pp. 4006–4013.

[Ras+20a] R. Rashad, J. Goerres, R. Aarts, J. B. C. Engelen, and S. Stramigioli. "Fully Actuated Multirotor UAVs: A Literature Review". In: *IEEE Robotics Automation Magazine* 27.3 (2020), pp. 97–107.

[Ras+20b] Ramy Rashad, Jelmer Goerres, Ronald Aarts, Johan BC Engelen, and Stefano Stramigioli. "Fully actuated multirotor UAVs: A literature review". In: *IEEE Robotics & Automation Magazine* 27.3 (2020), pp. 97–107.

[RBA15] Rethnaraj Rambabu, Muhammad Rijaluddin Bahiki, and Syaril Azrad. "Multi-sensor fusion based UAV collision avoidance system". In: *Jurnal Teknologi* 76.8 (2015).

[RBG12] Markus Ryll, Heinrich H Bulthoff, and Paolo Robuffo Giordano. "Modeling and control of a quadrotor UAV with tilting propellers". In: *2012 IEEE International Conference on Robotics and Automation*. IEEE. 2012, pp. 4606–4613.

[RF03] I Michael Ross and Fariba Fahroo. "Legendre pseudospectral approximations of optimal control problems". In: *New trends in nonlinear dynamics and control and their applications*. Springer, 2003, pp. 327–342.

[RH21] Marius L Ruh and John T Hwang. "Robust modeling and optimal design of rotors using blade element momentum theory". In: *AIAA AVIATION 2021 FORUM*. 2021, p. 2598.

[RL14] Mark Rollins and Mannie Lowe. *Beginning Lego Mindstorms Ev3*. Vol. 253. Springer, 2014.

[RLO18] Fabio Ruggiero, Vincenzo Lippiello, and Anibal Ollero. "Aerial manipulation: A literature review". In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1957–1964.

[RO+90] Mitchel Resnick, Stephen Ocko, et al. *LEGO/logo–learning through and about design*. Epistemology and Learning Group, MIT Media Laboratory Cambridge, 1990.

[Ros+17] A Roshanbin, Hussein Altartouri, Matej Karasek, and Andre Preumont. "COLIBRI: A hovering flapping twin-wing robot". In: *International Journal of Micro Air Vehicles* 9.4 (2017), pp. 270–282.

[Rus10] Radu Bogdan Rusu. "Semantic 3D object maps for everyday manipulation in human living environments". In: *KI-Kuenstliche Intelligenz* 24 (2010), pp. 345–348.

[Sab15] Francesco Sabatino. *Quadrotor control: modeling, nonlinearcontrol design, and simulation*. 2015.

[Sac16] Gottfried Sachs. "Comparison of power requirements: Flapping vs. fixed wing vehicles". In: *Aerospace* 3.4 (2016), p. 31.

[Sal+08] Sergio Salazar, Hugo Romero, Rogelio Lozano, and Pedro Castillo. "Modeling and real-time stabilization of an aircraft having eight rotors". In: *Unmanned Aircraft Systems*. Springer, 2008, pp. 455–470.

[San+20] Pedro J Sanchez-Cuevas, Antonio Gonzalez-Morgado, Nicolas Cortes, Diego B Gayango, Antonio E Jimenez-Cano, Anibal Ollero, and Guillermo Heredia. "Fully-Actuated Aerial Manipulator for Infrastructure Contact Inspection: Design, Modeling, Localization, and Control". In: *Sensors* 20.17 (2020), p. 4708.

[SE19] Tiago Lobato de Souza and Larissa Sato Elisiario. "Educational robotics teaching with Arduino and 3d print based on stem projects". In: *2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)*. IEEE. 2019, pp. 407–410.

[Sha+20] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping". In: *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2020, pp. 5135–5142.

[Sha+21] Tixiao Shan, Brendan Englot, Carlo Ratti, and Daniela Rus. "Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping". In: *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2021, pp. 5692–5698.

[She+18] Geyi Sheng, Minghui Min, Liang Xiao, and Sicong Liu. "Reinforcement learning-based control for unmanned aerial vehicles". In: *Journal of Communications and Information Networks* 3.3 (2018), pp. 39–48.

[Shi+15] T. Shimizu, S. Suzuki, T. Kawamura, H. Ueno, and H. Murakami. "Proposal of 6DOF multi-copter and verification of its controllability". In: *Society of Instrument and Control Engineers of Japan (SICE), 2015 54th Annual Conference of the* (July 2015), pp. 810–815. DOI: 10.1109/SICE.2015.7285456.

[SHR22] S Sorosh, TC Hutchinson, and KL Ryan. "NHERI TallWood 10-story Test Nonstructural, Part 4 of 4: Prefabricated Steel Stair Subassemblies". In: *Proceedings of the 12th National Conference in Earthquake Engineering*. 2022.

[Sil+17] Mario Silvagni, Andrea Tonoli, Enrico Zenerino, and Marcello Chiaberge. "Multipurpose UAV for search and rescue operations in mountain avalanche events". In: *Geomatics, Natural Hazards and Risk* 8.1 (2017), pp. 18–33.

[SK17] Vahram Stepanyan and Kalmanje S Krishnakumar. "Estimation, navigation and control of multi-rotor drones in an urban wind field". In: *AIAA information systems-AIAA infotech@ aerospace*. 2017, p. 0670.

[SKI05] Shingo Shimoda, Yoji Kuroda, and Karl Iagnemma. "Potential field navigation of high speed unmanned ground vehicles on uneven terrain". In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE. 2005, pp. 2828–2833.

[SL17] Dynamical Systems and Control Laboratory. *EduMIP ROS*. 2017. URL: https://dscl.lcsr.jhu.edu/home/courses/edumip_ros/.

[SL21] Seyed Omid Sajedi and Xiao Liang. "Uncertainty-assisted deep vision structural health monitoring". In: *Computer-Aided Civil and Infrastructure Engineering* 36.2 (2021), pp. 126–142.

[SN18] Johannes Schauer and Andreas Nuchter. "The peopleremover—removing dynamic objects from 3-d point cloud data by traversing a voxel occupancy grid". In: *IEEE robotics and automation letters* 3.3 (2018), pp. 1679–1686.

[Son+08] Arnold Song, Xiaodong Tian, Emily Israeli, Ricardo Galvao, Kristin Bishop, Sharon Swartz, and Kenneth Breuer. "Aeromechanics of membrane wings with implications for animal flight". In: *AIAA journal* 46.8 (2008), pp. 2096–2106.

[SPP18] Won Dong Shin, Jaejun Park, and Hae-Won Park. "Bio-inspired design of a gliding-walking multi-modal robot". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 8158–8164.

[SPP19] Won Dong Shin, Jaejun Park, and Hae-Won Park. "Development and experiments of a bio-inspired robot with multi-mode in aerial and terrestrial locomotion". In: *Bioinspiration & biomimetics* 14.5 (2019), p. 056009.

[SR13] KISUNG Seo and ALDEBARAN Robotics. "Using nao: introduction to interactive humanoid robots". In: *AldeBaran Robotics* (2013).

[Ste22] Robert F Stengel. *Flight dynamics*. Princeton university press, 2022.

[Sto04] RH Stone. "The T-wing tail-sitter unmanned air vehicle: from design concept to research flight vehicle". In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 218.6 (2004), pp. 417–433.

[Str+21a] James Strawson, Pengcheng Cao, Thomas Bewley, and Falko Kuester. "Rotor orientation optimization for direct 6 degree of freedom control of multirotors". In: *2021 IEEE Aerospace Conference (50100)*. IEEE. 2021, pp. 1–12.

[Str+21b] James Strawson, Pengcheng Cao, Danny Tran, Thomas Bewley, and Falko Kuester. "Monocoque Multirotor Airframe Design with Rotor Orientations Optimized for Direct 6-DoF UAV Flight Control". In: *AIAA AVIATION 2021 FORUM*. 2021, p. 2431.

[Str18a] J. R. Strawson. "Feedback Control Driven Mechanical Design Optimization." In: *UC San Diego Electronic Theses and Dissertations* (2018). URL: https://escholarship.org/uc/item/8w82m5n8.

[Str18b] James Robert Strawson. *Feedback Control Driven Mechanical Design Optimization*. University of California, San Diego, 2018.

[Sun+22] Jun Sun, Bo Peng, Cynthia Changxin Wang, Kunxiang Chen, Botao Zhong, and Jianjie Wu. "Building displacement measurement and analysis based on UAV images". In: *Automation in Construction* 140 (2022), p. 104367.

[Taz+18] M Lamine Tazir, Tawsif Gokhool, Paul Checchin, Laurent Malaterre, and Laurent Trassoudaine. "CICP: Cluster Iterative Closest Point for sparse–dense point cloud registration". In: *Robotics and Autonomous Systems* 108 (2018), pp. 66–86.

[TDA98] Richard W Thorington Jr, Karolyn Darrow, and C Gregory Anderson. "Wing tip anatomy and aerodynamics in flying squirrels". In: *Journal of Mammalogy* 79.1 (1998), pp. 245–250.

[Teu+10] Celine Teuliere, Laurent Eck, Eric Marchand, and Nicolas Guenard. "3D model-based tracking for UAV position control". In: *2010 IEEE/RSJ Interna-*

tional Conference on Intelligent Robots and Systems. IEEE. 2010, pp. 1084–1089.

[Thu+02] Benot Thuilot, Christophe Cariou, Philippe Martinet, and Michel Berducat. "Automatic guidance of a farm tractor relying on a single CP-DGPS". In: *Autonomous robots* 13.1 (2002), pp. 53–71.

[Tia+20] Yulun Tian, Katherine Liu, Kyel Ok, Loc Tran, Danette Allen, Nicholas Roy, and Jonathan P How. "Search and rescue under the forest canopy using multiple UAVs". In: *The International Journal of Robotics Research* 39.10-11 (2020), pp. 1201–1221.

[Tom+12] Teodor Tomic, Korbinian Schmid, Philipp Lutz, Andreas Domel, Michael Kassecker, Elmar Mair, Iris Lynne Grixa, Felix Ruess, Michael Suppa, and Darius Burschka. "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue". In: *IEEE robotics & automation magazine* 19.3 (2012), pp. 46–56.

[Tru+13] Agustin Trujillo-Pino, Karl Krissian, Miguel Aleman-Flores, and Daniel Santana-Cedres. "Accurate subpixel edge location based on partial area effect". In: *Image and vision computing* 31.1 (2013), pp. 72–90.

[TT01] Adrian LR Thomas and Graham K Taylor. "Animal flight dynamics I. Stability in gliding flight". In: *Journal of theoretical biology* 212.3 (2001), pp. 399–424.

[Tya+17] Maxim Tyan, Nhu Van Nguyen, Sangho Kim, and Jae-Woo Lee. "Comprehensive preliminary sizing/resizing method for a fixed wing–VTOL electric UAV". In: *Aerospace Science and Technology* 71 (2017), pp. 30–41.

[ULH16] Steffen Urban, Jens Leitloff, and Stefan Hinz. "Mlpnp - a real-time maximum likelihood solution to the perspective-n-point problem". In: *ISPRS Annals*

of Photogrammetry, Remote Sensing & Spatial Information Sciences. Vol. 3. 2016, pp. 131–138.

[Uni20] Middle East Technical University. 2020. URL: https://ocw.metu.edu.tr/pluginfile.php/6885/mod_resource/content/1/ch7/7-1.htm.

[UYB21] Hakan Ucgun, Ugur Yuzgec, and Cuneyt Bayilmis. "A review on applications of rotary-wing unmanned aerial vehicle charging stations". In: *International Journal of Advanced Robotic Systems* 18.3 (2021), p. 17298814211015863.

[Van+16] Fernando Vanegas, Duncan Campbell, Markus Eich, and Felipe Gonzalez. "UAV based target finding and tracking in GPS-denied and cluttered environments". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 2307–2313.

[VBS20] Daniel KD Villa, Alexandre S Brandao, and Mario Sarcinelli-Filho. "A survey on load transportation using multirotor UAVs". In: *Journal of Intelligent & Robotic Systems* 98.2 (2020), pp. 267–296.

[Vos+07] Roelof Vos, Ron Barrett, Roeland de Breuker, and Paolo Tiso. "Post-buckled precompressed elements: a new class of control actuators for morphing wing UAVs". In: *Smart materials and structures* 16.3 (2007), p. 919.

[Wal23] Walmart. *Power Your Fun Cyclone Mini Remote Control Stunt Race Speed Car*. 2023. URL: https://www.walmart.com/ip/Power-Your-Fun-Cyclone-Mini-Remote-Control-Stunt-Race-Speed-Car/695920888?athbdg=L1800.

[Wan+20] X Wang, CE Wittich, TC Hutchinson, Yehuda Bock, Dara Goldberg, Eric Lo, and F Kuester. "Methodology and validation of UAV-based video analysis approach for tracking earthquake-induced building displacements". In: *Journal of Computing in Civil Engineering* 34.6 (2020), p. 04020045.

[Wan+22] Xiang Wang, Eric Lo, Luca De Vivo, Tara C Hutchinson, and Falko Kuester. "Monitoring the earthquake response of full-scale structures using UAV vision-based techniques". In: *Structural control and health monitoring* 29.1 (2022), e2862.

[Wan+23] Ruhua Wang, Yanda Shao, Qilin Li, Ling Li, Jun Li, and Hong Hao. "A novel transformer-based semantic segmentation framework for structural condition assessment". In: *Structural Health Monitoring* (2023), p. 14759217231182303.

[Wat+20] Robert J Watson, S Gareth Pierce, Mina Kamel, Dayi Zhang, Charles N MacLeod, Gordon Dobie, Gary Bolton, Tariq Dawood, and Juan Nieto. "Deployment of Contact-Based Ultrasonic Thickness Measurements Using Over-Actuated UAVs". In: *European Workshop on Structural Health Monitoring*. Springer. 2020, pp. 683–694.

[Wen+21] Yufeng Weng, Jiazeng Shan, Zheng Lu, Xilin Lu, and Billie F Spencer Jr. "Homography-based structural displacement measurement for large structures using unmanned aerial vehicles". In: *Computer-Aided Civil and Infrastructure Engineering* 36.9 (2021), pp. 1114–1128.

[Wit+14] CE Wittich, TC Hutchinson, E Lo, D Meyer, and F Kuester. "The South Napa Earthquake of August 24, 2014: drone-based aerial and ground-based LiDAR imaging survey". In: *Structural Systems Research Project Report SSRP-2014* 9 (2014).

[WT10] Sonia Waharte and Niki Trigoni. "Supporting search and rescue operations with UAVs". In: *2010 International Conference on Emerging Security Technologies*. IEEE. 2010, pp. 142–147.

[Wyp+14] Tom Wypych, James Strawson, Vid Petrovic, Radley Angelo, Aliya Hoff, Matt Howland, Maurizio Seracini, Thomas Levy, and Falko Kuester. "Airborne

imaging for cultural heritage". In: *2014 IEEE Aerospace Conference*. IEEE. 2014, pp. 1–9.

[Xia+23] Xin Xia, Zonglin Meng, Xu Han, Hanzhao Li, Takahiro Tsukiji, Runsheng Xu, Zhaoliang Zheng, and Jiaqi Ma. "An automated driving systems data acquisition and analytics platform". In: *Transportation research part C: emerging technologies* 151 (2023), p. 104120.

[Xu+22] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. "Fast-lio2: Fast direct lidar-inertial odometry". In: *IEEE Transactions on Robotics* 38.4 (2022), pp. 2053–2073.

[XZ21] Wei Xu and Fu Zhang. "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3317–3324.

[Yan+18] Sheng Yang, Xiaoling Zhu, Xing Nian, Lu Feng, Xiaozhi Qu, and Teng Ma. "A robust pose graph approach for city scale LiDAR mapping". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1175–1182.

[Yan+19] Jiayao Yan, Ning Li, Tongji Luo, Michael T Tolley, and John T Hwang. "Optimal control and design of an underactuated ball-pitching robotic arm using large-scale multidisciplinary optimization". In: *AIAA Aviation 2019 Forum*. 2019, p. 3450.

[Yan+23] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. "Track anything: Segment anything meets videos". In: *arXiv preprint arXiv:2304.11968* (2023).

[YH09] Song Yanguo and Wang Huanjin. “Design of flight control system for a small unmanned tilt rotor aircraft”. In: *Chinese Journal of Aeronautics* 22.3 (2009), pp. 250–256.

[Zha+19] Fei Zhao, Wei Wang, Jingtao Zhang, Justyna Wyrwa, and Feng Sun. “Aerodynamic characteristics and pitching adjusting mechanism of the flying squirrel with deployed patagium”. In: *IEEE Access* 7 (2019), pp. 185554–185564.

[Zhe+20] Peter Zheng, Xinkai Tan, Basaran Bahadir Kocer, Erdeng Yang, and Mirko Kovac. “TiltDrone: A fully-actuated tilting quadrotor platform”. In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 6845–6852.

[Zho+21] Boyu Zhou, Yichen Zhang, Xinyi Chen, and Shaojie Shen. “Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 779–786.

[ZS14] Ji Zhang and Sanjiv Singh. “LOAM: Lidar odometry and mapping in real-time.” In: *Robotics: Science and systems*. Vol. 2. 9. Berkeley, CA. 2014, pp. 1–9.

[Zuo10] Zongyu Zuo. “Trajectory tracking control design with command-filtered compensation for a quadrotor”. In: *IET control theory & applications* 4.11 (2010), pp. 2343–2355.