A Novel Hybrid Aerial/Ground Mobile Manipulator

A NOVEL HYBRID AERIAL/GROUND MOBILE MANIPULATOR

BY

DAVID FINDLAY, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2017) McMaster University
(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	A Novel Hybrid Aerial/Ground Mobile Manipulator
AUTHOR:	David Findlay B.Eng., (Mechatronics Engineering) McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Shahin Sirouspour
NUMBER OF PAGES:	xi, 80

This work is dedicated to my parents who have been my compass in difficult times.

To my grandfather, Dr. Raymond Findlay, whose respect in the research community is an unrivaled inspiration.

Abstract

This research is concerned with the mechanical design of a mobile manipulating unmanned ground vehicle (MM-UGV) coupled with an existing commercial unmanned aerial vehicle (UAV) to create a novel hybrid aerial/ground mobile manipulator. A hybrid robotic system capable of manipulating in both aerial and ground environments is a new research direction for field robotics. The hybrid system has the potential of stimulating new research and engineering challenges as well as providing multipurpose robotic systems for industrial applications. A bilevel optimization-based strategy is presented for making important design choices, such as the selection of gear ratios, electric DC motors, manipulator link lengths, and UGV base length. The objective is to minimize the overall mass of the MM-UGV such that hybridization given certain constraints is possible. Constraints related to workspace, dynamic tip-over stability, actuator torque/force limits, and battery properties are incorporated into the formulation in order to ensure maneuverability of the system. Design specifications such as the expected range of end-effector forces, operating surface grade, and various position, velocity and acceleration variables are input to the optimization problem. The resulting problem is formulated as a robust mixed integer bilevel nonlinear program (MINBP), in which some of the constraints are derived from maximization/minimization over the operational variables to ensure constraint satisfaction in all possible ground/air operation scenarios. Optimizing over the operational variable space is a novel technique compared to current research that focuses on optimizing robotic mechanical components for a single trajectory. The Branch-and-Sandwich Bilevel optimization algorithm (BASBL) was used to find a solution to the optimization problem. A parallelized version of the algorithm was implemented and deployed on an IBM BladeCenter computer cluster. Speedup and parallel efficiency of the algorithm show significant improvement over the serial approach. Improving the run time of the optimization is critical for iterative engineering design with different input parameters. A prototype using an optimal set of design parameters is constructed and results of a preliminary flight experiment are reported. The proposed design optimization methodology is rather general and could be applied to other robotic systems.

Acknowledgements

I would like to extend my extreme gratitude and respect to Professor Shahin Sirouspour for his support, knowledge, and especially his patience throughout this process. I wish to express my sincerest appreciation to my friend and colleague, Mohammad Jafarinasab, who had a very strong influence in the development of the theory and experiments.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Unmanned ground vehicles (UGV) have been extensively used for decades to accomplish tasks that are dangerous and even impossible for human beings. The applications of mobile robots are abundant, growing in areas such as search and rescue missions, police operations, and hazardous site exploration. More recently, mobile manipulating unmanned ground vehicles (MMUGV) have emerged in the field of robotics. Clearly, MMUGVs are more useful because they have the ability of interacting with their surrounding environment. However, there are many environments that drastically limit the capabilities of both UGVs and MMUGVs. For example, ascending or descending stairs is an ongoing challenge for ground vehicles. Moreover, tipping-over due to uneven or rough terrain can render the ground vehicle damaged or useless. Tip-over avoidance algorithms have been implemented for control synthesis, however it has yet to be considered in the mechanical design process. The invention of the Unmanned Aerial Vehicle (UAV) solves both of the aforementioned problems, however they come with their own challenges. UAVs are capable of maneuvering through complex terrains much easier than UGVs. Again, robotic manipulators have been added to the UAV to create Mobile Manipulating Unmanned Aerial Vehicles (MMUAV), therefore allowing interaction with the environment. The caveat of using aerial robotics is the drastic limitation on payload capacity. The payload of UAVs is defined as the mass that they can carry in addition to the system mass. This substantially limits the maximum allowable payload that an MMUAV can manipulate. Although there has been great progress in field robotics, there is no existing hybrid system that is capable of manipulation in air and on the ground as a stand-alone Hybrid Mobile Manipulator (HMM). The goal is to leverage the benefits of both MMUGV and MMUAV into one hybrid HMM. Robotic systems capable of multi-environment object manipulation have the potential to significantly enhance our operational capabilities in remote, inaccessible, and hazardous environments with minimal risk to humans. A comprehensive strategy that could be employed to systematically design such a complex electro-mechanical systems to satisfy user and operational requirements/constraints is not available in the literature.

An example application for the HMM system would be hazardous site exploration of a multiple story building. The MMUGV mode can be used for investigation on the floor where it would be difficult for the MMUAV to stabilize due to ground effect caused by the propellers. The MMUAV mode will outperform the MMUGV mode in tasks such as door opening, stair climbing, and manipulation of objects on the wall/ceiling that would be unreachable if using the MMUGV mode.

1.2 Problem Statement

A hybrid system capable of operating in air and ground environments is unavailable. Furthermore, optimal mechanical design strategies for robotics currently only consider a single trajectory. This forces suboptimal behavior of the robot for general trajectories which could be unknown prior to the robot design.

1.3 Proposed Solution

A general robotic design strategy based on optimization is formulated in this research. The objective is to test the design strategy by employing it to design a novel HMM system. Although the design strategy is used to design the HMM, it should be noted that it is very general and could be applied to design any robotic system. This work is focused on the optimal mechanical design of an MMUGV that is capable of coupling/decoupling with an existing commercial UAV.

1.4 Contributions

The contributions of this research are enumerated as follows:

1. Introducing the concept of a novel hybrid aerial/ground mobile manipulator robotic system.

2. Proposing a general bilevel optimization strategy for choosing critical design parameters of the robot under a wide range of dynamic operating scenarios; the proposed methodology is applicable to other robotic systems.

3. Implementation of a prototype system with experimental results showing that the optimal design satisfies the prescribed design constraints and optimization inputs.

1.5 Publications

The following publications were the product of this research:

1. A conference paper accepted in the International Conference on Intelligent Robots and Systems (IROS) [8] which focused on the theoretical implications with numerical simulations.

2. A journal paper to be submitted to the IEEE/ASME Transactions on Mechatronics [9] which furthered the theoretical work and provided a full implementation of the hybrid system with experimental results.

1.6 Thesis Outline

The remainder of the thesis is organized as follows. Chapter 2 covers the literature review of UAV, MMUAV, UGV, MMUGV, and Optimal system design. Chapter 3 outlines the system and its different modes of functionality. Chapter 4 is focused on system modeling, where the full non-linear dynamic equations of the UAV are derived. The MMUAV equations are then described using Newton-Euler Recursive formulation. Lastly, the UGV stability is discussed in detail. Chapter 5 entails formulating the design problem as a bilevel optimization problem. The optimization results are subsequently presented and discussed in detail. The stage is set for Chapter

6, wherein the system development is discussed, namely the mechanical design, state estimation, and model identification. The hardware and software development is discussed in detail in Chapter 7. Experimental results of the system are shown in Chapter 8. Conclusive remarks and future work are outlined in Chapter 9.

Chapter 2

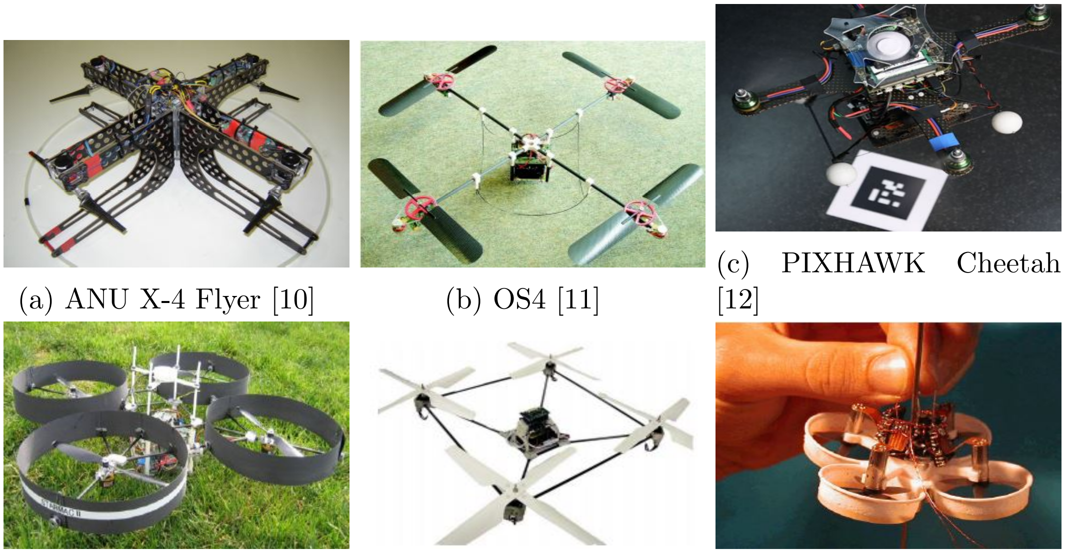
Literature Review

In this section state-of-the-art technology for each component of the HMM system is isolated and studied in detail.

2.1 Unmanned Aerial Vehicle

2.1.1 UAV System Design

The focus of this research is on multirotors, a unique type of UAV which is capable of vertical take off and landing (VTOL). Typically multirotors have an even number of brushless DC motors combined with fixed pitch propeller blades. Multirotors can be further categorized by the number of actuators: quadrotors (4), hexarotors (6), octorotors (8), etc. The advantage of this type of UAV is the maneuverability resulting from its dynamics. In general, a mechanical system that is unable to generate an instantaneous acceleration in any arbitrary direction is said to be underactuated. In other words, a system is underactuated if the number of control inputs is less than the number of degrees of freedom (DOF). In the case of a UAV, there are four control inputs: roll, pitch, yaw, and thrust, whereas there are six DOF. Fewer number of actuators reduces costs and weight, however synthesis of control schemes for underactuated robotics is a challenging problem resulting from complex dynamics and nonholonomic behavior. Universities have developed multirotors as test platforms for developing and deploying algorithms in many different research domains. There have been numerous successes in academia as shown in Figure 2.1.



(a) ANU X-4 Flyer [10]

(b) OS4 [11]

(c) PIXHAWK Cheetah [12]

(d) STARMAC [13]

(e) CEA X4-Flyer [14]

(f) Mesicopter [15]

Figure 2.1: Research Specific UAVs

The Australian National University produced the X-4 Flyer [10] as depicted in Figure 2.1a. The X-4 Flyer weighs 4.34 kg and is capable of carrying a payload of approximately 0.5 to 1.0 kg. The expected hover time is approximately 11 minutes assuming zero payload. The mechanical design was optimized for controlling the flight dynamics, including rotor flapping effects. Another exemplary system is the OS4 shown in Figure 2.1b which was developed at the École Polytechnique Fédérale de Lausanne (EPFL) [11]. The OS4 has a total mass of 0.65 kg, payload of 0.39 kg, and a maximum flight time of approximately 20 minutes. Design optimization, dynamic modeling and control were validated using a custom test-bed. The system has been used successfully to demonstrate numerous control designs including PID, LQR, and model-based controllers [16, 17, 18].

Stanford University developed the STARMAC quadrotor and test environment as shown in Figure 2.1d [13]. This quadrotor was designed for multiple quadrotor flight formation and collision avoidance. Reinforcement Learning Control have been implemented on this system in order to accommodate for nonlinear disturbances [19]. The X4-Flyer was developed within the French Atomic Energy Commission (CEA) and is depicted in Figure 2.1e. A prototype was developed with a total mass of 0.5 kg and flight time of 10 minutes. This quadrotor has double the number of blades on each motor, increasing the allowable payload. An adaptive flight controller was developed using a dynamic nonlinear model that is valid for quasi-stationary flight conditions [14]. The PIXHAWK Cheetah was developed at ETH Zurich and is shown in Figure 2.1c[12]. The system was designed for the development of on-board computer vision algorithms including simultaneous localization and mapping (SLAM), collision avoidance, and pattern recognition. Each motor is capable of producing 0.425 kg of thrust, allowing the 1.2 kg system to lift an additional payload of 0.4 kg. The total power consumption of the system at hover is 150-170W which allows 14-16 minutes of flight time. An example of a centimeter-scale quadrotor is the Mesicopter developed at Stanford University [15]. The total mass of the system is approximately 1.0 gram and customized wafer-cut molds were used to manufacture the rotors. This system was designed for inspection of very small areas, inaccessible by the larger aforementioned quadrotor systems.

Multirotors have also been an exciting topic in industry, for example Amazon has stared the delivery by drone experiments. The France-based Parrot ARDrone shown in Figure 2.2 is a popular commercial quadrotor that is often used in research due to its simplicity [1]. The system sensor suite includes two cameras, a sonar sensor, and an Inertial Measurement Unit (IMU). Two processing boards, one for I/O data processing (PIC) and another for algorithm execution (ARM based). The on-board battery supports a maximum of 1000mAh and discharge rate of 15C which allows the system to fly for 12-15 minutes. The 0.25 kg payload makes this system unfavorable if custom sensors or features are required.

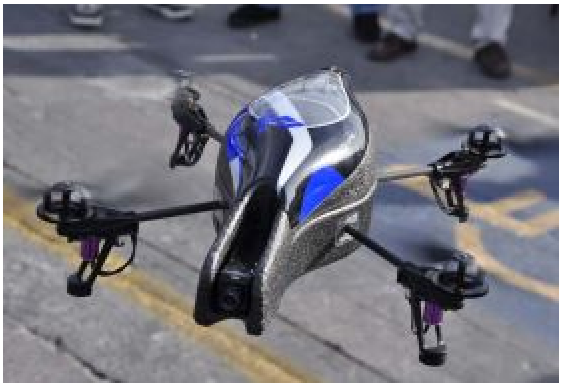
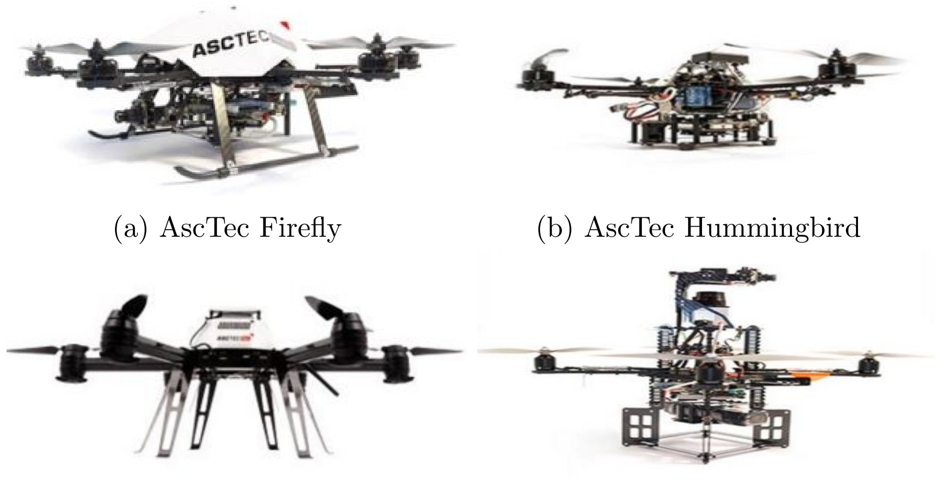


Figure 2.2: Parrot ARDrone [1]

Germany-based Ascending Technologies developed multiple notable quadrotors that are used in industry and as research platforms, see Figure 2.3 [2]. Among these quadrotors, the Pelican (see Figure 2.3d) and Hummingbird (see Figure 2.3b) have been more popular in the research community. The Hummingbird was designed to be light-weight and capable of highly dynamic motions. This small quadrotor has a total mass of 0.71kg and is capable of lifting a payload of only 0.2kg. The system can remain in flight for 20 minutes including the payload. The on-board computer encompasses an



(a) AscTec Firefly

(b) AscTec Hummingbird

(c) AscTec Neo

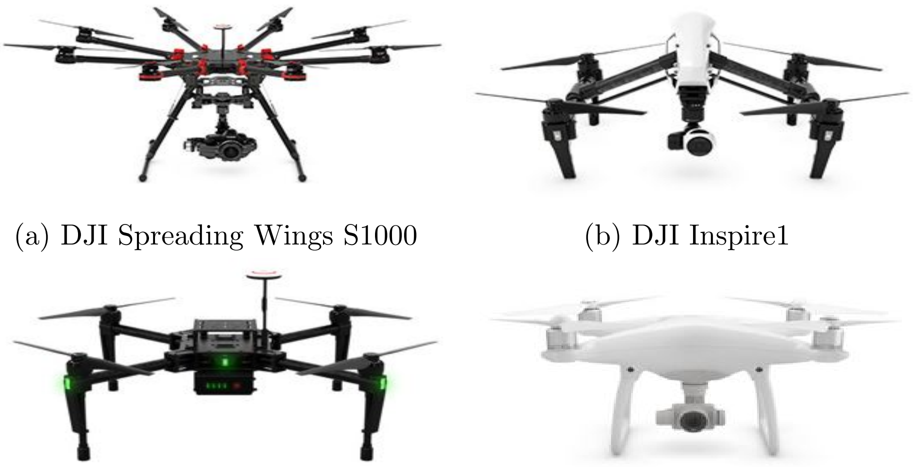
(d) AscTec Pelican

Figure 2.3: Ascending Technology UAVs [2]

Intel Atom processor Z530 and a 1kHz update rate IMU. The Hummingbird has been recently used for autonomous flight in unknown indoor environments using SLAM [20], optimization of air-ground robot communication networks [21], backstepping control [22], and passivity-based backstepping control [23]. The Pelican quadrotor is a larger vehicle, allowing for more customization of the on-board sensors and computer. The total mass of this quadrotor is 1.65 kg and can hoist a maximum payload of 0.65kg. The total flight time including the maximum payload is approximately 16 minutes. The on-board computer is an Intel Core i7 processor and a 1kHz update rate IMU. The Pelican has been successfully used in research for SLAM based autonomous navigation [24], passive fault-tolerant control [25], and aerial manipulation [26].

China-based Da-Jiang Innovations (DJI) is the global leader in aerial photography and cinematography. The DJI company has expanded significantly since the starting of this project. Some notable UAV products are shown in Figure 2.4 [3].

The Spreading Wings S1000 shown in Figure 2.4a is an octocopter that weighs



(a) DJI Spreading Wings S1000

(b) DJI Inspire1

(c) DJI Matrice100

(d) DJI Phantom4

Figure 2.4: DJI UAVs [3]

4.2kg with a maximum payload capacity of 6.8kg. The hover time with a 5.3kg payload is roughly 15 minutes. These properties make this model a highly favorable system in terms of adding on-board features. The Inspire1 quadrotor shown in Figure 2.4b was designed for GPS denied environments, relying on an on-board vision-based positioning system. The system has a total mass of 2.7kg, maximum payload of 0.68kg, and can hover for approximately 18 minutes. This system has been shown to monitor sea turtles in near-shore waters [27]. The Matrice100 is considered DJI's developer drone, see Figure 2.4c. This 2.431kg quadrotor can lift a 1.0kg payload for 16 minutes when equipped with a 5700mAh capacity battery with 22.8 VDC and 180W maximum power. The center frame allows flexibility when mounting additional components for adding functionality to the system. The Phantom4 quadrotor displayed in Figure 2.4d is the 10th revision of the Phantom series from DJI and was released in 2016. This quadrotor is intended for commercial cinematography and is capable of capturing 4K resolution video and 14 Megapixel photos. The total mass is 1.38kg and can hover for 28 minutes.

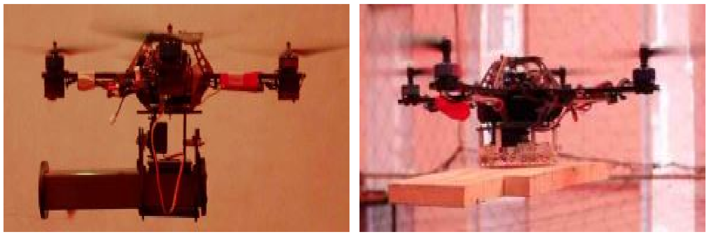
As a rapidly advancing field in robotics, many open-source projects have started in order to allow fast prototyping and development of new features. Some established projects are Pixhawk, Arducopter, Openpilot, Paparazzi, and Mikrokopter [28]. These projects offer on-board flight controllers, integrated sensor processing, and state-estimation algorithms.

The main research areas are modeling, control, perception, navigation, and autonomy. Moreover, multi-agent cooperation has attracted a great deal of attention. Advances in multirotor research has encouraged groups to start adding complexity and functionality to the system.

Teal Group Corporation is a U.S. based firm that gathers and publishes information about the aerospace and defense industry. Their 2015 market forecast on the UAV industry shows an annual growth from \$ 4 billion to \$ 14 billion, totaling \$ 93 billion in the next 10 years. Therefore, the impact of revolutionizing UAV technology is of extreme interest [29].

2.1.2 MMUAV System Design

Equipping a multirotor with a robotic manipulator allows interaction with the environment. Without a robotic manipulator, multirotors are limited to surveillance and data collection tasks. The simplest robotic manipulator that can be added to the system is a one degree-of-freedom (DOF) manipulator such as a gripper. Low-complexity grippers can allow a quadrotor to transport small loads and perch on tree branches or structure beams. There are two methods of grasping of interest when designing aerial grippers: impactive and ingressive. Impactive grippers use solid jaws to grasp objects whereas ingressive grippers rely on penetrating the surface of the object [30]. Of course, once the object has been grasped, the inertial parameters of the rigid body change. Consequently adaptive controllers that can improve the performance of the quadrotor flight have been developed. Impactive and ingressive grippers were installed on a quadrotor and multiple different payload inertia and mass were experimented (see Figure 2.5) [31].

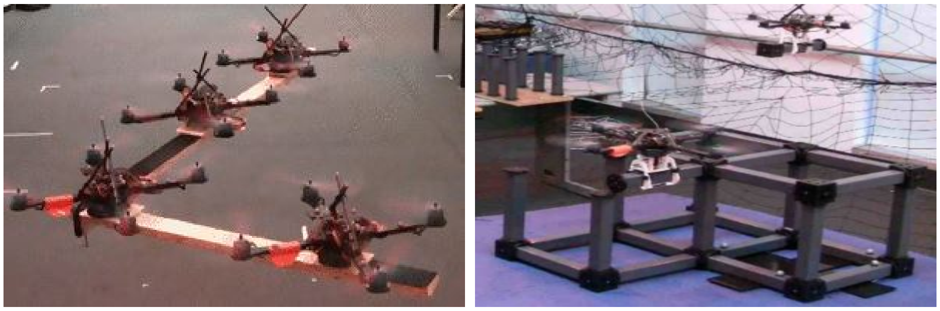


(a) Impactive gripper [30] (b) Ingressive gripper [31]

Figure 2.5: Quadrotor grippers

The design strategy for combining an open-source modular under-actuated robot hand with the aforementioned Parrot AR.Drone was discussed in [32]. Another group added a gripper to a UAV and designed a controller that was robust to dynamic load disturbances [33]. Since quadrotors are typically small (less than 4 kg), they have limited payload capacity making it impossible to lift heavy objects. A research team solved this problem by assigning the task to a team of quadrotors, each equipped with a gripper (see Figure 2.6a) [34]. Another example of cooperative quadrotor grippers was demonstrated in [35] and [36]. A team of quadrotors, each equipped with a magnetized gripper, assembled a truss-like structure from components equipped with magnets (see Figure 2.6b).

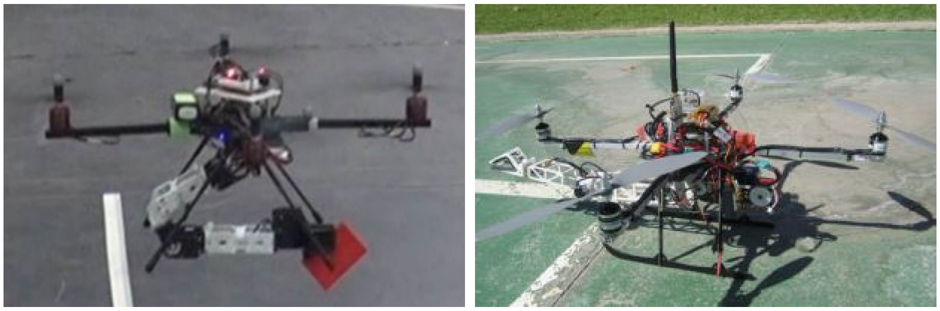
Although low-complexity grippers provide much more utility to the system, more functionality is needed for more complex tasks, therefore more DOF are required.



(a) Quadrotor team lifting object [34] (b) Quadrotor team building cube [35, 36]

Figure 2.6: Cooperative Quadrotor grippers

A quadrotor with a serial two-DOF manipulator is shown in Figure 2.7a [37]. An autonomous flight experiment was performed using adaptive sliding mode control in order to grasp and deliver an object. Another research group developed a serial three-DOF manipulator shown in Figure 2.7b [4]. This system was tested with PID control and variable parameter integral backstepping control. Outdoor experiements showed the validity of the proposed design and control methods.



(a) Quadrotor with a Two-DOF manipulator [37] (b) Quadrotor with a Three-DOF manipulator [4]

Figure 2.7: Quadrotors with a multi-link serial manipulator

Again the applications of this system become limited resulting from the simplicity of a serial robotic manipulator. This problem was solved by integrating a seven-DOF manipulator with an octorotor [4] shown in Figure 2.8. The research article focuses on the control design and implementation aspects. The octorotor was stabilized using a backstepping controller and the manipulator was controlled using an admittance control design. An experiment was performed wherein the robotic manipulator was equipped with a camera in order to track a moving object.

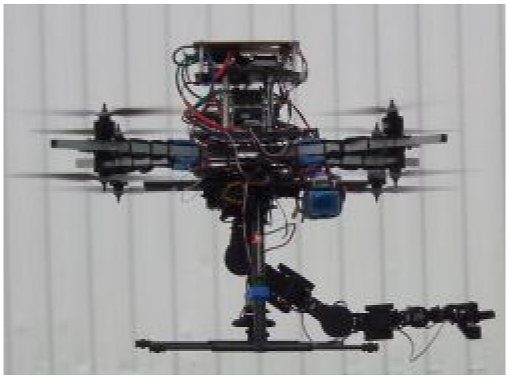


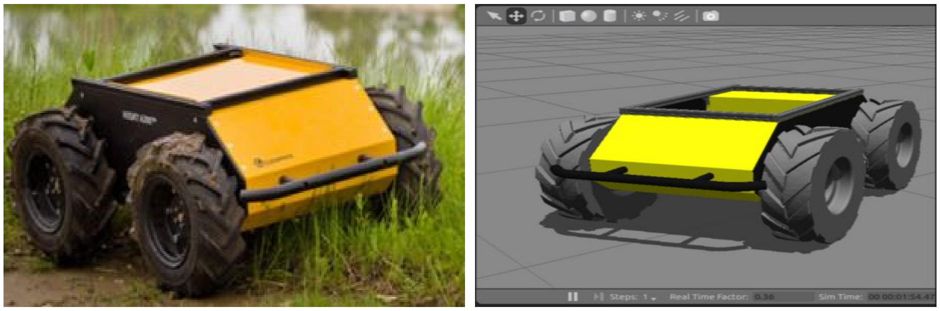
Figure 2.8: Octorotor with seven-DOF manipulator [4]

Some other groups are focused on multirotors equipped with multiple robot manipulators with multiple DOF. This allows one arm to stabilize the multirotor while the other arm manipulates a desired object [38], [39]. This principle was used in order to allow an aerial manipulator to perch and open a door [40].

2.2 Unmanned Ground Vehicle

Unmanned Ground Vehicles have attracted a tremendous amount of attention in recent years which has resulted in numerous autonomous self-driving cars. Technology leaders such as Google, Apple, IBM, and Tesla are researching and actively developing UGVs that are capable of obeying traffic laws. For the purpose of this research, the UGVs of interest are on a much smaller scale. The Husky UGV designed at Clearpath

Robotics is a four-wheeled small vehicle used for outdoor navigation (see Figure 2.9a) [5]. Clearpath makes it easy to implement autonomous control algorithms by providing the Gazebo simulation model (see Figure 2.9b). Gazebo is a robotics simulation environment that allows rapid algorithm testing, robotic resign, and regression testing under realistic scenarios [41]. The Husky UGV has a total mass 50kg and can carry a 75kg payload. Although there is a lot of open-source support, it is not possible to integrate this UGV with a small multirotor due to the payload limitation.



(a) Husky UGV outdoor (b) Husky UGV simulation

Figure 2.9: Husky UGV from Clearpath Robotics [5]

Tracked mobile robots provide the advantage of addition stability. NASA developed the urban robot nicknamed Urbie shown in Figure 2.10. The front legs are capable of rotating 360 degrees in order to help the robot climb stairs and flip-over. The stereo-vision camera allows the robot to navigate city terrain autonomously.

The Matilda tracked mobile robot was employed by the U.S. Army Marine Corps in order to clear tunnels and for pipe exploration [42]. This UGV has a total mass of 18kg to allow soldiers to carry the robot between zones. QuinetiQ North America developed the Dragon Runner 10 that weighs only 4.5kg. Thus UGV can sustain high impact maneuvers such as throwing it through a window and is small enough to fit inside a backpack [43]. Other well-designed tracked mobile robots include iRobot's

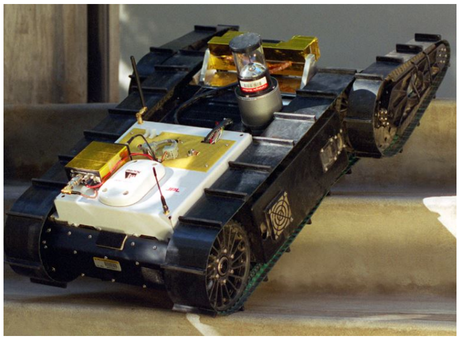


Figure 2.10: NASA Urbie UGV

PackBot [44], AZIMUT which combines tracks and legs [45], and HELIOS [46] which has allowed wheelchairs to ascend/descend stairs. Although the scope of the UGV review was limited, it is evident that the utility of such systems is limited without a robotic manipulator.

2.2.1 MMUGV System Design

MMUGVs have proven to be extremely useful in nuclear disaster cleanup. The Defense Advanced Research Projects Agency (DARPA) holds robotics obstacle course challenges that are similar to the environment faced by disasters such as the Fukushima Daiichi nuclear reactor disaster in 2011. The 710 Kobra MMUGV developed by iRobot was used to dispose of hazardous material by controlling a vacuum cleaner connected to its robotic manipulator [47].

MMUGVs have also shown great potential in military applications. Foster-Miller's TALON MMUGV was used in lieu of American soldiers and played a key role in their conflicts with Afghanistan and Iraq [48]. Moreover, the Kuka Youbot MMUGV supports more than one manipulator and has been a popular influence in industrial applications. All of the aformentioned MMUGVs are prone to tipping over and becoming useless. A team from the University of Toronto addressed this issue by combining locomotion and manipulation [49]. Although these designs perform well in rough and uneven terrain, they are not optimized for minimal mass and therefore are inappropriate for UAV hybridization. The MMUGV should have minimal mass in order to allow a UAV to hoist it.

Tip-over stability is a mojor concern for MMUGV systems. Algorithms have been developed to measure tip-over stability, predict tip-over conditions, and take actions to avoid tipping over. Three main algorithms have been developed: the Zero-Moment Point (ZMP), Force-Angle (FA) stability measure, and Moment-Height stability (MHS). The ZMP is a coordinate on the ground where all external forces and torques acting on the robot are replaced with one force vector [50]. FA computes the stability measurement based on the angle of external force on the center of mass of the robot [51]. The MHS algorithm takes into consideration the inertia of the robot about each axis of a convex polygon which is described by the contact points of the robot and the ground [52]. A study was done in [53] to compare each stability criteria and determine their effectiveness with some experimental insights.

2.2.2 Optimal Design

Structural optimization is a category of optimization problems where the objective function and constraints are evaluated based on a structural analysis. The optimization design variable describes the geometry of the system, typically system dimensions. The objective function is often the system weight and the constraints impose limits on the system dimensions. A comprehensive review of structural optimization strategies and algorithms focusing on optimal mechanical development can be found in [54]. Structural optimization can be further divided into four categories: sizing optimization, material optimization, shape optimization, and topology optimization [55]. Sizing optimization is concerned with determining the dimensions of certain components in the system. Material optimization is used to determine the material composition of the system that optimizes the objective function, for example the stiffness of a beam. Shape optimization can be used to modify the original system shape, for example holes can be drilled in the structure to reach optimal objective function. The characteristics of these holes are decided by the optimization algorithm. Topology optimization is concerned with modifying the relationships between elements in the system. For example, this can be used to determine the optimal number of robotic manipulator linkages given an objective function and constraints.

Research groups have applied structural optimization to industrial robotic manipulators. Drive-train optimization of a robotic manipulator was explored in [56], using commercially available components for motors and gearbox elements. Similar research was done in [57] with different different problem formulation. Kinematic Isotropy and optimal kinematic design of robotic manipulators has been done to determine robotic manipulator link lengths [58], [59]. A concurrent mechanical design method is described in [60], which combines control tuning and mechanical design as an iterative approach for a legged robotic system. All of the aforementioned optimization formulations consider a very specific trajectory for the robot. This is highly problematic if the robot path is not known a priori. These approaches limit the robotic design to a single trajectory and suffer from sub-optimality if other trajectories are pursued.

Chapter 3

System Overview

3.1 Modes of Operation

Figure 3.1 depicts the operation of the hybrid aerial/ground robotic manipulator and its various modes of operation.

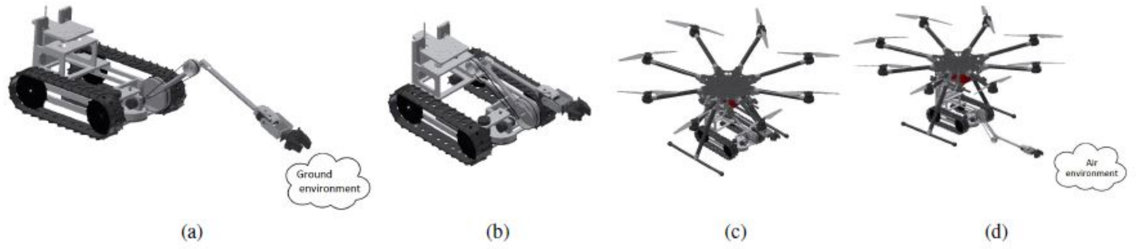


Figure 3.1: Conceptual operation of the hybrid vehicle in different modes.

Consider a scenario where a manipulation task first requires ground manipulation followed by aerial manipulation. First the system operates in the ground mode as a MM-UGV as depicted in Figure 3.1a. Once the manipulation task is complete, the manipulator arm folds back as the MM-UGV prepares to couple with the UAV, shown in Figure 3.1b. The hybrid vehicle can take off after coupling with the manipulator stowed and navigate to the aerial task site as depicted in Figure 3.1c. The MM-UAV subsequently performs an aerial manipulation task which is unreachable from the ground or by a regular MM-UGV as shown in Figure 3.1d.

3.2 Component Layout

The component layout for the MM-UGV is highlighted in Figure 3.2.

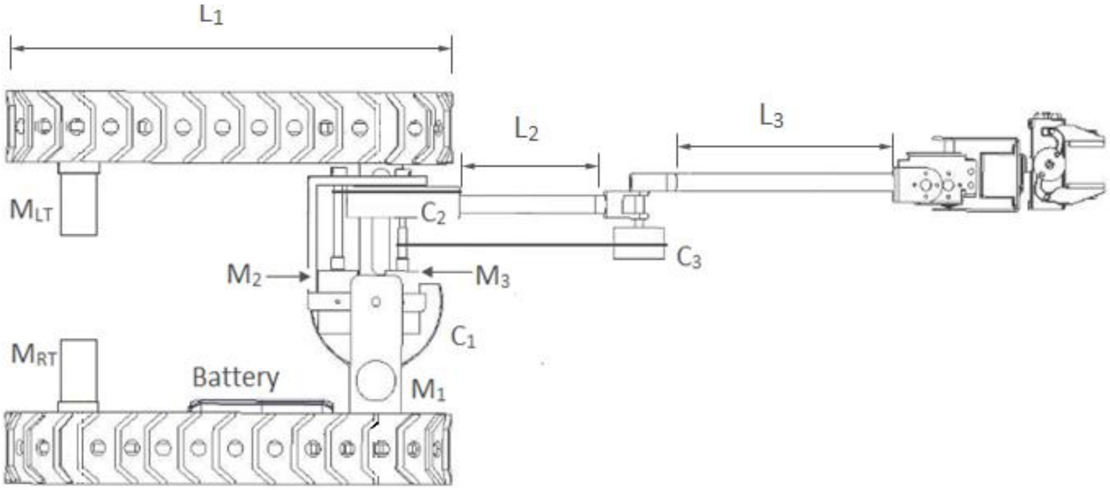


Figure 3.2: MM-UGV CAD drawing of component layout

This figure is important because it gives insight into the system design variables. Note that for clarity, the latching mechanism is removed from the figure. The motors M_1, M_2, M_3 actuate the revolute joints of the robotic manipulator. The motors that drive the tracks and provide differential driving capability of the MM-UGV are labeled as M_{LT}, M_{RT}. The manipulator joints are actuated through cable driven capstan gear mechanisms, C_1, C_2, C_3. A threaded motor-shaft extension allows the Capstan to be coupled to the motor-shaft through a cable. Capstan drives are light weight, suffer from little losses, and allow remote actuation of the links making the motors safer during a crash landing. The Capstan gear mechanism highly influenced the configuration of the three revolute joins. The parallel second and third joint axes allowed for convenient cable placement and functionality. Although the proposed robotic manipulator has three joints, only the second and third link lengths are used as design variables L_2, L_3. The length of base tracks is also used in the design, L_1. Lastly, the battery is selected since the battery mass is considerable and therefore should be minimized if possible.

An isometric view of the CAD model MM-UGV is shown in Figure 3.3. The joint axes are illustrated for the tracks and manipulator joints J_i where $i \in \{LT, RT, 1, 2, 3\}$. The tracks are differentially actuated using the torques produced by the left and right track motor and gearbox combinations. The required torques at the track joints are

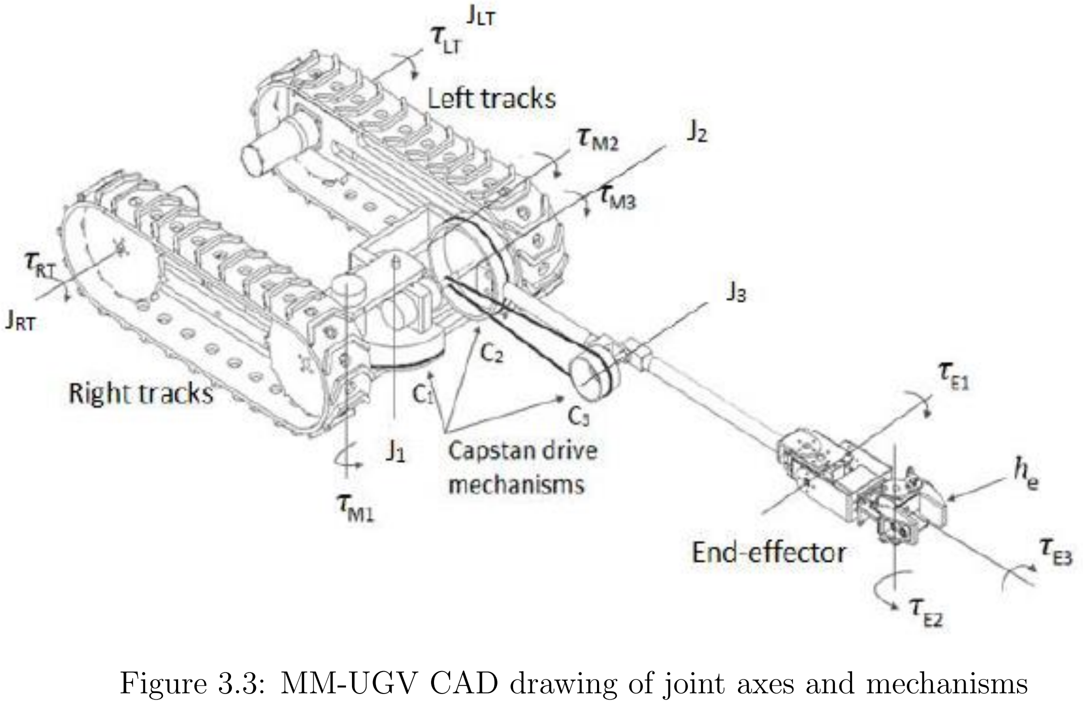


Figure 3.3: MM-UGV CAD drawing of joint axes and mechanisms

denoted by τ_{LT} and τ_{RT} respectively. The manipulator links are actuated using the torques τ_{Mj} produced by the drive system for each link, where $j \in \{1, 2, 3\}$. The end-effector experiences an interaction force vector with the environment $\boldsymbol{h}_e \in R^3$. The end-effector is actuated using torques τ_{Ei} where $j \in \{1, 2, 3\}$.

3.3 Latching Mechanism

The latching mechanism was designed to allow for alignment offset in 6 degrees of freedom (DOF). In other words, within a design tolerance, the MMUGV can successfully latch with the stationary UAV if there is misalignment in all 3 orientations and 3 positions in space. This is a critical design implication because the MMUGV must be able to couple with the UAV when the terrain is not perfectly level. Moreover, the latching mechanism is designed such that back-drivability is not possible in order to guarantee that the MMUGV does not slip out of the latch. The latching components were not considered in the mass minimization optimization algorithm presented in this research. However, the latch was designed specifically such that a small lead-screw stepper motor could be used to facilitate the latching process.

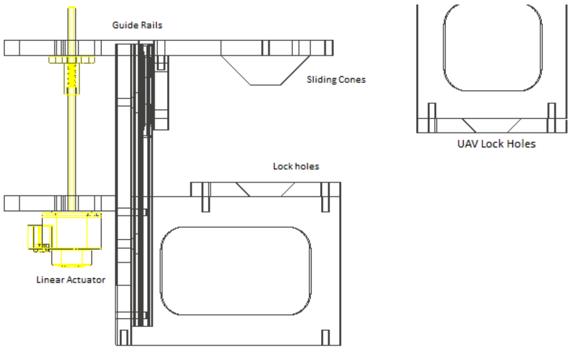


Figure 3.4: Latching Mechanism, CAD drawing.

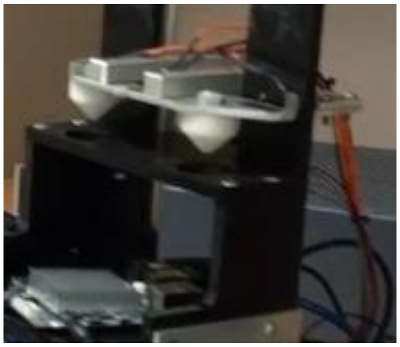


Figure 3.5: Latching Mechanism.

As shown in the CAD model in Figure 3.4, the upper latching surface is allowed to move vertically, actuated through a lead-screw stepper motor and the motion is restricted to one axis using guide rails. Notice the two latching cones on the latching surface. These cones are used to guide the latching surfaces together by first aligning with the UAV Lock Holes and subsequently the MMUGV Lock Holes. The lower latching surface has two beveled holes that have minimal tolerance of 1mm and therefore allow the cones to have a snug fit. This is important because looseness in the cone-to-hole connection could result in latch failure (dropping the MMUGV during flight) or degraded performance of the manipulator during flight (due to vibrations). Note that the cone material selected was white acetal copolymer to allow the small tolerance without the concern for jamming. Once the cones are fully inserted into the latching surface holes, the surfaces are driven together. Lastly, the Neoprene (synthetic rubber) with crisscross texture was placed between the latching surfaces in order to dampen vibration during flight. The constructed latching mechanism is shown in Figure 3.5

Chapter 4

System Modeling

The dynamics of the hybrid system are analyzed in its two main modes of operation: aerial mode when operating as a MMUAV and ground mode when operating as a MMUGV. The dynamic equations are used in the formulation of the optimization problem, for example the motor/gearbox required torques and dynamic tip-over stability requires the dynamics of the robot to be known.

Figure 4.1 depicts the coordinate frames required for describing the motion equations. Wherein, $\mathcal{O}_W$ represents the world (inertial) frame, and $\mathcal{O}_G$ and $\mathcal{O}_A$ are fixed-body frames attached to the UGV and UAV centers of masses, respectively. Furthermore, a fixed-body frame $\mathcal{O}_i, i \in (1, 2, 3)$ is attached to the center of mass of each manipulator link. Lastly, $\mathcal{O}_{e1}$ frame and $\mathcal{O}_{e2}$ are assigned to the end-effector center of mass. The frame orientations are assigned arbitrarily [6].

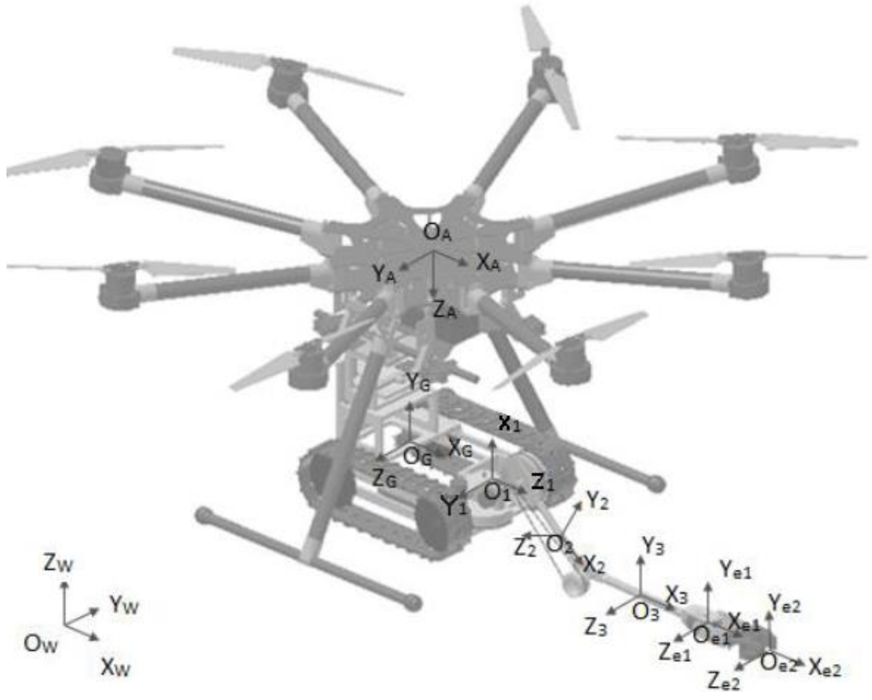


Figure 4.1: Frames of reference for system dynamics.

4.1 Manipulator Dynamics

In both modes of operation, the recursive Newton-Euler formulation of solid-body dynamics is employed to derive the inverse dynamic equations [6]. The dynamics analysis is performed in two steps. First, given the motion variables of all system DOFs (position, velocity, and acceleration) and beginning from the UGV/UAV, a forward recursion is done to determine the velocities/accelerations of all bodies in the system using the Newton-Euler equations. Second, starting from the end-effector forces, a backward force recursion is performed to derive the interaction forces/moments between the bodies up to the external forces/moments applied to the UGV/UAV. Before continuing, Table 4.1 highlights the variables that will be used in this discussion. Figure 4.2 is an illustrative representation of the variables described in Table

Variable	Description
$\boldsymbol{\omega}_i$	Angular velocity of link
$\boldsymbol{\omega}_{mi}$	Angular velocity of motor
$\dot{\boldsymbol{\omega}}_i$	Angular acceleration of link
$\dot{\boldsymbol{\omega}}_{mi}$	Angular acceleration of motor
$\dot{\boldsymbol{p}}_i$	linear velocity of link
$\dot{\boldsymbol{p}}_{C_i}$	linear velocity of the center of mass of link
$\ddot{\boldsymbol{p}}_i$	linear acceleration of link
$\ddot{\boldsymbol{p}}_{C_i}$	linear acceleration of the center of mass of link
θ_i	joint angle
$\dot{\theta}_i$	joint angular velocity
$\ddot{\theta}_i$	joint angular acceleration
$\boldsymbol{r}_{i-1,i}$	vector from origin of $\mathcal{O}_{i-1}$ to origin of $\mathcal{O}_i$
$\boldsymbol{r}_{i-1,C_i}$	vector from origin of $\mathcal{O}_{i-1}$ to center of mass C_i
$\boldsymbol{r}_{i,C_i}$	vector from origin of $\mathcal{O}_i$ to center of mass C_i
I_{r_i}	Moment of inertia of rotor i
$\boldsymbol{I}_{m_i}$	Inertia tensor of motor i
m_i	Mass of augmented link i

Table 4.1: Newton-Euler Equation variable descriptions [6]

4.1 borrowed from [6] and allows us to clearly understand how to handle the case where the motors do not align with the joint axes. This is particularly interesting in our case because of the cable-driven capstan mechanisms used as gear elements. Furthermore, when a variable is superscript of a variable denotes the frame in which it is described. The rotation matrix $\boldsymbol{R}_i^j$ denotes the rotation matrix of frame i with respect to frame j. For example, if we want to express a vector $\boldsymbol{p}^i$ in frame $i-1$, then the vector is pre-multiplied by $\boldsymbol{R}_i^{i-1}$.

The forward recursion equations for revolute joints are summarized as follows. First the angular velocity of link i in frame i is written as,

$$\boldsymbol{\omega}_i^i = (\boldsymbol{R}_i^{i-1})^T \left(\boldsymbol{\omega}_{i-1}^{i-1} + \dot{\theta}_i \boldsymbol{z_G}\right) \tag{4.1}$$

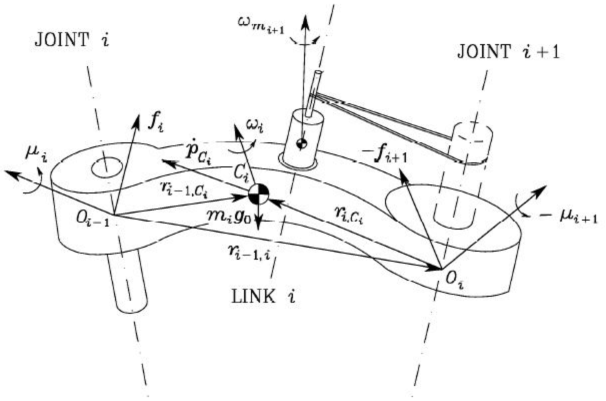


Figure 4.2: Newton-Euler formalism on link variable outline [6]

Subsequently, angular acceleration of link i in frame i can be computed,

$$\dot{\boldsymbol{\omega}}_i^i = (\boldsymbol{R}_i^{i-1})^T \left(\dot{\boldsymbol{\omega}}_{i-1}^{i-1} + \ddot{\theta}_i \boldsymbol{z}_G + \dot{\theta}_i \boldsymbol{\omega}_{i-1}^{i-1} \times \boldsymbol{z}_G \right) \tag{4.2}$$

The linear acceleration of link i in frame i can be computed using,

$$\ddot{\boldsymbol{p}}_i^i = (\boldsymbol{R}_i^{i-1})^T \ddot{\boldsymbol{p}}_{i-1}^{i-1} + \dot{\boldsymbol{\omega}}_i^i \times \boldsymbol{r}_{i-1,i}^i + \boldsymbol{\omega}_i^i \times \left(\boldsymbol{\omega}_i^i \times \boldsymbol{r}_{i-1,i}^i \right) \tag{4.3}$$

Next the linear acceleration of the center of gravity of link i in frame i is written as,

$$\ddot{\boldsymbol{p}}_{C_i}^i = \ddot{\boldsymbol{p}}_i^i + \dot{\boldsymbol{\omega}}_i^i \times \boldsymbol{r}_{i,C_i}^i + \boldsymbol{\omega}_i^i \times \left(\boldsymbol{\omega}_i^i \times \boldsymbol{r}_{i,C_i} \right) \tag{4.4}$$

Followed by the computation of the angular acceleration of motor i in frame $i-1$,

$$\dot{\boldsymbol{\omega}}_{m_i}^{i-1} = k_{ri} \ddot{\boldsymbol{\theta}}_i \boldsymbol{z}_{m_i}^{i-1} + k_{ri} \dot{\theta}_i \boldsymbol{\omega}_{i-1}^{i-1} \times \boldsymbol{z}_{m_i}^{i-1} \tag{4.5}$$

Note that $\boldsymbol{q} = [\theta_i, \theta_{i+1}, ..., \theta_n]$ for $i \in \{1, 2, ..., n\}$ for an n-joint system. Once equations 4.1 - 4.5 have been propagated to the end-effector, the stage is set for the backwards recursion. The force exerted by link $i-1$ on link i is given by,

$$\boldsymbol{f}_i^i = \boldsymbol{R}_{i+1}^i \boldsymbol{f}_{i+1}^{i+1} + m_i \ddot{\boldsymbol{p}}_{C_i}^i \tag{4.6}$$

Further, the moment exerted by link $i-1$ on link i with respect to the origin of $\mathcal{O}_{i-1}$ is given by,

$$\begin{aligned} \boldsymbol{\mu}_i^i = -\boldsymbol{f}_i^i \times \left(\boldsymbol{r}_{i-1,i}^i + \boldsymbol{r}_{i,C_i}^i\right) + \boldsymbol{R}_{i+1}^i \boldsymbol{\mu}_{i+1}^{i+1} + \boldsymbol{R}_{i+1}^i \boldsymbol{f}_{i+1}^{i+1} \times \boldsymbol{r}_{i,C_i}^i + \bar{\boldsymbol{I}}_i^i \dot{\boldsymbol{\omega}}_i^i + \\ \boldsymbol{\omega}_i^i \times \left(\bar{\boldsymbol{I}}_i^i \boldsymbol{\omega}_i^i\right) + k_{r,i+1} \ddot{\theta}_{i+1} I_{r_{i+1}} \boldsymbol{z}_{m_{i+1}}^i + k_{r,i+1} \dot{\theta}_{i+1} I_{r_{i+1}} \boldsymbol{\omega}_i^i \times \boldsymbol{z}_{m_{i+1}}^i \end{aligned} \tag{4.7}$$

Finally, the torque resulting at the joints can be calculated as,

$$\boldsymbol{\tau}_i = \boldsymbol{\mu}_i^T \boldsymbol{z}_{i-1} + k_{ri} I_{r_i} \dot{\boldsymbol{\omega}}_{m_i}^T \boldsymbol{z}_{m_i} + \boldsymbol{F}_{v_i} \dot{\theta}_i + \boldsymbol{F}_{s_i} \mathrm{sgn}\left(\dot{\theta}_i\right) \tag{4.8}$$

The Newton-Euler formulation just described can be implemented on the hybrid system. The equations are solved for the aerial mode (acting as an MMUAV) as follows. First, we assume that the angular velocity $\boldsymbol{\omega}_G^G = \boldsymbol{\nu}$, angular acceleration $\dot{\boldsymbol{\omega}}_G^G = \dot{\boldsymbol{\nu}}$, and linear acceleration $\ddot{\boldsymbol{p}}_G^G = \ddot{\boldsymbol{\xi}}$ of the MMUAV are given. Moreover, the vector of forces and moments exerted on the end-effector, $\boldsymbol{h}_e = [\boldsymbol{f}_4^4 \; \boldsymbol{\mu}_4^4]^{\mathrm{T}}$, is known. The rotation matrices used in the formulation are constructed as follows. The rotation matrix from the world frame to the UAV frame is written as,

$$\boldsymbol{R}_G^W = \boldsymbol{R}_x(\phi) \boldsymbol{R}_y(\theta) \boldsymbol{R}_z(\psi)$$

The roll angle, ϕ, defines the rotation around the x-axis, the pitch angle, θ, defines the rotation about the y-axis, and the yaw angle , ψ, defines the rotation about the z-axis.

$$\boldsymbol{R}_1^G = \boldsymbol{R}_z\left(\theta_1\right) = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{R}_2^1 = \boldsymbol{R}_x(\frac{\pi}{2})\boldsymbol{R}_z\left(\theta_2\right) = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 \\ 0 & 0 & -1 \\ \sin(\theta_2) & \cos(\theta_2) & 0 \end{bmatrix}$$

$$\boldsymbol{R}_3^2 = \boldsymbol{R}_z\left(\theta_3\right) = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{R}_4^3 = \boldsymbol{I}_{3\times 3}$$

It is assumed that the rotor axis i coincides with joint axis i, $\boldsymbol{z}_{m_i}^{i-1} = \boldsymbol{z}_G = [0\ 0\ 1]^T$. Typically, in Newton-Euler formulation the augmented inertia of link i, $\bar{\boldsymbol{I}}_i$, represents the inertia tensor of link i plus the inertia tensor of motor $i+1$. However, the joints are remotely actuated through a cable-driven capstan mechanism. Therefore, the augmented inertia of each link as well as the base is written as follows,

$$\bar{\boldsymbol{I}}_G = \boldsymbol{I}_{L_G} + \boldsymbol{I}_{m_1}$$

$$\bar{\boldsymbol{I}}_1 = \boldsymbol{I}_{L_1} + \boldsymbol{I}_{m_2} + \boldsymbol{I}_{m_3}$$

$$\bar{\boldsymbol{I}}_2 = \boldsymbol{I}_{L_2}$$

$$\bar{\boldsymbol{I}}_3 = \boldsymbol{I}_{L_3}$$

Since the length of link 1 is small, the overall required torque of each motor is reduced. The stage is now set to carry out the forward and backward recursion steps. It would be excessive to write each step and would not provide any additional information about the problem and therefore it was omitted. The Newton Euler formulation in its recursive form is beneficial computationally with a complexity of $O(n^2)$. Moreover, we now have the required joint torques for a given prescribed motion. We also have the interaction force that the manipulator exerts on the UAV, $\boldsymbol{f}_1^1 = \boldsymbol{F}_{\mathrm{M}}$, and the interaction moment, $\boldsymbol{\mu}_1^1 = \boldsymbol{\mu}_{\mathrm{M}}$. This interaction force/torque is considered a disturbance to the UAV dynamics.

4.2 UAV Dynamics

The stage is set for deriving the UAV dynamics. The linear position of the UAV is defined in $\mathcal{O}_W$,

$$\boldsymbol{\xi} = [x\ y\ z]^{\mathrm{T}}$$

The attitude of the UAV, $\boldsymbol{\eta}$ is also defined in $\mathcal{O}_W$ using the Euler angle notation,

$$\boldsymbol{\eta} = [\phi\ \theta\ \psi]^{\mathrm{T}}$$

The relationship between the angular velocity of the UAV and the attitude rate (derivative of the Euler angle vector $\boldsymbol{\eta}$) is realized through a transformation matrix $\boldsymbol{G}$. As in [61],

$$\dot{\boldsymbol{\eta}} = \boldsymbol{G}^{-1}\boldsymbol{\nu} \tag{4.9}$$

where the $\boldsymbol{G}$ matrix is given by,

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & \mathrm{s}(\theta) \\ 0 & \mathrm{c}(\phi) & \mathrm{c}(\theta)\mathrm{s}(\phi) \\ 0 & -\mathrm{s}(\phi) & \mathrm{c}(\theta)\mathrm{c}(\phi) \end{bmatrix}$$

And its inverse,

$$\boldsymbol{G}^{-1} = \begin{bmatrix} 1 & \mathrm{s}(\phi)\mathrm{t}(\theta) & \mathrm{t}(\theta)\mathrm{c}(\phi) \\ 0 & \mathrm{c}(\phi) & -\mathrm{s}(\phi) \\ 0 & \mathrm{s}(\phi)/\mathrm{c}(\theta) & \mathrm{c}(\phi)/\mathrm{c}(\theta) \end{bmatrix}$$

where $c(\cdot)$, $s(\cdot)$, and $t(\cdot)$ represent cosine, sine, and tangent respectively. The Newton-Euler equations allow us to express the linear dynamics of the UAV in $\mathcal{O}_W$ as follows,

$$m_{\mathrm{U}}\ddot{\boldsymbol{\xi}} = -g[0\ 0\ 1]^{\mathrm{T}} + \boldsymbol{R}_W^G \boldsymbol{F}_{\mathrm{U}} + \boldsymbol{R}_W^G \boldsymbol{R}_G^1 \boldsymbol{F}_{\mathrm{M}} \tag{4.10}$$

where m_{U} is the mass of the UAV combined with UGV and $\boldsymbol{F}_{\mathrm{U}}$ represents the thrust vector associated with the propellers. The angular dynamics of the UAV in the UAV body frame $\mathcal{O}_G$ is expressed as,

$$\boldsymbol{I}_{\mathrm{U}}\dot{\boldsymbol{\nu}} + \boldsymbol{\nu} \times (\boldsymbol{I}_{\mathrm{U}}\boldsymbol{\nu}) = \boldsymbol{\tau}_{\mathrm{U}} + \boldsymbol{R}_G^1 \boldsymbol{\mu}_1^1 \tag{4.11}$$

where $\boldsymbol{I}_{\mathrm{U}}$ denotes the inertia tensor of the UAV and UGV combined and $\boldsymbol{\tau}_{\mathrm{U}}$ denotes the external torque exerted on the UAV as a result of the propellers. In order to complete the angular dynamics, the relationship in 4.9 must be employed. The angular acceleration is related to the Euler angle acceleration as follows,

$$\ddot{\boldsymbol{\eta}} = \frac{d}{dt}(\boldsymbol{G}^{-1}\boldsymbol{\nu}) \tag{4.12}$$

The chain-rule is implemented,

$$\ddot{\boldsymbol{\eta}} = \dot{\boldsymbol{G}}^{-1}\boldsymbol{\nu} + \boldsymbol{G}^{-1}\dot{\boldsymbol{\nu}} \tag{4.13}$$

In order to solve for the derivative of the rotation matrix, the skew-symmetric matrix is employed,

$$\dot{\boldsymbol{G}}^{-1} = \boldsymbol{S}(\boldsymbol{\nu})\boldsymbol{G}^{-1}$$

The skew symmetric matrix is formed such that its symmetric elements with respect to the main diagonal represent the components of $\boldsymbol{\nu}$ as follows,

$$\boldsymbol{S}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & -\nu_z & \nu_y \\ \nu_z & 0 & -\nu_x \\ -\nu_y & \nu_x & 0 \end{bmatrix}$$

The complete angular dynamics of the UAV are written as follows,

$$\ddot{\boldsymbol{\eta}} = \boldsymbol{S}(\boldsymbol{\nu})\boldsymbol{G}^{-1}\boldsymbol{\nu} + \boldsymbol{G}^{-1}\boldsymbol{I}_{\mathrm{U}}^{-1}\left(-\boldsymbol{\nu} \times (\boldsymbol{I}_{\mathrm{U}}\boldsymbol{\nu}) + \boldsymbol{\tau}_{\mathrm{U}} + \boldsymbol{R}_{G}^{1}\boldsymbol{\mu}_{1}^{1}\right) \tag{4.14}$$

At this point it is appropriate to mention that the octorotor used in this study is modeled as a quadrotor in order to solve the redundancy problem of eight propellers. Adjacent rotors with the same spinning direction are paired together to form a quadrotor system. The rotor speed of the propellers are denoted by $\omega_{r,i}$ and the propeller produced forces F_i which as mentioned before construct the thrust vector $\boldsymbol{F}_{\mathrm{U}}$. It has been shown [62] that the propeller produced forces are proportional to the square angular speed of the rotor. This relationship is shown,

$$F_i = k\omega_{r,i}^2 \tag{4.15}$$

where $k > 0$ is the lift constant and depends on the number of blades on the propeller, the blade chord length, the air density and the cubed rotor blade radius [62]. The total thrust T_U is given by,

$$T_U = k\sum_{i=1}^{4}\omega_{r,i}^2 \tag{4.16}$$

Therefore, the thrust vector defined in $\mathcal{O}_G$ is defined as,

$$\boldsymbol{F}_{\mathrm{U}} = \begin{bmatrix} 0 \\ 0 \\ T_U \end{bmatrix} \tag{4.17}$$

The torque produced by the propellers, τ_{M_i} acts in the opposing direction the rotor spin. The propeller produced torque is given by,

$$\tau_{M_i} = b\omega_{r,i}^2 \tag{4.18}$$

where b depends on the same factors as k. Note that F_i is linearly related to τ_{M_i} through a torque-force mapping constant λ,

$$\frac{\tau_{M_i}}{F_i} = \frac{b}{k} = \lambda \tag{4.19}$$

The torque vector $\boldsymbol{\tau}_{\mathrm{U}}$ as mentioned earlier in the dynamic equations, consists of three components τ_ϕ acting about the x-axis, τ_θ acting about the y-axis, and τ_ψ acting about the z-axis. Note that these axes are the principal axes of $\mathcal{O}_G$. The torque vector is derived as follows,

$$\boldsymbol{\tau}_{\mathrm{U}} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} -l_{U1} & -l_{U1} & l_{U1} & l_{U1} \\ -l_{U2} & l_{U2} & l_{U2} & -l_{U2} \\ \lambda & -\lambda & \lambda & -\lambda \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{4.20}$$

It is customary to concatenate the force and torque vectors described in 4.20 and 4.16 to form the following,

$$\begin{bmatrix} T_U \\ \boldsymbol{\tau}_{\mathrm{U}} \end{bmatrix} = \boldsymbol{A} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -l_{U1} & -l_{U1} & l_{U1} & l_{U1} \\ -l_{U2} & l_{U2} & l_{U2} & -l_{U2} \\ \lambda & -\lambda & \lambda & -\lambda \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{4.21}$$

Figure 4.3 shows the free-body diagram for calculating the torque resulting from the propeller produces thrust.

As mentioned in the introduction, there are six degrees of freedom of the UAV and only four control inputs $F_i = k\omega_{r,i}^2$. The force-torque equation shown in 4.21 is visible to the dynamic equations. This vector is used to design controllers to stabilize the attitude and subsequently the position of the UAV. The force-torque vector is

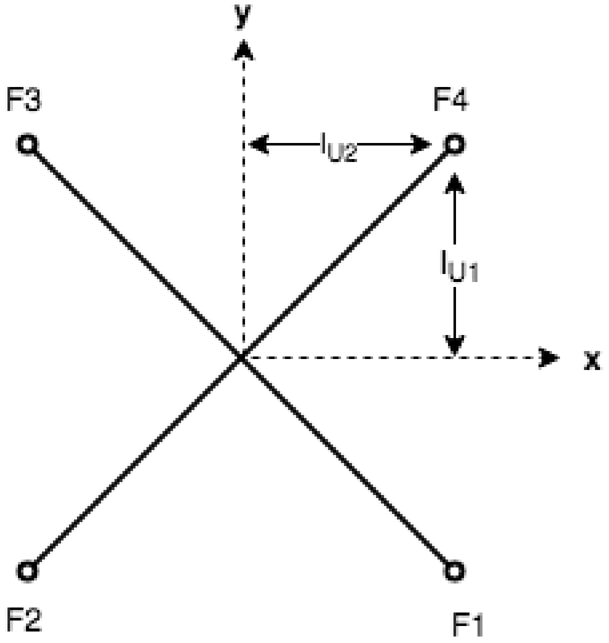


Figure 4.3: Thrust diagram; thrust vectors are pointing out of the page.

mapped to the propeller produced forces through the mapping matrix $\boldsymbol{A}$ as follows,

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = \boldsymbol{A}^{-1} \begin{bmatrix} T_U \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{4.22}$$

4.3 MMUGV Dynamics

The dynamic equations that describe the motion of the MMUGV are derived in the same manner as the MMUAV scenario. However, in the case of ground operation, we are only concerned with the linear motion in x and y directions and angular motion about the z-axis in $\mathcal{O}_W$. The MMUGV is rotated into frame $\mathcal{O}_G$ as a result of a surface grade β. Again the Newton-Euler equations are used to derive the required joint torques and the interaction force between the base and the manipulator. The dynamic formulation follows [63] with the extension of the manipulator and considering the aforementioned assumptions. The acceleration of the mobile base center of mass in the base frame $\mathcal{O}_G$ is derived by studying nonholonomic mechanics [64],

$$\boldsymbol{a}_{\boldsymbol{c}}^{\boldsymbol{G}} = \dot{\boldsymbol{v}}_{\boldsymbol{c}}^{\boldsymbol{G}} - \boldsymbol{J}_2 \boldsymbol{v}_{\boldsymbol{c}}^{\boldsymbol{G}} \dot{\theta} \tag{4.23}$$

where $\boldsymbol{v}_{\boldsymbol{c}}^{\boldsymbol{G}} \in \mathbb{R}^2$ is the velocity of the center of mass of the base in $\mathcal{O}_G$, θ is the heading, and $\boldsymbol{J}_n \in \mathbb{R}^n$ is the reverse identity matrix. The linear dynamics are written as follows,

$$m_B \boldsymbol{a}_c^G = \boldsymbol{F}_L + \boldsymbol{F}_R + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \left(\boldsymbol{R}_G^1 \boldsymbol{F}_M - \boldsymbol{R}_G^W \boldsymbol{z}_W g \right) \tag{4.24}$$

where m_B is the mass of the mobile base, $\boldsymbol{F}_R \in \mathbb{R}^2$ is the force that the right wheel exerts on the base and $\boldsymbol{F}_L \in \mathbb{R}^2$ is the force that the left wheel exerts on the base. The mobile robot is assumed to exhibit pure rolling (no slipping) in the x-direction of $\mathcal{O}_G$ and no sliding in the y-direction of $\mathcal{O}_G$. These non-holonomic constraints are considered by setting the velocity of point A to zero. This decouples the components of v_c^G, which can be expressed as follows,

$$v_{c(y)}^G = d\dot{\theta} \tag{4.25}$$

Taking equation 4.25 into account and incorporating torque, 4.24 can be rewritten as,

$$m_B\dot{v}^0_{c(x)}\dot{\theta} - m_B d_B\dot{\theta}^2 - \begin{bmatrix}1 & 0 & 0\end{bmatrix}\left(\boldsymbol{R}^1_G\boldsymbol{F}_M - \boldsymbol{R}^W_G\boldsymbol{z}_W g\right) = \frac{\tau_L + \tau_R}{r_T} \tag{4.26}$$

where d_B is the distance between the center of mass and the track wheel joint axis in the body x-direction, $\mathcal{O}_G$. The variable r_T describes the radius of the driving wheel.

The angular dynamics are denoted by,

$$I_z\ddot{\theta} = [-w_B/2 \ \ -d_B]\boldsymbol{F}_L + [w_B/2 \ d_B]\boldsymbol{F}_R + [0\ 0\ 1]\boldsymbol{R}^1_G\boldsymbol{\mu}_M \tag{4.27}$$

where w_B is the distance between the driving wheels and I_z is the inertia of the mobile robot about the z-axis of $\mathcal{O}_G$. The angular dynamics expressed in equation 4.27 can also be reduced using the non-holonomic constraint to yield the following formula,

$$(m_B d_B^2 + I_z)\ddot{\theta} + m_B d v^G_{c(x)}\dot{\theta} - [0\ 0\ 1]\boldsymbol{R}^1_G\boldsymbol{\mu}_M = \frac{w_B}{2r_T}(\tau_R - \tau_L) \tag{4.28}$$

This completes the discussion about dynamic equations of the hybrid system in its two main modes of operation. Given the motion variables, the required joint torque of the manipulator and mobile robot wheels can be derived in the case of ground operation. Moreover, in the aerial case, the manipulator torques can be calculated and the required propeller-produced force to stabilize the UAV can be determined.

The next step is to determine a tip-over stability criteria that will alert the optimization algorithm that the current design variables and motion variable limits may incur a tip-over incident. The next subsection is aimed at deriving the so-called

Moment Height Stability criteria [52].

4.4 Tip-over Stability

Tipping over is one of the most problems addressed when controlling a MMUGV. The unknown terrain and also unknown inertial/mass properties of the manipulated object can easily cause a MMUGV to tip-over. Therefore, a dynamic estimation of tip-over called Moment-Height-Stability (MHS) is investigated [52]. Note that this parameter is used during the design process for the first time in this work.

The MHS algorithm first starts with defining a support boundary polygon (SBP). The SBP is derived by creating a convex surface formed by the points of the mobile robot that are in direct contact with the ground. Only pure rotation (no sliding) about a support boundary edge is considered for loss of stability. Recalling the nonholonomic constraints imposed on the system when analyzing the MMUGV dynamics, the yaw stability or stability about the z-axis of $\mathcal{O}_G$ can be neglected. Therefore the roll and pitch stability are the focus of the MHS algorithm. The support boundary edges are defined as $\hat{\mathbf{e}}_i$ where $i \in \{1, 2, ..., n\}$ and the support boundary vertices are denoted by v_i where $i \in \{1, 2, ..., n\}$. In this case the support boundary polygon is a rectangle and therefore $n = 4$. The edges can be define as unit vector as follows,

$$\begin{aligned} \hat{\mathbf{e}}_i &= \frac{v_{i+1} - v_i}{||v_{i+1} - v_i||} \\ \hat{\mathbf{e}}_n &= \frac{v_1 - v_n}{||v_1 - v_n||} \end{aligned} \tag{4.29}$$

where $i \in \{1, 2, ..., n-1\}$. The moment about each vertex, M_{v_i} can be computed by considering the manipulator interaction force/moment, the gravitational force, and the inertial force. Accordingly,

$$\boldsymbol{M}_{v_i} = \boldsymbol{R}_G^1 \boldsymbol{\mu}_1^1 + \boldsymbol{r}_{v_{i,cm}}^G \times \begin{bmatrix} 0 \\ 0 \\ m_B g \end{bmatrix} + \boldsymbol{r}_{v_{i,cm}}^G \begin{bmatrix} \dot{\boldsymbol{v}}_c^G \\ 0 \end{bmatrix} + \boldsymbol{r}_{v_{i,G}}^G \times (\boldsymbol{R}_G^1 \boldsymbol{f}_1^1) \tag{4.30}$$

The vertices moments are projected to the support boundary edges through a dot product as follows,

$$M_{e_i} = \boldsymbol{M}_{v_i} \cdot \hat{\boldsymbol{e}}_i \tag{4.31}$$

Once the edge moments have been computed, the stability about each edge can be calculated as follows,

$$\alpha_i = (I_{e_i})^{\sigma_i} M_{e_i} \tag{4.32}$$

where,

$$\sigma_i = \begin{cases} 1, & \text{if} M_{e_i} > 0 \\ -1, & \text{otherwise} \end{cases} \tag{4.33}$$

The minimum edge stability is used because it determines the worst case stable edge. Lastly the height of the center of mass of the base is incorporated into the stability criteria as follows,

$$\psi = (h_{cm})^{\lambda} \min \alpha_i \tag{4.34}$$

where,

$$\lambda = \begin{cases} -1, & \text{if} \min \alpha_i > 0 \\ 1, & \text{otherwise} \end{cases} \tag{4.35}$$

Note that the center of mass height h_{cm} depends on the kinematics of the manipulator.

The meaning of the stability criteria can be understood as follows. If $\psi > 0$, the MMUGV is still stable. If $\psi = 0$, the MMUGV is still stable but the onset of instability is imminent. If $\psi < 0$, then the MMUGV is unstable and is in the process of tipping over.

Now that the dynamic equations of the system operating in both MMUGV and MMUAV modes have been derived, as well as the MHS stability criteria, the stage has been set for the formulation of the optimization problem.

Chapter 5

Design Optimization Problem

The design problem is formulated as a robust bilevel programming problem (BLPP)[65, 66]. The formulation of the BLPP consists of an upper-level optimization problem and multiple lower-level optimization problems, where the lower-level optimization problems act as constraints on the upper-level optimization problem.

5.1 Optimization Formulation

The upper-level optimization problem is formulated as the minimization of the MMUGV mass. Recall that it was decided that the UAV that the MMUGV couples with will be a commercial model and therefore we do not have control over its mass. The minimization of the MMUGV as the upper-level cost function was done in order to increase the suitability of coupling capability. The drive-chain components (such as DC motors and gearboxes), manipulator links, UGV chassis, capstan mechanisms, and battery contribute to the mass of the system. To reduce complexity, the latching mechanism is considered a constant mass and left out of the optimization. The upper-level optimization problem is subject to a set of constraints such as the manipulator workspace, manipulator link length, propeller produced forces, battery properties, and tip-over stability. Multiple lower-level optimization problems are present including the motor stall torque, motor peak speed, gearbox stall torque, gearbox peak speed. These lower-level optimization problems are taken in both the ground case G and the aerial case A. Moreover, the variable y in this case is the motion variable space (e.g. positions, velocities, accelerations) and the end-effector interaction forces. Therefore the robustification of the lower-level optimization problem is taken into account by finding the worst-case values over all possible combinations of the motion variables. This approach ensures that the designed MMUGV can operate within the expected operational limits, without violating actuator constraints or tipping over.

The upper-level decision variables include manipulator link lengths l_i for $i = 2, 3$, base length l_1, Capstan gear ratios (continuous variables), DC motors, gearboxes, and battery (integer variables). The integer variables are selected from a list of commercially-available components $\boldsymbol{u_m}$, $\boldsymbol{u_{gb}}$, and $\boldsymbol{u_b}$ respectively. The hierarchical structure of the optimization problem is shown here. Note that the presence of the integer variables technically makes the problem a more difficult mixed-integer bilevel programming problem (MIBLPP), which complicates the selection of the solution algorithm. Equations VII to XIV represent the lower-level optimization problems written in short-hand in order to preserve clarity. The objective to be minimized, $f(\boldsymbol{x})$, is the total mass of the system, which consists of the combined manipulator mass and UGV chassis mass m_s, the motor mass $m_{m,i}$ and gearbox mass $m_{gb,i}$ used to drive the manipulator links and UGV tracks, and the battery mass m_b.

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = \textstyle\sum_i \left(m_{m,i}(\boldsymbol{u_m}) + m_{gb,i}(\boldsymbol{u_{gb}})\right) + m_s + m_b(\boldsymbol{u_b})$$

subject to:

$$
\begin{aligned}
& l_2 l_3 \geq \rho && \text{(I)}\\
& l_2 < \gamma && \text{(II)}\\
& 0 < T_j \leq T^{\max} \;;\; j = 1, ..., 4 && \text{(III)}\\
& \mathcal{C}_\alpha > 0 && \text{(IV)}\\
& I_{B,\max} \geq \textstyle\sum_i I_{M,i}^{\text{stall}} && \text{(V)}\\
& V_B \geq \max\{V_i\} && \text{(VI)}\\
& \mathcal{C}_{\tau_{M,i}}^G \leq T_{M,i}^{\text{stall}} \;,\; \mathcal{C}_{\tau_{M,i}}^A \leq T_{M,i}^{\text{stall}} && \text{(VII, VIII)}\\
& \mathcal{C}_{\omega_{M,i}}^G \leq \omega_{M,i}^{\text{peak}} \;,\; \mathcal{C}_{\omega_{M,i}}^A \leq \omega_{M,i}^{\text{peak}} && \text{(IX,X)}\\
& \mathcal{C}_{\tau_{gb,i}}^G \leq T_{gb,i}^{\text{stall}} \;,\; \mathcal{C}_{\tau_{gb,i}}^A \leq T_{gb,i}^{\text{stall}} && \text{(XI,XII)}\\
& \mathcal{C}_{\omega_{gb,i}}^G \leq \omega_{gb,i}^{\text{peak}} \;,\; \mathcal{C}_{\omega_{gb,i}}^A \leq \omega_{gb,i}^{\text{peak}} && \text{(XIII, XIV)}
\end{aligned}
$$

where:

$$i \in \{LT, RT, 1, 2, 3\}$$

$$\boldsymbol{x} = [\boldsymbol{u_m}, \boldsymbol{u_{gb}}, \boldsymbol{u_b}, l_1, l_2, l_3, C_1, C_2, C_3]$$

5.2 Upper Level Constraints

The constraints are the manipulator dexterous workspace with respect to the UGV (I), maximum link length (II), the propellers produced forces (III), the dynamic tip-over stability margin (IV), the maximum current output of the battery (V), the battery voltage (VI) and the peak torque and speed of the motors and gearboxes in both the aerial mode and the ground mode (VII - XIV). The lower bound ρ on the size of workspace in (I) ensures that the resulting manipulator would have have a useful workspace, while attempting to minimize its mass. The Link 2 length is bounded above by γ in (II) so that the manipulator arm could be retracted in the aerial mode and to reduce the possibility of its colliding with the under-belly of the UAV. The UAV propellers must be able to produce sufficient thrust forces and moments in order to hoist the MM-UGV and control the manipulator through the entire range of desired motions. The upper and lower limits on the UAV propeller thrust forces are given in (III). Given the decision variable contains a motor selection, a battery can be chosen in order to ensure that the sum of motor stall currents do not exceed the maximum allowable battery current. This constraint is displayed in Constraint (V). Moreover, the battery voltage is selected such that all the motors can operate at their nominal voltage, shown in Constraint (VI). Next, the lower-level optimization problems are discussed in detail which elaborates on the aforementioned short-notation.

5.3 Lower Level Optimization Problems

To determine the worst-case stability measure (MHS stability criteria) over all possible combinations of motion variables and end-effector forces (lower-level decision variable y),

$$\mathcal{C}_\alpha = \left\{ \begin{array}{l} \min\limits_{\boldsymbol{y}} \{\Upsilon(\boldsymbol{x}, \boldsymbol{y})\} \\ \\ \text{s.t. } \boldsymbol{g}_j(\boldsymbol{y}) \leq 0 \ \ ; \ j = 1, ..., 6 \end{array} \right\} > 0 \tag{5.1}$$

with $\boldsymbol{y} = [q_i, \dot{q}_i, \ddot{q}_i, \beta, \boldsymbol{h_{e,i}}, a_x]; i = 1, ..., 3$. The set of inequality constraints $\boldsymbol{g} \in \mathbb{R}^6$ is given by,

$$\begin{array}{ll} g_1 = |\ddot{q}| - \ddot{q}^{\text{G,max}} & g_4 = |\boldsymbol{h}_{\text{e},\boldsymbol{i}}| - h_e^{\text{G,max}} \\ g_2 = |\dot{q}| - \dot{q}^{\text{G,max}} & g_5 = |\beta| - \beta^{\text{max}} \\ g_3 = |q| - q^{\text{G,max}} & g_6 = |a_x| - a_x^{\text{max}} \end{array}$$

where a_x is the linear acceleration of the UGV in the x-direction of $\mathcal{O}_G$. Note that the superscript G indicates a ground mode value. Later, a superscript A will indicate an aerial mode value. Note that the joints q are not treated separately when defining the constraints for simplicity.

The battery is selected to ensure that the sum of the motor stall currents does not exceed the maximum allowable battery current as in Constraint (V). Moreover, the battery voltage is selected such that all the motors can operate at their nominal voltage, see Constraint (VI). Voltage regulating circuits are used in the physical implementation to ensure motor safety.

Selecting motors typically requires the investigation of three criteria: continuous torque rating, peak torque rating, and maximum allowable speed. However, since the entire motion-space is explored, a trajectory is not prescribed and therefore the continuous torque rating is not examined. To simplify the drive-train model, the gears and shafts are assumed to be stiff and the backlash of the gears is assumed negligible. Given the joint torque calculated by the Newton-Euler algorithm, the required i^{th} motor torque is given by,

$$\tau_{M,i} = (J_{M,i} + J_{gb,i})\ddot{q}_i + \frac{\tau_i}{k_i \eta_{gb} \eta_m}; i = 1, ..., 5 \tag{5.2}$$

where $J_{M,i}$ is the motor rotor inertia, $J_{gb,i}$ is the gearbox rotor inertia, $\ddot{q}_i$ is the acceleration of the output shaft, k_i is the gear ratio, $\eta_{m,i}$ is the efficiency of the motor, and $\eta_{gb,i}$ is the efficiency of the gearbox. When selecting a motor, the required peak torque over the entire motion-space must be less than the stall torque limit of the motor. Therefore, the worst-case motor torque for the ground mode is formulated,

$$\mathcal{C}^{G}_{\tau_{M,i}} = \left\{ \begin{array}{l} \max\limits_{\boldsymbol{y}}\{|\tau_M(\boldsymbol{x},\boldsymbol{y})|\} \\ \text{s.t. } \boldsymbol{g}_j(\boldsymbol{y}) \leq 0 \ \ ; \ j = 1, ..., 6 \end{array} \right\}_i \leq T^{stall}_{M,i} \qquad (5.3)$$
$$; i = 1, ..., 5$$

The worst-case motor torque for the aerial mode is also formulated,

$$\mathcal{C}^{A}_{\tau_{M,i}} = \left\{ \begin{array}{l} \max\limits_{\boldsymbol{z}}\{|\tau_M(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})|\} \\ \text{s.t. } \boldsymbol{h}_j(\boldsymbol{z}) \leq 0 \ \ ; j = 1, ..., 15 \end{array} \right\}_i \leq T^{stall}_{M,i} \qquad (5.4)$$
$$; i = 1, ..., 5$$

Note that the joints q are not treated separately when defining the constraints for simplicity. The parameter vector is defined by,

$$\boldsymbol{z} \ = \ [q_i, \dot{q}_i, \ddot{q}_i, \phi_i, \dot{\phi}_i, \ddot{\phi}_i, \theta_i, \dot{\theta}_i, \ddot{\theta}_i, \psi_i, \dot{\psi}_i, \ddot{\psi}_i, \boldsymbol{h_{e,i}}, \dot{x}, \ddot{x}, \dot{y}, \ddot{y}, \dot{z}, \ddot{z}]; \quad i \ = \ 1, ..., 3 \qquad (5.5)$$

The set of inequality constraints $\boldsymbol{h} \in \mathbb{R}^{19}$ are given by,

$$\begin{array}{ll}
h_1 = |\ddot{q}| - \ddot{q}^{A,\max} & h_{11} = |\psi| - \psi^{\max} \\
h_2 = |\dot{q}| - \dot{q}^{A,\max} & h_{12} = |\dot{\psi}| - \dot{\psi}^{\max} \\
h_3 = |q| - q^{A,\max} & h_{13} = |\ddot{\psi}| - \ddot{\psi}^{\max} \\
h_4 = |\boldsymbol{h_{e,i}}| - {h_e}^{A,\max} & h_{14} = |\dot{x}| - \dot{x}^{\max} \\
h_5 = |\phi| - \phi^{\max} & h_{15} = |\ddot{x}| - \ddot{x}^{\max} \\
h_6 = |\dot{\phi}| - \dot{\phi}^{\max} & h_{16} = |\dot{y}| - \dot{y}^{\max} \\
h_7 = |\ddot{\phi}| - \ddot{\phi}^{\max} & h_{17} = |\ddot{y}| - \ddot{y}^{\max} \\
h_8 = |\theta| - \theta^{\max} & h_{18} = |\dot{z}| - \dot{z}^{\max} \\
h_9 \ = |\dot{\theta}| - \dot{\theta}^{\max} & h_{19} = |\ddot{z}| - \ddot{z}^{\max} \\
h_{10} = |\ddot{\theta}| - \ddot{\theta}^{\max} &
\end{array}$$

Furthermore, the required peak speed must not exceed the maximum permissible peak speed of the motor. In the ground mode,

$$\mathcal{C}^{G}_{\omega_M,i} = k_i \dot{q}^{G,\max} \leq \omega^{\text{peak}}_{M,i}; i = 1, ..., 5 \tag{5.6}$$

Similarly, in the aerial mode,

$$\mathcal{C}^{A}_{\omega_M,i} = k_i \dot{q}^{A,\max} \leq \omega^{\text{peak}}_{M,i}; i = 1, ..., 5 \tag{5.7}$$

A similar approach is taken when investigating the gearbox peak torque rating. The permissible peak output torque of the gearbox cannot be exceeded by the required peak torque of the gear box. First, the ground mode is investigated,

$$\mathcal{C}^{G}_{\tau_{gb},i} = \left\{ \begin{array}{l} \max\limits_{\boldsymbol{y}}\{|\tau(\boldsymbol{x}, \boldsymbol{y})|\} \\ \text{s.t. } \boldsymbol{g}_j(\boldsymbol{y}) \leq 0 \;\; ; j = 1, ..., 6 \end{array} \right\}_i \leq T^{stall}_{gb,i} \\ ; \; i = 1, ..., 5 \tag{5.8}$$

Second, the aerial mode is analyzed,

$$\mathcal{C}^{A}_{\tau_{gb},i} = \left\{ \begin{array}{l} \max\limits_{\boldsymbol{z}}\{|\tau(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})|\} \\ \text{s.t. } \boldsymbol{h}_j(\boldsymbol{z}) \leq 0 \;\; j = 1, ..., 19 \end{array} \right\}_i \leq T^{stall}_{gb,i} \\ ; i = 1, ..., 5 \tag{5.9}$$

The required peak speed of the gearbox output shaft must not exceed the allowable peak speed of the gearbox. In the ground mode,

$$\mathcal{C}^{G}_{\omega_{gb},i} = k_i \dot{q}^{G,\max} \leq \omega^{\text{peak}}_{M,i}; \; i = 1, ..., 5 \tag{5.10}$$

Similarly, in the aerial mode,

$$\mathcal{C}^{A}_{\omega_{gb},i} = k_i \dot{q}^{A,\max} \leq \omega^{\text{peak}}_{M,i};\ i = 1, ..., 5 \tag{5.11}$$

5.4 Computation Approach

Global optimality is a key concern when determining an algorithm that can solve the proposed formulation. It is critical that the lower-level optimization problems are solved globally because they are constraints for the upper-level optimization problem. Moreover, it is important in order to ensure proper operation of the gear-train components over all possible motion variables. If the lower-level optimization problems get trapped in a local optimal solution, the operation of the robot could fail. Although it is desired that the upper-level optimization is solved globally, it is not critical to the operation of the robot. Nevertheless, a global optimization approach is taken for both upper-level and lower-level optimization problems.

The Branch-and-Sandwich Bilevel optimization algorithm (BASBL) was used to deterministically solve the optimization problem [67], [68], [69]. BASBL can solve general MINBP with finite ϵ-convergence to a global optimal solution. The lower and upper problems are split into two solution spaces and explored using a single branch-and-bound tree with a customized branching strategy. The branch-and-bound strategy performs exhaustive partitioning on the domain created from lower and upper level optimization variables, which makes BASBL a strong candidate for parallel execution to reduce the run time. Taking advantage of the open-source implementation, the algorithm was parallelized using a shared-memory model known as OpenMP in C++. Previous work has been done on parallel implementations of the standard

Branch-and-Bound algorithm [70, 71]. The OpenMP implementation of BASBL was deployed on a 140 node IBM BladeCenter cluster shown in Table 5.1.

Grid	Nodes	CPU	Memory (GB)
1-4	4	dual 3.33 GHz Dual Core Xeon	6
5-14	10	dual 2.4 GHz Pentium 4 Xeon	8
15-64	50	dual 2.4 GHz Pentium 4 Xeon	1
65-69	5	dual 2.6 GHz Pentium 4 Xeon	1
70	1	dual 2.6 GHz Pentium 4 Xeon	4
71-84	14	dual 2.0 GHz dual core Opteron	2
85-94	10	dual 2.2 GHz dual core Opteron	4
95-98	4	dual 2.2 GHz dual core Opteron	8
99-140	42	dual 3.06 GHz Pentium 4 Xeon	1

Table 5.1: IBM Cluster Node Hardware Specification

Speedup is a metric for analyzing the benefits of converting a sequential algorithm into a parallel/distributed algorithm. Speedup is defined as,

$$S = T_s/T_p \tag{5.12}$$

where T_s is the fastest practical sequential algorithm wall-time execution, and T_p is the wall-time for p processors. Figure 5.1 shows the speed up for the parallel BASBL algorithm as well as a linear speedup reference point.

Due to the complexity of the robotic system in terms of mass and inertia, it was decided to use a modeling software derive the inertia and mass properties of the robotic components. An initial design was implemented in Inventor, where the inertial and mass properties are derived. This CAD model was exported in XML format and transferred to SimMechanics in Matlab/Simulink. This SimMechanics model can be used for simulation and carry out the evaluation of derived dynamic equations required for the optimization.

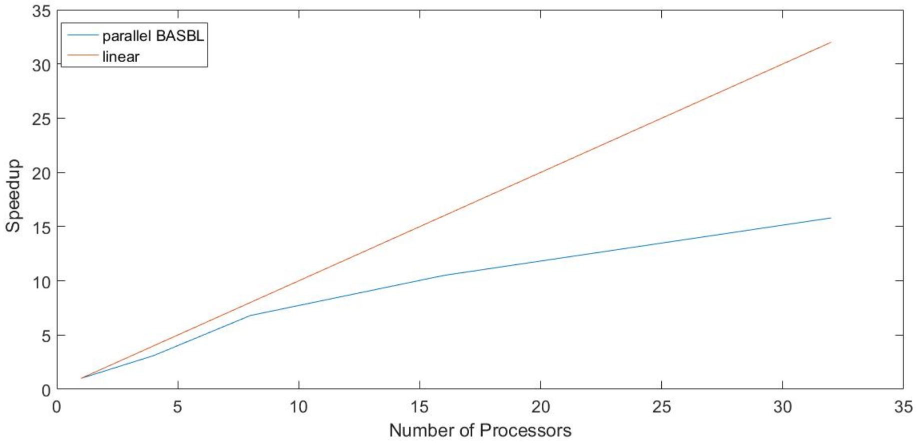


Figure 5.1: Speedup for parallel BASBL

5.5 Numerical Results

The DJI Spreading Wings S1000 is employed as the UAV platform because of its large payload capacity compared to other commercially-available UAVs [3] The set of DC motors used in the design optimization are selected from the Maxon Motors database [72]. The battery selection is made from a database of RC batteries [73]. Acetal black copolymer material was used to construct the frame of the mobile base. The limits in the inequality constraints $\boldsymbol{h}$ and $\boldsymbol{g}$ are assigned as follows. The manipulator motion limits are all 2π (rad, rad/s, rad/s^2) (i.e., $q^{G,\max}$, $\dot{q}^{G,\max}$, $\ddot{q}^{G,\max}$, $q^{A,\max}$, $\dot{q}^{A,\max}$ and $\ddot{q}^{A,\max}$). Moreover, the grade slope is restricted by $\beta^{\max} = \frac{\pi}{6}$ (rad), the linear acceleration of the MM-UGV by $a_x^{\max} = 2$ (m/s^2), and the end-effector force by $h_e^{\max} = 2$ (N). The Euler angle limits, and their first and second derivative limits are all set to $\frac{\pi}{9}$ (rad), (rad/s) and (rad/s^2) respectively. The derivative and second derivative limits for translation are set to 1 (m/s) and 1 (m/s^2) respectively. The minimum workspace was set to $\rho = 0.02$ (m^2).

The initial design was unable to satisfy all constraints for any of the 1000 Monte

Carlo simulations, whereas the optimal design had no violations across all simulations.

The following optimization results were achieved using these motion, configuration, and end-effector force limits. Table I shows the initial and optimal geometry selected for the design.

Table 5.2: Optimization results for lengths

Design variable	Initial	Optimal
l_1	0.3000	0.3316
l_2	0.1500	0.1296
l_3	0.1500	0.1732

The drive-train initial and optimal configurations are shown in Table II.

Table 5.3: Optimization results for drive-trains

Drive-Train	Initial motor	Initial gearbox	Optimal motor	Optimal gearbox
joint 1	RE-max 29	C_1= 30	A-max 26	C_1= 9.18
joint 2	RE-max 29	C_2= 20	RE-max 29	C_2= 10.55
joint 3	RE-max 29	C_3= 10	RE-max 29	C_3= 6.24
left tracks	A-max 26	GS 50 A	A-max 26	GS 30 A
right tracks	A-max 26	GS 50 A	A-max 26	GS 30 A

The initial design and optimal design use the same 11.1V-35C-2200mAh battery, which weighs 190g. The optimized MM-UGV weighs 4.45 Kg, whereas the initial MM-UGV weighs 6.26 Kg. This mass reduction of approximately 29% demonstrates that the optimized design is clearly much more compatible with the UAV than the initial one.

5.6 Monte Carlo Simulations

To validate the design, two separate simulation examples are carried out. First, a ground simulation is done to verify the MM-UGV design. Second, an aerial simulation is considered in order to verify that the UAV can successfully hoist the MM-UGV while the manipulator is controlled through a given motion. Since the design must be valid for all trajectories within the prescribed motion variable constraints, a Monte Carlo simulations approach was adopted. 1000 Monte Carlo Simulations were carried out by sampling the motion variables from a Gaussian distribution randomly between the allowable limits. The motor torque constraints and the dynamic stability tip-over stability margin are shown in Table 5.4. The motor torque constraints are reformulated to be greater or equal to zero. The propeller produced forces constraint is compared against the experimentally determined maximum and reformulated to be greater than or equal to zero for consistency. The results from the ground simulations and the aerial simulations are paired and shown in Table 5.4.

Table 5.4: Monte Carlo Simulation Results

Constraint	Violations (optimal design)	Violations (initial design)
$\mathcal{C}_\alpha$	0	1000
$\mathcal{C}^G_{\tau_{M,1}}$	0	783
$\mathcal{C}^G_{\tau_{M,2}}$	0	762
$\mathcal{C}^G_{\tau_{M,3}}$	0	779
T_1	0	1000
T_2	0	1000
T_3	0	1000
T_4	0	998

5.7 Dynamic Simulations

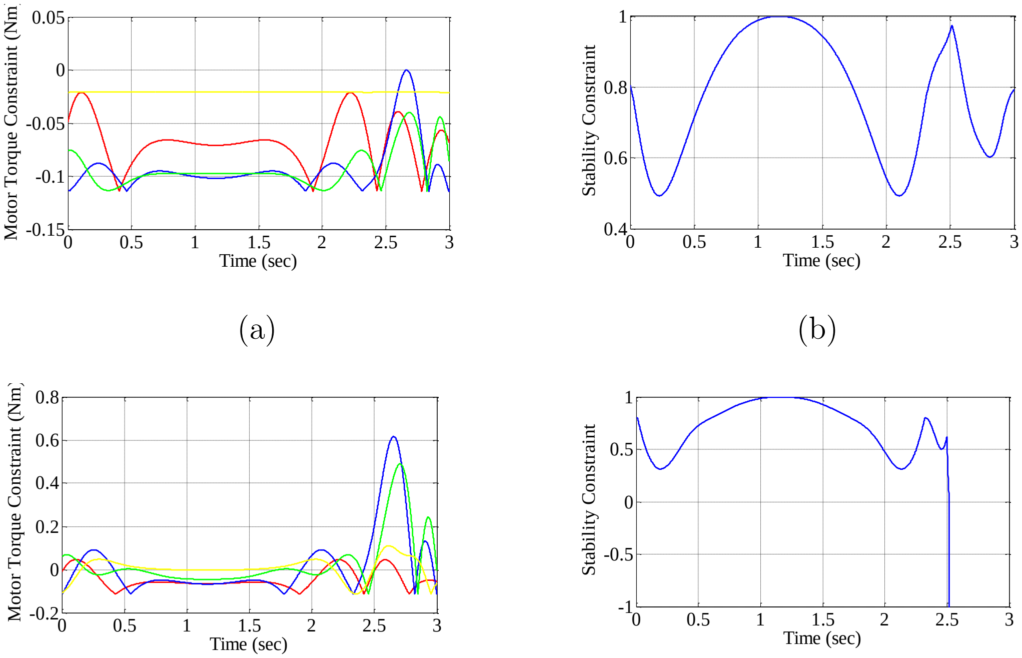
To validate the design, two separate simulation examples are carried out. First, a ground simulation is done to verify the MM-UGV design. Second, an aerial simulation is considered in order to verify that the UAV can successfully hoist the MM-UGV and control the manipulator through some given motion.

5.7.1 Ground Simulation

The MM-UGV moves linearly along a slope of $\beta = \frac{\pi}{6}$ (rad) with respect to $\boldsymbol{X_W}$. The acceleration is set to be the maximum allowable linear deceleration -2 (m/s^2), the initial velocity v_i is 1 (m/s) and the initial position x_i is 0 (m). The motion is therefore described by,

$$x(t) = -t^2 + v_i t$$

The motion of each link is described using a cubic trajectory. For the i^{th} joint, the initial angle $\theta_{i,i}$ is 0 (rad), the final angle $\theta_{f,i}$ is π (rad), the initial angular velocity $\dot{\theta}_{i,i}$ is 0 (rad/s) and the final angular velocity is 2π (rad/s). The magnitude of force applied to the end-effector during the trajectory is 1.5 (N). The motor torque constraints and the dynamic stability tip-over stability margin are investigated for the prescribed motion. Here, the motor torque constraints are reformulated to be strictly less than or equal to zero in order to visualize each motor torque simultaneously. Figure 5.2a and Figure 5.2b show the constraints for the optimized design, whereas Figure 5.2c and Figure 5.2d show the constraints for the initial design. It can be observed from these plots that the optimal design is feasible because it conforms to the constraints. It is important to note that the initial design is not feasible because the torque constraints are violated and the MM-UGV tips over at 2.5 seconds.



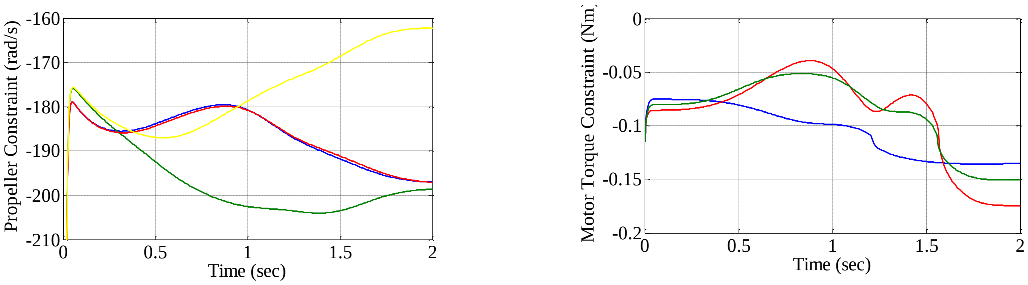
(a) (b)

(c) (d)

Figure 5.2: Actuators and tip-over stability constraints in ground simulations. a) torque constraints using optimal design (must be less than zero); b) dynamic tip-over stability constraint using optimal design (must be positive); c) torque constraints using initial design; d) dynamic tip-over stability constraint using initial design.

5.7.2 Aerial Simulation

The UAV is held fixed, hovering with constant altitude and zero attitude. Similar to the previous simulation, the manipulator undergoes a cubic trajectory. For the i^{th} joint, the initial angle $\theta_{i,i}$ is $-\pi$ (rad), the final angle $\theta_{f,i}$ is 0 (rad), the initial angular velocity $\dot{\theta}_{i,i}$ is 0 (rad/s) and the final angular velocity is 2π (rad/s). In this case, the propeller speed constraints are examined to verify that the UAV can carry the MM-UGV and control the manipulator through the designed trajectory. Again, the constraints plotted in Figure 5.3 are reformulated to be less than or equal to zero. Fig. Figure 5.3a shows that the propeller speed constraints are satisfied. Therefore, the UAV can produce enough thrust force to carry the MM-UGV while controlling the manipulator. Moreover, Fig. Figure 5.3b illustrates that the motor torques of the manipulator remain within the desired constraint limits.



(a) (b)

Figure 5.3: Actuators constraints in aerial simulations: a) propeller speed constraints using optimal design; b) torque constraints using optimal design.

Chapter 6

Hardware and Software Design

The hardware architecture used for control and sensing are described in detail in this section. It is evident that the MMUGV and UAV must have separate hardware components to facilitate individual operation as well as cooperation.

6.1 MMUGV Hardware

Figure 6.1 shows the fully connected hardware component diagram of the MM-UGV. An ODROID-C2 was used as the central hub for sensors and actuators as well as the main controller. The computational resources and I/O resources of the ODROID-C2 outperform other comparable computers such as the Gumstix collection and the Raspberry Pi. The four USB ports makes this computer a favorable choice for robotics applications since it is typical to have many sensors and actuators that require I/O. The out-of-box Linux OS on the computer however is not a suitable platform for robotics because it could cause control or sensing delays. This could lead to catastrophic instabilities if the sensor signals are not sampled at correct intervals or if the

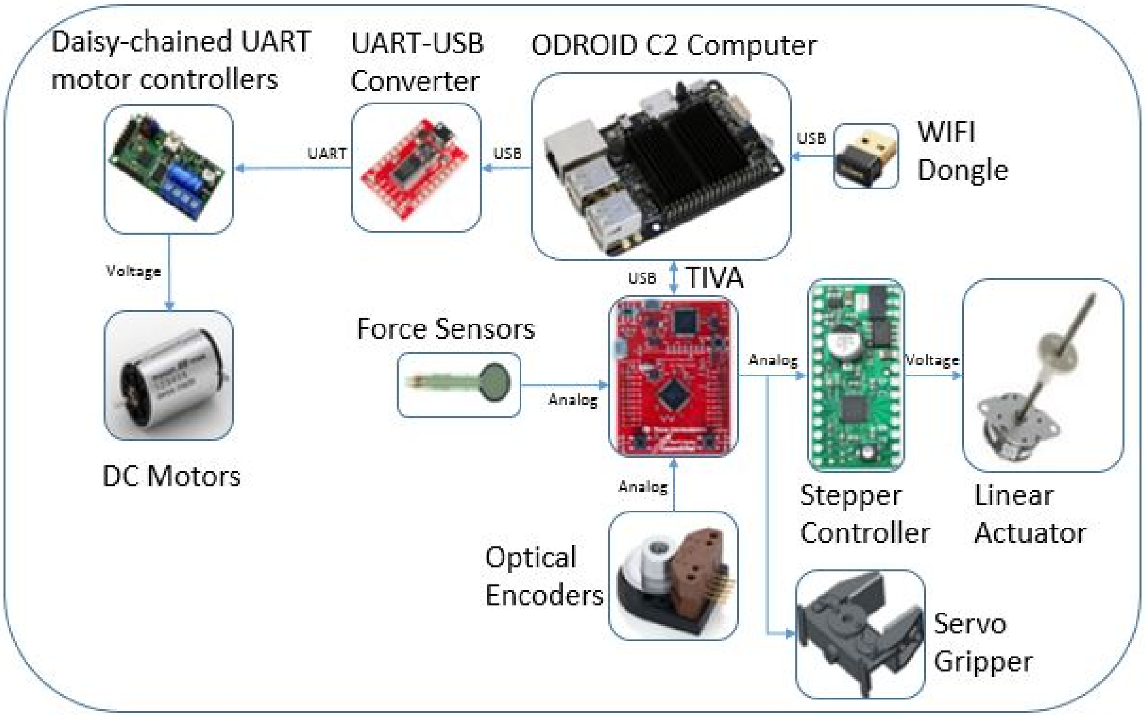


Figure 6.1: MM-UGV hardware.

control signals are delayed. Therefore, the full preemptive kernel patch 3.10.80-rt88 of real-time Linux (RTLinux) was used in order to guarantee reliable response time on critical software components.

A first in first out (FIFO) scheduler policy was used to determine which process to give CPU time next. The RT priority of sensing and control processes are set to 98, it is discouraged to set a maximum priority level of 99 since watchdog threads and other resource managers require this priority level. Other processes such as measurements and logging are set to lowest priority (1) since it is not critical to the robotic operation. Each RT critical application has its virtual address space locked to physical RAM in order to reduce page faults, which can lead to significant latencies. The real-time operation of the applications was examined using [74] which provides routines for analyzing both cyclic timing and the publish-subscribe mechanism of ROS. The CPU was overclocked from 1.5GHz to 1.728 GHz and an external fan was added to allow proper cooling of the overclocked CPU. The Linux-based OS allows for the utilization of the Robotic Operating System (ROS) middleware, discussed in further detail in Chapter 7.

Each DC motor was controlled using a Simple Motor Controller 18v7 from Pololu. A daisy-chained UART connection was used in order to only consume a single USB port on the ODROID-C2. These motor controllers are voltage controllers, which is not ideal. For more accurate control, torque controllers which use current as the control source must be used. However, since PID controllers are used for this robotic system rather than model-based controllers, these motor controllers are an effective choice.

The motors are equipped with E2 optical encoders from US DIGITAL with 1000 counts per revolution (CPR). The encoders are used in quadrature mode which allows for 4000 counts per revolution, which is 0.09 degrees of resolution. The Texas Instruments TIVA TM4C123G microcontroller based on the ARM Cortex-M4F 80MHz processor was selected to process the encoder pulses and convert them into digital counts for the ODROID-C2. Each TIVA board has two embedded encoder interface circuits capable of handling one optical encoder each called the Quadrature Encoder Interface (QEI). Three TIVA microcontrollers were used in order to process the encoder pulses from 3 manipulator motors and 2 track motors. The output of the QEI is rotor position, velocity and direction. The velocity is captured using built-in timer registers, the direction is determined using a position integrator, and the position is obtained using quadrature encoder hardware. The QEI is robust up to 20MHz frequency of the encoder input channels, which is more than satisfactory considering that the designed trajectories boundaries do not need high velocities or accelerations of the joints.

The gripper is actuated with a servo motor which is controlled using a PWM channel on the TIVA board. The control signal is generated at the ODROID-C2, which propagates the TIVA board and subsequently to the servo motor.

A latching mechanism is mounted on the MM-UGV that allows the coupling and decoupling with the UAV. The latching mechanism is actuated using a 36000 Series Linear Actuator from Haydon Motion Solutions. The E36441-05 external model with standard flange is controlled using the TIVA board combined with a A4988 stepper motor driver from Pololu. The driver provides a simple step and direction control interface of the stepper motor based linear actuator. The latching mechanism is equipped with 3 force sensor resistors in order to detect when the latch has closed sufficiently for a secure connection. The FSR has 0.5 inches of sensing area and therefore are used to ensure robustness and stability of the connection between the MM-UGV and the UAV. A 10 Kohm pull-down resistor was used to limit the output of the FSR to a 5V range.

A 5GHz Wifi Dongle was used to communicate the latching state between MM-UGV and UAV ODROID computers. Note that both the MMUGV and UAV hardware suites have a wifi dongle. For the purpose of this experiment, wifi was used as the channel of communication between the two systems. This must be changed to a different mode of operation when wifi is not available, for example during field operation.

6.2 MMUAV Hardware

Figure 6.2 shows the fully connected hardware component diagram of the MM-UAV.

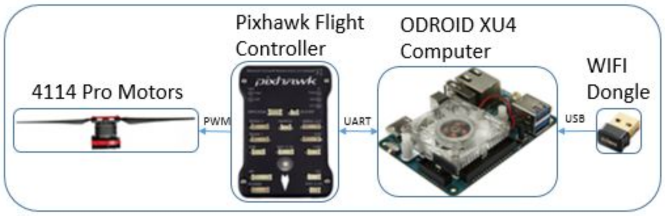


Figure 6.2: MM-UAV hardware.

The UAV is equipped with an ODROID-XU4 main controller which is more powerful compared to the ODROID-C2. This Octa Core Linux Computer was also upgraded to an RTLinux OS. A more powerful computer was selected for the UAV control because in future work, state estimation using cameras will also reside on this computer. Both control and estimation are computationally complex which justifies the requirement for stronger compute power. A Pixhawk from 3D Robotics was used as the flight controller for the experiments performed in this paper. Mavros, a ROS package designed specifically for the pixhawk, was used to relay commands and sensor measurements between the ODROID-XU4 and the pixhawk through a UART connection. The standard PID controller was used which controls the attitude at a rate of 500 Hz. A teleoperator controls the UAV with a 6-channel DX6i Spektrum Radio. The signals are processed by the receiver which is connected directly to the Pixhawk. The receiver inputs are then relayed back to the ODROID-XU4. The Pixhawk controller processes radio commands at 50Hz. The UAV is also equipped with a 5GHz Wifi Dongle in order to connect with the MM-UGV, as mentioned in the previous section.

6.3 Software Design

ROS middleware was used for message passing between software components in the distributed computing system. Figure 6.3 shows the simplest version of a ROS framework.

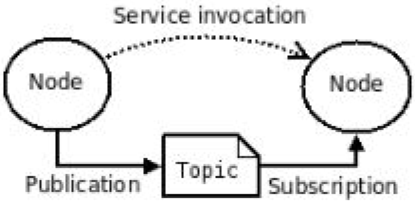


Figure 6.3: ROS communication concept [7].

The nodes are ROS processes/applications that perform computation. This allows for modular decomposition of the computation across different nodes which has proven to be very useful for debugging hardware/software issues. The nodes exchange information using messages, which are data structures containing typed fields. Publishers and Subscribers compose the ROS graph, where publishers write messages of a certain type to a topic and subscribers that are reading from the topic will receive the messages. The topic is restricted to one type of message in order to maintain consistency. There can be many different types, subscribers can read from different topics, and publishers can write to different topics. This flexible framework allows for easy integration of sensor and control processes with hardware components. ROS Indigo for ARM architecture was installed on both ODROID computers. Figure 6.4 shows the software structure used for implementation of the hybrid system. The TIVA node, written in Python, reads the position, velocity, and direction from the TIVA board through a serial USB interface which has already converted the encoder

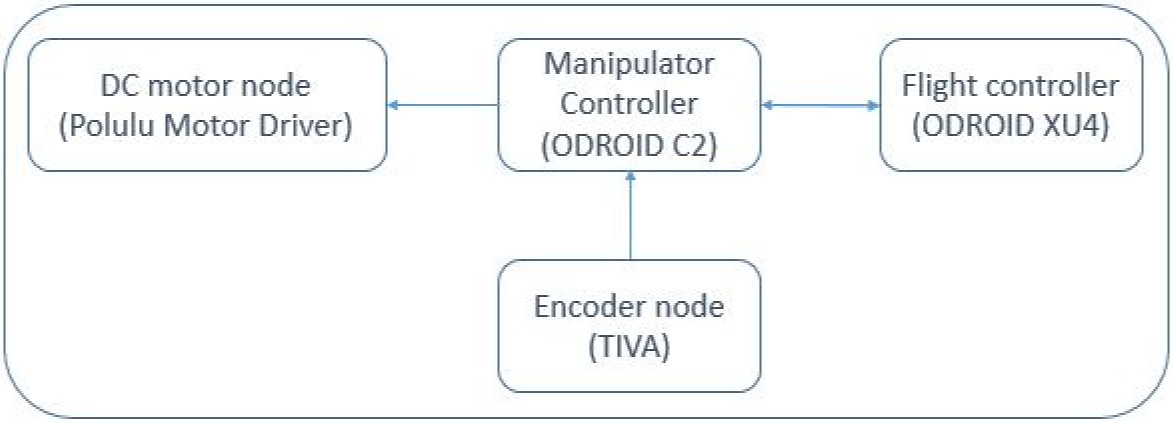


Figure 6.4: Software Diagram; The hardware components are in brackets.

pulses into digital signals. The node then publishes the the position, velocity and direction of each motor to the motor-state topic. The manipulator controller node was developed in Matlab R2016a Simulink which provides a ROS toolbox. The Matlab Coder was used to convert Matlab code to optimized C++ code automatically with minimal code changes. In addition to using a real time OS, process priority, cpu affinity, reserved process memory, and locking virtual address space into RAM in order to ensure real time execution of the controller. This allowed the controller to run at a maximum latency of 112 microseconds. The controller node subscribes to the motor-state topic to read encoder information. The PID control signals generated by the controller are published to the control-signal topic. The motor driver node, written in C, subscribes to the control-signal topic to read the control commands, and writes them to the pololu boards through a USB interface. The Pixhawk reads signals from the teleoperator radio and uses them to control the attitude and altitude of the UAV. The IMU measurements are sent to the Mavros ROS node residing on the ODROID-XU4 for investigation purposes.

Chapter 7

Experimental Results

The experimental work is separated into a thrust experiment and a flight experiment.

7.0.1 Thrust Experiment

The torque-force mapping constant λ derived in Chapter 4 is repeated here for convenience, $\frac{\tau_{M_i}}{F_i} = \frac{b}{k} = \lambda$. An experiment was designed in order to numerically determine the value of λ for the commercial UAV used. This is critical because it is used in the dynamic formulation of the UAV and subsequently in the optimization problem.

The propeller produced force was measured using an ATI Mini40 force/torque sensor combined with a 16-bit DAQ. Measurements were captured at 1 kHz sampling rate and a resolution of 0.01 N. A 2 ms PWM pulse was sent to the built-in electronic speed controller (ESC) of the DJI 4114 PRO 400KV motor. The experiment is shown in Figure 7.1 which starts from rest and ramps up to full throttle after the PWM command was issued. The force averaged from 4s to 10s was 24.37N and since the UAV is modeled as a quadrotor, $T^{\max} = 48.74N$. This experiment shows the advantage of using an octorotor as the commercial UAV.

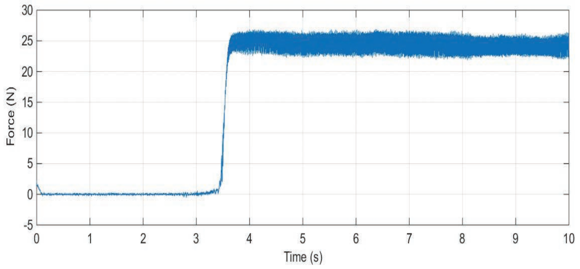


Figure 7.1: Thrust Experiment to determine T^{max}

In order to determine the torque-force mapping constant λ, 60 experiments such as the one shown in Figure 7.1 were performed. The steady-state force vector and torque vector for all experiments are $\boldsymbol{T}_{\text{ss}} \in \mathbb{R}^{60\text{x}1}$ and $\boldsymbol{\tau}_{\text{ss}} \in \mathbb{R}^{60\text{x}1}$ respectively. The ratio is determined using the left pseudo-inverse as follows,

$$\lambda = (\boldsymbol{T}_{\text{ss}}^T \boldsymbol{T}_{\text{ss}})^{-1} \boldsymbol{T}_{\text{ss}}^T \boldsymbol{\tau}_{\text{ss}} \tag{7.1}$$

which takes on the value of 9.06×10^{-4}.

7.0.2 Flight Experiment

The hybrid system is demonstrated through a flight experiment available at `https://youtu.be/SfIunbfv7Uc`. Figure 7.2 shows the MM-UAV in flight where with the manipulator extended. This experiment was carried out to validate that physical operation of the constructed robot obeys the constraints designed in the optimization problem.

The UAV is controlled by a human operator using a 6-channel Spektrum RC transmitter. The manipulator is controlled through the same cubic trajectory consecutively. The third joint is held fixed throughout the experiment. The first and

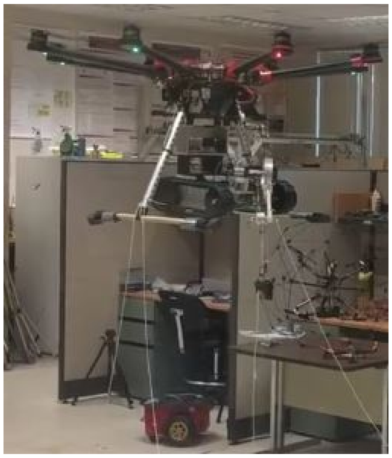


Figure 7.2: Flight Experiment

second joints undergo the same trajectory; the initial angle is 0 (rad), the final angle is $\pi/4$ (rad), the initial angular velocity and final velocity are 0 (rad/s).

The desired trajectory for the first joint and second joint are shown in Figures 7.3 and 7.4, respectively. The manipulator joint errors are shown in Figure 7.5. The PID controller gains were tuned using a trial and error procedure. The two large disturbances that start at 15 seconds into the experiment results from tuning errors with the PID controller. Reduction of this error requires more diligent gain tuning and consistent flight maneuvers by the operator. Another source of error is the tension in the Capstan drive mechanism cables. If there is any cable looseness or cable slippage, the controller will lose track of the correct position and the encoders will no longer track the true position of the motor shafts.

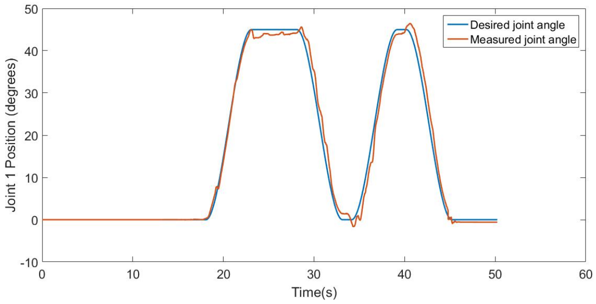


Figure 7.3: Manipulator joint 1 position desired and measured

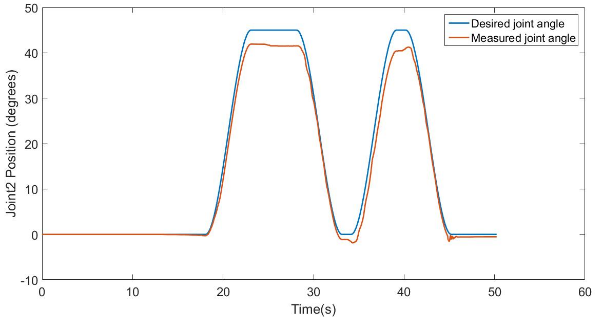


Figure 7.4: Manipulator joint 2 position desired and measured

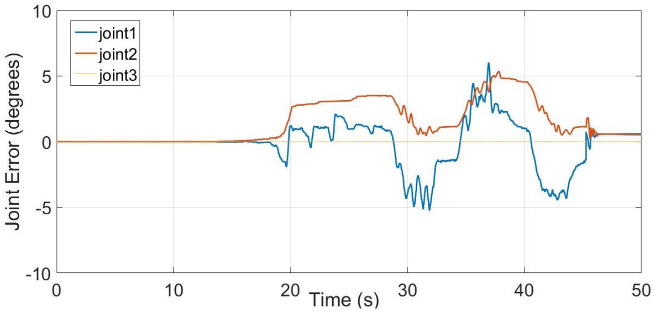


Figure 7.5: Manipulator joint position errors

Chapter 8

Conclusion

This research solved the problem that a hybrid system capable of operating in air and ground environments is unavailable. Moreover, the single-trajectory robotic design optimization limitation which leads to suboptimal robots for general trajectories was also solved.

A novel hybrid aerial/ground mobile manipulator and a novel optimization-based strategy for choosing critical design parameters in such robotic systems were the highlights of this thesis. The system was designed by proposing a general bilevel optimization strategy for choosing critical design parameters of the robot under a wide range of dynamic operating scenarios, which was generalized to other robotic system as well. The design problem was formulated as a mixed-integer bilevel optimization where the main objective was to minimize the overall robot mass. Dynamic models for the manipulator, the MM-UGV and the MM-UAV were used in the formulation of the constraints in the design optimization. The size of manipulator's workspace, actuator force/torque limits, and dynamic tip-over stability under all possible dynamic/static motion scenarios were some of the constraints considered in the optimization. The resulting nonlinearly constrained optimization was solved with a parallel Branch-and-Sandwich to obtain globally optimal design parameters.

A physical prototype of the hybrid aerial/ground mobile robot with a self-latching mechanism was developed and demonstrated in a flight experiment showing that the optimal design satisfies the prescribed design constraints and optimization inputs. Results of Monte Carlo simulations confirmed statistically that the system with the optimal parameters satisfies all the design requirements.

The research also resulted in a conference paper (IROS) which focused on the theoretical implications of the design formulation [8]. Another article to be submitted to the IEEE/ASME Transactions on Mechatronics [9] focuses on the implementation of the hybrid system described herein.

8.1 Future Research

The hybrid robotic system can now serve as a research platform for further contributions. Some directions include:

1. Autonomous coupling/decoupling of the MMUGV with the UAV.
2. Model-based control of the MMUAV.
3. Further hybridization to other environments such as underwater.
4. Considering overheating of the motor in the optimization formulation.
5. Teleoperation of the robotic manipulator.
6. Vision-based grasping.

Bibliography

[1] Pierre-Jean Bristeau, François Callou, David Vissiere, Nicolas Petit, et al. The navigation and control technology inside the ar. drone micro uav. In *18th IFAC world congress*, volume 18, pages 1477–1484, 2011.

[2] Ascending Technologies. Research UAVs, march 2016.

[3] DJI. Dji, march 2015.

[4] AE Jimenez-Cano, J Martin, Guillermo Heredia, Aníbal Ollero, and R Cano. Control of an aerial robot with multi-link arm for assembly tasks. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4916–4921. IEEE, 2013.

[5] Clearpath Robotics. Self-driving vehicles for industry, mar 2016.

[6] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[7] Ros concepts. `http://wiki.ros.org/ROS/Concepts`.

[8] David Findlay, Mohammad Jafarinasab, and Shahin Sirouspour. Optimization-based design of a novel hybrid aerial/ground mobile manipulator. In *Proceedings*

of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 2467–2472. IEEE, 2015.

[9] David Findlay, Mohammad Jafarinasab, and Shahin Sirouspour. Optimization-based design and prototyping of a new hybrid aerial/ground mobile manipulator. To be submitted.

[10] Paul Edward Ian Pounds. *Design, construction and control of a large quadrotor micro air vehicle*. Australian National University, 2007.

[11] Samir Bouabdallah and Roland Siegwart. Towards intelligent miniature flying robots. In *Field and Service Robotics*, pages 429–440. Springer, 2006.

[12] Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer, and Marc Pollefeys. Pixhawk: A system for autonomous flight using onboard computer vision. In *Robotics and automation (ICRA), 2011 IEEE international conference on*, pages 2992–2997. IEEE, 2011.

[13] Gabe Hoffmann, Dev Gorur Rajnarayan, Steven L Waslander, David Dostal, Jung Soon Jang, and Claire J Tomlin. The stanford testbed of autonomous rotorcraft for multi agent control (starmac). In *Digital Avionics Systems Conference, 2004. DASC 04. The 23rd*, volume 2, pages 12–E. IEEE, 2004.

[14] N Guenard, T Hamel, and V Moreau. Dynamic modeling and intuitive control strategy for an" x4-flyer". In *Control and Automation, 2005. ICCA'05. International Conference on*, volume 1, pages 141–146. IEEE, 2005.

[15] Ilan Kroo, Fritz Prinz, Michael Shantz, Peter Kunz, Gary Fay, Shelley Cheng,

Tibor Fabian, and Chad Partridge. The mesicopter: A miniature rotorcraft concept–phase ii interim report. *Stanford University, USA*, 2000.

[16] Samir Bouabdallah, Andre Noth, and Roland Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2451–2456. IEEE, 2004.

[17] Samir Bouabdallah and Roland Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2247–2252. IEEE, 2005.

[18] Samir Bouabdallah and Roland Siegwart. Full control of a quadrotor. In *Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ international conference on*, pages 153–158. IEEE, 2007.

[19] Steven L Waslander, Gabriel M Hoffmann, Jung Soon Jang, and Claire J Tomlin. Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3712–3717. IEEE, 2005.

[20] Abraham Bachrach, Ruijie He, and Nicholas Roy. Autonomous flight in unknown indoor environments. *International Journal of Micro Air Vehicles*, 1(4):217–228, 2009.

[21] Stephanie Gil, Mac Schwager, Brian J Julian, and Daniela Rus. Optimizing communication in air-ground robot networks using decentralized control. In *Robotics*

and Automation (ICRA), 2010 IEEE International Conference on, pages 1964–1971. IEEE, 2010.

[22] Dongjun Lee, Changsu Ha, and Zhiyuan Zuo. Backstepping control of quadrotor-type UAVs and its application to teleoperation over the internet. In *Intelligent Autonomous Systems 12*, pages 217–225. Springer, 2013.

[23] ChangSu Ha, Zhiyuan Zuo, Francis B Choi, and Dongjun Lee. Passivity-based adaptive backstepping control of quadrotor-type UAVs. *Robotics and Autonomous Systems*, 62(9):1305–1315, 2014.

[24] Roland Brockers, Sara Susca, David Zhu, and Larry Matthies. Fully self-contained vision-aided navigation and landing of a micro air vehicle independent from external sensor inputs. In *SPIE Defense, Security, and Sensing*, pages 83870Q–83870Q. International Society for Optics and Photonics, 2012.

[25] Abdel-Razzak Merheb, Hassan Noura, and Francois Bateman. Passive fault tolerant control of quadrotor uav using regular and cascaded sliding mode control. In *Control and Fault-Tolerant Systems (SysTol), 2013 Conference on*, pages 330–335. IEEE, 2013.

[26] Matteo Fumagalli, Roberto Naldi, Alessandro Macchelli, Francesco Forte, Arvid QL Keemink, Stefano Stramigioli, Raffaella Carloni, and Lorenzo Marconi. Developing an aerial manipulator prototype: Physical interaction with the environment. *Robotics & Automation Magazine, IEEE*, 21(3):41–50, 2014.

[27] Elizabeth Bevan, Thane Wibbels, Blanca MZ Najera, Marco AC Martinez, Laura AS Martinez, Francisco I Martinez, Javier M Cuevas, Tiffany Anderson,

Amy Bonka, Mauricio H Hernandez, et al. Unmanned aerial vehicles (uavs) for monitoring sea turtles in near-shore waters. *Marine Turtle Newsletter*, (145):19, 2015.

[28] Hyon Lim, Jaemann Park, Daewon Lee, and H Jin Kim. Build your own quadrotor: Open-source projects on unmanned aerial vehicles. *Robotics & Automation Magazine, IEEE*, 19(3):33–45, 2012.

[29] Teal Group Corporation. Press release, mar 2015.

[30] Gareth J Monkman, Stefan Hesse, Ralf Steinmann, and Henrik Schunk. *Robot grippers*. John Wiley & Sons, 2007.

[31] Daniel Mellinger, Quentin Lindsey, Michael Shomin, and Vijay Kumar. Design, modeling, estimation and control for aerial grasping and manipulation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2668–2673. IEEE, 2011.

[32] Agisilaos G Zisimatos, Minas V Liarokapis, Christoforos I Mavrogiannis, and Kostas J Kyriakopoulos. Open-source, affordable, modular, light-weight, underactuated robot hands. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3207–3212. IEEE, 2014.

[33] Paul EI Pounds, Daniel R Bersak, and Aaron M Dollar. Stability of small-scale uav helicopters and quadrotors with added payload mass under pid control. *Autonomous Robots*, 33(1-2):129–142, 2012.

[34] Daniel Mellinger, Michael Shomin, Nathan Michael, and Vijay Kumar. Cooperative grasping and transport using multiple quadrotors. In *Distributed autonomous robotic systems*, pages 545–558. Springer, 2013.

[35] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. Construction of cubic structures with quadrotor teams. *Proc. Robotics: Science & Systems VII*, 2011.

[36] Quentin Lindsey, Daniel Mellinger, and Vijay Kumar. Construction with quadrotor teams. *Autonomous Robots*, 33(3):323–336, 2012.

[37] Suseong Kim, Seungwon Choi, and H Jin Kim. Aerial manipulation using a quadrotor with a two dof robotic arm. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4990–4995. IEEE, 2013.

[38] Marin Kobilarov. Trajectory control of a class of articulated aerial robots. In *Proceedings International Conference on Unmanned Aircraft Systems*, pages 958–965, 2013.

[39] Matko Orsag. Mobile manipulating unmanned aerial vehicle (mm-uav): Towards aerial manipulators. *system*, 13:14.

[40] Hideyuki Tsukagoshi, Masahiro Watanabe, Takahiro Hamada, Dameitry Ashlih, and Ryuma Iizuka. Aerial manipulator with perching and door-opening capability. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4663–4668. IEEE, 2015.

[41] Open Source Robotics Foundation. Gazebo, mar 2016.

[42] Naval Studies Board et al. *Autonomous vehicles in support of naval operations*. National Academies Press, 2005.

[43] Richard Bloss. Unmanned vehicles while becoming smaller and smarter are addressing new applications in medical, agriculture, in addition to military and security. *Industrial Robot: An International Journal*, 41(1):82–86, 2014.

[44] Brian M Yamauchi. Packbot: a versatile platform for military robotics. In *Defense and Security*, pages 228–237. International Society for Optics and Photonics, 2004.

[45] François Michaud, Dominic Létourneau, Martin Arsenault, Yann Bergeron, Richard Cadrin, Frédéric Gagnon, Marc-Antoine Legault, Mathieu Millette, Jean-François Paré, Marie-Christine Tremblay, et al. Azimut, a leg-track-wheel robot. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 3, pages 2553–2558. IEEE, 2003.

[46] Shigeo Hirose, Edwardo F Fukushima, Riichiro Damoto, and Hideichi Nakamoto. Design of terrain adaptive versatile crawler vehicle helios-vi. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 3, pages 1540–1545. IEEE, 2001.

[47] iRobot. Warrior 710 kobra, mar 2016.

[48] Peter Wells and Dan Deguire. Talon: A universal unmanned ground vehicle platform, enabling the mission to be the focus. In *Defense and Security*, pages 747–757. International Society for Optics and Photonics, 2005.

[49] Pinhas Ben-Tzvi, Andrew A Goldenberg, and Jean W Zu. Design, simulations and optimization of a tracked mobile robot manipulator with hybrid locomotion

and manipulation capabilities. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2307–2312. IEEE, 2008.

[50] Miomir Vukobratović and Branislav Borovac. Zero-moment pointthirty five years of its life. *International Journal of Humanoid Robotics*, 1(01):157–173, 2004.

[51] EG Papadopoulos and Daniel A Rey. A new measure of tipover stability margin for mobile manipulators. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 4, pages 3111–3116. IEEE, 1996.

[52] Shady Ali, A Moosavian, and Khalil Alipour. Stability evaluation of mobile robotic systems using moment-height measure. In *Robotics, Automation and Mechatronics, 2006 IEEE Conference on*, pages 1–6. IEEE, 2006.

[53] Philip R Roan, Aaron Burmeister, Amin Rahimi, Kevin Holz, and David Hooper. Real-world validation of three tipover algorithms for mobile robots. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4431–4436. IEEE, 2010.

[54] Kazuhiro Saitou, Kazuhiro Izui, Shinji Nishiwaki, and Panos Papalambros. A survey of structural optimization in mechanical product development. *Journal of Computing and Information Science in Engineering*, 5(3):214–226, 2005.

[55] Ole Sigmund. Topology optimization: a tool for the tailoring of structures and materials. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 358(1765):211–227, 2000.

[56] Lelai Zhou, Shaoping Bai, and Michael Rygaard Hansen. Design optimization

on the drive train of a light-weight robotic arm. *Mechatronics*, 21(3):560–569, 2011.

[57] Marcus Pettersson and Johan Olvander. Drive train optimization for industrial robots. *Robotics, IEEE Transactions on*, 25(6):1419–1424, 2009.

[58] Manja Kirćanski. Kinematic isotropy and optimal kinematic design of planar manipulators and a 3-dof spatial manipulator. *The International journal of robotics research*, 15(1):61–77, 1996.

[59] Kourosh E Zanganeh and Jorge Angeles. Kinematic isotropy and the optimum design of parallel manipulators. *The International Journal of Robotics Research*, 16(2):185–197, 1997.

[60] KM Digumarti, C Gehring, S Coros, J Hwangbo, and R Siegwart. Concurrent optimization of mechanical design and locomotion control of a legged robot. *Mobile Service Robotics: CLAWAR 2014*, 12:315, 2014.

[61] Bernard Etkin and Loyd Duff Reid. *Dynamics of flight: stability and control*, volume 3. Wiley New York, 1996.

[62] Gabriel Hoffmann, Haomiao Huang, Steven Waslander, and Claire Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6461, 2007.

[63] Rached Dhaouadi and A Abu Hatab. Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework. *Advances in Robotics & Automation*, 2(2):1–7, 2013.

[64] Anthony M Bloch, J Baillieul, P Crouch, and J Marsden. Nonholonomic mechanics and control. interdisciplinary applied mathematics 24. *Systems and Control*, 2003.

[65] Zeynep H Gümüş and Christodoulos A Floudas. Global optimization of nonlinear bilevel programming problems. *Journal of Global Optimization*, 20(1):1–31, 2001.

[66] Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256, 2007.

[67] Polyxeni-M Kleniati and Claire S Adjiman. A generalization of the branch-and-sandwich algorithm: From continuous to mixed-integer nonlinear bilevel problems. volume 72, pages 373–386. Elsevier, 2015.

[68] Polyxeni-M Kleniati and Claire S Adjiman. Branch-and-sandwich: a deterministic global optimization algorithm for optimistic bilevel programming problems. part ii: Convergence analysis and numerical results. volume 60, pages 459–481. Springer, 2014.

[69] Polyxeni-Margarita Kleniati and Claire S Adjiman. Branch-and-sandwich: a deterministic global optimization algorithm for optimistic bilevel programming problems. part i: Theoretical development. volume 60, pages 425–458. Springer, 2014.

[70] Teodor Gabriel Crainic. Parallel branch-and-bound algorithms. pages 1–28. John Wiley & Sons, Inc., 2006.

[71] Lucio Barreto and Michael Bauer. Parallel branch and bound algorithm - a

comparison between serial, openmp and mpi implementations. volume 256, page 012018, 2010.

[72] Maxon motor products catalogue. http://www.maxonmotor.com/maxon/view/catalog/.

[73] Thunderpowerrc catalogue. http://www.thunderpowerrc.com/Products.

[74] Jonas Sticha. Validating the real-time capabilities of the ros communication middleware, 2014.