Controlled Manipulation Using Autonomous Aerial Systems

by

Manohar B. Srikanth

M.Sc.(Engg.), Computer Systems, Indian Institute of Science (2007)
B.E., Electronics & Communication, University of Mysore (2001)

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2013

Author: ....
Department of Mechanical Engineering
November 15, 2012

Certified by : ....
Anuradha M. Annaswamy
Senior Research Scientist, Department of Mechanical Engineering
Thesis Supervisor

Accepted by :..
David E. Hardt
Chairman, Department Committee on Graduate Students

Controlled Manipulation Using Autonomous Aerial Systems

by

Manohar B. Srikanth
November 15, 2012

Submitted to the Department of Mechanical Engineering

The main focus of the thesis is to design and control Autonomous Aerial Systems, also referred to as Unmanned Aerial Vehicles (UAVs). UAVs are able to hover and navigate in space using the thrust forces generated by the propellers. One of the simplest such vehicles that is widely used is a Quadrotor. While UAVs have been predominantly used for "fly and sense" applications, very few investigations have focused on using them to perform manipulation by contact. The latter is challenging because of the dual goal of performing manipulation and maintaining stable flight. Because Quadrotors can quickly reach a location, their ability to manipulate can be impactful in many scenarios. While efficient flight control of Quadrotor has been an active research area, using Quadrotor to perform manipulation is novel and challenging. In this thesis, a range of Quadrotor designs and control strategies are proposed in order to carry out autonomous manipulation of objects.

We first derive a dynamic model of the Quadrotor that accounts for the presence of contact, object dynamics and kinematics. To improve manipulation performance, a passive light-weight end-effector interface between the Quadrotor and the object is proposed. The complexity of the dynamics is systematically reduced by making certain assumptions. The resulting dynamic model is divided into nonlinear subsystems on the basis of their degrees of freedom, for each of which separate controllers are designed. An efficient docking approach is proposed that permits fast and aggressive docking, even at very high speeds. Because a single Quadrotor UAS is limited in manipulation capability, a multi Quadrotor cooperative manipulation scheme is proposed.

Control strategies are proposed to deal with kinematic and parametric uncertainties. A manipulation scheme to open a door with unknown hinge location is proposed. A nonlinear adaptive controller is implemented to perform efficient tracking in the presence of parametric uncertainty.

In order to improve robustness to accidental contacts, a novel flexible Quadrotor, denoted as ParaFlex, is designed. The advantages of ParaFlex over a rigid Quadrotor are demonstrated. A Simulation, Test and Validation Environment (STeVE) is developed to facilitate smooth and efficient transition from design process to simulation to experiments.

Thesis Supervisor & Committee Chair:
A. M. Annaswamy, Senior Research Scientist, Department of Mechanical Engineering, MIT.

Thesis Committee:
Prof. Jean-Jacques Slotine, Professor, Department of Mechanical Engineering, MIT
Prof. Kamal Youcef-Toumi, Professor, Department of Mechanical Engineering, MIT
Prof. Nicolas Roy, Associate Professor, Department of Aeronautics and Astronautics, MIT
Dr. Eugene Lavretsky, Senior Technical Fellow, Boeing Research and Technology, CA.

Acknowledgments

First and foremost, I would like to express my deep and heartfelt gratitude to my research advisor Dr. Anuradha M. Annaswamy, who has been an incredible source of support, knowledge, and inspiration throughout the course of my stay at MIT. Right from the beginning, Dr. Annaswamy encouraged novel ideas, even if they deviated from the main stream of work in the laboratory. As openly as she accepted these proposed ideas, she critically forged them to ensure that they really qualified to be challenging, and important. Given that I come from a non-controls background, it was through her able guidance, that I could quickly put myself on the tracks of controls, and at the same time, was able to use my previous skill-set. I consider myself lucky to have her as my Ph.D. mentor, and hope to build further on the concrete scientific foundation inherited by being a part of her group.

Mandayam A. Srinivasan has provided tremendous support since the beginning of my stay at MIT. His suggestions, the resources available as well continuous and open access to the Touch Lab have been a significant help.

I take this opportunity to thank my thesis committee members: Prof. Jean-Jacques Slotine, Prof. Kamal Youcef-Toumi, Prof. Nicolas Roy, and Dr. Eugene Lavretsky for their valuable inputs, which have collectively helped me to shape my research better.

I would like to thank Prof. Slotine, who always found time to provide me with guidance and inputs in the domain of robotics, and for teaching me one of the key courses in non-linear control design.

Prof. Youcel-Toumi's course on modeling, and his time to time suggestions regarding my research, have been extremely helpful for my thesis. Thank you, Sir, for taking time out of your busy schedule to guide me as and when I needed.

My sincere thanks to Prof. Nicolas Roy for giving me the opportunity to participate in his group meetings, which were indeed very helpful, especially during my early work. I would also like to thank him for letting me use the experimental facility, which was critically important for my project.

Support and advice by Dr. Eugene Lavretsky as an industry expert, was distinctly helpful. It provided me with the necessary check-points to validate the practical relevance of my ideas and methods, and in effect, boosted my confidence.

I am truly grateful to my labmates for essential tips on a wide spectrum of topics starting with how to crack an exam, to the best restaurants in the neighborhood! Zac Dydek, Jinho Jang, and Yildidray Yiadiz, Travis Gibson, Megumi Matsustani, Benjamin Jenkins, Amith Somnath, Yoav Sharon and, Damoon S., you all have been a great company!

I wish I could write a page about each of the visiting students, who worked with me. Their help in carrying out the actual experiments, as well as exchange of ideas, is sincerely acknowledged. Many thanks to Frieder Whittmann, Albert Soto, Erica Nwankwo, Kenneth Guttierez, and Eduardo Steed.

I am fortunate to have a wide circle of friends here at MIT. The time spent with them- having conversations ranging from heated discussions on contemporary topics to light-hearted chitchat over umpteen cups of coffee, as well as loads of fun activities- has always been quite exciting and refreshing. All those moments will be cherished.

I would specifically like to mention my friends and colleagues at the Student Art Association, The Tech newspaper, and Technique club of MIT, with whom, I could discover myself as a photographer, and derive enormous creative pleasure through our joint activities and workshops. Thank you so much, pals!

Last but not the least; I would like to thank my family members and well-wishers for their constant encouragement and support. My parents, and my wife- Monali, have been great impetuses behind my efforts, and I feel blessed to have them.

I especially want to thank my wife, Monali Manohar, who has been a tremendous driving force, especially during the second half of my doctoral work.

Thank You!!

Table of Contents

Table of Figures

1 Introduction

1.1 The Big Picture

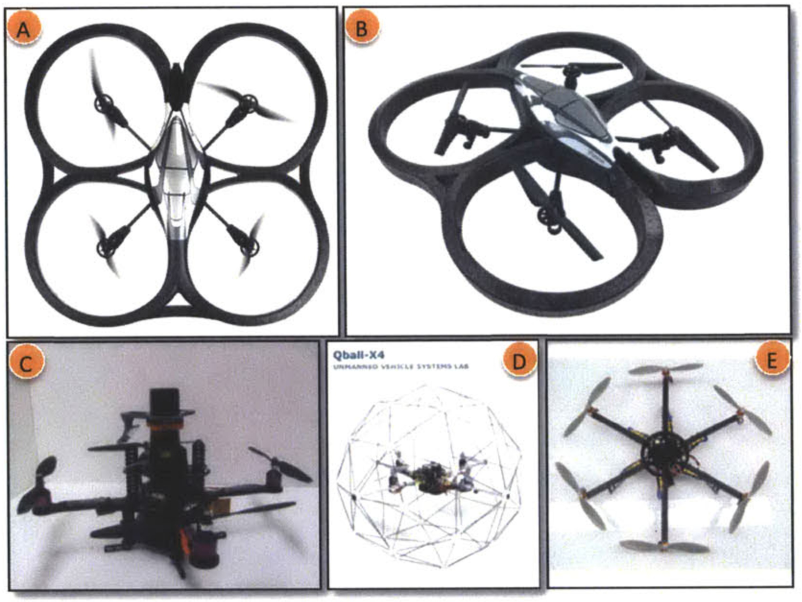

Figure 1.1: Examples of Autonomous Aerial Systems that can perform Vertical Takeoff and Landing. (A,B) Parrot AR Drone [1], (C) Ascending Tech Quadrotor, (D) Quansar QBall4, Quadrotor with a protective cage, (E) ArduCopter with six rotors [2].

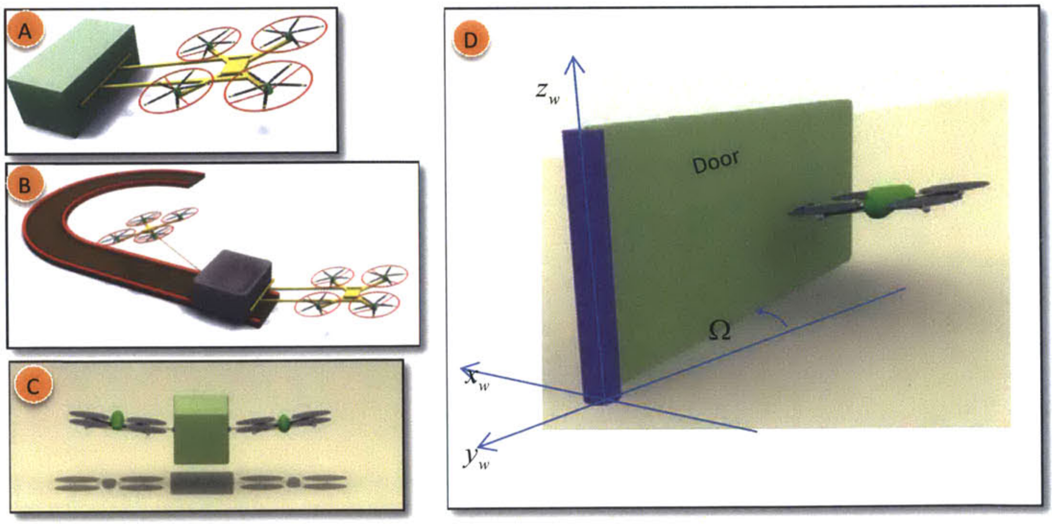

Figure 1.2: Examples of manipulation using Autonomous Aerial Systems (AAVs) or Unmanned Aerial Vehicles (UAVs). (A) A Quadrotor UAV pushing an Object. (B,C) Two Quadrotor UAVs pushing-pulling an Object, (D) Quadrotor UAV opening a door.

The fundamental focus of this thesis is to technologically empower the aerial vehicles, such as the ones shown in Figure 1.1, to perform manipulation tasks as shown in Figure 1.2. Autonomous Aerial Vehicles

(AAVs), also referred to as Unmanned Aerial Vehicles (UAVs) shown in Figure 1.1, are able to hover and navigate in space using the thrust forces generated by the propellers. While AAVs/UAVs have been predominantly used for "fly and sense" applications, they can also be used for "contact" based applications. With intelligent redesign, the thrust forces can also be utilized to apply forces to a target Object, thus inducing a motion. The UAVs first fly close to the Object and safely docks to it. While maintaining stable flight, a controlled amount of force is applied to the Object. As the Object starts to move, the UAV continuously ensures flight stability as well as docking stability. This is illustrated in Figure 1.3. While a single UAV is limited in the manipulation capability, multiple UAVs can cooperatively perform more complex manipulation tasks, as shown in Figure 1.2(B,C).

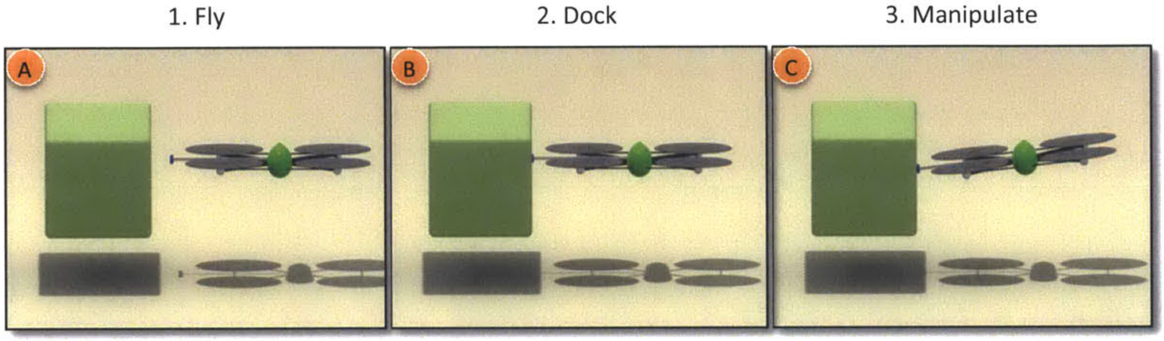

Figure 1.3: The basic approach in manipulation. (A) The UAV flies to close proximity of the target Object. (B) UAV docks to the object by means of an end-effector, and (C) UAV performs manipulation by applying a force. In (A) the UAV is in flight mode with flight dynamics at play, and in (B,C) the UAV is in manipulation mode where both flight and manipulation dynamics are at play. After the manipulation process, the UAV undocks and enter into flight mode as in (A).

While efficient flight control of UAVs has been an active research area, using them to perform manipulation by contact is novel and challenging because of the dual goal: perform manipulation and maintain stable flight. The focus of this thesis is the development of a number of Quadrotor designs and control strategies in order to carry out autonomous manipulation.

In the proposed approach, the UAV uses a lightweight, passive end-effector mechanism to interface with the Object, as shown in Figure 1.3(C). As a result, the dynamics of the UAV is coupled with that of the Object with several free degrees of freedom. Careful design choices have to be made to ensure that the aerodynamic coupling is minimized, which otherwise would complicate the dynamics. A nominal flight controller fails to stabilize the UAV during a contact event. Designing stable and efficient manipulation schemes requires a thorough understanding of the new dynamics. Unlike in free-flight scenarios, the UAV has to perform more precise maneuvers in order to carry out the manipulation task, thus placing stringent requirements on controller performance. When multiple UAVs manipulate a single object, the resulting complex dynamics has to be systematically simplified and modularized in order to make control design possible. When some of the system parameters are unknown, carryout manipulation becomes inefficient and sometimes impossible. Adaptive control schemes have to be designed to address this issue. While intentional contact is used to perform manipulation, accidental contacts can occur during a free-flight operation and may potentially damage the UAV. In order to robustify the UAV to accidental contacts, control based solution can be obtained, but this alone will not suffice. Therefore, a UAV design solution has to be provided. To address this, a flexible Quadrotor design, designated as ParaFlex, is proposed. These discussions imply that new methods have to be developed for the design and development of novel Quadrotors that are able to successfully manipulate objects while maintaining stable and robust flight. This is the focus of the thesis.

We first derive a dynamic model of the Quadrotor that accounts for the presence of contact, Object dynamics and kinematics. To improve manipulation performance, a passive light-weight end-effector interface between the Quadrotor and the Object is proposed. The complexity of the dynamics is systematically reduced by making certain assumptions. The resulting dynamic model is divided into nonlinear subsystems on the basis of their degrees of freedom, for each of which separate controllers are designed. An efficient docking approach is proposed that permits fast and aggressive docking, even at very high speeds. Because a single Quadrotor UAS is limited in manipulation capability, a multi Quadrotor cooperative manipulation scheme is proposed.

Several control strategies are proposed to deal with uncertainty, particularly in the presence of kinematic and parametric uncertainties. A Quadrotor manipulation and control scheme to open an unknown door is designed, where the door kinematics is assumed to be unknown. When the Quadrotor's parameters are unknown, a nonlinear adaptive controller is implemented to perform efficient trajectory tracking.

While establishing contact was intentional in the above manipulation tasks, contacts can occur accidentally that may lead to a crash. In order to improve robustness to accidental contacts, a novel flexible Quadrotor design is proposed to further improve the robustness.

Figure 1.4: Illustration: A disaster stuck place (Oslo, Norway, 22 July, 2011). Overlaid on top are the Aerial Vehicles, equipped with manipulation capability, can readily fly and reach the top floor where search and rescue operation can be executed.

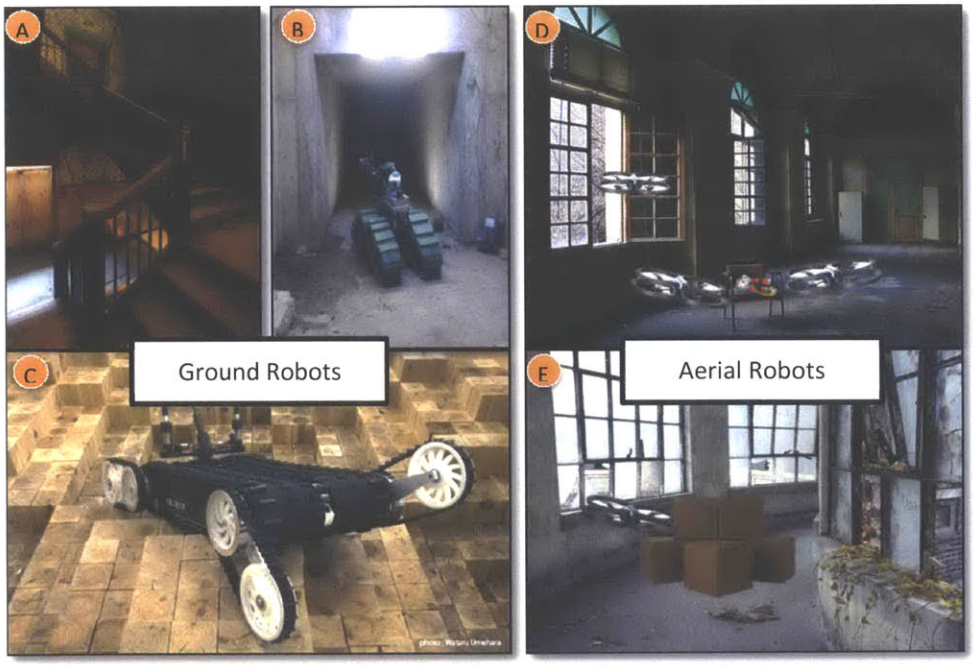

Figure 1.5: Performance comparison of Ground Robot with Aerial Robots during a search and rescue mission. In (A,B,C), a ground robot has to traverse a rough and unknown terrain, and may take prohibitively long time to reach the location. In (D,E) a set of Quadrotor UAV reach the location much faster than the ground robots. Because they can perform manipulation, they cooperatively open the window and let themselves inside. They then perform the necessary manipulation tasks, such as moving hazardous items out of the way, or clearing the space.

A simulation, test and validation environment (STeVE) is developed to facilitate smooth transition from design to simulation to experiments. STeVE consists of several software and hardware tools, some of which are developed in house, and includes Hardware-in-the-loop test facility and actual flight systems.

In addition to validation of the Quadrotor designs and control strategies using STeVE, the efficacy of various manipulation and control schemes is demonstrated through extensive flight experiments.

1.2 Potential Applications

With the ability to fly and manipulate objects, several novel applications can be proposed. The ability to fly permits the UAVs to reach a target location with speed and efficiency. The ability to manipulate while being in the air enables them to perform tasks which otherwise would be extremely inefficient or impossible. Consider the scenario shown in Figure 1.4. A number of UAVs that are manipulation capable can readily fly to the top floor of the building where a search and rescue mission has to be performed. If for instance a window or a door needs to be opened to enter, the AAVs can perform such an operation. The main advantage here is the time to reach the location, which is in the order seconds. If a ground robot, like the ones shown in Figure 1.5 (A,B,C) have to reach the location, they may take several minutes to reach because of traversability issues. Once inside the building, UAVs can perform various tasks, some examples shown in Figure 1.5 (D,E).

Another example is remote assembling, where a set of UAVs can move and assemble deployable tents before humans could reach there. UAVs can perhaps help disabled open doors or clear their path as they navigate an unknown environment. They can also be used for contact based inspection or repair, for example, ship or building exterior inspection. They can also aid other larger systems, for example a heavy crane can lift a large object while smaller UAVs can push the object to precision maneuver.

The proposed approach utilizes dynamics contacts in order to dock and apply forces to another object. Recently, several groups have used other method to perform manipulation. The following section summarizes these methods and compares them to the methods proposed in this thesis.

1.3 Prior Art

In this section, we start with prior art on UAVs in general and then present related work on utilizing UAVs for manipulation tasks.

There has been extensive research on the Autonomous Aerial Vehicles, commonly referred as Unmanned Aerial Vehicles (UAVs). Quadrotor or Multi rotor based UAVs that have the capability of hover have particularly seen more interest in recent research. Modern day mini UAVs can be battery operated for several minutes, and weigh anywhere from around few grams to few kilograms. With wireless capability, onboard sensing and powerful onboard computation capability, they are now being extensively used indoors, urban spaces, and outdoors. As of today, their main limitation are the flight time, payload capacity, and global state estimation [cite]. With the battery technology improving every day, and progress in state estimation research, the UAVs may be much more ubiquitous in the coming days. The original idea of a Quadrotor was proposed in early 19th century [3]. Some of the early work on mini battery operated Quadrotors [4-6] focused on understanding the dynamics and proposed control designs. Approximation of the dynamics is often necessary in order to design a controller and therefore it is important to understand the relative importance of various components of the dynamics. For instance, in most cases the aerodynamics effects and propeller nonlinearities are ignored. [7-11] provides an extensive coverage of state of the art in UAV design, modeling and control.

While a number a research groups have focused on flight controls, only a handful of research groups have investigated the idea of utilizing UAVs for manipulation. GRASP Laboratory at University of Pennsylvania [12, 13] proposed two main methods for object manipulation using UAVs, presented in Figure 1.6. Multiple Quadrotors are connected to a single Object using cable as shown in Figure 1.6(A). An optimum trajectory is designed for each of the Quadrotor in order to carry the object [cite paper]. Although this method is well suited for pick and place task, the method requires that the cables are firmly connected before the start of manipulation, and cables have to be detached after the manipulation, demanding additional mechanism or human assistance. Another method proposed by the same group, shown in Figure 1.6(B), is to use active grippers on the Quadrotor. The gripper is driven by servo motor and can grasp an object of suitable size, weight and shape. [cite the paper] presents the control algorithms, and assembling algorithms to build a structure which is shown in Figure 1.6(C). Although quite useful in several situations, several drawbacks of this system prohibit general applicability. Firstly, the active gripper is an added weight and consumes power. For a UAV that weighs about 0.5kg, the typical payload capacity is approximately 40% (lift capacity of around 200grams with a reasonable flight time of ~15mins). The addition of the gripper (~50-100grams) reduces to the payload capacity by a significant amount. The remaining payload capacity, of the order of 20% of the UAV weight, limits the kinds of objects that can be lifted. Further, the object should have a matching feature for the gripper to obtain a successful grasp, which is very restrictive. Lastly, during lifting, the object should be grasped at a point closer to the center of gravity. Otherwise, the off-center mass has to be countered by a torque from the UAV, which is already a torque-limited system. From the dynamics point of view, once grasped, the UAV and the Object forms a rigid body with a new mass and inertia properties resulting in a relatively straightforward control design. A minor point to add is that during picking process, the UAV is directly above the object, which means the air flow of the UAV is significantly disturbed by the shape of the Object. Control design has to account for the modified aerodynamics in order to produce corrective inputs.

Figure 1.7: (A) shows the work done at Yale University [14, 15], where an outdoor helicopter is equipped with a gripper to pick up an object. The approach is similar to that of Figure 1.6(B) [12] and suffers from the drawbacks which have been listed above. A minor point to note is that, because this is a large outdoor helicopter, picking an object on ground can be very challenging because of the ground effect.

Work done at ETHZ [16-18] , shown in Figure 1.7: (B) utilizes modular UAVs to perform self-docking to become larger UAV structure. Although UAV is in contact, the goal is to create a larger UAV and not manipulation.

[16] proposes using Quadrotors to perform juggling with a ball, as shown in Figure 1.7: (C). The position of the ball is tracked and predicted. An "impact point" is estimated where the Quadrotor needs to move to pitch the ball up. The Quadrotor is commanded to arrive at the impact point with a desired velocity such that the next hit of the ball produces a desired motion of the ball. [18] proposes using Quadrotors (equipped with a gripper) to assemble a structure with bricks. Again, the idea is similar to that shown in Figure 1.6(A).

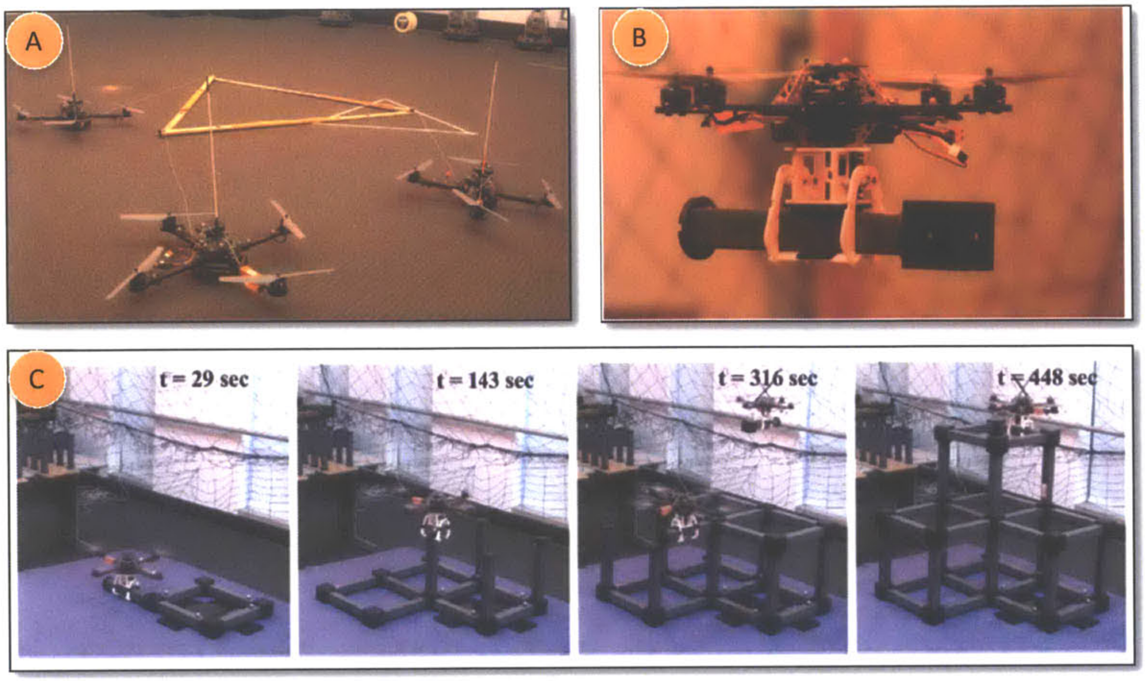

Figure 1.6: AAV based manipulation carried out at GRASP Lab of UPenn [12, 13, 19-21]. (A) Three Quadrotors cooperatively carry a payload using cables are the interface. (B) A Quadrotor is equipped with a gripper that can grasp an object. (C) Using gripper, the Quadrotor assembles the individual pieces to form a structure.

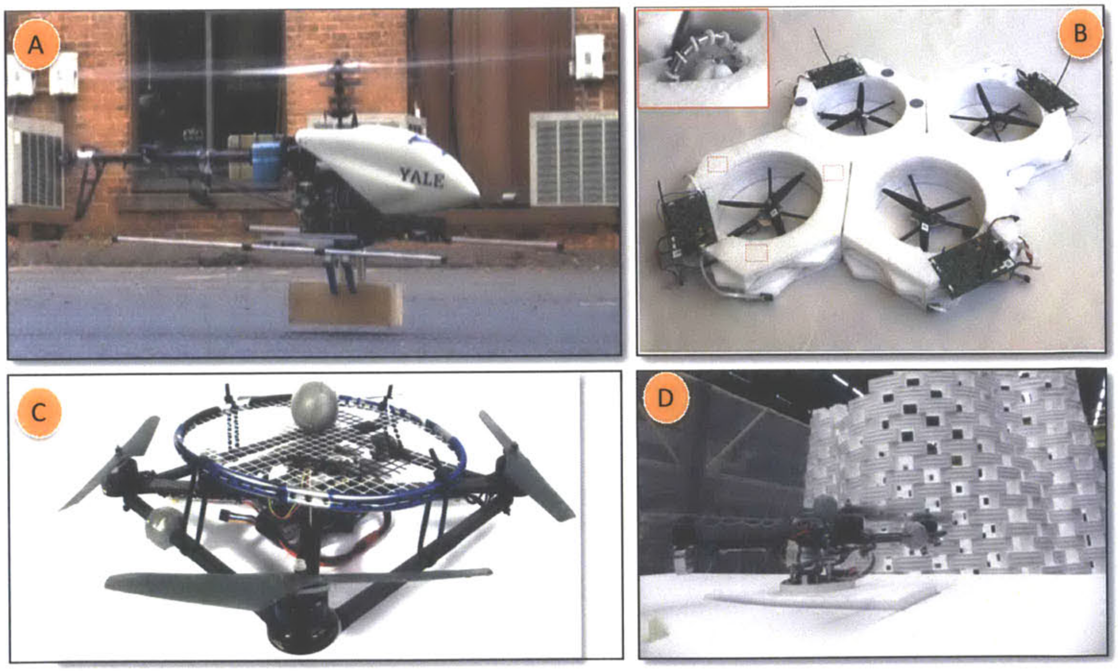

Figure 1.7: (A) A helicopter with gripper designed at Yale to pick and carry a payload [15]. (B) A group of aerial vehicles designed at ETHZ to dock with each other to form a larger platform [17, 18]. (C) A ball juggling Quadrotor designed at ETHZ [16].

1.4 Key Features of the Thesis Approach

The approach used is one where the UAV contacts the object by docking and manipulates it by a combination of push forces. The first step therefore, is the selection of a suitable end-effector that facilitates safe and efficient docking and accurate and agile manipulation. The end-effector also dictates the dynamics of the overall UAV and hence the control design. Aerodynamics that enters the picture when the UAV is in close proximity to the object is also a factor. The aerodynamic effects and the selection of the most suitable end-effector are discussed in Chapter 2.

The next component of our approach is the derivation of the underlying dynamics. As this is significantly different in the case when the UAV is in free-flight versus when it is in contact with the object, we separately derive the dynamic models in both of these cases. In each, case, since the system is underactuated, we propose methods to group the dynamics into two sets, one which describes the states that influence manipulation, and those do not. These models are used both for deriving contact stability while docking and during manipulation, and for deriving control strategies for the manipulation tasks themselves. This is presented in Chapter 3.

The problem of manipulation with a single UAV is first discussed. Insights from this case are used to address manipulation using two UAVs. One of the main bottlenecks in manipulation is slow docking. We propose an aggressive docking strategy where the Quadrotor can impact an object while maintaining stable flight. This strategy is also helpful when there is an uncertainty in object's location. Chapter 4 discusses manipulation schemes using a single and two Quadrotors.

We propose two manipulation schemes when the system parameters are partially known. In the first case, we assume the object parameters, like mass, are unknown and in the second case we assume the Object kinematic constrains are unknown. The latter is illustrated through tasks such as Quadrotor opening a door whose hinge location is unknown, as well as manipulation of an object with a more arbitrary kinematic constraint. Chapter 5 discusses the control solutions in the presence of uncertainty.

To address accidental contacts, and to further improve robustness, we design a flexible Quadrotor, denoted as *ParaFlex*. The main feature of Paraflex is a rigid frame with flexible joints, which allows a safe deformation of the vehicle with the on-set of collisions. This is addressed in Chapter 6, where the underlying dynamics, its response to collisions, and advantages over a rigid Quadrotor are discussed.

In all of the above cases, the proposed modeling approach and control schemes are validated using realistic numerical simulation followed by flight tests wherever possible. An integrated suite of tools is developed to facilitate simulation, hardware-in-the-loop tests and flight tests. This setup is denoted as Simulation Test and Validation Environment, STeVE for short. STeVE consists of various – off-the-shelf and in-house – developed software and hardware components. The main advantage of STeVE is that it enables smooth transition from conceptual design to simulation to flight experiments. The details of STeVE are presented in Chapter 2. The simulation and experimental results for each of the previously mentioned manipulation schemes are presented in the relevant chapters.

1.5 Advantages of the Proposed Approach

The proposed method utilizes dynamic contacts and performs push-and-pull operation only. Also it does not involve air lifting. Many of the tasks in real-world can be accomplished using a combination of pushing and pulling. Example tasks are the opening of a door or pushing a cart. Because the contact is established dynamically, it neither requires preparatory work like cable attachment nor does it require specific features to grasp on to. Moreover, since we do not lift the object, the payload capacity is virtually unlimited. For instance, a Quadrotor weighing 0.5kg can typically push with a 2N force. Object weighing 10s of kilograms can be easily pushed with such forces. The only limiting factor, however, is the stick friction, which can be dealt to some degree using impact manipulation. Multiple UAVs can cooperatively push-pull an object to achieve higher degree of freedom manipulation and producing larger collective forces and torques. As will be shown later, masses of up to 4Kgs are easily manipulated using a low cost Quadrotor weighing only 400grams. The push operation is power efficient because the UAV virtually consumes almost the same amount of power as it consumes to hover.

Another advantage of our approach is due to the fact that the UAV is on the side of the Object as opposed to being on the top. As a result, the air-flow of the UAV is undisturbed. In addition, the end-effector interface design proposed in this thesis is a passive, light-weight (~10grams or 2% of UAV Weight) mechanism, which translates to significantly more flight time.

Also, the methods proposed in this thesis can also be utilized for applications requiring air lifting by having one powerful UAV lift the Object using a cable system with several light weight UAVs performing precise manipulation using the approach of this thesis.

1.6 Thesis Contributions

- The thesis presents the selection of the nature of manipulation scheme using UAVs through a combination of free-flight and docking. Through docking and application of controlled forces, the thesis presents multi-UAV, multi-degree of manipulation schemes.
- The thesis proposes design of passive, light-weight end-effector that facilitates stable docking and efficient manipulation of Objects.
- Through simulation and experiments, the thesis validates the proposed control solutions for several types of manipulation
    - Dynamics for a single UAV manipulating an Object is proposed and a control scheme is presented. A reference input is design procedure based on the constraints imposed by the contact stability condition is presented. The modeling approach and control schemes are validated through simulation and flight tests.
    - Dynamics for two UAVs manipulating an Object is presented. To ease the control design, a systematically reduced order, fully actuated dynamical system is derived. The necessary contact stability conditions are derived and a method to design the reference input that satisfies these conditions is presented. The proposed approach is validated using simulation and flight tests.
    - The thesis proposes manipulation strategies in the presence of uncertainty. Using path of least resistance approach, a scheme to manipulate objects that have unknown kinematic constrains is presented. Through simulation and experiments, a Quadrotor opening a door with unknown hinge location is presented. When the system parameters such as the mass are unknown, the thesis experimentally demonstrates the performance of adaptive nonlinear tracking controller.
    - The thesis presents a novel design for a flexible Quadrotor denoted as ParaFlex. The robustness of ParaFlex to accidental contacts is shown using simulation and experiments.
- For the purposes of validation of above mentioned schemes, a Simulation, Test and Validation Environment (STeVE) is developed. The thesis presents the components and interconnection of STeVE and how it permitted fast and efficient validation.

1.7 Thesis Organization

The thesis is organized into the following seven chapters.

Chapter 2: We present the architecture and components of Simulation, Test and Validation Environment (STeVE). Details of how the hardware-in-the-loop and flight test setup were used are presented. Various end-effector designs are presented, and their relative performances are compared.

Chapter 3: We delve in to mathematical modeling of the Quadrotor manipulation system. We present the dynamics of Quadrotor when it is in free-flight and when it is in Contact. We then derive the dynamics when two Quadrotors are manipulating a single object. We show how the complexity of the equations of motion is reduced by splitting the overall dynamics into sub systems. We derive contact stability conditions for the Quadrotors to remain docked.

Chapter 4: Using the equations of motion, we proceed to design control algorithms to perform tracking and in turn to perform manipulation. We demonstrate the efficacy of the controller through simulation and flight experiments. We demonstrate a single and two Quadrotors pushing an Object.

Chapter 5: Here we proposed solutions for more complex manipulation, like the Quadrotor opening a door. We propose control strategies to perform manipulation when some of the system parameters are unknown. We demonstrate the proposed design to manipulate objects with unknown kinematic constraints and unknown mass.

Chapter 6: While ideas presented in preceding chapters proposed stabilizing a Quadrotor when it is in contact and to perform manipulation, in this chapter we present a Quadrotor design solution. The design provides robustness to the Quadrotor in the face of an impact. We present flexible Quadrotor design and demonstrate its efficacy through numerical simulation and flight experiments.

Chapter 7: We conclude the thesis by summarizing the results from each of the chapters. We briefly comment about the potential future work, listing specific directions.

2 Simulation, Test and Validation Environment

Simulation and Flight experiments are necessary to validate the proposed UAV designs, manipulation schemes and controls schemes. Apart from the function of validation, they also serve as a tool to aid in research. A number of findings happen within the simulation, thus cutting down expensive flight tests. Further, transitioning from simulation to flight tests can bring in unexpected hurdles, for example the Quadrotor models used in simulation are always an approximation to the real Quadrotor. A control system designed and tuned in the simulation thus may not perform well in the flight experiment. In order to address this problem, a standard practice is to use Hardware-in-the-loop (HIL) setups. HIL essentially adds realism to the simulation by replacing the model of the plant with actual components of the plant (in this case a Quadrotor). HIL permits smooth transition from simulation to flight test, isolating any design issues in the early phase. Another advantage of HIL is its ability to provide access to states of the Quadrotor which is not directly measurable in real flight experiments. For instance, forces generated by the actuators can be measured.

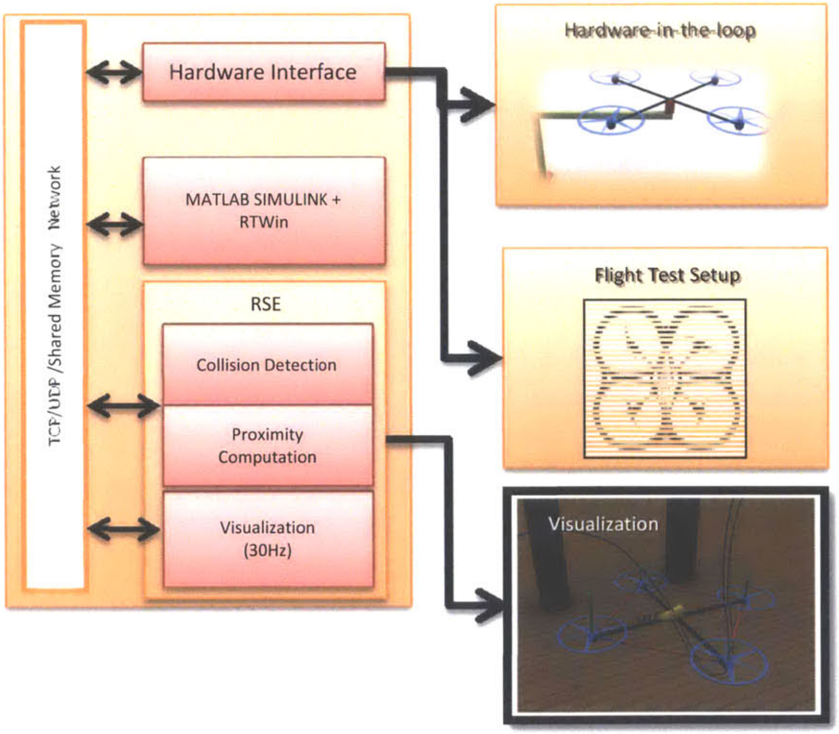

Figure 2.1: A block diagram representation of STeVE showing some of its key components.

The proposed Simulation, Test and Validation Environment, STeVE for short, is a collection of hardware and software tools to address the above mentioned goal. In this chapter, the components of STeVE are presented, and some of key performance metrics are reported. Results from HIL experiments and how it helped make design choices are presented.

2.1 Simulation Subsystem

We used MATLAB Simulink [22] to perform most of the core computation including simulation of Quadrotor and also implementation of control algorithms. Since the Quadrotor is expected to make contact with an object, collision events have to be detected and collision forces have to be included in the simulation. This particular feature is difficult to implement in MATLAB. In order to perform collision based simulation, we resorted to another simulation tool PhysX [23], a free rigid-body simulation environment (RSE for short) package. We also extensively used MSC Nastran [24] as a replacement for PhysX. PhysX is available as a C++ library and requires extensive programming in order to build and simulate a Quadrotor and the Objects in the environment. Nastran, however provides a modeling environment to interactively build the Quadrotor body and other environmental objects. The main advantage of RSE is that it allows modeling of rigid bodies of arbitrary shapes and can perform collision based simulation very efficiently at high rate and accuracy. We could achieve frame rate of up to 500Hz in our case using PhysX. Nastran however was non real-time, but more accurate than PhysX. The control algorithm was implemented in MATLAB Simulink, and MATLAB communicated with RSE though a communication interfaces that was developed using C++ and Communication Library Functions.

2.2 Hardware-In-The-Loop Test Facility

Although RSE handled collision within the simulation, it still lacks the realism of a real Quadrotor. Modeling all of the aspects of a real Quadrotor is time consuming and often impossible. To address this issue, we incorporate some of the hardware components of the Quadrotor into the simulation in order to make the overall simulation more realistic. The hardware component is interfaced to the simulation through necessary digital-to-analog and analog-to-digital converter interfaces. We propose two types of HIL setup. The first one uses a force-torque sensor is shown in the Figure-2.2. The second one uses a motion sensor as shown in Figure 2.5.

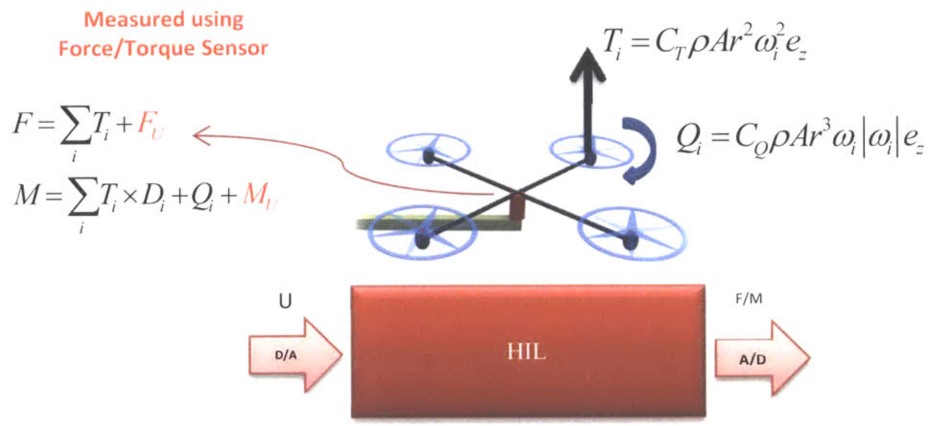
Measured using Force/Torque Sensor
$F=\sum_i T_i + F_U$
$M=\sum_i T_i \times D_i + Q_i + M_U$
$T_i = C_T \rho A r^2 \omega_i^2 e_z$
$Q_i = C_Q \rho A r^3 \omega_i |\omega_i| e_z$
U
D/A
HIL
F/M
A/D

Figure 2.2: HIL Setup Utilizing a Force-Torque sensor to measure the Actuator generated forces. Equations show the Thrust and moment generated as a function of propeller speed and other system constants.

2.2.1 **Force Based HIL**

In the simulation the Quadrotor is modeled using a rigid body. The actuators are modeled as a second order system with quadratic nonlinearities which arise due to propeller aerodynamics [4]. These models, however sophisticated, fail to include all of the nonlinearities of the actuator that arise due to the electronic speed controller and propeller aerodynamics. Moreover, when the Quadrotor is flying close to an object or another Quadrotor, the air flow in the propeller get disrupted, as a result causing disturbances. These aspects are hard to model in the simulation. As a solution, we replace the simulation model of the actuator with a real hardware actuator. The Quadrotor actuators, along with the frame of the Quadrotor are mounted on a force-torque sensor. The developed hardware setup is shown in Figure 2.3 and its function is illustrated in Figure 2.2. Using the HIL setup of Figure 2.3, we performed a number of experiments including (1) system identification of the actuators (2) compare different types of propellers and motors in terms of noise/vibration performance, and (3) proximity effects – when there is an obstruction to the flow of air, assess the changes in lift forces.

Since the actuator is speed controlled, the motor model was assumed to be

$$H(s) = e^{-T_D s} \frac{1}{(1+\tau s)} \tag{2.1}$$

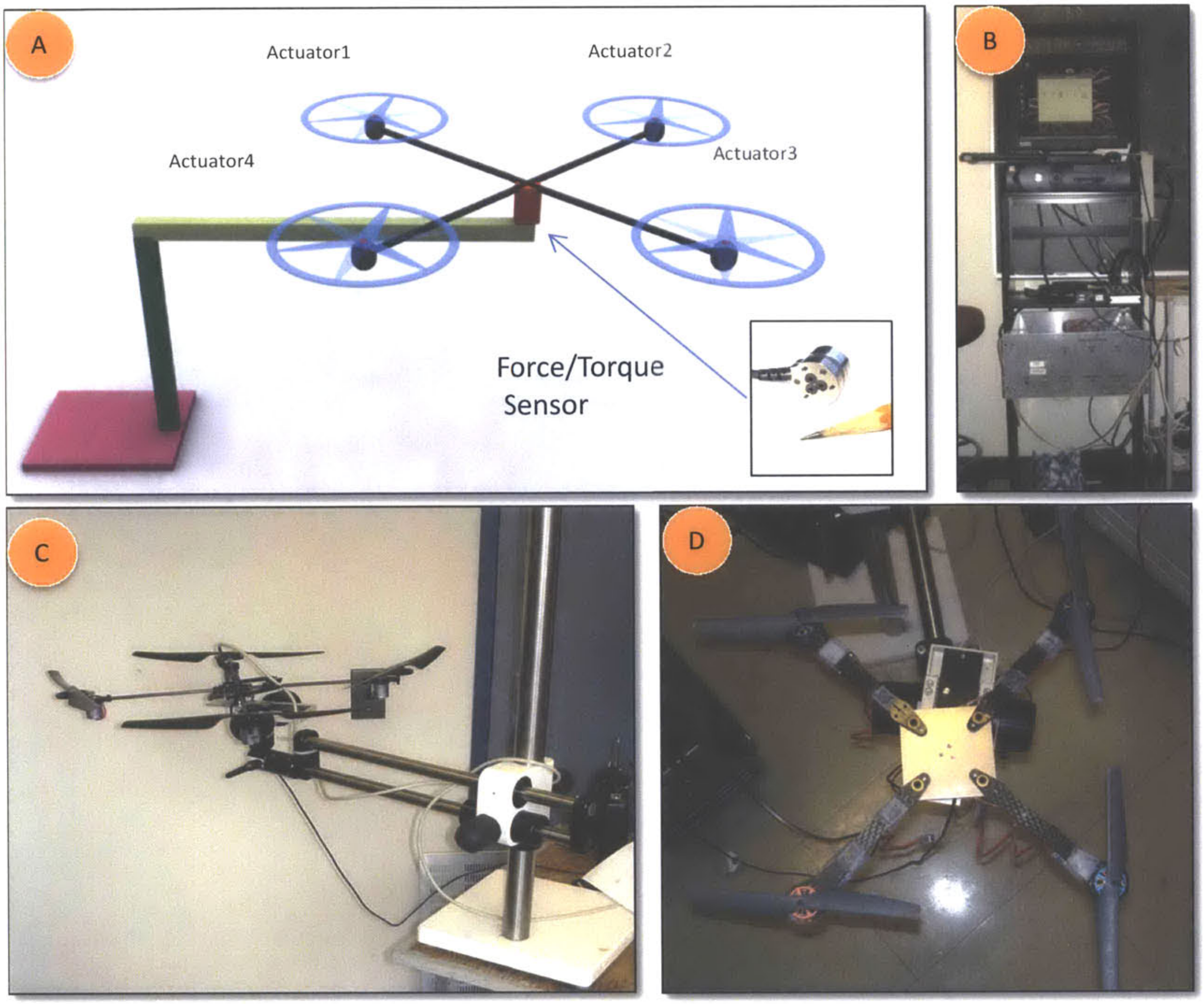

Figure 2.3: The fabricated setup of HIL with a Force-Torque sensor. (A) shows the schematic with the actual force-torque sensor shown in the inset. (B) Shows the instrumentation system mainly consisting of analog-to-digital and digital-to-analog converters. The large size of is due to large power drivers for the motors. The computer housed the LabView NID DAQ cards for data acquisition. (C) Shows a DRAGAN FLY Quadrotor frame consisting of brushed DC Motors. (C) Shows Ascending Tech Quadrotor frame with brushless 3-phase motors.

The time delay T_D and the motor time constant τ were identified using the system identification tool box of Matlab. The quadratic response of the thrust force as a function of propeller speed was verified, that is $F_{lift} = kw_{speed}^2$. A quadratic function was fitted as

$$F_{lift} = a + bv + cv^2 \tag{2.2}$$

Where F_{lift} the lift is force and a,b,c are the constants, and v is the input to the actuator. Equation-(2.2) then was used to linearize the quadratic around the operating point in order to perform control design. In the experimental study, the best motor was found to be of brushless 3-phase type, made by Hacker. Other motors included geared brushless motors, and geared brushed DC motors.

Proximity effects were studies using the Force based HIL. When the motor was running a fixed speed, an obstacle (a flat piece of wood) was introduced 12 inches below the propeller. This resulted in an increased lift force, as shown in the experimental result in Figure 2.4. This effect is also known as ground effect.

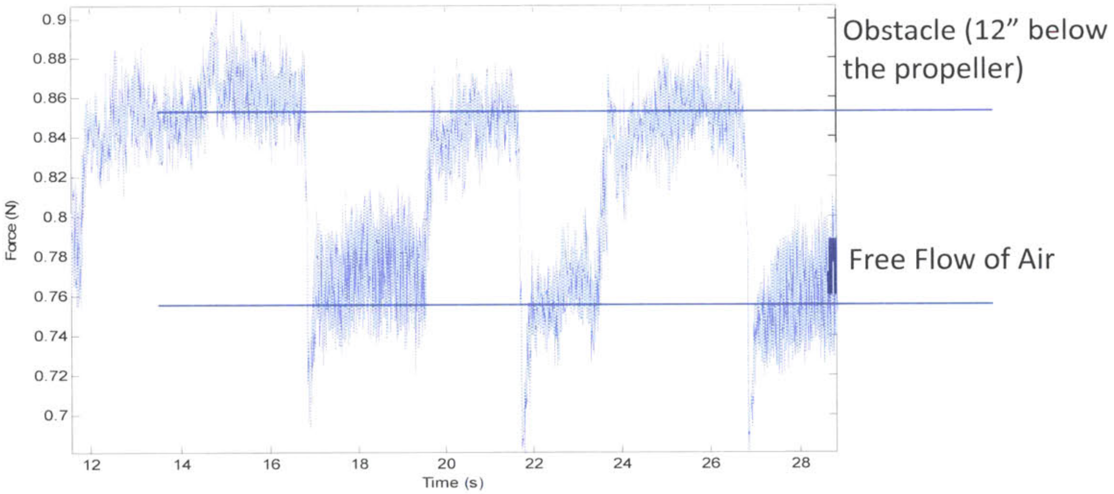

Figure 2.4: Experimental result of force measurement. The actuator is given an a constant input to generate a fixed lift force of 0.76N. When an obstacle is introduced 12 inches below the propeller, lift force increased by about 0.1N. The plot shows the fluctuation of the lift force as the obstacle was repeatedly introduced.

2.2.2 Motion Based HIL

In the case of Force based HIL, the HIL replaced the actuators and not the Quadrotor itself. The rigid body dynamics had to be simulated by applying the forces measured from the HIL. To take the HIL one step further, a motion based HIL was developed. In the motion based HIL, instead of a fixed Quadrotor frame, the Quadrotor was free to move in a confined space, and the motion of the Quadrotor was measured. The forces generated by the actuators now act on the frame, and the simulation system uses the measured position and orientation.

The experimental setup is shown in Figure 2.5. A Phantom Haptic Device [25], a mechanical 6DoF motion sensing (and force feedback) device is used to sense the motion of the Quadrotor. It is essentially an articulated 6DoF mechanism with 6 joint angle sensors. The Quadrotor frame is rigidly mounted on the end-effector of the Phantom device. The Phantom device is equipped with high resolution (4096 counts per rotation) optical encoders at each of the joints. The encoder reading translated to angle so the joints. Using the measured angles and performing forward kinematics, the end-effector location was determined. The end-effector location corresponds the Quadrotors position and orientation, which is used in the simulation.

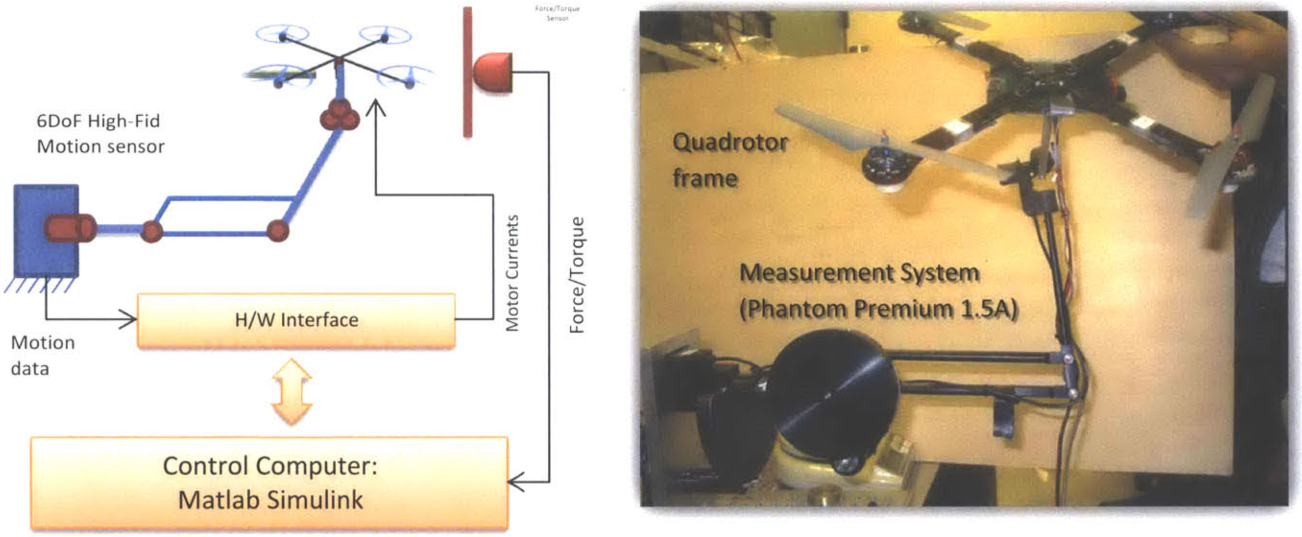
Force/Torque Sensor
6DoF High-Fid Motion sensor
Motor Currents
Force/Torque
H/W Interface
Motion data
Control Computer: Matlab Simulink
Quadrotor frame
Measurement System (Phantom Premium 1.5A)

Figure 2.5: HIL Test Facility showing Quadrotor mounted on a motion sensing platform. Here the Quadrotor is free to move in a confined space.

The key feature of the above mentioned HIL is it's agility to sense the position and orientation as a high rate of 5000Hz. Such measurement rates are usually hard to achieve in real flight experiments. The main drawback of this HIL is the parasitic dynamics that comes because of the mechanical motion sensor. Another disadvantage of course is the range of motion achievable. The workspace was limited to approximately 200mm cube. The HIL was used to evaluate end-effector designs and also to perform adaptive control based manipulation experiments, which is presented in Chapter-5.

2.3 End-Effector Design

In order to perform manipulation, the UAV has to first establish a mechanical coupling. The interface between the object and the UAV is termed as the End-Effector. The End-effector is part of the UAV, which is used to dock to the object. Since the UAV is on the periphery of the object, the air flow is not disturbed, which otherwise would have caused aerodynamic disturbances as noted in the experimental result of Figure 2.4.

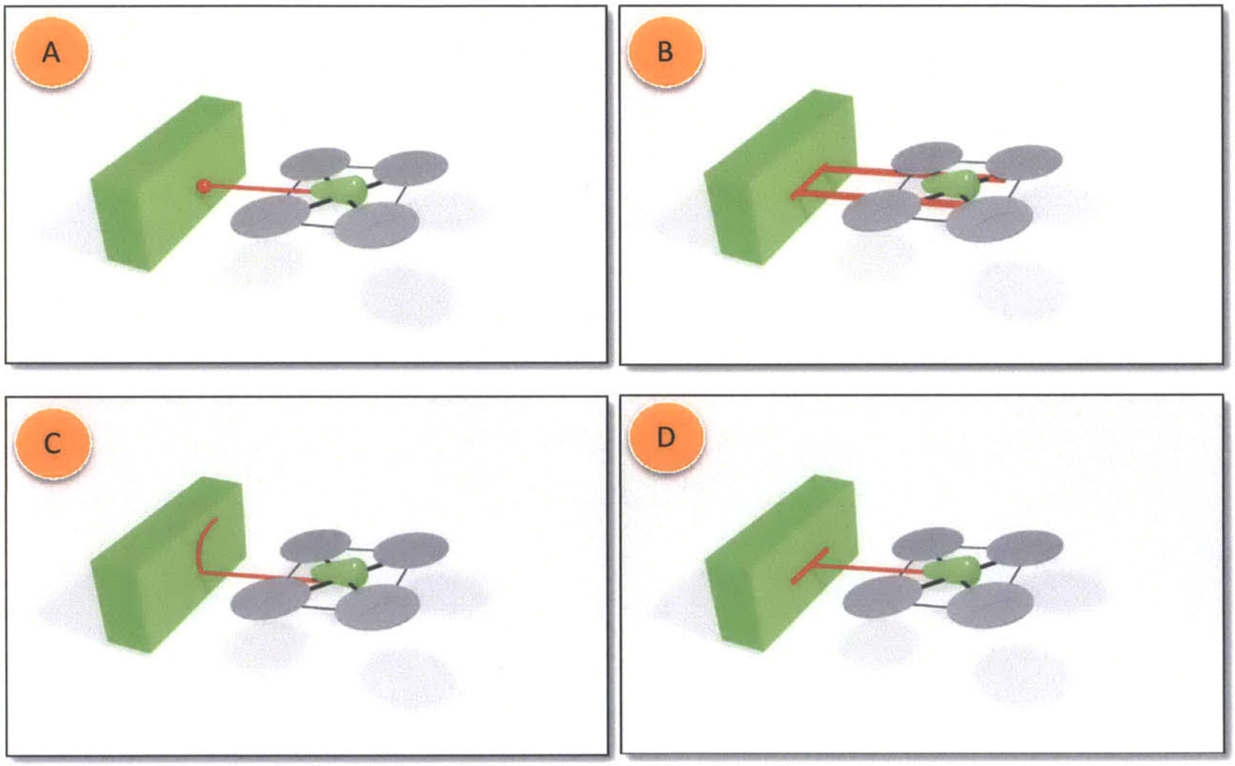

Figure 2.6: End-effector designs.

We proposed and evaluated a number of End-effector designs shown in Figure 2.6. The first design is in Figure 2.6(A), which is point-on-surface constraint. Although it is simple, because of point contact, the end-effector is not robust. The second design shown in Figure 2.6(B) is more robust because two beams and line-on-surface contact. However, when the UAV rolls about the end-effector axis, the bar touching the surface of the object slides, which produces friction torque. This is undesirable because the UAV needs to freely roll to correct for any disturbances. Obstruction to roll can lead to inefficient manipulation preformation. In both Figure 2.6(A) and (B), the contact reaction force introduces a torque on the UAV, which has to be countered through appropriate control design. As an alternative, Figure 2.6(C) shows a curved segment at the end-effector. As the UAV banks to apply the necessary force, the contact point moves in such a way that the contact reaction force does not introduce a torque. This simplifies the dynamics and also the manipulation process. The main disadvantage of this mechanism is the compliance. Because the contact reaction force is in line with the UAV center, an aggressive docking may lead to instability. Further, the design in Figure 2.6(C) does not have the soft compliance which exists in the design of Figure 2.6(A) and (B). The final design which captures of the best features of Figure 2.6(A,B,C) is the design in Figure 2.6(D). Design of Figure 2.6(D) is essentially similar to that of the design in Figure 2.6(A), except that it has two revolute joints as shown in Figure 2.7. The joints permit the UAV to perform roll around the x-axis, and also to swivel around the z-axis (point outward from the plane of Figure 2.7). Figure 2.8 shows the experimental setup utilizing two promising End-effector designs.

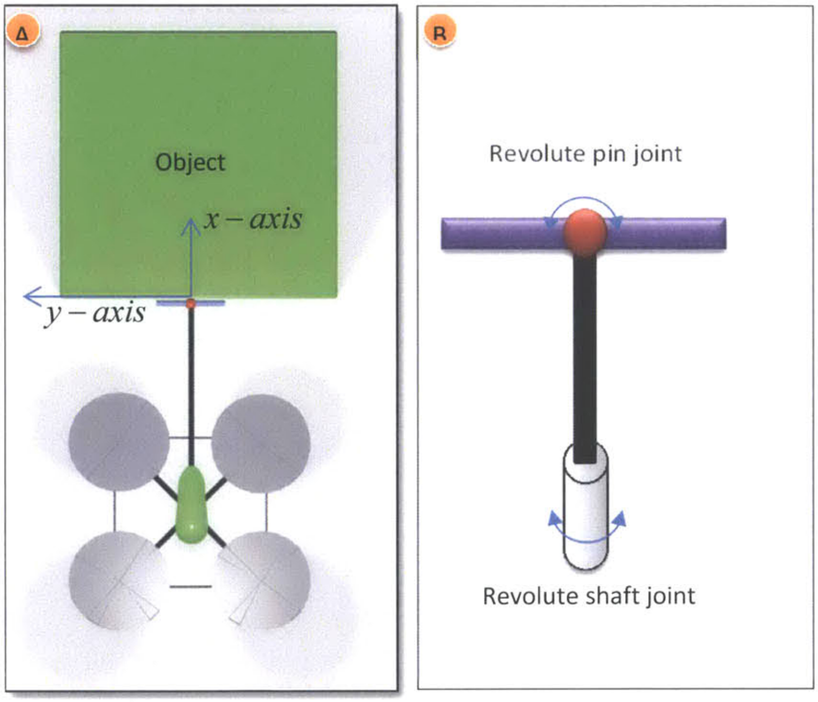

Figure 2.7: End-Effector design showing the joints and degrees-of-freedom.

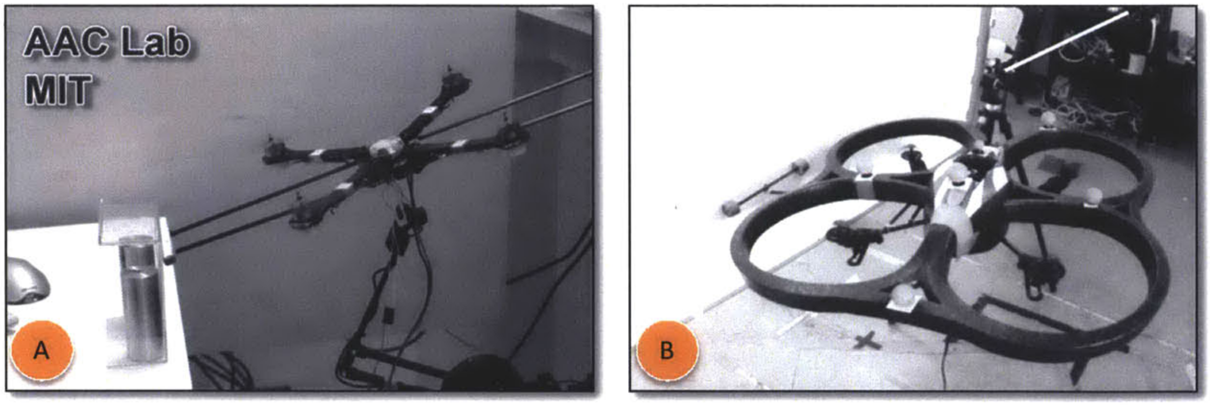

Figure 2.8: End-effectors used in experiments. (A) Application of design shown in Figure 2.6(B). (B) Use of design shown in Figure 2.6 (D).

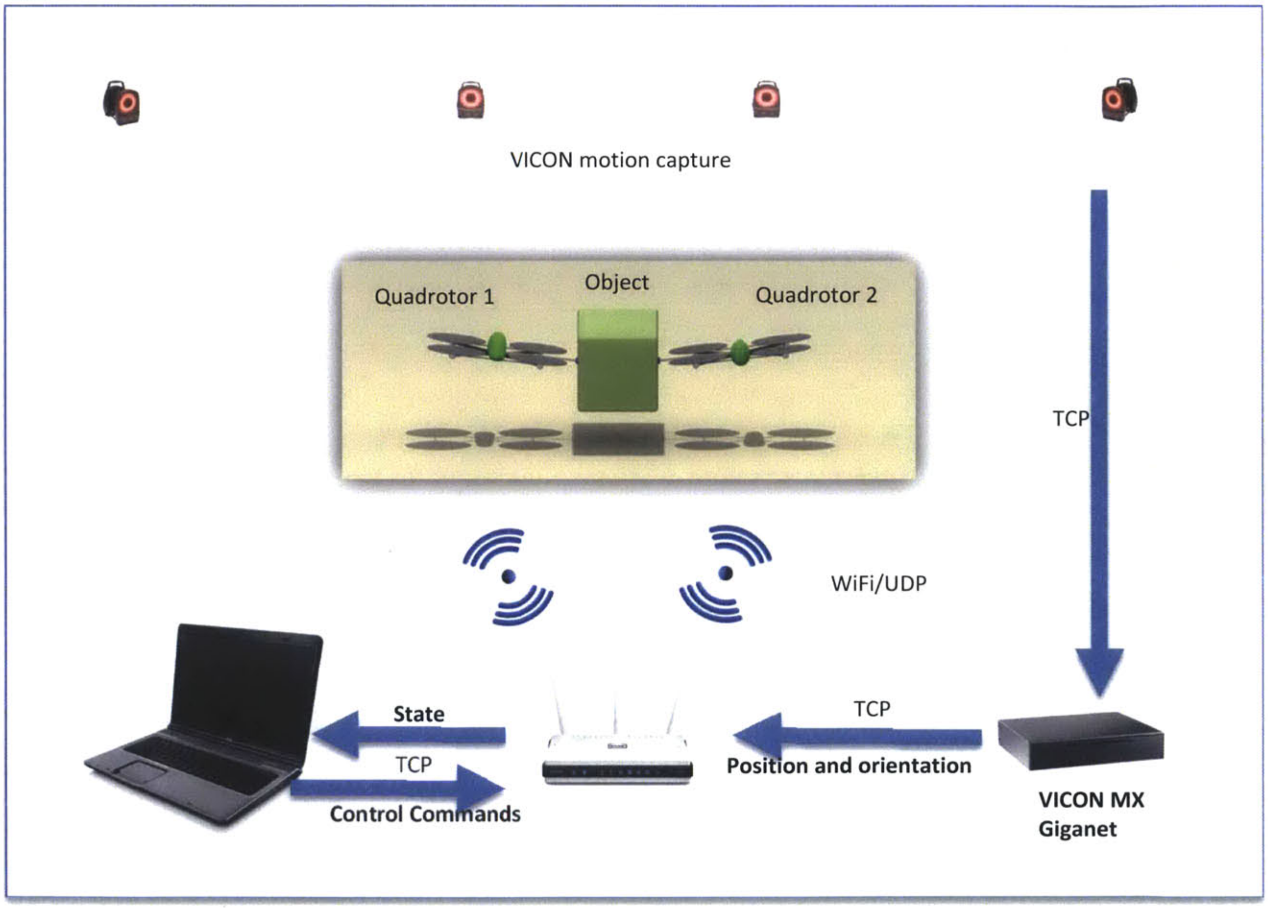

Figure 2.9: The architecture of flight test environment. The position and orientation of the UAVs are sensed by Vicon Motion Capture System [26]. An interface program reads the position and orientation from the network and forwards it to the control/simulation module. The control system then computes the control commands which are then forwarded to the UAVs via WiFi via the interface program. The interface program manages various events including initialization, take-off and landing. It is also responsible for visualization and data logging.

2.4 Flight Test Environment

The final validation of the control algorithms is conducted in a flight test environment. The architecture of the flight test setup is shown in Figure 2.9. The position and orientation of the UAVs are sensed at a high rate (up to 250Hz) by the Vicon Motion Capture system [26]. The date reaches the control computer by means of a TCP/IP network. An interface program developed in C++ acts as a junction block for various data streams. The position and orientation is formatted and forwarded to the control system. Matlab Simulink handles the incoming state information and computes the control command for the UAVs, which is then forward to the interface program. The interface program formats the command data and forwards it to the UAVs using UDP protocol through WiFi. All of these events happen synchronously and it is ensured that the delays is as least as possible. The interface program also provides GUI for the user and performs various additional tasks such as takeoff, landing and emergency landing.

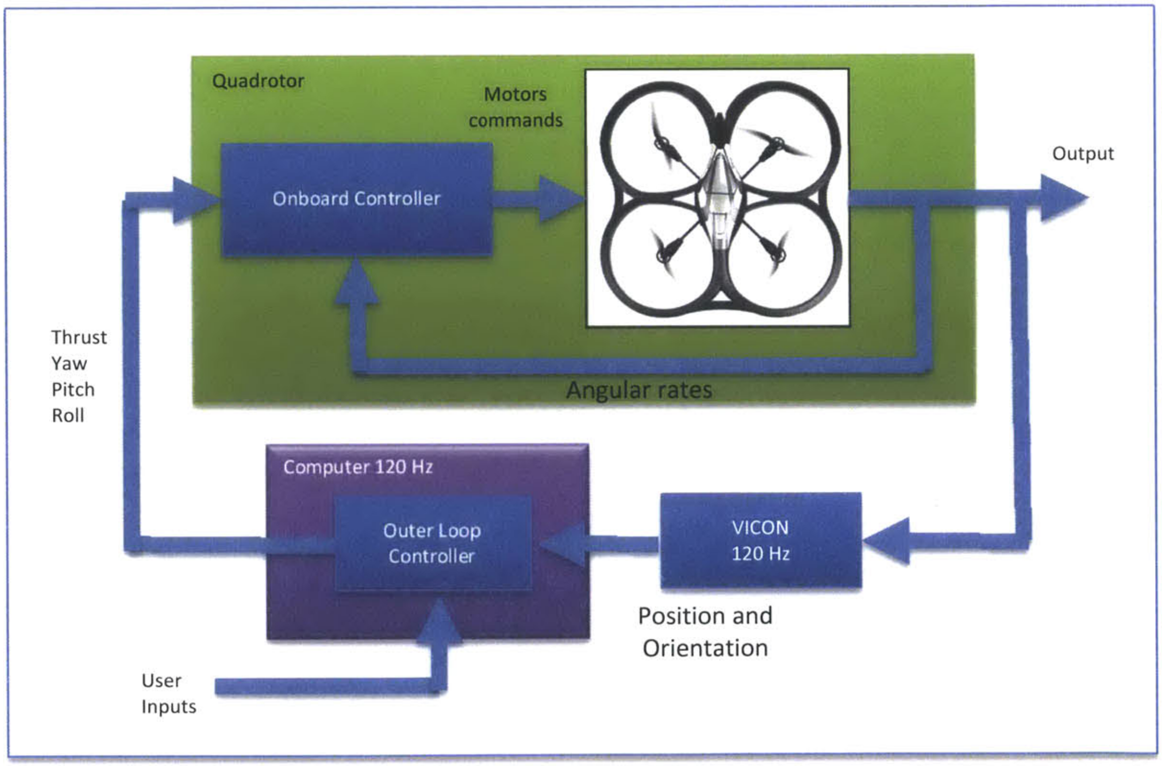

Figure 2.10: Quadrotor Control Loop.

From the control loop perspective, there are mainly two main loops. The Quadrotor has onboard gyros that prove angular velocities. A part of the control loop, termed as onboard-control is directly implemented on the Quadrotor. The outer loop consists of more complicated controllers which are implemented on a Desktop Computer. The control loops are sown in Figure 2.10. The AR Drone Quadrotor is equipped with an onboard computer that runs Linux operating system. A control application is written in C++ that resides onboard. The onboard control software communicates with the interface program using UDP protocol, and also perform local control computation at 200Hz. The position and orientation of the object being manipulated are also sensed and reported to the interface program.

2.5 Summary

This chapter proposed the architecture and components of STeVE, and various end-effector designs. STeVE permitted smooth transition from simulation to experiments, cutting down significant amount of software and hardware duplication. Hardware-in-the-loop setup enabled evaluation of various end-effector designs and permitted selection of a promising design. It also permitted evaluate of adaptive control algorithms. Simulation and experimental results using STeVE are presented in Chapter 4, 5 and 6. The End-effector design proposed in this chapter is used for manipulation tasks proposed in Chapter 3, 4 and 5.

In the next chapter, we will focus on deriving the underlying dynamics when the Quadrotor is in contact with the Object.

3 Dynamics of Quadrotor In contact

When the Quadrotor is docked to an object, its dynamics is much different from that of a free-flight dynamics. In order to design a controller to stabilize the UAV when it is docked, it is necessary to understand the dynamics, which is the focus of this chapter. First we present the dynamics of the Quadrotor in free flight, and linearize it around hover position. We then derive the nonlinear equations of motion when the Quadrotor is docked to the object. We retain the most dominant components of the dynamics and ignore effects due to aerodynamics and actuator dynamics. The reader is referred to [15-26] for more complex Quadrotor models and control solutions [14, 20, 27-36].

3.1 Dynamics of the Quadrotor in Free-Flight

The Quadrotor has four inputs, which are given by $F_{zb}, M_{\phi}, M_{\theta}, M_{\psi}$, the collective thrust force in body frame along z_b axis of the body, and the three body moments respectively. These are also known as Helicopter style inputs, which are resolved into individual actuator commands through a simple linear transform.

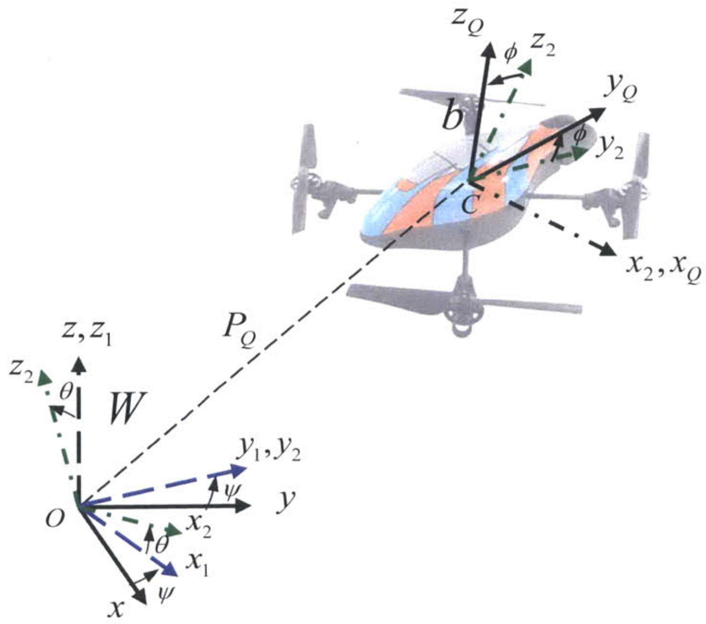

Figure 3.1: Coordinate system for the Quadrotor UAV

The coordinate system is shown in Figure 3.1. Let $P_Q = \left[x_Q, y_Q, z_Q\right]^T$ be the center of the Quadrotor and ϕ, θ, ψ be the roll, pitch and yaw angle of the Quadrotor with ZYX ordering. Let $P_E = \left[x_E \quad y_E \quad z_E\right]^T$ be the end-effector location in world frame W . In the local body frame, the end-effector location is given by $\left[r \quad 0 \quad 0\right]$, where r is the length of the End-effector. The governing equations for the translational acceleration is given by

$$m_Q \ddot{P}_Q = \begin{bmatrix} 0 \\ 0 \\ -m_Q g \end{bmatrix} + R_W^b \begin{bmatrix} 0 \\ 0 \\ F_{zb} \end{bmatrix} \tag{3.1}$$

where, g is acceleration due to gravity, m_Q is the mass of the Quadrotor, F_{zb} is the thrust input for the Quadrotor defined along the body z-axis, and R_W^b is the rotation matrix given by

$$R_W^b = \begin{bmatrix} c\theta c\psi - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & +c\phi c\psi & s\psi s\theta - c\theta s\phi c\psi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} \tag{3.2}$$

where, $c\theta = \cos(\theta), s\theta = \sin(\theta)$ and so on.

The translational dynamics in Equation-(3.1) can be expanded to obtain

$$\begin{aligned} \ddot{x}_Q &= \left(c\psi s\theta + c\theta s\phi s\psi\right)\frac{F_{zb}}{m_Q} \\ \ddot{y}_Q &= \left(s\psi s\theta - c\theta s\phi c\psi\right)\frac{F_{zb}}{m_Q} \end{aligned} \tag{3.3}$$

$$\ddot{z}_Q = -g + \left(c\phi \mathrm{c}\theta\right)\frac{F_{zb}}{m_Q} \tag{3.4}$$

The rotational dynamics is given by

$$J\dot{\omega} = -\omega \times J\omega + \begin{bmatrix} M_\phi \\ M_\theta \\ M_\psi \end{bmatrix} \tag{3.5}$$

As explained before, $\begin{bmatrix} M_\phi & M_\theta & M_\psi \end{bmatrix}^T$ is the roll, pitch, yaw input to the Quadrotor along the body axes, $\omega = \begin{bmatrix} p & q & r \end{bmatrix}^T$ is the angular velocity of the Quadrotor, and J is the moment of inertia measured in body frame, and is given by $J = diag\begin{bmatrix} J_x & J_y & J_z \end{bmatrix}$. Because of the symmetry in Quadrotor frame, we have $J_z = 2J_x = 2J_y$. The rate of Euler angles is related to the angular velocity as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = N_{\phi\theta\psi}\omega \tag{3.6}$$

where,

$$N_{\phi\theta\psi} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ \sin\theta\tan\phi & 1 & -\cos\theta\tan\phi \\ -\dfrac{\sin\theta}{\cos\phi} & 0 & \dfrac{\cos\theta}{\cos\phi} \end{bmatrix} \tag{3.7}$$

Taking the derivative of Equation-(3.6)

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = N_{\phi\theta\psi}\dot{\omega} + \dot{N}_{\phi\theta\psi}\omega \tag{3.8}$$

Equation-(3.5) can be simplified as

$$\dot{\omega} = \begin{bmatrix} \dfrac{J_y - J_z}{J_x} qr \\ \dfrac{J_z - J_x}{J_y} pr \\ \dfrac{J_x - J_y}{J_z} pq \end{bmatrix} + \begin{bmatrix} \dfrac{1}{J_x} M_\phi \\ \dfrac{1}{J_y} M_\theta \\ \dfrac{1}{J_z} M_\psi \end{bmatrix} \tag{3.9}$$

Using Equation-(3.9) in Equation-(3.8), and using the symmetry in J, we can derive the rotational dynamics in Euler angle representation as

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = N_{\phi\theta\psi} \begin{bmatrix} \dfrac{1}{J_x} M_\phi \\ \dfrac{1}{J_y} M_\theta \\ \dfrac{1}{J_z} M_\psi \end{bmatrix} + N_{\phi\theta\psi} \begin{bmatrix} -qr \\ pr \\ 0 \end{bmatrix} + \dot{N}_{\phi\theta\psi} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.10}$$

Thus, the complete dynamics of the Quadrotor is given by Equation-(3.3), (3.4) and (3.10).

3.2 Approximate Flight Dynamics

We assume the Quadrotor is close to hover position. This implies that the two Euler angles ϕ and θ, are small with the yaw angle ψ arbitrary, and $F_{zb} \approx m_Q g$. Using this we approximate the translational dynamics of x_Q and y_Q in Equations-(3.3) as

$$\begin{aligned} \ddot{x}_Q &\approx (c\psi\theta + s\psi\phi) g \\ \ddot{y}_Q &\approx (s\psi\theta - c\psi\phi) g \end{aligned} \tag{3.11}$$

The altitude dynamics however is retained in the original form as

$$\ddot{z}_Q \approx -g + (c\phi \, c\theta) \frac{F_{zb}}{m_Q} \tag{3.12}$$

While deriving Equation-(3.11) and (3.12), it can be seen that $\sin\theta$ is approximated as θ only in Equation-(3.3), whereas $\cos\theta$ is not approximated as 1 in Equation-(3.12).

With small angles, $N_{\phi\theta\psi} \approx I_{3\times 3}$ (identify), and therefore the angular dynamics given in Equation- (3.10) can be approximated, as

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \approx \begin{bmatrix} \frac{1}{J_x} M_\phi \\ \frac{1}{J_y} M_\theta \\ \frac{1}{J_z} M_\psi \end{bmatrix} \tag{3.13}$$

This is because the altitude dynamics z_Q is fast, with both terms on the r.h.s of Equation-(3.12) contributing in equal measure and therefore cannot be simplified. In comparison, under hover condition, x_Q and y_Q vary slowly and therefore $\sin\theta$ can be approximated as θ.

3.3 Dynamics of the Quadrotor in Contact

In this section we derive the unified dynamics of a Quadrotor when it is configured to be in contact with another object. Here we assume the object is free to move only along the $x - axis$.

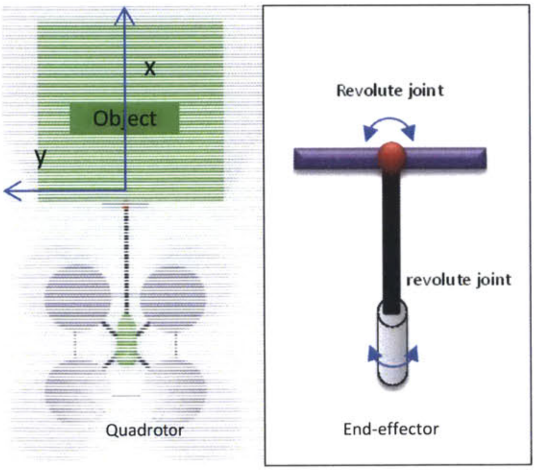

Figure 3.2: (Left) Top view of Quadrotor docked to the object showing the end-effector in contact with the object. (Right) The end-effector mechanism degrees of freedom.

The end-effector we assumed here is one of the various designs we presented in Chapter 1. This design is shown in Figure 3.2. Essentially, the end-effector is a "point-on-surface" constraint. This implies that the end-effector permits 5 degrees of freedom – three angular degrees and two surface-sliding translational degrees.

We see that the combined system, that is, the Quadrotor and the Object still has 6 degrees of freedom. Since the Object is restricted to move along one degree of freedom, we can say the position of the Object can be described by x_E. Note that the end-effector location P_E is related to center of the Quadrotor P_Q through a rigid transformation matrix (4x4).

Modeling of the complete system for the 6 degrees of freedom can be quite challenging. We separate the total degrees of freedom and group them in to X_{free} and X_{manip} as

$$X_{free} = \begin{bmatrix} y_Q \\ \phi \\ \psi \end{bmatrix} \tag{3.14}$$

$$X_{manip} = \begin{bmatrix} x_E \\ z_E \\ \theta \end{bmatrix} \tag{3.15}$$

We note that we have four control inputs and 6 degrees of freedom, which make this system underactuated. We associate $\begin{bmatrix} M_\phi & M_\psi \end{bmatrix}^T$ with X_{free} and $\begin{bmatrix} F_{zb} & M_\theta \end{bmatrix}^T$ with X_{manip} .

X_{free} corresponds to degrees of freedom that do not contribute towards pushing the Object, while X_{manip} corresponds to degrees that push the Object and hence permits manipulation. The assumption, however, is that the two dynamics are decoupled from one another under certain conditions that are satisfied during nominal operation. One key assumption is that ψ is small during manipulation.

In the following sections we derive dynamics for X_{free} and X_{manip}.

3.3.1 X_{free} Dynamics

We start from first principles, as we did to derive the equations of motion in Section-3.1. We make small angle approximation for ϕ . θ , however is not small because in order to apply force on the Object, the Quadrotor makes steep angles as shown in Figure 3.3. With these assumptions, the dynamics of X_{free} can be derived as

$$\ddot{X}_{free} = \begin{bmatrix} \ddot{y}_Q \\ \ddot{\phi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} (\tan\theta \sin\psi - \phi\cos\psi) g \\ \dfrac{M_\phi}{J_x}\cos\theta + \dfrac{M_\psi}{J_z}\sin\theta \\ -\dfrac{M_\phi}{J_x}\sin\theta + \dfrac{M_\psi}{J_z}\cos\theta \end{bmatrix} \tag{3.16}$$

3.3.2 X_{manip} Dynamics

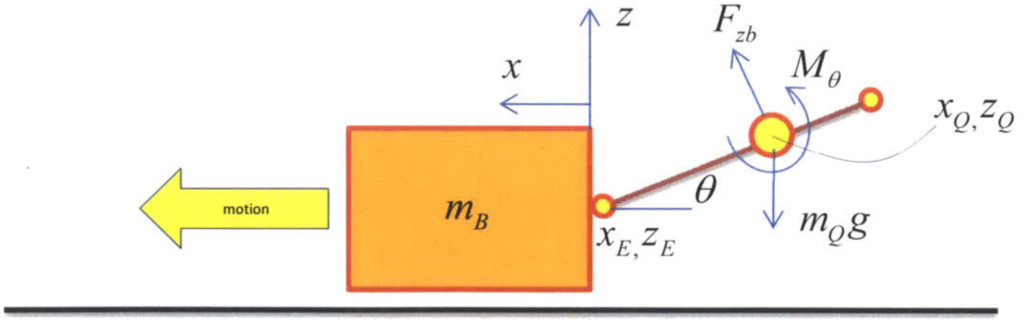

Figure 3.3: Simplified model for X_{manip} dynamics.

Again, starting from first principles we derive the dynamics of the X_{manip}, whose free body diagram is shown in Figure 3.3. Using Euler-Lagrange approach we derive the Equation of motion in the standard manipulator form as

$$\mathbf{H}\ddot{X}_{manip} + \mathbf{C}\dot{X}_{manip} + \mathbf{G} = \mathbf{B}\begin{bmatrix} F_{zb} \\ M_{\theta} \end{bmatrix} \tag{3.17}$$

where

$$\mathbf{B} = \begin{bmatrix} \sin\theta & 0 \\ \cos\theta & 0 \\ r & 1 \end{bmatrix} \tag{3.18}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & m_Q r\dot{\theta}\cos\theta \\ 0 & 0 & -m_Q r\dot{\theta}\sin\theta \\ 0 & 0 & -m_Q r\left(\dot{x}_E\cos\theta - \dot{z}_E\sin\theta\right) \end{bmatrix} \tag{3.19}$$

$$\mathbf{G} = \begin{bmatrix} 0 \\ m_Q g \\ m_Q g r\cos\theta \end{bmatrix} \tag{3.20}$$

$$\mathbf{H} = \begin{bmatrix} m_B + m_Q & 0 & m_Q r\sin\theta \\ 0 & m_Q & m_Q r\cos\theta \\ m_Q r\sin\theta & m_Q r\cos\theta & m_Q r^2 + J_y \end{bmatrix} \tag{3.21}$$

where m_B is the mass of the Object and r is the end-effector length. The acceleration can be found by simply inverting the dynamics as

$$\ddot{X}_{manip} = \mathbf{H}^{-1}\left(\mathbf{B}\begin{bmatrix} F_{zb} \\ M_{\theta} \end{bmatrix} - \mathbf{C}\dot{X}_{manip} - \mathbf{G}\right) \tag{3.22}$$

3.3.3 Contact Stability and Push Force

The contact force between the end-effector and the Object dictates whether or not the contact will be sustained. This is important because if the contact is lost, the model in Equation-(3.17) is invalid, and hence the controller that is designed based on Equation-(3.17) is also invalid. In general the contact force has to follow the condition

$$F_{contact,\min} \leq F_{contact} \leq F_{contact,\max} \tag{3.22}$$

For a non-prehensile contact, $F_{contact,\min} = 0$, meaning only a push force can be applied, and for a prehensile contact $F_{contact,\min} < 0$, meaning a pull force can also be applied. In our experimental setup we considered only a non-prehensile contact. The end-effector can include a magnet if the object being manipulated has a magnetic surface, in which case we have a prehensile contact. In this paper we propose to open a door, so we only considered non-prehensile case because the door is expected to be made of non-magnetic material. Further, a non-prehensile contact means x_E cannot be controlled to reach a desired location because the actuation is only unidirectional.

The force appearing on the object is termed as Push force, designated by F_{push}. The push force appearing on the Object is given by

$$F_{push} = m_B \ddot{x}_E \tag{3.23}$$

Because F_{push} is the only force acting on the object, we can say the contact force between the Object and the end-effector is $F_{contact} = F_{push}$. The expression for F_{push} can be obtained using Equation-(3.22). Ignoring the coriolis terms, we have

$$F_{push} \approx \eta \frac{F_{zb} J_y \cos\theta - M_\theta m_Q r \sin\theta}{J_y + \eta m_Q r^2 \sin^2\theta} \tag{3.24}$$

where $\eta = \dfrac{m_B}{m_B + m_Q}$.

The control inputs F_{zb}, M_θ can be thought of as decomposed into the static gravity compensation part and the dynamic part as $F_{zb} = F_{zb,static} + F_{zb,\Delta}, M_\theta = M_{\theta,static} + M_{\theta,\Delta}$. It can be shown that

$$\begin{aligned} F_{zb,static} &= \frac{m_Q g}{\cos\theta} \\ M_{\theta,static} &= -m_Q g r\left(\cos\theta - \frac{1}{\cos\theta}\right) - rF_{zb,\Delta} \end{aligned} \tag{3.25}$$

we assume $0 \le \theta \ll \dfrac{\pi}{2}$ (this is ensured though proper choice of reference input θ_{des}).

Using Equation-(3.25) and (3.24), we get

$$F_{push} \approx \eta g m_Q \tan\theta \left(\frac{J_y + m_Q r^2 \sin^2\theta}{J_y + \eta m_Q r^2 \sin^2\theta} \right) + \eta \frac{F_{zb,\Delta} \sin\theta\left(J_y + m_Q r^2\right) - M_{\theta,\Delta}\left(m_Q r \sin\theta\right)}{J_y + \eta m_Q r^2 \sin^2\theta} \tag{3.26}$$

During steady state of θ, z_E we have $F_{zb,\Delta} \to 0, M_{\theta,\Delta} \to 0$, so we can further approximate Equation-(3.26) as

$$F_{push} \approx \eta g m_Q \tan\theta \tag{3.27}$$

Note, the accuracy of Equation-(3.27) reduces as θ increases. In practical scenarios, typically we ensure $|\theta| \le \frac{\pi}{6}$. With this condition, for a Quadrotor with $m_Q = 0.5kg$ and $\eta \approx 1.0$, we can expect a maximum force of about $2.8N$.

3.3.4 An Alternate Representation of X_{manip} dynamics

The dynamical system in Equation-(3.17) is underactuated because only two control inputs drive a three degree of freedom system. To address this, we derive a two-input, two-degree of freedom system by eliminating x_E from Equation-(3.17). Let us call the new system degree of freedom vector as

$$\tilde{X}_{manip} = \begin{bmatrix} z_Q \\ \theta \end{bmatrix} \tag{3.28}$$

Note we are using the Quadrotor center height z_Q, which is related to z_E as

$$z_Q = z_E + r\sin\theta \tag{3.29}$$

Starting from Equation-(3.17), we can derive the new system dynamics as

$$\tilde{\mathbf{H}}\ddot{\tilde{X}}_{manip} + \tilde{\mathbf{C}}\dot{\tilde{X}}_{manip} + \tilde{\mathbf{G}} = \tilde{\mathbf{B}}\begin{bmatrix} F_{zb} \\ M_\theta \end{bmatrix} \tag{3.30}$$

where,

$$\tilde{\mathbf{H}}=\begin{bmatrix} m_Q & 0 \\ \eta m_Q r\cos\theta & I_Q+\eta m_Q r^2\sin^2\theta \end{bmatrix} \tag{3.31}$$

$$\tilde{\mathbf{C}}=\begin{bmatrix} 0 & 0 \\ 0 & \eta m_Q r^2\dfrac{\sin(2\theta)}{2}\dot{\theta} \end{bmatrix} \tag{3.32}$$

$$\tilde{\mathbf{G}}=\begin{bmatrix} m_Q g \\ \eta m_Q g r\cos\theta \end{bmatrix} \tag{3.33}$$

$$\tilde{\mathbf{B}}=\begin{bmatrix} \cos(\theta) & 0 \\ \eta r & 1 \end{bmatrix} \tag{3.34}$$

and $\eta=\dfrac{m_B}{m_B+m_Q}$.

In the derivation of Equation-(3.32), $m_Q r\cos\theta\dot{x}_E$ term has been ignored from $\widetilde{\mathbf{C}}(3,3)$ owing to the fact that $\left|\dot{x}_E\dot{\theta}\right|\ll 0$. The push force is computed as before can be shown to be Equation-(3.27).

The main advantage of representing the system in the form of Equation-(3.30) is its suitability for inverse dynamics control and adaptive control design using Slotine and Li Adaptive control [37, 38] with unknown η . This is discussed in detail in Chapter 5.

3.3.5 Dynamics of Two Quadrotors Manipulating an Object

When two Quadrotors are docked to the Object, as shown in Figure 3.4, the Object can be manipulated in either direction of $x-axis$, which was not possible in there single UAV case. There are two approaches we can take to deriving the dynamics of such a system. One approach is to think of each UAV as a force producing entity just as in the single UAV case and design controllers to manipulate the Object. The other approach is to derive complete dynamics of the unified system, which of course is much more complicated. We first present the complete unified dynamics and present conditions under which we can make approximations to reduce it to independent sub systems.

As before, we first consider the total degrees of freedom of the system in Figure 3.4, given by

$$X_{manip,dual} = \begin{bmatrix} x_E & z_{E,1} & z_{E,2} & \theta_1 & \theta_2 \end{bmatrix}^{\mathrm{T}} \tag{3.35}$$

We also note that each Quadrotor as its own X_{free}

$$X_{free,1} = \begin{bmatrix} y_{E,1} \\ \phi_1 \\ \psi_1 \end{bmatrix}, X_{free,2} = \begin{bmatrix} y_{E,2} \\ \phi_2 \\ \psi_2 \end{bmatrix} \tag{3.36}$$

$X_{free,1}$ and $X_{free,2}$ dynamics can be derived in the same was as we did for the single Quadrotor case.

The dynamics for $X_{manip,dual}$ is given by

$$\mathbf{H}_{dual}\ddot{X}_{manip,dual} + \mathbf{C}_{dual}\dot{X}_{manip,dual} + \mathbf{G}_{dual} = \mathbf{B}_{dual}\begin{bmatrix} F_{zb,1} \\ F_{zb,2} \\ M_{\theta,1} \\ M_{\theta,2} \end{bmatrix} \tag{3.37}$$

where

$$\mathbf{H}_{dual}=\begin{bmatrix} m_B+2m_Q & 0 & 0 & m_Q r\sin\theta_1 & -m_Q r\sin\theta_2 \\ 0 & m_Q & 0 & m_Q r\cos\theta_1 & 0 \\ 0 & 0 & m_Q & 0 & m_Q r\cos\theta_2 \\ m_Q r\sin\theta_1 & m_Q r\cos\theta_1 & 0 & m_Q r^2+J_y & 0 \\ -m_Q r\sin\theta_2 & 0 & m_Q r\cos\theta_2 & 0 & m_Q r^2+J_y \end{bmatrix} \tag{3.38}$$

$$\mathbf{C}_{dual}=\begin{bmatrix} 0 & 0 & 0 & m_Q r\dot{\theta}_1\cos\theta_1 & -m_Q r\dot{\theta}_2\cos\theta_2 \\ 0 & 0 & 0 & -m_Q r\dot{\theta}_1\sin\theta_1 & 0 \\ 0 & 0 & 0 & 0 & -m_Q r\dot{\theta}_2\sin\theta_2 \\ 0 & 0 & \begin{matrix}-\dot{z}_{E,1}m_Q r\sin\theta_1 \\ +\dot{x}_E m_Q r\cos\theta_1\end{matrix} & 0 & 0 \\ 0 & 0 & 0 & 0 & \begin{matrix}-\dot{z}_{E,2}rm_Q\sin\theta_2 \\ -\dot{x}_E rm_Q\cos\theta_2\end{matrix} \end{bmatrix} \tag{3.39}$$

$$\mathbf{G}_{dual}=\begin{bmatrix} 0 \\ m_Q g \\ m_Q g \\ m_Q grcos\theta_1 \\ m_Q grcos\theta_2 \end{bmatrix} \tag{3.40}$$

$$\mathbf{B}_{dual}=\begin{bmatrix} sin\theta_1 & sin\theta_2 & 0 & 0 \\ cos\theta_1 & 0 & 0 & 0 \\ 0 & cos\theta_2 & 0 & 0 \\ r & 0 & 1 & 0 \\ 0 & r & 0 & 1 \end{bmatrix} \tag{3.41}$$

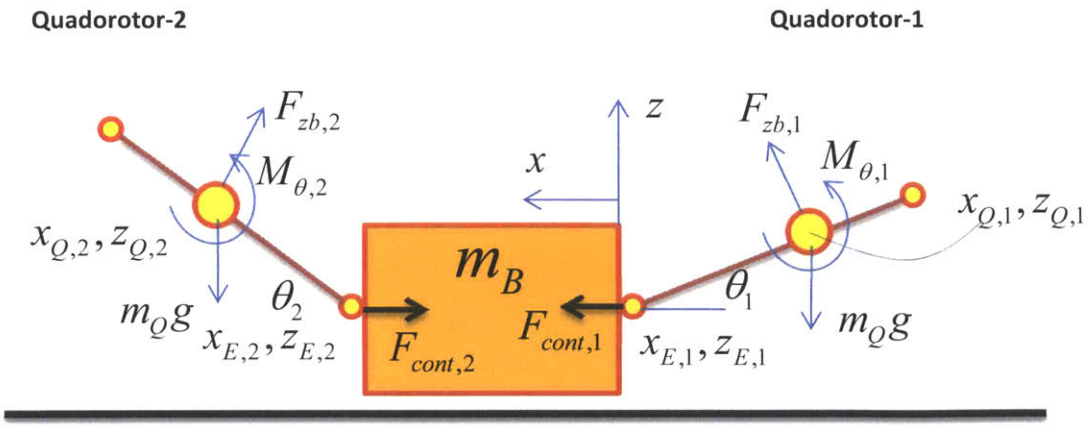

Figure 3.4: Two Quadrotors docked to the Object on either side apply a collective force to achieve bilateral manipulation. The top schematic is the top-view and the bottom schematic is the side-view.

The contact force at the right side end-effector (Quadrotor-1), with the convention followed in Figure 3.4, can be derived as

$$
\begin{aligned}
F_{cont,1} &= \left(\frac{m_B + m_Q}{m_B + 2m_Q}\right)\left(F_{zb,1}\sin\theta_1 - m_Q r\ddot{\theta}_1 \sin\theta_1 - m_Q r\dot{\theta}_1^2 \cos\theta_1\right) \\
&+ \left(\frac{m_Q}{m_B + 2m_Q}\right)\left(F_{zb,2}\sin\theta_2 - m_Q r\ddot{\theta}_2 \sin\theta_2 - m_Q r\dot{\theta}_2^2 \cos\theta_2\right)
\end{aligned}
\tag{3.42}
$$

and the contact force on the left side end-effector (Quadrotor-2) is given by

$$
\begin{aligned}
F_{cont,2} &= \left(\frac{m_B + m_Q}{m_B + 2m_Q}\right)\left(F_{zb,2}\sin\theta_2 - m_Q r\ddot{\theta}_2 \sin\theta_2 - m_Q r\dot{\theta}_2^2 \cos\theta_2\right) \\
&+ \left(\frac{m_Q}{m_B + 2m_Q}\right)\left(F_{zb,1}\sin\theta_1 - m_Q r\ddot{\theta}_1 \sin\theta_1 - m_Q r\dot{\theta}_1^2 \cos\theta_1\right)
\end{aligned}
\tag{3.43}
$$

Note that we have retained $\ddot{\theta}_1$ and $\ddot{\theta}_2$ terms in Equation-(3.42) and(3.43). These terms can be replaced as a function of velocities and control inputs, however the final result would be more complex. Using Equation-(3.42) and (3.43), the total force acting on the Object can be written as

$$F_{Object} = \left(F_{cont,1} + F_{cont,2}\right) = m_B \ddot{x}_E =$$
$$\left(\frac{m_B}{m_B + 2m_Q}\right)\begin{bmatrix} \left(F_{zb,1}\sin\theta_1 + F_{zb,2}\sin\theta_2\right) \\ -\left(m_Q r\ddot{\theta}_1 \sin\theta_1 + m_Q r\dot{\theta}_1^2\cos\theta_1\right) \\ +\left(m_Q r\ddot{\theta}_2 \sin\theta_2 + m_Q r\dot{\theta}_2^2\cos\theta_2\right) \end{bmatrix} \tag{3.44}$$

3.3.6 **An Alternate Representation**

We see that the dynamics in Equation-(3.37) is underactuated. An alternative representation to $X_{manip,dual}$ dynamics is to split the system into fully actuated subsystems. We consider a change in coordinate system and use

$$\tilde{X}_{manip,dual} = \begin{bmatrix} x_E & z_{Q,1} & z_{Q,2} & \theta_1 & \theta_2 \end{bmatrix}^{\mathrm{T}} \tag{3.45}$$

We split the $\tilde{X}_{manip,dual}$ vector into three sub systems: x_E , $\tilde{X}_{manip,1} = \begin{bmatrix} z_{Q,1} & \theta_1 \end{bmatrix}^T$ and $\tilde{X}_{manip,2} = \begin{bmatrix} z_{Q,2} & \theta_2 \end{bmatrix}^T$. x_E Corresponds to manipulation degree of freedom of the Object, $\tilde{X}_{manip,1}, \tilde{X}_{manip,2}$ correspond do the dynamics of the two Quadrotors. We then write the dynamics of the two sub systems as

$$\tilde{\mathbf{H}}_1 \ddot{\tilde{X}}_{manip,1} + \tilde{\mathbf{C}}_1 \dot{\tilde{X}}_{manip,1} + \tilde{\mathbf{G}}_1 = \tilde{\mathbf{B}}_1 \begin{bmatrix} F_{zb,1} \\ M_{\theta,1} \end{bmatrix} + \tilde{\mathbf{B}}_{ext,1} F_{ext,1} \tag{3.46}$$

and

$$\tilde{\mathbf{H}}_2 \ddot{\tilde{X}}_{manip,2} + \tilde{\mathbf{C}}_2 \dot{\tilde{X}}_{manip,2} + \tilde{\mathbf{G}}_2 = \tilde{\mathbf{B}}_2 \begin{bmatrix} F_{zb,2} \\ M_{\theta,2} \end{bmatrix} + \tilde{\mathbf{B}}_{ext,2} F_{ext,2} \tag{3.47}$$

We account for the coupling between the two systems by introducing the coupling terms $\tilde{\mathbf{B}}_{ext,1}, \tilde{\mathbf{B}}_{ext,2}$. The derivation of Equation-(3.46) and (3.47), and the expressions for $\tilde{\mathbf{H}}_i, \tilde{\mathbf{C}}_i, \tilde{\mathbf{G}}_i, \tilde{\mathbf{B}}_i, i = 1,2$ are presented in Section-3.3.7.

We will now focus now the contact forces and the contact stability. As we derived in Equation-(3.27), the push force generated by each of the Quadrotor is given by

$$F_{push,1} \approx m_Q g \tan\theta_1 \tag{3.48}$$

$$F_{push,2} \approx m_Q g \tan\theta_2 \tag{3.49}$$

The contact forces (both nominally positive, acting into the Object is considered positive) are given by

$$F_{cont,1} \approx \eta F_{push,1} - (1-\eta) F_{push,2} \tag{3.50}$$

$$F_{cont,2} \approx -\eta F_{push,2} + (1-\eta) F_{push,1} \tag{3.51}$$

where

$$\eta = \frac{m_B + m_Q}{m_B + 2m_Q} \tag{3.52}$$

The contact stability condition for the two Quadrotor case is given by

$$\begin{aligned} F_{cont,\min} \le F_{cont,1} \le F_{cont,\max} \\ F_{cont,\min} \le F_{cont,2} \le F_{cont,\max} \end{aligned} \tag{3.53}$$

By ensuring $\theta_1 \to \theta_{des,1}$ and $\theta_2 \to \theta_{des,2}$ through an appropriate controller, we can ensure tracking of desired push forces. That is, $F_{push,1} \to F_{push,1,des}$, $F_{push,2} \to F_{push,d,des}$.

We can compute the total force acting on the Object as

$$
\begin{aligned}
F_{object} &= F_{cont,1} - F_{cont,2} \\
F_{object} &= (2\eta - 1)\left(F_{push,1} + F_{push,2}\right) \approx (2\eta - 1)\left(m_Q g \tan\theta_1 + m_Q g \tan\theta_2\right)
\end{aligned}
\tag{3.54}
$$

and the acceleration of the object is F_{object}/m_B .

It is also worth noting that an internal grasp force appears on the Object given by

$$
\begin{aligned}
F_{object,grasp} &= \frac{F_{cont,1} + F_{cont,2}}{2} \\
F_{object,grasp} &= (2\eta - 1)\frac{F_{push,1} - F_{push,2}}{2} \approx (2\eta - 1)\frac{\left(m_Q g \tan\theta_1 - m_Q g \tan\theta_2\right)}{2}
\end{aligned}
\tag{3.55}
$$

This grasp force can be used to ensure firmness in holding the object in the face of external disturbances.

We see that because of the constraint given by Equation-(3.53), the push and grasp force given by Equations-(3.54) and (3.55) respectively, will also have a constraint. Derivation of this constraint is carried out in Section-3.3.8. With the knowledge of feasible $\begin{bmatrix} F_{object} \\ F_{object,grasp} \end{bmatrix}$, one can arbitrarily pick a desired value for $\begin{bmatrix} F_{object} \\ F_{object,grasp} \end{bmatrix}$ (within the feasible region) that ensures contact stability of Equation-(3.53). This is particularly useful in design of manipulation scheme presented in Chapter 4.

3.3.7 Complete Derivation of $\tilde{X}_{manip,1}, \tilde{X}_{manip,2}$ Dynamics

Our goal in this section is to start from the dynamics given by Equation-(3.37) and derive the approximate dynamics in the form of Equation-(3.46) and Equation-(3.47).

We first make a change of coordinate to replace $z_{E,i}$ with $z_{Q,i}, i = 1,2$. From Figure 3.4, we see that

$$\begin{aligned} z_{Q,1} &= z_{E,1} + r\sin\theta_1 \\ z_{Q,2} &= z_{E,2} + r\sin\theta_2 \end{aligned} \tag{3.56}$$

We will now derive the dynamics for the new variable

$$\tilde{X}_{manip,dual} = \begin{bmatrix} x_E & z_{Q,1} & z_{Q,2} & \theta_1 & \theta_2 \end{bmatrix}^{\mathrm{T}} \tag{3.57}$$

After incorporating the change of coordinate according to Equation-(3.56), we perform the following matrix operation on the new dynamical system:

$$\begin{aligned} row_4 &\leftarrow row_4 - row_2\left(r\cos\theta_1\right) \\ row_5 &\leftarrow row_5 - row_3\left(r\cos\theta_2\right) \end{aligned} \tag{3.58}$$

to obtain the final dynamics as

$$\tilde{\mathbf{H}}_{dual}\ddot{\tilde{X}}_{manip,dual} + \tilde{\mathbf{C}}_{dual}\dot{\tilde{X}}_{manip,dual} + \tilde{\mathbf{G}}_{dual} = \tilde{\mathbf{B}}_{dual}\begin{bmatrix} F_{zb,1} \\ F_{zb,2} \\ M_{\theta,1} \\ M_{\theta,2} \end{bmatrix} \tag{3.59}$$

where

$$\tilde{\mathbf{H}}_{dual} = \begin{bmatrix} m_B + 2m_Q & 0 & 0 & m_Q r\sin\theta_1 & -m_Q r\sin\theta_2 \\ 0 & m_Q & 0 & 0 & 0 \\ 0 & 0 & m_Q & 0 & 0 \\ m_Q r\sin\theta_1 & 0 & 0 & m_Q r^2\sin^2\theta + J_y & 0 \\ -m_Q r\sin\theta_2 & 0 & 0 & 0 & m_Q r^2\sin^2\theta + J_y \end{bmatrix} \tag{3.60}$$

$$\tilde{\mathbf{G}}_{dual} = \begin{bmatrix} 0 \\ m_Q g \\ m_Q g \\ 0 \\ 0 \end{bmatrix} \tag{3.61}$$

$$\tilde{\mathbf{B}}_{dual} = \begin{bmatrix} \sin\theta_1 & \sin\theta_2 & 0 & 0 \\ \cos\theta_1 & 0 & 0 & 0 \\ 0 & \cos\theta_2 & 0 & 0 \\ r\sin^2\theta & 0 & 1 & 0 \\ 0 & r\sin^2\theta & 0 & 1 \end{bmatrix} \tag{3.62}$$

We will defer the derivation of contents of coriolis matrix $\tilde{\mathbf{C}}_{dual}$ until later.

We are now ready to partition the dynamical system in Equation-(3.59) into three sub-systems with the following degrees of freedom: x_E , $\tilde{X}_{manip,1} = \begin{bmatrix} z_{Q,1} & \theta_1 \end{bmatrix}^T$ and $\tilde{X}_{manip,2} = \begin{bmatrix} z_{Q,2} & \theta_2 \end{bmatrix}^T$.

We can write the dynamics for the three sub systems as

$$\begin{gathered} \left(m_B + 2m_Q\right)\ddot{x}_E + m_Q r\sin\theta_1\ddot{\theta}_1 - m_Q r\sin\theta_2\ddot{\theta}_2 + \\ m_Q r\left(\dot{\theta}_1\cos\theta_1 - \dot{\theta}_2\cos\theta_2\right) = F_{zb,1}\sin\theta_1 + F_{zb,2}\sin\theta_2 \end{gathered} \tag{3.63}$$

$$\mathbf{H}_1\ddot{\tilde{X}}_{manip,1} + \begin{bmatrix} 0 \\ m_Q r\sin\theta_1\ddot{x}_E \end{bmatrix} + \mathbf{C}_1\dot{\tilde{X}}_{manip,1} + \mathbf{G}_1 = \mathbf{B}_1\begin{bmatrix} F_{zb,1} \\ M_{\theta,1} \end{bmatrix} \tag{3.64}$$

$$\mathbf{H}_2\ddot{\tilde{X}}_{manip,2} + \begin{bmatrix} 0 \\ -m_Q r\sin\theta_2\ddot{x}_E \end{bmatrix} + \mathbf{C}_2\dot{\tilde{X}}_{manip,2} + \mathbf{G}_2 = \mathbf{B}_2\begin{bmatrix} F_{zb,2} \\ M_{\theta,2} \end{bmatrix} \tag{3.65}$$

Where,

$$\mathbf{H}_i = \begin{bmatrix} m_Q & 0 \\ 0 & m_Q r^2 \sin^2\theta_i + J_y \end{bmatrix}$$

$$\mathbf{G}_i = \begin{bmatrix} m_Q g \\ 0 \end{bmatrix} \tag{3.66}$$

$$\mathbf{B}_i = \begin{bmatrix} \cos\theta_i & 0 \\ r\sin^2\theta_i & 1 \end{bmatrix}$$

$$i = 1,2$$

We will not derive $\mathbf{C}_1, \mathbf{C}_2$ as they can be inferred from $\mathbf{H}_1, \mathbf{H}_2$ respectively.

Our goal now is to eliminate $\ddot{x}_E$ from Equations-(3.64) and(3.65). From Equation-(3.63), we have

$$\ddot{x}_E = \frac{F_{zb,1}\sin\theta_1 - m_Q r\sin\theta_1\ddot{\theta}_1}{m_B + 2m_Q} + \frac{F_{zb,2}\sin\theta_2 + m_Q r\sin\theta_2\ddot{\theta}_2}{m_B + 2m_Q} - \frac{m_Q r\left(\dot{\theta}_1\cos\theta_1 - \dot{\theta}_2\cos\theta_2\right)}{m_B + 2m_Q} \tag{3.67}$$

Equation-(3.64) can be expanded to

$$m_Q\ddot{z}_{Q,1} + \mathbf{C}_{1,z_Q} + m_Q g = F_{zb,1}\cos\theta_1 \tag{3.68}$$

and

$$\left(m_Q r^2\sin^2\theta_1 + J_y\right)\ddot{\theta}_1 + m_Q r\sin\theta_1\ddot{x}_E + \mathbf{C}_{1,\theta} = F_{zb,1} r\sin^2\theta + M_{\theta,1} \tag{3.69}$$

Where $\mathbf{C}_{1,z_Q}$ and $\mathbf{C}_{1,\theta}$ are coriolis terms. We now use Equation-(3.67) in Equation-(3.69) to obtain

$$\begin{aligned}&\left(\eta m_Q r^2 \sin^2\theta_1 + J_y\right)\ddot{\theta}_1 + \mathbf{C}_{1,\theta,2} = \\ &F_{zb,1}\eta r \sin^2\theta + M_{\theta,1} - m_Q r \sin\theta_1 \underbrace{\left(\frac{F_{zb,2}\sin\theta_2 + m_Q r \sin\theta_2\ddot{\theta}_2 + m_Q r\cos\theta_2\dot{\theta}_2}{m_B + 2m_Q}\right)}_{\text{External Input}}\end{aligned} \tag{3.70}$$

where

$$\eta = \frac{m_B + m_Q}{m_B + 2m_Q} \tag{3.71}$$

We now multiply Equation-(3.68) with $\eta r \cos\theta_1$ and add it to Equation-(3.70) to arrive at the final dynamics of the system for $\tilde{X}_{manip,1} = \begin{bmatrix} z_{Q,1} & \theta_1 \end{bmatrix}^T$ as

$$\tilde{\mathbf{H}}_1 \ddot{\tilde{X}}_{manip,1} + \tilde{\mathbf{C}}_1 \dot{\tilde{X}}_{manip,1} + \tilde{\mathbf{G}}_1 = \tilde{\mathbf{B}}_1 \begin{bmatrix} F_{zb,1} \\ M_{\theta,1} \end{bmatrix} + \mathbf{B}_{ext,1} F_{ext,1} \tag{3.72}$$

where

$$\begin{aligned}\tilde{\mathbf{H}}_1 &= \begin{bmatrix} m_Q & 0 \\ \eta m_Q r \cos\theta_1 & \eta m_Q r^2 \sin^2\theta_1 + J_y \end{bmatrix} \\ \tilde{\mathbf{G}}_1 &= \begin{bmatrix} m_Q g \\ \eta m_Q g r \cos\theta_1 \end{bmatrix} \\ \tilde{\mathbf{B}}_1 &= \begin{bmatrix} \cos\theta_1 & 0 \\ \eta r & 1 \end{bmatrix}\end{aligned} \tag{3.73}$$

and

$$\mathbf{B}_{ext,1} = \begin{bmatrix} 0 \\ (1-\eta)\sin\theta_1 \end{bmatrix} \tag{3.74}$$

$$F_{ext,1} = F_{zb,2}\sin\theta_2 + m_Q r \sin\theta_2 \ddot{\theta}_2 + m_Q r \cos\theta_2 \dot{\theta}_2 \tag{3.75}$$

According to Equation-(3.72), we see that the input to the dynamics of θ_1 is

$$\eta r F_{zb,1} + M_{\theta,1} - (1-\eta) r \sin\theta_1 \underbrace{\left(F_{zb,2}\sin\theta_2 + m_Q r \sin\theta_2 \ddot{\theta}_2 + m_Q r \cos\theta_2 \dot{\theta}_2\right)}_{\text{External Input}} \tag{3.76}$$

In practical cases, the object mass $m_B \gg m_Q$ which means, $\eta \approx 1$. Further, we limit θ_1 such that $|\theta_1| < \frac{\pi}{6}$ This allows us to make reasonable approximations to the external input part in Equation-(3.76) by ignoring $\ddot{\theta}_2, \dot{\theta}_2$ and assuming $F_{zb,2} \approx \frac{m_Q g}{\cos\theta_1}$. Thus, Equation (3.76) can be re written as

$$\eta r F_{zb,1} + M_{\theta,1} - (1-\eta) r \sin\theta_1 \underbrace{\left(m_Q g \tan\theta_2\right)}_{\text{External Input}} \tag{3.77}$$

Thus, we can approximate Equation-(3.75) as

$$F_{ext,1} \approx m_Q g \tan\theta_2 \tag{3.78}$$

The coriolis matrix $\tilde{\mathbf{C}}_1$ can be derived from the inertia matrix $\tilde{\mathbf{H}}_1$ using the following definition [37]

$$\tilde{\mathbf{C}}_{ij} = \frac{1}{2}\dot{\tilde{\mathbf{H}}}_{ij} + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial \tilde{\mathbf{H}}_{ik}}{\partial q_j} - \frac{\partial \tilde{\mathbf{H}}_{jk}}{\partial q_i}\right)\dot{q}_k \tag{3.79}$$

Where $q = \tilde{X}_{manip,1}$. Using Equation-(3.79), we can show that

$$\tilde{\mathbf{C}}_1 = \begin{bmatrix} 0 & 0 \\ -\eta m_Q r \dfrac{\sin(\theta_1)}{2}\dot{\theta}_1 & \eta m_Q r^2 \dfrac{\sin(2\theta_1)}{2}\dot{\theta}_1 \end{bmatrix} \tag{3.80}$$

Similarly, we can derive the dynamics of the second Quadrotor as

$$\tilde{\mathbf{H}}_2 \ddot{X}_{manip,2} + \tilde{\mathbf{C}}_2 \dot{X}_{manip,2} + \tilde{\mathbf{G}}_2 = \tilde{\mathbf{B}}_2 \begin{bmatrix} F_{zb,2} \\ M_{\theta,2} \end{bmatrix} + \mathbf{B}_{ext,2} F_{ext,2} \tag{3.81}$$

where

$$\tilde{\mathbf{H}}_2 = \begin{bmatrix} m_Q & 0 \\ \eta m_Q r \cos\theta_2 & \eta m_Q r^2 \sin^2\theta_2 + J_y \end{bmatrix}$$

$$\tilde{\mathbf{C}}_2 = \begin{bmatrix} 0 & 0 \\ -\eta m_Q r \frac{\sin(\theta_2)}{2}\dot{\theta}_2 & \eta m_Q r^2 \frac{\sin(2\theta_2)}{2}\dot{\theta}_2 \end{bmatrix}$$

$$\tilde{\mathbf{G}}_2 = \begin{bmatrix} m_Q g \\ \eta m_Q g r \cos\theta_2 \end{bmatrix}$$

$$\tilde{\mathbf{B}}_2 = \begin{bmatrix} \cos\theta_2 & 0 \\ \eta r & 1 \end{bmatrix}$$

$$\mathbf{B}_{ext,2} = \begin{bmatrix} 0 \\ -(1-\eta)\sin\theta_2 \end{bmatrix}$$

$$F_{ext,2} \approx m_Q g \tan\theta_1 \tag{3.82}$$

3.3.8 **Contact Stability Condition for the Two Quadrotor Case**

Here we provide the contact stability condition for the case of two Quadrotor manipulation. That is, we seek to determine a region of valid combinations of object force (Equation-(3.54)) and grasp force (Equation-(3.55)) such that the contact stability conditions below are satisfied.

$$\begin{gathered} F_{cont,\min} \le F_{cont,1} \le F_{cont,\max} \\ F_{cont,\min} \le F_{cont,2} \le F_{cont,\max} \end{gathered} \tag{3.83}$$

The force contribution from each of the Quadrotors is computed using Equation-(3.48)-(3.49) as

$$F_{push,1} \approx m_Q g \tan\theta_1 \tag{3.84}$$

$$F_{push,2} \approx m_Q g \tan\theta_2 \tag{3.85}$$

The contact forces are always acting into the body with reference to the Quadrotor. Therefore, we choose to have positive sign convention for both the contact forces, that is, to ensure that the end-effector does not detach from the Object, we need to ensure $F_{cont,1} > 0, F_{cont,2} > 0$. This permits us to write a general contact stability condition of the form given in Equation-(3.53).

The contact forces are related to push forces as

$$F_{cont,1} \approx \eta F_{push,1} - (1-\eta) F_{push,2} \tag{3.86}$$

$$F_{cont,2} \approx -\eta F_{push,2} + (1-\eta) F_{push,1} \tag{3.87}$$

We note that

$$\frac{1}{2} \le \eta \le 1 \tag{3.88}$$

Let us assume a non-prehensile contact, that is $F_{cont,\min} = 0$. The contact stability condition of Equation-(3.53) becomes

$$\begin{aligned} 0 \le F_{cont,1} \le F_{cont,\max} \\ 0 \le F_{cont,2} \le F_{cont,\max} \end{aligned} \tag{3.89}$$

We also need to ensure that $|\theta_1| < \theta_{\max}, |\theta_2| < \theta_{\max}$ in order to ensure stability of the system. $\theta_{\max}$ is determined by empirical rules based on maximum allowed angles.

Thus, we have the following condition on the push forces as

$$\begin{bmatrix} -m_Q g \tan(\theta_{\max}) \\ -m_Q g \tan(\theta_{\max}) \end{bmatrix} \leq \begin{bmatrix} F_{push,1} \\ F_{push,2} \end{bmatrix} \leq \begin{bmatrix} m_Q g \tan(\theta_{\max}) \\ m_Q g \tan(\theta_{\max}) \end{bmatrix} \tag{3.90}$$

From Equation-(3.86) we see that

$$F_{cont,\max} = m_Q g \tan(\theta_{\max}) \tag{3.91}$$

Hence, we have

$$\mathbf{0} \leq \begin{bmatrix} F_{cont,1} \\ F_{cont,2} \end{bmatrix} \leq \begin{bmatrix} m_Q g \tan(\theta_{\max}) \\ m_Q g \tan(\theta_{\max}) \end{bmatrix} \tag{3.92}$$

Let us construct a matrix form of Equation-(3.86) and (3.87) as

$$\begin{bmatrix} F_{cont,1} \\ F_{cont,2} \end{bmatrix} = \underbrace{\begin{bmatrix} \eta & -(1-\eta) \\ (1-\eta) & -\eta \end{bmatrix}}_{=M} \begin{bmatrix} F_{push,1} \\ F_{push,2} \end{bmatrix} \tag{3.93}$$

Our goal now is to find a subspace for $\begin{bmatrix} F_{cont,1} \\ F_{cont,2} \end{bmatrix}$ and $\begin{bmatrix} F_{push,1} \\ F_{push,2} \end{bmatrix}$ such that Equation-(3.90) ,(3.92) and (3.93) are all valid.

This subspace is determined using a geometric approach which is briefly summarized below. See Figure 3.5 for an illustration of all regions defined below:

1. We start with Equation-(3.90) and denote it as ***region1***.
2. We apply the transformation M defined in (2,11) to ***region1*** to result in ***region2***.
3. Since Eq. (2.10) has to be satisfied, the governing region is a subset of ***region2***, and is denoted as ***region3***. This region provides the bounds for $\begin{bmatrix} F_{cont,1} \\ F_{cont,2} \end{bmatrix}$.
4. Using ***region3*** and M^{-1}, we arrive at ***region4*** which illustrates the bounds on $\begin{bmatrix} F_{push,1} \\ F_{push,2} \end{bmatrix}$ for which contact stability is guaranteed.

Using ***region4***, we now derive bounds on $\begin{bmatrix} F_{object} \\ F_{object,grasp} \end{bmatrix}$ for which contact stability is guaranteed. The net force acting on the Object is given by Equation-(3.54)

$$\begin{aligned} F_{object} &= F_{cont,1} - F_{cont,2} \\ F_{object} &= (2\eta - 1)\left(F_{push,1} + F_{push,2}\right) \approx (2\eta - 1)\left(m_Q g \tan\theta_1 + m_Q g \tan\theta_2\right) \end{aligned} \tag{3.94}$$

The net grasp force acting on the Object is given by Equation-(3.55)

$$\begin{aligned} F_{object,grasp} &= \frac{F_{cont,1} + F_{cont,2}}{2} \\ F_{object,grasp} &= (2\eta - 1)\frac{F_{push,1} - F_{push,2}}{2} \approx (2\eta - 1)\frac{\left(m_Q g \tan\theta_1 - m_Q g \tan\theta_2\right)}{2} \end{aligned} \tag{3.95}$$

We can write Equation-(3.54) and (3.55) in matrix form as

$$\begin{bmatrix} F_{object} \\ F_{object,grasp} \end{bmatrix} = (2\eta - 1)\begin{bmatrix} 1 & 1 \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}\begin{bmatrix} F_{push,1} \\ F_{push,2} \end{bmatrix} \tag{3.96}$$

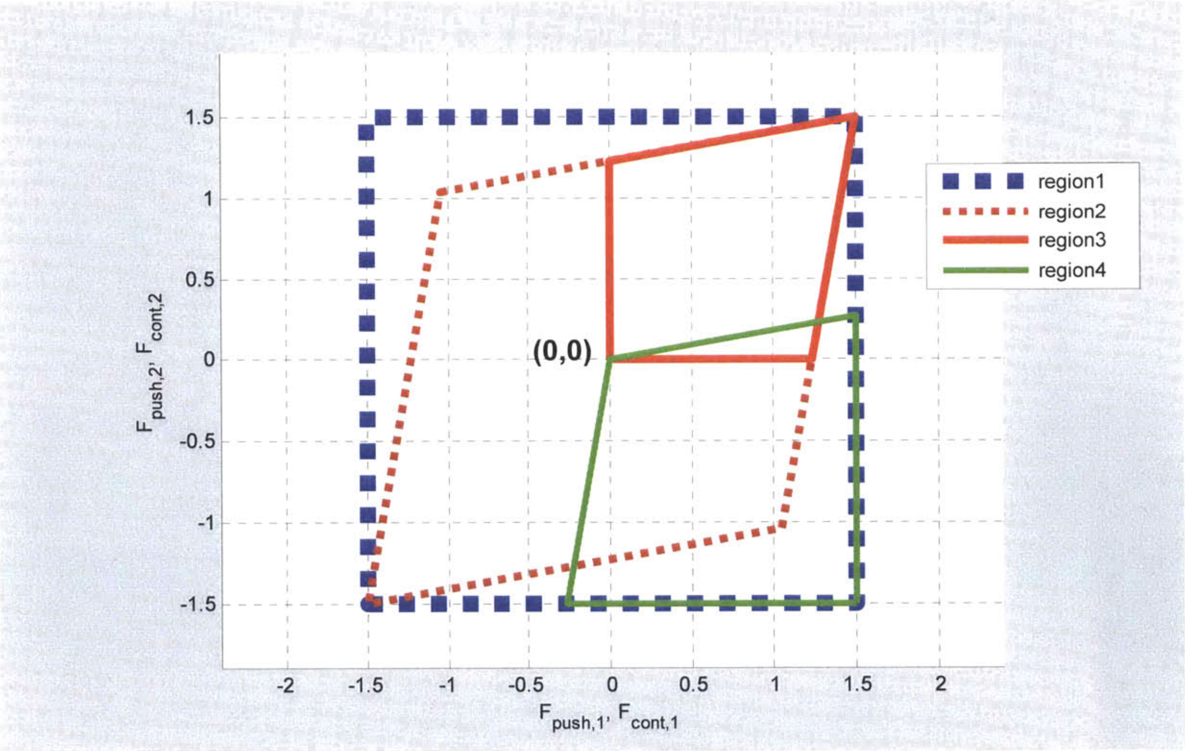

Figure 3.5: Region of stability

Using Equation-**Error! Reference source not found.** and ***region4***, we can find the subspace for $\begin{bmatrix} F_{object} \\ F_{object,grasp} \end{bmatrix}$ that ensures contact stability. This subspace is shown as ***region5*** plotted in black trace in Figure 3.6. It is easy to see that $\begin{bmatrix} F_{object} \\ F_{object,grasp} \end{bmatrix}$ has to lie in ***region5*** indicated in Figure 3.6 and Figure 3.7.

We can explicitly solve for the vertices of ***region5***, as shown in Figure 3.7, as

$$W_1 = \begin{bmatrix} \dfrac{(2\eta-1)}{n} \\ \dfrac{(2\eta-1)^2}{2n} \end{bmatrix} m_Q g \tan(\theta_{\max}) \tag{3.97}$$

$$W_3 = \begin{bmatrix} -\dfrac{(2\eta-1)}{n} \\ \dfrac{(2\eta-1)^2}{2n} \end{bmatrix} m_Q g \tan(\theta_{\max}) \tag{3.98}$$

$$W_4 = \begin{bmatrix} 0 \\ (2\eta-1) \end{bmatrix} m_Q g \tan(\theta_{\max}) \tag{3.99}$$

$$W_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} m_Q g \tan(\theta_{\max}) \tag{3.100}$$

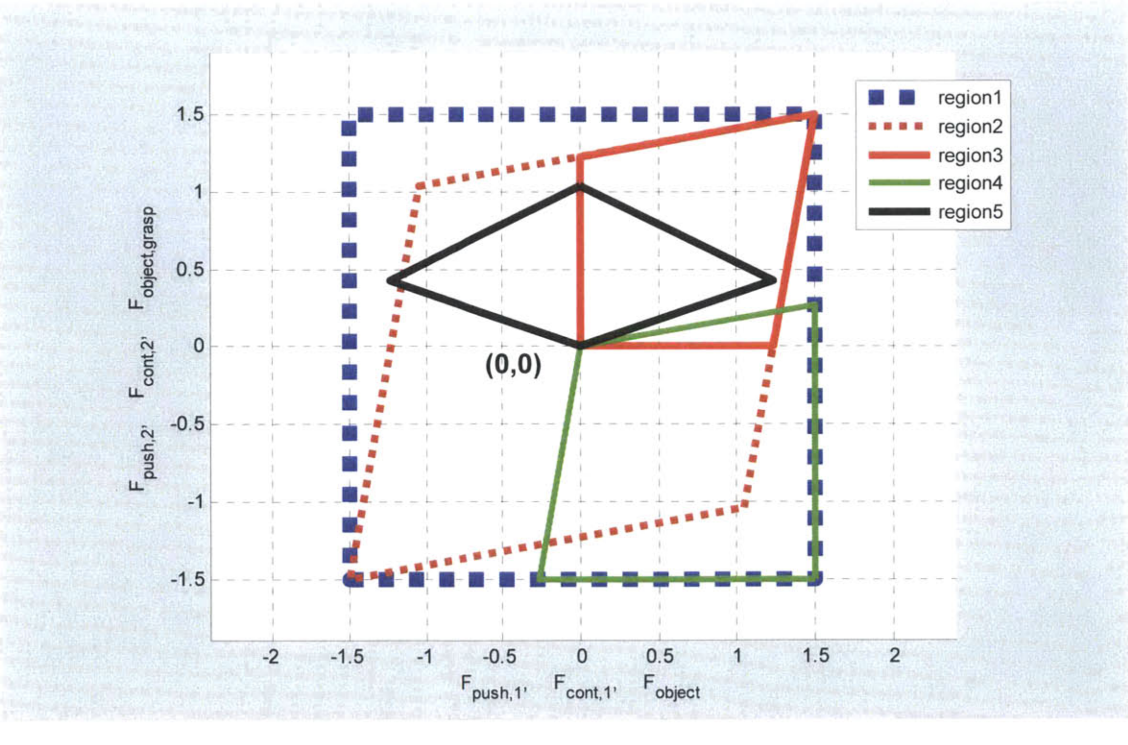

Figure 3.6: Region of stability. The black trace enclosing *region5* corresponds to the subspace of $\begin{bmatrix} F_{object} \\ F_{object,grasp} \end{bmatrix}$ that satisfies contact stability. As expected, $F_{object,grasp} > 0$ suggesting a positive grasp force, and F_{object} has both positive and negative excursions.

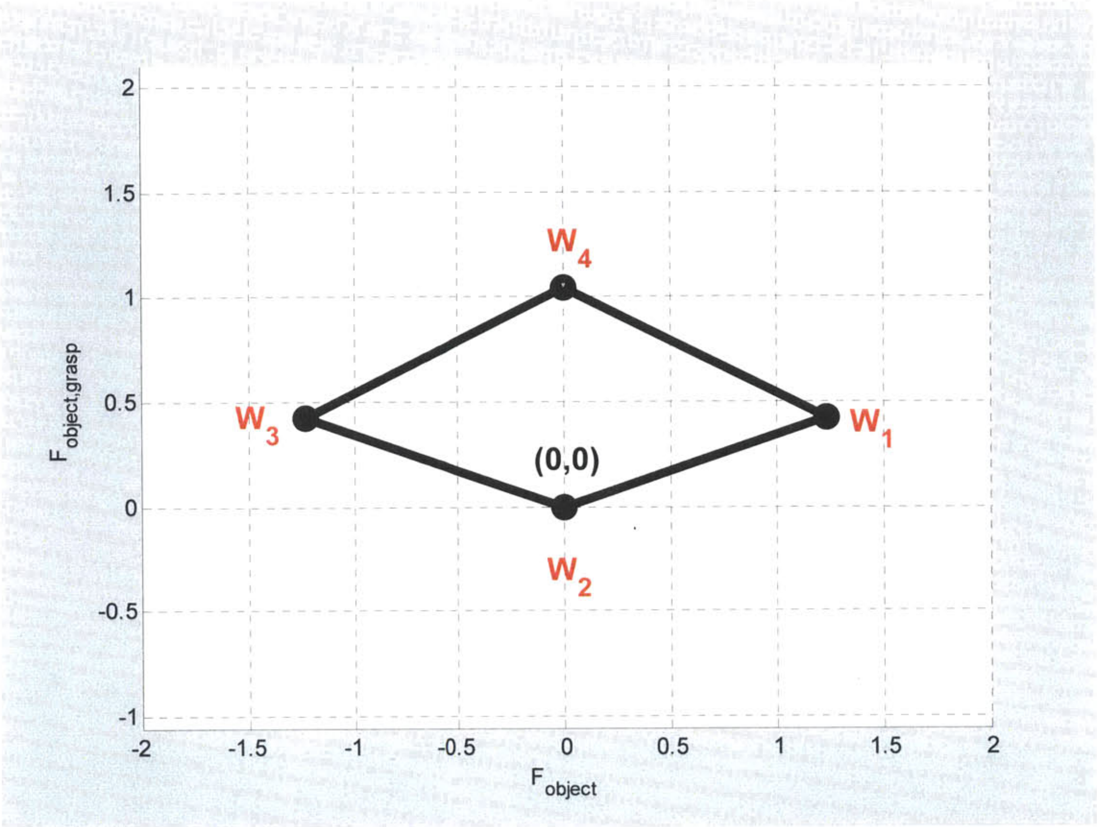

Figure 3.7: region5 enclosed within the black trace is the region of permissible subspace for $\begin{bmatrix} F_{object} \\ F_{object,grasp} \end{bmatrix}$.

In summary, during a manipulation process, $\begin{bmatrix} F_{object} \\ F_{object,grasp} \end{bmatrix}$ demanded by the control algorithm should essentially lie within ***region5*** in order to ensure contact stability and also to ensure $|\theta_1| < \theta_{\max}, |\theta_2| < \theta_{\max}$

.

In case the demanded point lies outside of ***region5***, a nearest point inside of ***region5*** can be solved for using a geometric method, for example.

3.4 **Summary**

In this section we derived the dynamics for the single and two Quadrotors manipulating an Object. In the next chapter, we will use these dynamical models to design control schemes in order to perform a given manipulation task. We will use the conditions we derived for the Object force and the grasp force to design shape reference inputs that satisfy contact stability conditions.

4 Manipulation Using Single and Multiple Quadrotors

In this chapter we propose several manipulation schemes using Quadrotors. At the heart of the manipulation is the contact based control design. In the previous chapter, we derived the dynamics. Here we present the necessary control algorithms and study the performance in simulation and experiments. We note the three basic steps of manipulation, illustrated in Figure 4.1.

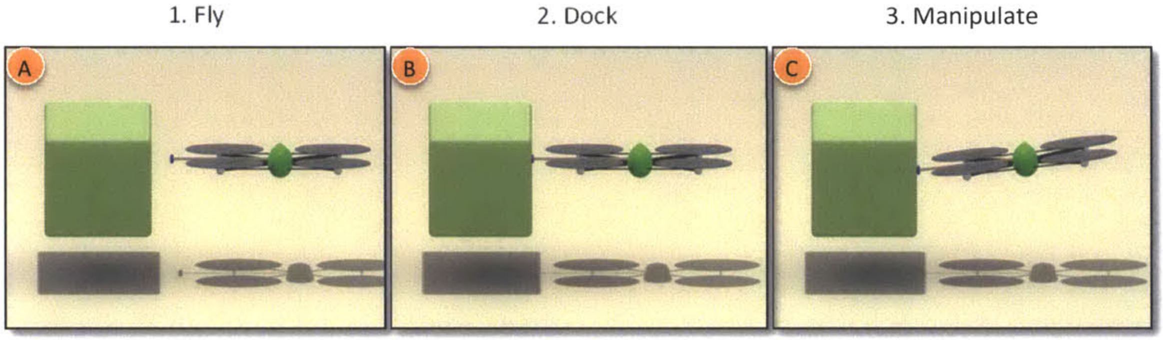

Figure 4.1: Basic steps in manipulation. (A) The UAV flies close to the object and ensures the end-effector is located at the designated docking position on the object. Here the UAV is in free-flight mode and uses a free-flight controller to perform this step. (B) The UAV docks to the object, at which point the contact dynamics is applicable. From this point onwards, the controller is switched to manipulation controller. (C) By applying a desired amount of force, the UAV performs the manipulation. After the manipulation, the UAV is undocked and the controller is switched to free-flight controller.

We first present a very basic version of free-flight controller. Control of a free-flying Quadrotor has been dealt in a number of research articles [4, 5, 19, 28-33, 35, 39-42]. These controllers however are not readily applicable to our case because the dynamics of the Quadrotor in contact differs significantly.

We then present the manipulation controller that we design based on the dynamics derived in 3. We assume that the object is constrained to move along x-axis, which is the desired direction of manipulation. In chapter-5, we remove this restriction and the object is free to move in x-y plane.

We make an assumption that we can precisely switch the controller from flight-controller to manipulation-controller at the instant of docking. This requires that we accurately determine the occurrence of docking event. Any uncertainty in physical parameters, such as the object location can hinder the switching process. To address this uncertainty, we propose a robust docking scheme. This scheme not only addresses the above mentioned uncertainty, but also permits docking at high speeds. We demonstrate the efficacy of the proposed controller in simulation and experimental setup. STEVE system presented in Chapter-2 is used to perform the experiments.

4.1 Control of Quadrotor in Free Flight

Here we summarize the controller for the Quadrotor to perform free-flight maneuvers. The simplified dynamics of the Quadrotor are given by Equations-(3.11)-(3.13). A Quadrotor command consists of desired location and a heading angle, denoted as $[x_{des}, y_{des}, z_{des}, \psi_{des}]^T$.

Altitude of the Quadrotor z_Q is controlled using a PID controller with gravity compensation as

$$F_{zb} = \frac{m_Q g + C_{PID}^{Altitide}\left(z_{Q,des} - z_Q\right)}{\cos\theta\cos\phi} \tag{4.1}$$

We use a generic PID control structures as

$$C_{PID}\left(b-a\right) = K_P\left(b-a\right) + K_D\left(\dot{b}-\dot{a}\right) + K_I\int\left(b-a\right) \tag{4.2}$$

The control gains in Equation-(4.2) are designed using classical methods and are fine-tuned through experimental trials.

The translation controller that moves the Quadrotor to desired location x_{des}, y_{des} is designed using a PID control structure. A coordinate transform matrix $\mathbf{R}_\psi$ is involved to account for yaw angle ψ. The controller has the form

$$\begin{bmatrix} \phi_{des} \\ \theta_{des} \end{bmatrix} = \mathbf{C}_{PID}^{Translation}\left(\mathbf{R}_\psi \begin{bmatrix} x_Q - x_{des} \\ y_Q - y_{des} \end{bmatrix}\right) \tag{4.3}$$

We now implement the PID controller for angle sub-system as

$$\begin{bmatrix} M_\phi \\ M_\theta \\ M_\psi \end{bmatrix} = \mathbf{C}_{PID}^{Angular}\left(\begin{bmatrix} \phi-\phi_{des} \\ \theta-\theta_{des} \\ \psi-\psi_{des} \end{bmatrix}\right) \tag{4.4}$$

The translation system provides the desired roll and pitch angles that feed the angle-sub system in Equation-(4.4). Complete description of a Quadrotor controller can be found in [4, 5].

4.2 Control of Quadrotor in Contact

Here we present controllers for the dynamics we derived in Equation-(3.16) and Equation-(3.30). That is, we design controller for X_{free} and $\tilde{X}_{manip}$, which accounts for a total of five degrees of freedom. The remaining one degree of freedom, x_E, is left uncontrolled. Instead, we control the force along this degree of freedom by appropriate choice of $\tilde{X}_{manip,des}$. The following sections present the control of the two sub systems.

4.2.1 X_{free} Controller

The goal of X_{free} controller is to ensure

$$X_{free} \to X_{free,des} \tag{4.5}$$

where

$$X_{free,des} = \begin{bmatrix} y_{Q,des} \\ \phi_{des} \\ \psi_{des} \end{bmatrix} \tag{4.6}$$

Note that according to Equation-(3.16), y_Q depends on ϕ. Therefore, we cannot independently specify $y_{Q,des}$ and ϕ_{des}. We specify $y_{Q,des}$ only and ϕ_{des} follow according to the control law defined in Equation-(4.9).

Using Equation-(3.16), we can design a feedback linearizing controller for M_ϕ and M_ψ as

$$\begin{bmatrix} M_\phi \\ M_\psi \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} -k_{d\phi}\left(\dot{\phi}-\dot{\phi}_{des}\right)-k_{p\phi}\left(\phi-\phi_{des}\right) \\ -k_{d\psi}\left(\dot{\psi}-\dot{\psi}_{des}\right)-k_{p\psi}\left(\psi-\psi_{des}\right) \end{bmatrix} \tag{4.7}$$

By suitably picking the gains, we can ensure $\psi \to \psi_{des}$ and $\phi \to \phi_{des}$.

With $\psi \to \psi_{des}, \phi \to \phi_{des}$, and noting that $F_{zb} \approx m_Q g$ (which will be shown in next section), and using Equation-(3.16), we note that

$$\ddot{y}_Q = \left(\tan\theta \sin\psi - \phi\cos\psi\right) g \tag{4.8}$$

Since the dynamics of $\ddot{y}_Q$ depends on ϕ, we see that we cannot simultaneously control both y_Q and ϕ . Since our goal is to control y_Q, we find a control function for ϕ_{des} that ensures $y_Q \to y_{Q,des}$ as

$$\phi_{des}\cos\psi = \left(k_{dy}\left(\dot{y}_Q - \dot{y}_{Q,des}\right) + k_{py}\left(y_Q - y_{Q,des}\right)\right) - g\tan\theta\sin\psi \tag{4.9}$$

ϕ_{des} obtained in Equation-(4.9) is used in Equation-(4.7). With $\phi \to \phi_{des}$, we see that $y_Q \to y_{Q,des}$. Note that when $\psi = \frac{\pi}{2}$ (which is a singularity point) Equation-(4.9) cannot be used to solve for ϕ_{des}. This problem however can be overcome by making a suitable transformation to $[x_Q, y_Q]^T$ as done elsewhere in the references given in the beginning of the chapter. We also note the presence of θ in Equation-(4.7)-(4.9), which are external inputs as far as system of Equation-(3.16) is concerned.

4.2.2 $\tilde{X}_{manip}$ Controller

The goal of $\tilde{X}_{manip}$ controller is to ensure

$$\tilde{X}_{manip} \rightarrow \tilde{X}_{manip,des} \tag{4.10}$$

where

$$\tilde{X}_{manip,des} = \begin{bmatrix} z_{Q,des} \\ \theta_{des} \end{bmatrix} \tag{4.11}$$

Since we have the system in the standard manipulator form in Equation-(3.30), we can design a dynamic inversion controller as

$$\begin{bmatrix} F_{zb} \\ M_{\theta} \end{bmatrix} = \tilde{\mathbf{B}}^{-1}\left(\tilde{\mathbf{H}}\tilde{V} + \tilde{\mathbf{C}}\dot{\tilde{X}}_{manip} + \tilde{\mathbf{G}}\right) \tag{4.12}$$

where

$$\tilde{V} = K_{Xp}\left(\tilde{X}_{manip,des} - \tilde{X}_{manip}\right) + K_{Xd}\left(\dot{\tilde{X}}_{manip,des} - \dot{\tilde{X}}_{manip}\right) + \left(\ddot{\tilde{X}}_{manip,des}\right) \tag{4.13}$$

K_{Xp}, K_{Xd} are positive diagonal matrices.

It can be seen that during the steady state the control inputs are given by

$$\begin{bmatrix} F_{zb} \\ M_\theta \end{bmatrix}_{steady} = \tilde{\mathbf{B}}^{-1}\tilde{\mathbf{G}} = \begin{bmatrix} \dfrac{m_Q g}{\cos\theta} \\ \eta m_Q g r\left(\cos\theta - \dfrac{1}{\cos\theta}\right) \end{bmatrix} \tag{4.14}$$

These values suggest the operating point for the Quadrotor and permits knowing the limits for θ given the actuator saturation.

4.2.3 $\tilde{X}_{manip,1}$ and $\tilde{X}_{manip,2}$ Controllers for Two Quadrotor Case

Here we focus on designing the controllers for the two Quadrotor case. We derived the dynamics as Equation-(3.46) and (3.47). The main difference in comparison to single Quadrotor case as in Equation-(3.30) is the presence of external force input. Since this external input is a known force, we account for this in the inverse dynamic controller as

$$\begin{bmatrix} F_{zb,1} \\ M_{\theta,1} \end{bmatrix} = \tilde{\mathbf{B}}_1^{-1}\left(\tilde{\mathbf{H}}_1\tilde{V}_1 + \tilde{\mathbf{C}}_1\dot{\tilde{X}}_{manip,1} + \tilde{\mathbf{G}}_1 + \tilde{\mathbf{B}}_{ext,1}F_{ext,1}\right) \tag{4.15}$$

$$\begin{bmatrix} F_{zb,2} \\ M_{\theta,2} \end{bmatrix} = \tilde{\mathbf{B}}_2^{-1}\left(\tilde{\mathbf{H}}_2\tilde{V}_2 + \tilde{\mathbf{C}}_2\dot{\tilde{X}}_{manip,2} + \tilde{\mathbf{G}}_2 + \tilde{\mathbf{B}}_{ext,2}F_{ext,2}\right) \tag{4.16}$$

We note that $X_{free,1}$ and $X_{free,2}$ controller will be same as the one presented in Section-0. In the following section we present how these trajectories are designed within the constraints of the system in order to perform a given manipulation task.

4.3 **Desired Trajectories for Manipulation**

In the previous section we presented the controllers for X_{free} and $\tilde{X}_{manip}$ that ensured tracking, flight stability and generating the necessary push force. Here we design manipulation schemes using single and two Quadrotors using the controllers we designed. That is, we explicitly construct the desired commands $X_{free,des}$ and $\tilde{X}_{manip,des}$ discussed in Section-4. Since the Object is constrained to move only along $x-axis$, we can assume

$$X_{free,des} = \begin{bmatrix} y_{E,des} = 0 \\ \phi_{des} = 0 \\ \psi_{des} = 0 \end{bmatrix} \qquad (4.17)$$

This ensures that the Quadrotors are aligned with the Object surface during the manipulation. Any disturbance along X_{free} degrees are handled by the X_{free} Controller presented in the preceding section.

We will now construct $\tilde{X}_{manip}$ as follows. Since we do not want the end-effector to slide on the surface of the Object during manipulation, we select $z_{E,des} = 0$. Since $\tilde{X}_{manip}$ has z_Q, we select

$$z_{Q,des} = z_{E,des} + r\sin\theta_{des} \qquad (4.18)$$

which ensures $z_E \to z_{E,des}$ as $z_{Q,des} \to z_Q, \theta \to \theta_{des}$. The choice of θ_{des} is based on the desired force input on the object, which we will derive separately for single and two Quadrotor cases in Section-5.1 and 5.2 respectively.

Remark: In order for the Quadrotor to apply forces and sustain flight stability, not only it is necessary that the contact stability is met, but also that the Quadrotor push force should act normal to the surface of the Object. If the push force is not normal, then the end-effector will start to slide on the surface causing a yaw moment to on the Quadrotor. If the magnitude of the sliding moment is greater than what the Quadrotor yaw-controller can counter, the flight stability cannot be ensured. In what follows, we assume that the push force always acts normal to the surface of the object.

4.3.1 **Manipulation Using a Single Quadrotor**

In the case of single UAV manipulation, the only force acting on the Object is due to the Quadrotor. This force is given by Equation-(3.27). Given a desired push force $F_{push,des}$, we design θ_{des} as

$$\theta_{des} \approx \tan^{-1}\left(\frac{F_{push,des}}{\eta m_Q g}\right) \tag{4.19}$$

The acceleration of the Object $\ddot{x}_E$ is related to the applied force by Equation-(3.23). Given a desired trajectory for the Object $x_{E,des}(t)$, we compute the force necessary to perform trajectory tracking using a PID control law on the tracking error as

$$F_{push,des} = C_{PID}\left(x_{E,des} - x_E\right) \tag{4.20}$$

Equation-(4.20), however can cause $F_{push,des}$ to have both positive and negative excursions for an arbitrary $x_{E,des}$. Assuming a prehensile contact (ex.: end-effector with a magnetic contact), it implies that $F_{contact,min}$ can be negative. Using an estimate of $F_{push,des}$ for a given prehensile end-effector, this lower limit $F_{contact,min}$ is computed. Based on empirical evidence, a θ_{max} for which stable contact is ensured for all $\theta \leq \theta_{max}$ is computed and the upper limit on $F_{push,des}$ is determined as

$$F_{contact,\max} = \eta m_Q g \tan\left(\theta_{\max}\right) \tag{4.21}$$

We now modify Equation-(4.20) to account for the contact stability constraints as

$$F_{push,des} = \begin{cases} C_{PID}\left(x_{E,des} - x_E\right) & if \quad F_{contact,\min} \le C_{PID}\left(x_{E,des} - x_E\right) \le F_{contact,\max} \\ F_{contact,\min} & if \quad F_{contact,\min} > C_{PID}\left(x_{E,des} - x_E\right) \\ F_{contact,\max} & if \quad F_{contact,\max} < C_{PID}\left(x_{E,des} - x_E\right) \end{cases} \tag{4.22}$$

We now summarize the manipulation scheme with following steps

1. Given $x_{E,des}(t)$, compute desired push force using Equation-(4.22)
2. Compute the desired angle θ_{des} Equation-(4.19)
3. Compute the desired altitude using Equation-(4.18)

We note that step (2) and (3) completely specify $\tilde{X}_{manip,des}$. The complete control system that specifies the control inputs $\begin{bmatrix} F_{zb} \\ M_\theta \end{bmatrix}$ and $\begin{bmatrix} M_\phi \\ M_\psi \end{bmatrix}$ are determined by Equation-(4.12) and Equation-(4.7) respectively.

4.3.2 **Manipulation using Two Quadrotors**

Here we assume non-prehensile contact because we can perform bilateral manipulation with two Quadrotors. Given a desired Object trajectory $x_{E,des}\left(t\right)$, using a PID control law, we compute the desired Object force as

$$F_{object,des} = C_{PID}^{dual}\left(x_{E,des} - x_E\right) \tag{4.23}$$

We then pick the desired grasp force $F_{object,grasp,des}$ to be a constant. The goal here is to design $\begin{bmatrix} F_{push,1,des} \\ F_{push,2,des} \end{bmatrix}$. As discussed in Section 3.5.1, these forces in turn generate contact forces $\begin{bmatrix} F_{cont,1,des} \\ F_{cont,2,des} \end{bmatrix}$ which then need to be such that they satisfy the contact stability conditions in Equation-(3.53).

In Figure 3.7 in particular showed the feasible region where $\begin{bmatrix} F_{object,des} \\ F_{object,grasp,des} \end{bmatrix}$ should lie. It is therefore possible to choose $F_{object,grasp,des}$ to have a constant value that lies in this region. The center of the region is one possible value that can be chosen.

We can summarize the manipulation scheme with following steps:

1. Given $x_{E,des}(t)$, compute the desired object force using Equation-(4.23)
2. Pick a grasp force $F_{object,grasp,des} = \frac{(2\eta-1)^2}{2n} m_Q g \tan(\theta_{\max})$, which is according to Equation-(3.98) . This ensures we have a maximum swing for $F_{object,des}$.
3. Ensure $\begin{bmatrix} F_{object} \\ F_{object,grasp} \end{bmatrix}$ is inside ***region5*** defined by the polygon with vertices W_1, W_2, W_3, W_4 defined in Equations-(3.97) to (3.100) .
4. If $\begin{bmatrix} F_{object} \\ F_{object,grasp} \end{bmatrix}$ is outside the polygon, project it to the nearest interior point.
5. Resolve the desired push forces by each of the Quadrotor using Equations-(3.96)
6. With $\begin{bmatrix} F_{push,1,des} \\ F_{push,2,des} \end{bmatrix}$, compute the desired angle using Equation-(3.84) and (3.85).
7. Compute the desired altitudes using Equation-(4.18)
8. We now have a complete description for $\tilde{X}_{manip,des,1}, \tilde{X}_{manip,des,2}$
9. Compute control inputs using Equation-(4.15)and (4.16)
10. Using $X_{free,des}$ Equation-(4.17), compute the control inputs using Equation-(4.7).

With the above recipe for manipulation, we proceed to the following sections that deal with simulation and experiments.

4.4 Simulation and Experiments

Using STEVE described in Chapter-2, we will proceed to conduct several simulations and experiments to demonstrate the performance of the proposed control schemes. In our analysis, we assumed two phases of dynamics. The first is when, the Quadrotor was in free-flight mode and the other in which the

Quadrotor is already docked and performing manipulation. In reality, during the phase transition, the Quadrotor and the object collide with each other, undergoing impacts. Depending on the state of the Quadrotor and the Object at the instant of impact, the resulting dynamics may or may not be stable. This analysis is beyond the scope of this paper. However, the contact dynamics due to impact is included in the simulation studies. The key parameters of the system are listed below:

- Mass of the Quadrotor $m_Q = 440grams$
- Moment of Inertia $J_x = J_y = J_z/2 = 0.001kgm^2$
- Mass of the Object = 2Kgs, End-effector length $r = 0.35m$

We present the simulation of a single Quadrotor pushing an object given a desired push force $F_{push,des}$. We assume the Quadrotor is already docked to the Object. We then present the simulation of docking when the free-flying Quadrotor impacts the Objects with various velocities and angles θ. We then derive an empirical rule that dictates the stable combinations of impact velocity $x_{Q,t_{imp}}$ and angle $\theta_{t_{imp}}$. We propose a scheme that permits stable aggressive docking even in the presence of uncertainty in switching of the controller. We then present the simulation of two Quadrotor manipulating an Object with a given reference trajectory.

4.4.1 **Single Quadrotor Manipulation**

Here we present the Simulation of a Single Quadrotor pushing an Object using the Controller implemented in Section-4.2.2 and 4.2.3. The desired push force $F_{push,des}$, is a series of step functions which leads to a series of steps functions for θ_{des} according to Equation-(4.19), as shown in Figure 4.2. The plot shows the resulting angle $\theta(t)$ and also the actual push force $F_{push}(t)$. We can notice the spikes in the $F_{push}(t)$ that leads to zero push force momentarily. This behavior can be attributed to the step input and ignored dynamics in deriving Equation-(3.27). The Object trajectory is shown in Figure 4.3.

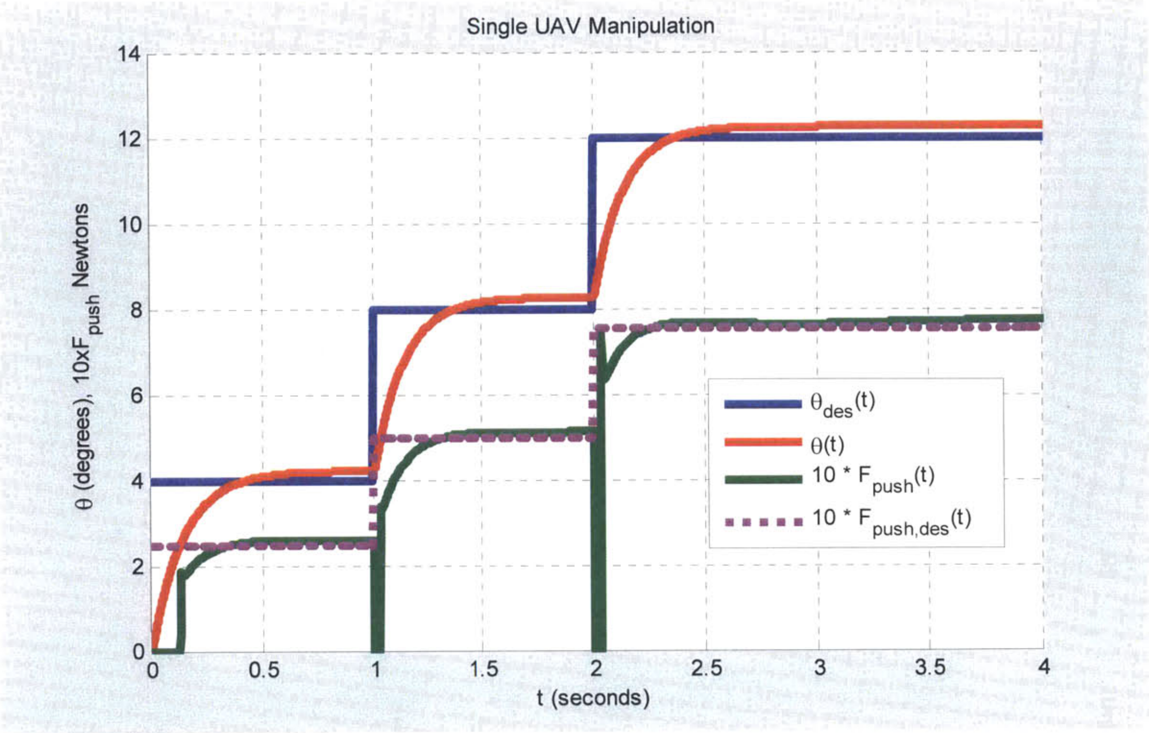

Figure 4.2: Single UAV pushing an Object with a desired push force. The plot shows the desired angle $\theta_{des}(t)$ that corresponds to a desired push force $F_{push,des}$. The resulting $\theta(t)$ and the push foces $F_{push}(t)$ are shown. The forces are maginified 10 times for visibility in the plot.

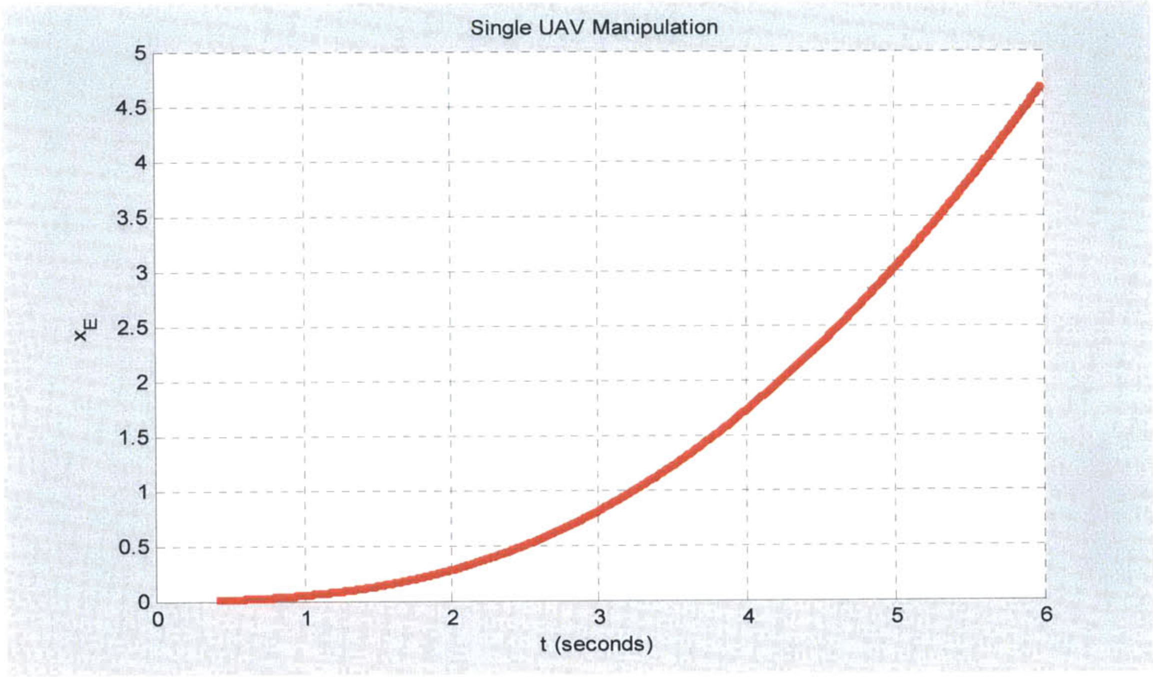

Figure 4.3: Motion of the Object x_E due to a single Quadrotor pushing.

4.4.2 **Two Quadrotor Manipulation**

We follow the recipe presented in Section-4.3.2 to design a tracking controller and to pick the free parameters. We select a constant grasp force as

$$F_{object,grasp,des} = \frac{(2\eta - 1)^2}{2n} m_Q g \tan(\theta_{\max}) = 0.4405N \tag{4.24}$$

with $\theta_{\max} = 20^\circ$ and $\eta = \frac{2+0.44}{2+2\times 0.44} = 0.8472$.

We design a PD controller for computing the object force in Equation-(4.23). We present two simulation results: (1) with $x_{E,des}$ as sinusoidal input and (2) with $x_{E,des}$ as Square wave input. The result of sinusoidal input trajectory is shown in Figure 4.5. Figure 4.5 shows a more aggressive control action for a square wave reference input that leads to a counter intuitive configuration of $\theta_1 < 0$ and $\theta_2 > 0$.

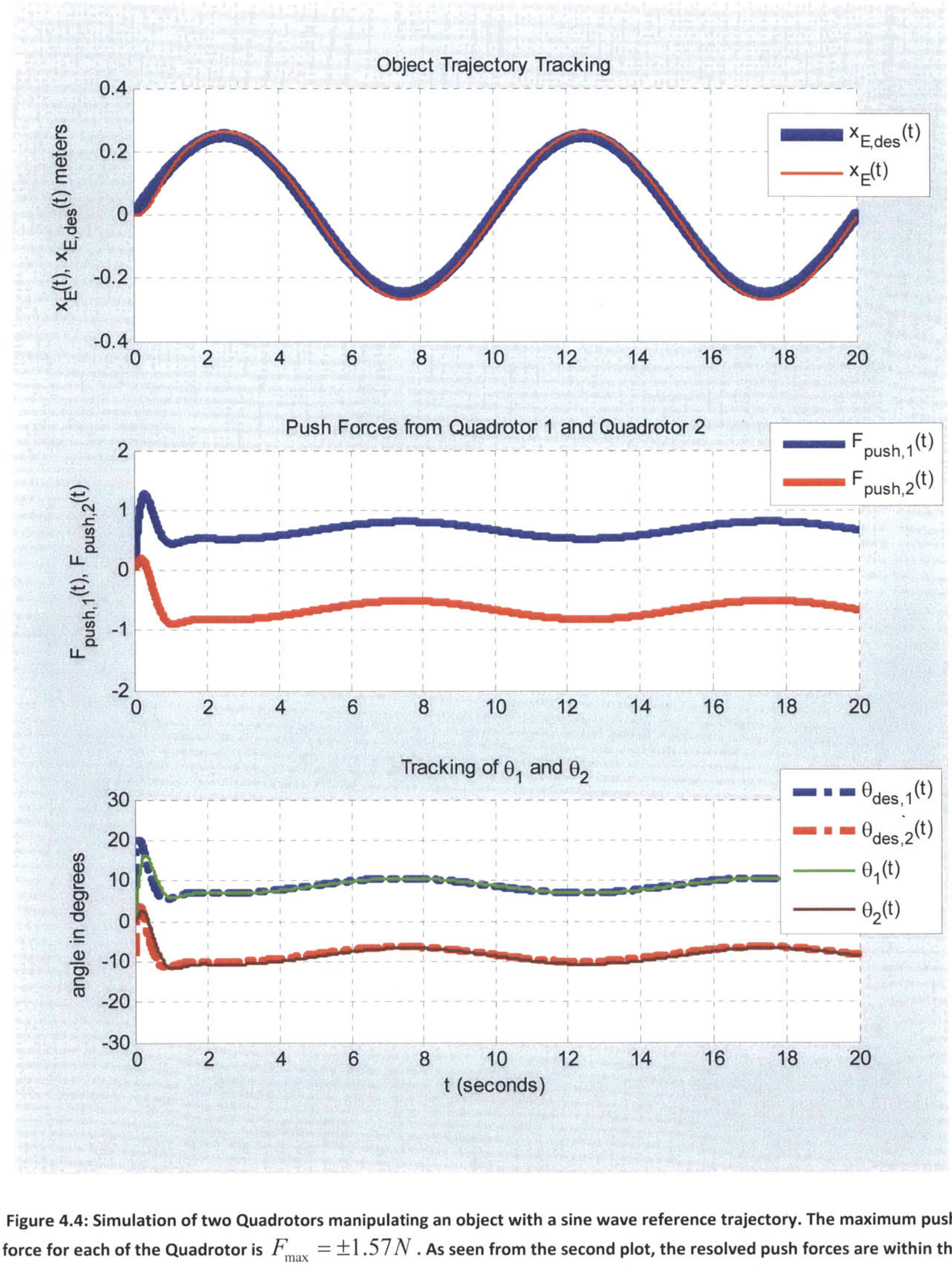

Figure 4.4: Simulation of two Quadrotors manipulating an object with a sine wave reference trajectory. The maximum push force for each of the Quadrotor is $F_{\max} = \pm 1.57N$. As seen from the second plot, the resolved push forces are within the limits.

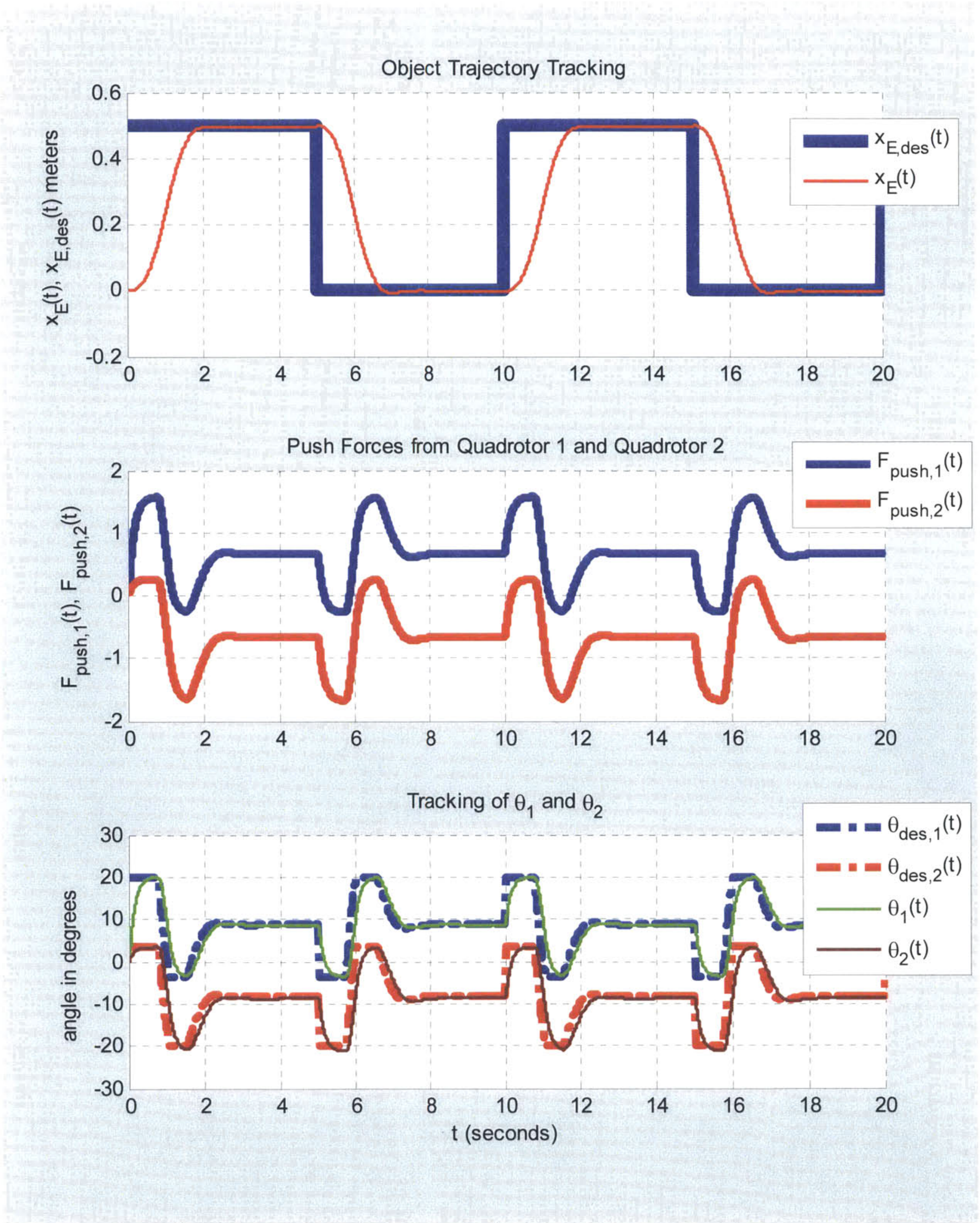

Figure 4.5: Simulation of two Quadrotors manipulating an object with a square wave reference trajectory. The maximum push force for each of the Quadrotor is $F_{\max} = \pm 1.57N$. As seen from the second plot, the resolved push forces are within the limits. However, θ_1, θ_2 are occasionally exceeding $\theta_{\max} = 20^\circ$ because of the ignored dynamics.

4.4.3 Docking and Controller Switching Approach

As mentioned already, the overall manipulation scheme consists of two steps. The first step consisted of a free-flight controller which culminated when the Quadrotor end-effector arriving precisely at the Object surface. The second step consists of a contact-based controller which assumed that docking had occurred. The precise determination of whether or not such a docking has occurred is however quite difficult to determine, and therefore prevents an accurate implementation of a switching controller. Also, if contact were to occur prior to the actual switching of the controller, then a stable motion of the Quadrotor can occur only if the free-flight controller is robust to collisions. Such robustness can be ensured only at very low impact velocities and angles in the Quadrotor. This limits the Quadrotor from any aggressive docking tasks. We first examine the consequences of failing to switch the controller after the moment of impact. We then compare it to the case when the controller is switched to manipulation controller.

A Quadrotor is made to approach and impact a fixed object (non-movable). In the first case, the Quadrotor is controlled with the free-flight controller presented in Section-4.1. Several such simulation experiments are conducted with various θ and impact velocity $\dot{x}_E$. In each case we note if the Quadrotor was stable after the impact, that is, if it reached a stable configuration with $\theta \to \theta_{des}, \dot{x}_E \to 0$. The red-trace in the plot of Figure 4.6 ($x-axis$ represents the pitch angle θ and the $y-axis$ indicates the impact velocity) indicates the 'separatrix' that separates the region of stable combinations of θ and $\dot{x}_E$. Any point below the red-curve results in a stable docking and any point above the red-curve leads to instabilities.

In the second simulation experiments, we perform the same experiment as above, but with the manipulation Controller presented in Section-4.2. The result is plotted in the blue-trace of Figure 4.6.

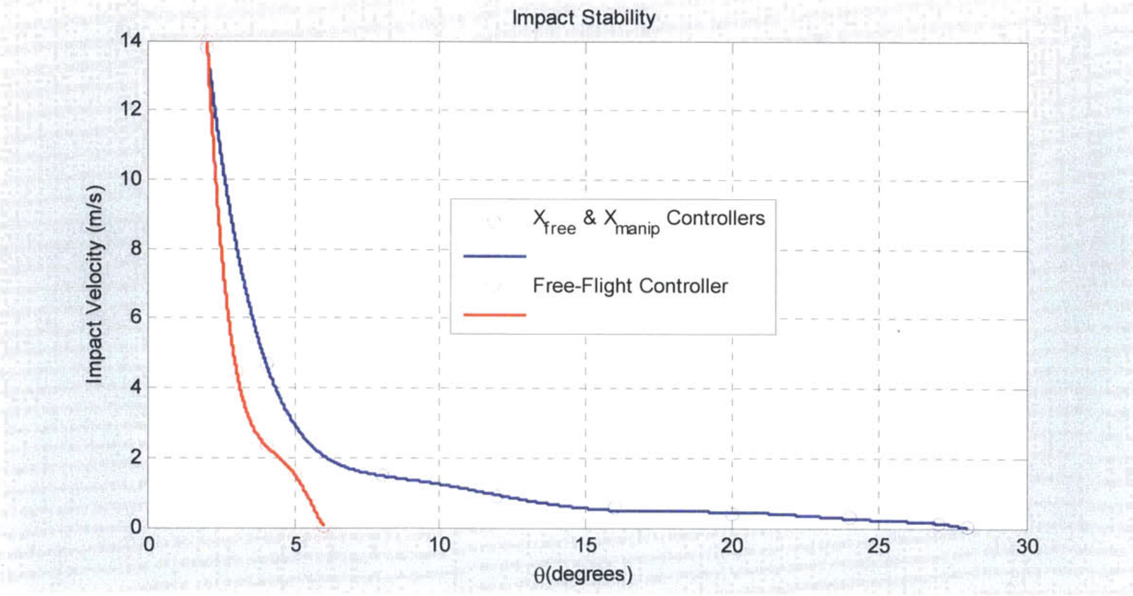

Figure 4.6: Impact experiment to compare the performance of Free-flight controller with Manipulation Controller.

Figure 4.7 shows the time evolution of $\theta(t)$ for the two simulation experiments with $\theta_{des} = 10^{\circ}$ and an impact velocity of $\dot{x}_E = 1m/s$. For this operating point, the free-flight controller becomes unstable while the Manipulation controller keeps the Quadrotor stable.

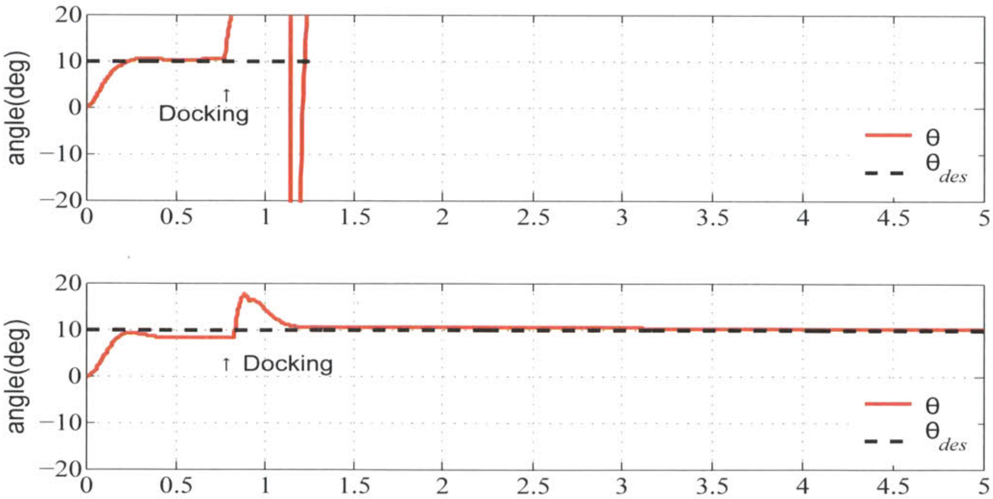

Figure 4.7: $\theta(t)$ plots for Free-Flight (above) and Manipulation (below) controllers.

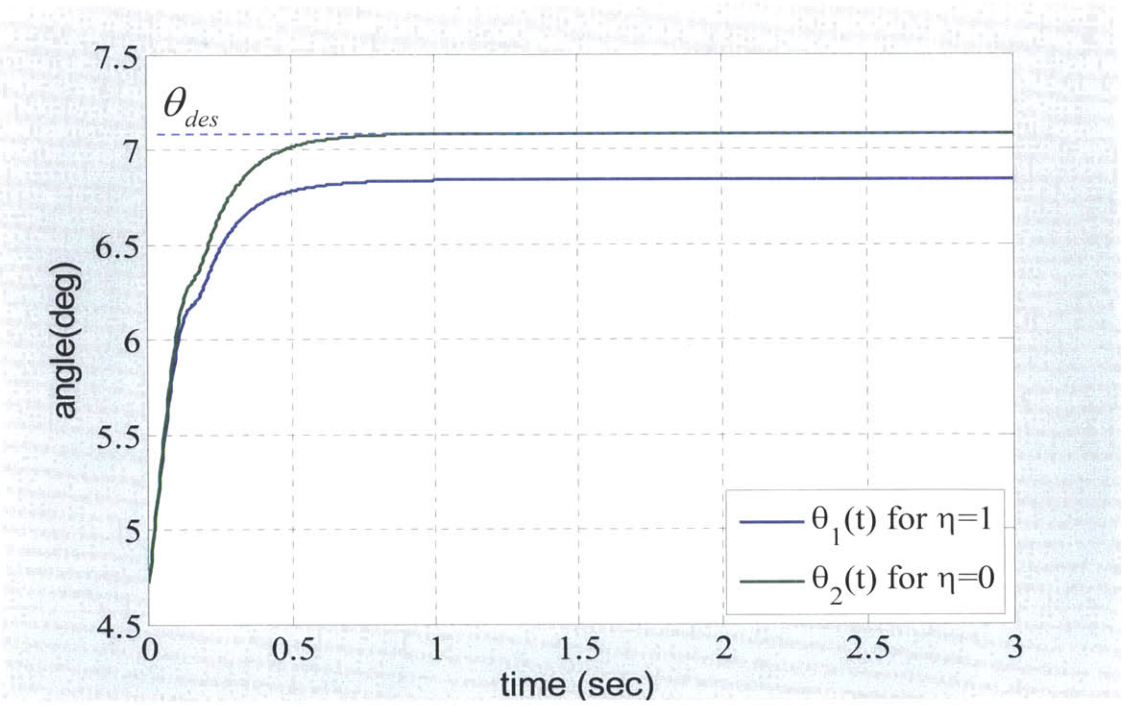

Figure 4.8: Green trace – Quadrotor is in free-flight with free-flight controller Blue trace – Quadrotor is in free-flight, however is commanded by the manipulation controller. The reduction in performance is acceptable in view of the benefit of robustness.

It can therefore be inferred from the plot of Figure 4.6 that the manipulation controller performs significantly better in terms of stabilizing the Quadrotor for a wider range of θ and $\dot{x}_E$ combinations.

Note that in the second simulation experiment involving the manipulation controller, we switched the controller exactly at the instant of docking. This is not realistic as it is difficult to ascertain the exact moment at which docking occurs. We therefore present an alternate switching scheme. Instead of switching the controller from free-flight controller to manipulation controller at the instant of docking, we switch it well before the docking. We will study the consequences of such a premature switching through simulation. One benefit of doing so is that it enables dealing with the uncertainties in switching process. Precise switching at the instant of docking requires precise determination of Object distance from the Quadrotor, which might be challenging in many cases. For instance, if the Object is closer to the Quadrotor end-effector than assumed, the docking process may lead to instability if the θ and $\dot{x}_E$ combination at the moment of impact is above the red-curve of Figure 4.6. With the proposed alternate scheme of premature switching, this uncertainty can be handled. When the end-effector of Quadrotor is approximately d_{obj} distance away from the object, we switch the controller to manipulation controller setting $\eta = 1$. In this mode the Quadrotor is still in free-flight, however, it is commanded by the manipulation controller. The main advantage of this method is that the Quadrotor is now robust to contacts even if the actual distance to object is less than the assumed distance. However, we now have to understand the implications of turning on the manipulation controller when the Quadrotor is still in flight phase. As the numerical simulation shows, the Quadrotor can still perform flight maneuvers with a slightly reduced tracking performance. However, the loss in tracking performance is overweighed by the fact that the Quadrotor is now robust to aggressive docking (docking at high θ and $\dot{x}_E$) which otherwise would have been impossible. The flight performance is shown in Figure 4.8. After docking, we change the value of η to its true value and continue the manipulation process. The simulation result in the second plot of Figure 4.7 was in fact performed using premature switching. The proposed method of premature switching is validated using the experimental setup shown in (B,C) of Figure 4.9.

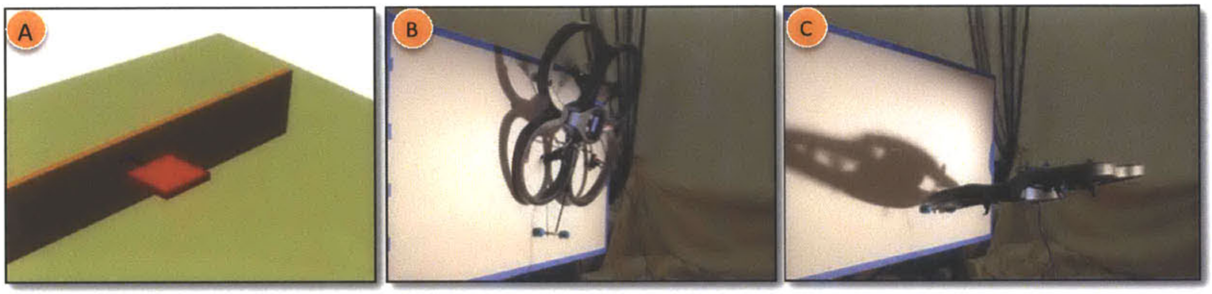

Figure 4.9: (A) Shows the docking simulation setup. (B) The Quadrotor running the free-flight controller becomes unstable and flips over. (C) The Quadrotor running the manipulation controller remains stable.

A recipe to design the docking process is as follows

- Given the anticipated far object distance $\tilde{d}_{obj}$ (such that $\tilde{d}_{obj} > d_{obj}$), compute the anticipated impact velocity $\dot{\tilde{x}}_{E,imp}$ given θ_{des} as

$$\dot{\tilde{x}}_{E,imp} \approx \sqrt{2\tilde{d}_{obj}\theta_{des}g}\ {}^{m}\!/_{s} \tag{4.25}$$

- Since we know the stable combinations of θ and $\dot{x}_E$ from Figure 4.6, we can pick θ_{des} given $\tilde{d}_{obj}$ and $\dot{\tilde{x}}_{E,imp}$ using Equation-(4.25).

4.5 Experimental Results

The experiments are carried out using STeVE setup. AR Drone UAVs [1] are suitably modified by replacing the onboard controller firmware with our own controller. The Quadrotor's position and orientation are tracked in the motion capture setup [26] that is set to run at $120Hz$ (which is also the control rate). The commands are sent to the Quadrotor over dedicated Wifi network using UDP protocol. The onboard controller only performs the derivative control while rest of the control computation is done off-board on a Windows PC.

4.5.1 Single Quadrotor Manipulation

The single Quadrotor manipulation setup is shown in Figure 4.10 and a close up is shown in Figure 4.11. The Object is free to move on a straight track which is located at some elevation. The tracks are made with two aluminum rods supported by sturdy tripods. The cart rides on the track using smooth ball bearings. The Quadrotor and the cart are fitted with markers that are tracked using the motion capture setup, thus providing the position and orientation data.

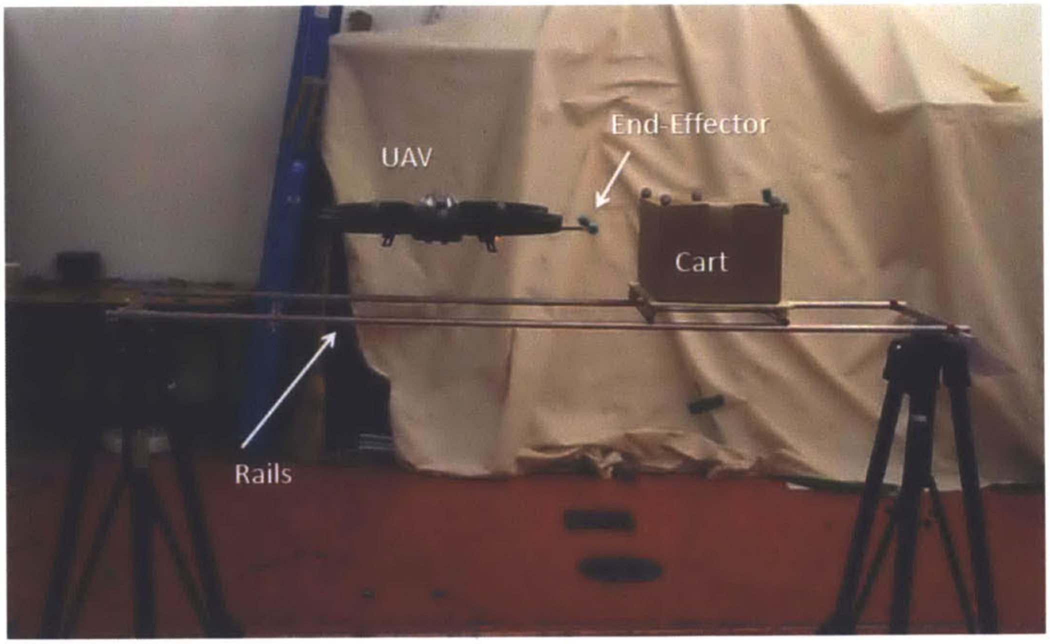

Figure 4.10: Single Quadrotor experimental setup showing the Quadrotor and the Object. The Object is free to move on the track. The white balls on the cart and UAV are the retro reflectors that are used by the motion capture setup to track the position and orientation.

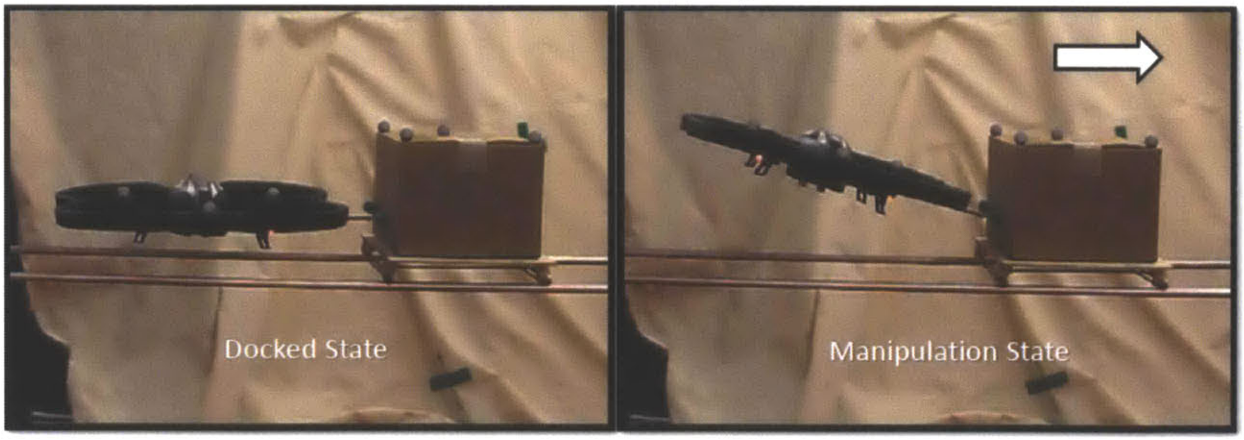

Figure 4.11: The Quadrotor is docked (left) and is pushing the cart (right) by making $\theta > 0$.

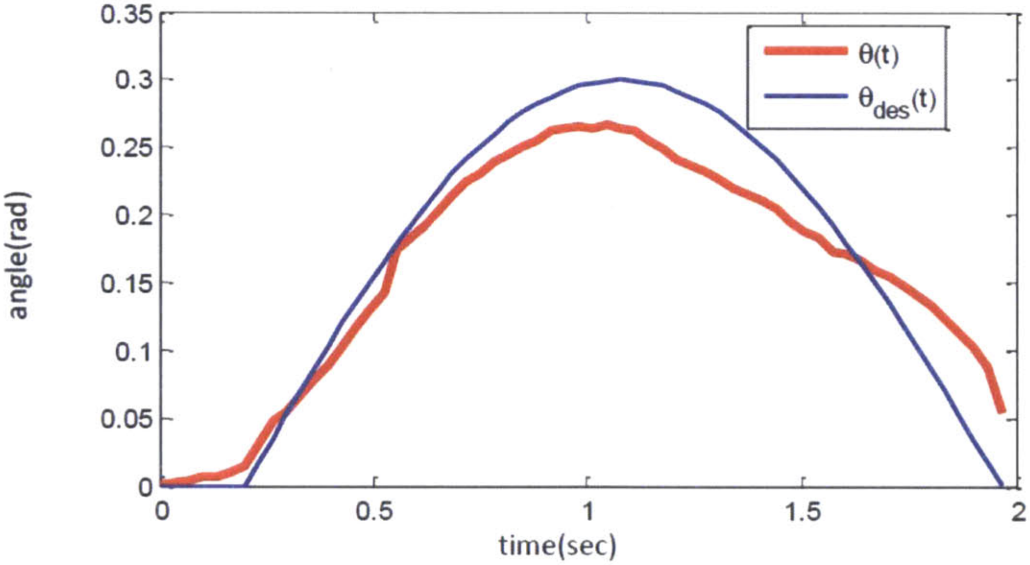

Figure 4.12: Experimental result of θ tracking.

The plot in Figure 4.12 shows the tracking performance of θ for a half cycle of sinusoid. Plot in Figure 4.13 shows the plot of $F_{push,des}(t)$ and $F_{push}(t)$. Figure 4.14 shows the position of the cart due to the push force.

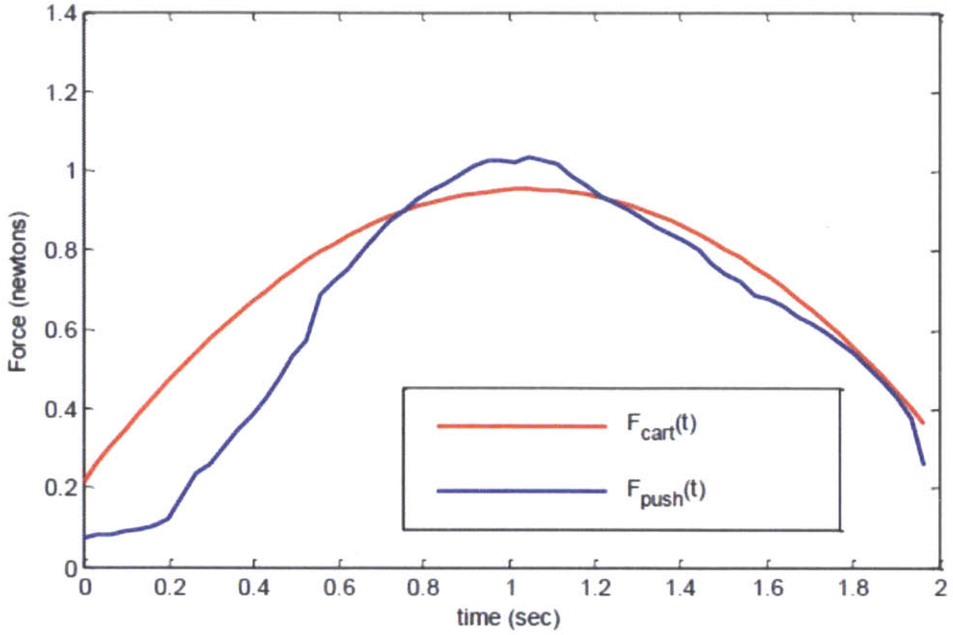

Figure 4.13: Desired force $F_{push,des}(t)$ and the actual push force $F_{push}(t)$

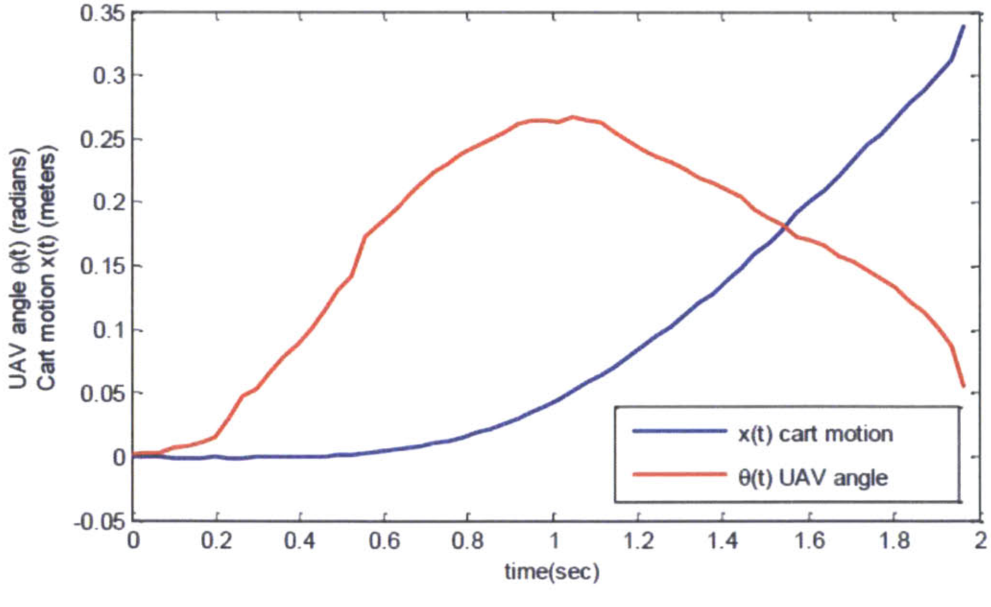

Figure 4.14: The position of the Object x_E and the angle θ

4.5.2 **Two Quadrotor Manipulation**

Figure 4.15 shows the two Quadrotors docked to the object in the center (two white boards). The total weight of Object was matched to be around 4Kgs. The manipulation scheme and the control scheme presented in in this chapter are used to perform trajectory tracking of a sinusoid. The result of experimental plot is shown in Figure 4.16. The sinusoid starts at $t = 15sec$ with a peak value of $\pm 500mm$ and a bias of $1000mm$. The tracking performance is similar to the simulation results presented in Figure 4.4. The deviation from the reference command can be attributed to the aerodynamic disturbances and presence of sag in the aluminum rails.

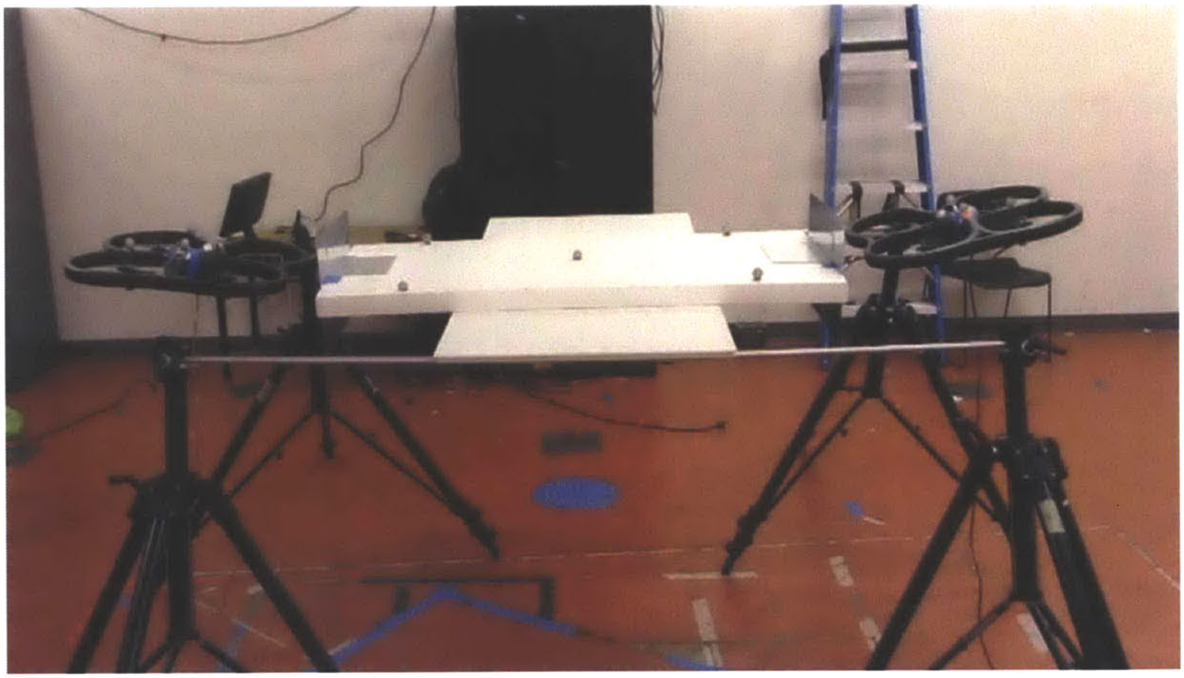

Figure 4.15: Experimental setup for two Quadrotor manipulation system. The Object is a flat foam core weighing about 4Kgs. The two Quadrotors are in docked state.

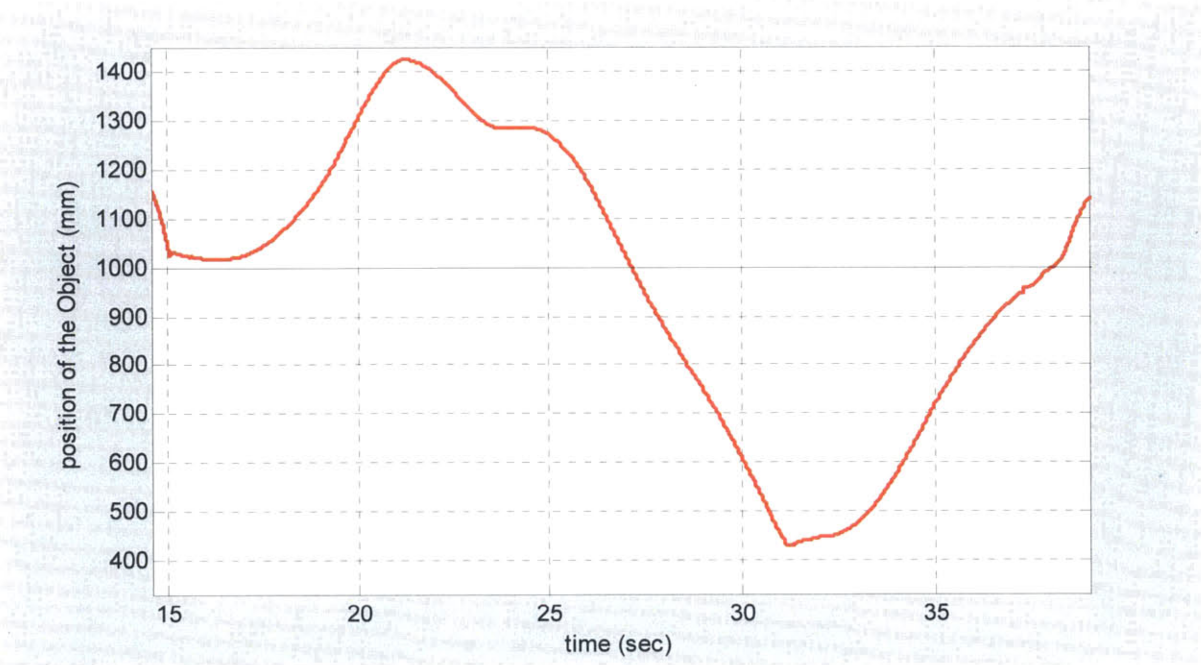

Figure 4.16: Object position x_E due to a commanded sinusoidal input. The two Quadrotors are docked at t=15sec. One complete cycle of sinusoid position command is executed. The corresponding simulation result is shown in Figure 4.5.

4.6 Summary

In this chapter, we presented the control scheme based on the dynamics that was derived in Chapter-3. In the case of a single Quadrotor manipulation, selection of push force was a direct approach. While in the two Quadrotor case, the push forces from each of the Quadrotors was based on the feasible region computed from the contact stability conditions and the tracking error. The simulation and experimental results demonstrate the validity of the model and the control schemes.

5 Manipulation In the Presence of Uncertainties

In the previous chapter, we performed manipulation of an Object assuming perfect knowledge of the system parameters. In a practical scenario, some of the parameters of the system may be unknown. This chapter presents control strategies to deal with uncertainties, particularly in the presence of (1) Parametric uncertainty and (2) Kinematic uncertainty. We first focus on parametric uncertainly where we assume the mass parameters are unknown. We start from the dynamics derived in Chapter 3.

5.1 Manipulation In the Presence of Parametric Uncertainty

We propose a design a controller for the Quadrotor manipulation problem assuming the mass parameters of the system: η and m_Q are unknown. We assume we only know the lower and upper bounds of these parameters. There are two implications because of this uncertainty. The first one being the tracking performance of θ, and the second being the dependence of push force F_{push} on η and m_Q. We consider the dynamics of a single Quadrotor manipulating an Object, given by Equation-(3.30), rewritten here

$$\tilde{\mathbf{H}}\ddot{\tilde{X}}_{manip} + \tilde{\mathbf{C}}\dot{\tilde{X}}_{manip} + \tilde{\mathbf{G}} = \tilde{\mathbf{B}}\begin{bmatrix} F_{zb} \\ M_\theta \end{bmatrix} \tag{5.1}$$

We notice that $\tilde{\mathbf{H}}, \tilde{\mathbf{C}}, \tilde{\mathbf{G}}$ are all functions of the parameters η and m_Q, which are assumed to be unknown. Our goal is design a tracking controller for $\tilde{X}_{manip}$ such that it follows a given trajectory $\tilde{X}_{manip,des}$ in the presence of uncertainty in η and m_Q .

One can easily assume limits on the uncertain terms based on practical considerations as

$$\begin{aligned} &\eta_{lower} \leq \eta \leq 1 \\ &m_{Q,lower} \leq m_Q \leq m_{Q,upper} \end{aligned} \tag{5.2}$$

The maximum value of η cannot exceed unity by definition. As derived in Chapter-3, we recall that the force applied by the Quadrotor on the object is given by

$$F_{push} \approx \eta m_Q g \tan\theta \tag{5.3}$$

In order to ensure we are applying a desired amount of force to an object, it is essential to know the system parameters η and m_Q. Lack of complete knowledge of these terms means we cannot compute θ_{des} using the Equation-3.X, which is

$$\theta_{des} \approx \tan^{-1}\left(\frac{F_{push,des}}{\eta m_Q g}\right) \tag{5.4}$$

This makes an adaptive solution mandatory and it is proposed in the next section.

5.1.1 **Adaptive Nonlinear Control**

We first propose to address the tracking performance in the presence of the above said uncertainty. We proposed to use Slotine and Li adaptive tracking controller [37]. The design of the controller requires the dynamics to be first written in the form of Equation-(5.1) and that $\dot{\tilde{\mathbf{H}}} - 2\tilde{\mathbf{C}}$ is skew symmetric. We see that this condition is not met with the terms given by

$$\tilde{\mathbf{H}} = \begin{bmatrix} m_Q & 0 \\ \eta m_Q r\cos\theta & J_y + \eta m_Q r^2 \sin^2\theta \end{bmatrix} \tag{5.5}$$

$$\tilde{\mathbf{C}} = \begin{bmatrix} 0 & 0 \\ 0 & \eta m_Q r^2 \dfrac{\sin(2\theta)}{2}\dot{\theta} \end{bmatrix} \tag{5.6}$$

$$\tilde{\mathbf{G}} = \begin{bmatrix} m_Q g \\ \eta m_Q g r\cos\theta \end{bmatrix} \tag{5.7}$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} \cos(\theta) & 0 \\ \eta r & 1 \end{bmatrix} \tag{5.8}$$

Since the coriolis matrix $\tilde{\mathbf{C}}$ is not unique, we can re derive it to ensure $\dot{\tilde{\mathbf{H}}} - 2\tilde{\mathbf{C}}$ is skew symmetric using the expression

$$\tilde{\mathbf{C}}_{ij} = \frac{1}{2}\dot{\tilde{\mathbf{H}}}_{ij} + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial\tilde{\mathbf{H}}_{ik}}{\partial q_j} - \frac{\partial\tilde{\mathbf{H}}_{jk}}{\partial q_i}\right)\dot{q}_k \tag{5.9}$$

So, we have

$$\tilde{\mathbf{C}} = \begin{bmatrix} 0 & 0 \\ -\eta m_Q r\dfrac{\sin(\theta)}{2}\dot{\theta} & \eta m_Q r^2 \dfrac{\sin(2\theta)}{2}\dot{\theta} \end{bmatrix} \tag{5.10}$$

For notational simplicity, in this section we will use $\tilde{X}$ instead of $\tilde{X}_{manip}$. Before we present the adaptive control design, we will first define the terms

$$\begin{aligned} \tilde{q} &= \tilde{X} - \tilde{X}_{des} \\ s &= \dot{\tilde{q}} + \Lambda\tilde{q} \\ \dot{q}_r &= \dot{\tilde{X}}_{des} - \Lambda\tilde{q} \end{aligned} \tag{5.11}$$

where Λ is a positive diagonal matrix. We design the reference command as

$$\tilde{X}_{des} = \begin{bmatrix} z_{E,des} + r\sin\theta_{des} \\ \theta_{des} \end{bmatrix} \tag{5.12}$$

where $Z_{E,des}$ is the desired end-effector location, which is kept constant (and equal to zero without loss of generality) to avoid sliding on the object surface. So, we have

$$\begin{aligned} \tilde{X}_{des} &= \begin{bmatrix} r\sin(\theta_{des}) \\ \theta_{des} \end{bmatrix} \\ \dot{\tilde{X}}_{des} &= \begin{bmatrix} r\cos(\theta_{des})\dot{\theta}_{des} \\ \dot{\theta}_{des} \end{bmatrix} \end{aligned} \tag{5.13}$$

The goal is to derive an adaptive control law such that $\tilde{X}_{des} \rightarrow \tilde{X}$. To this effect, we define a constant vector of unknown parameters and call it a. The parameter estimation error is given by

$$\tilde{a} = \hat{a} - a \tag{5.14}$$

where $\hat{a}$ is the estimate of a. We consider the Lyapunov candidate function

$$V(t) = \frac{1}{2}\left[s^T \tilde{\mathbf{H}} s + \tilde{a}^T \Gamma^{-1} \tilde{a}\right] \tag{5.15}$$

where Γ is a symmetric positive definite matrix. Differentiating Equation-(5.15) yields

$$\dot{V}(t) = s^T \left(\tilde{\mathbf{H}} \ddot{\tilde{X}} - \tilde{\mathbf{H}} \ddot{q}_r\right) + \frac{1}{2} s^T \dot{\tilde{\mathbf{H}}} s \tag{5.16}$$

We note that in our case,

$$\dot{\tilde{\mathbf{H}}} - 2\tilde{\mathbf{C}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.17}$$

Using Equation-(5.17) and Equation-(5.1), we eliminate $s^T \dot{\tilde{\mathbf{H}}} s$ term in Equation-(5.16), to arrive at

$$\dot{V}(t) = s^T \left(\tilde{\mathbf{B}} \begin{bmatrix} F_{zb} \\ M_\theta \end{bmatrix} - \tilde{\mathbf{H}} \ddot{q}_r - \tilde{\mathbf{C}} \dot{q}_r - \tilde{\mathbf{G}}\right) + \dot{\hat{a}}^T \Gamma^{-1} \hat{a} \tag{5.18}$$

We also note that with a proper choice of the unknown parameter vector a, we can argue that all of the system matrices $\tilde{\mathbf{H}}, \tilde{\mathbf{C}}, \tilde{\mathbf{G}}$ are linear functions of a. This permits us to define a known matrix $Y = Y\left(\tilde{X}, \dot{\tilde{X}}, \dot{q}_r, \ddot{q}_r\right)$ such that

$$\tilde{\mathbf{H}}\ddot{q}_r + \tilde{\mathbf{C}}\dot{q}_r + \tilde{\mathbf{G}} = \tilde{\mathbf{B}}\begin{bmatrix} F_{zb} \\ M_\theta \end{bmatrix} = Y\left(\tilde{X}, \dot{\tilde{X}}, \dot{q}_r, \ddot{q}_r\right)a \tag{5.19}$$

Taking control law to be

$$\begin{bmatrix} F_{zb} \\ M_\theta \end{bmatrix} = \tilde{\mathbf{B}}^{-1}\left(Y\hat{a} - K_D s\right) \tag{5.20}$$

leads to

$$\dot{V}(t) = s^T Y\hat{a} - s^T K_D s + \dot{\hat{a}}^T \Gamma^{-1}\hat{a} \tag{5.21}$$

Updating the parameter estimate $\hat{a}$ using the integral

$$\hat{a}(t) = -\int_{-\infty}^{t} \left(\Gamma Y^T s\right) dt \tag{5.22}$$

then yields,

$$\dot{V}(t) = -s^T K_D s \leq 0 \tag{5.23}$$

where K_D is a positive definite matrix. This ensures the both global stability and convergence of tracking error.

Two assumptions have been made while arriving at Equation-(5.23), both pertaining to the gain matrix $\tilde{\mathbf{B}}$ in Equation-(5.20). We note that when θ is close to $\pm\frac{\pi}{2}$, the condition number of $\tilde{\mathbf{B}}$ approaches 0, which in turn means the matrix inverse will seize to exist. Therefore, necessary limits have to be placed on θ in order to ensure the control signals are bounded (and within the limits of actual system). The second assumption made is that $\tilde{\mathbf{B}}$ is fully known. This is not true because it is a function of η, which is stated in the beginning that is it is not fully known. To resolve this issue, we set $\eta = 1$ and introduce a matrix

$$\tilde{\mathbf{B}}_1 = \begin{bmatrix} \cos(\theta) & 0 \\ r & 1 \end{bmatrix} \tag{5.24}$$

and use this in Equation-(5.20). Implications of this are now studied. We note that

$$\tilde{\mathbf{B}}^{-1} = \begin{bmatrix} \dfrac{1}{\cos\theta} & 0 \\ \dfrac{-\eta r}{\cos\theta} & 1 \end{bmatrix} \text{ and } \tilde{\mathbf{B}}_1^{-1} = \begin{bmatrix} \dfrac{1}{\cos\theta} & 0 \\ \dfrac{-r}{\cos\theta} & 1 \end{bmatrix} \tag{5.25}$$

So, Equation-(5.20) can we rewritten as

$$\begin{bmatrix} F_{zb} \\ M_\theta \end{bmatrix} = \tilde{\mathbf{B}}_1^{-1}\left(Y\hat{a} - K_D s\right) + \begin{bmatrix} 0 & 0 \\ (1-\eta)\dfrac{r}{\cos\theta} & 0 \end{bmatrix}\left(Y\hat{a} - K_D s\right) \tag{5.26}$$

We propose a new control law based on $\tilde{\mathbf{B}}_1^{-1}$ as

$$\begin{bmatrix} F_{zb} \\ M_\theta \end{bmatrix} = \tilde{\mathbf{B}}_1^{-1}\left(Y\hat{a} - K_D s\right) \tag{5.27}$$

We now see that

$$\dot{V}(t) = -s^T K_D s + s^T \begin{bmatrix} 0 & 0 \\ (n-1)r & 0 \end{bmatrix} \begin{bmatrix} F_{zb} \\ M_\theta \end{bmatrix} \tag{5.28}$$

Let us expand the above equation term by term

$$\dot{V}(t) = -s_1^2 K_{D_1} - s_2^2 K_{D_2} - s_2(1-\eta) r F_{zb} \tag{5.29}$$

We rearrange the terms in Equation-(5.29) to find the condition under for $\dot{V}(t)$ to be negative definite. We have

$$K_{D_1} s_1^2 + K_{D_2}\left(s_2 + \frac{(1-\eta) r F_{zb}}{2K_{D_2}}\right)^2 > \frac{(1-\eta)^2 r^2 F_{zb}^2}{4K_{D_2}} \tag{5.30}$$

We note that $0 \le \eta \le 1$ and $r > 0$. We also note that F_{zb} is bounded $0 \le F_{zb} \le F_{\text{actuator-max}}$ because of the physical limitations of the actuator. We can then conclude that the right hand side of Equation-(5.30) is always positive semi definite. The region of stability is shown in Figure 5.1, and a more conservative condition is given by Equation-(5.31).

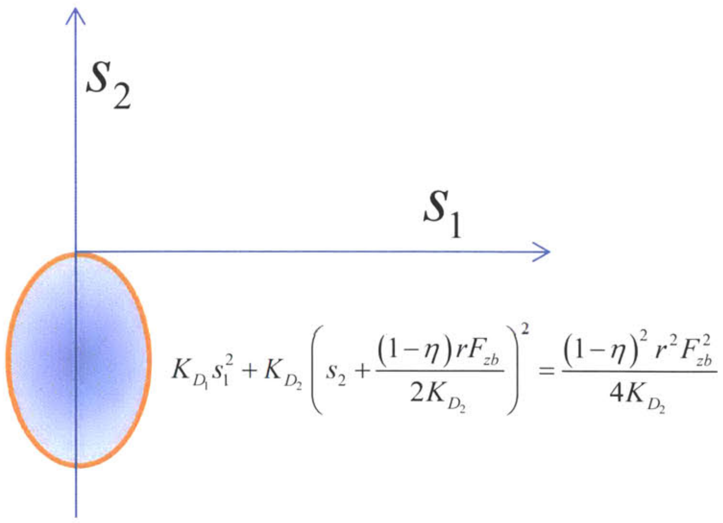
s_2
s_1
$K_{D_1}s_1^2 + K_{D_2}\left(s_2+\frac{(1-\eta)rF_{zb}}{2K_{D_2}}\right)^2 = \frac{(1-\eta)^2 r^2 F_{zb}^2}{4K_{D_2}}$

Figure 5.1: Region of instability shown in shaded area within the ellipsoid.

$$\begin{aligned} &s_2 \geq 0 \\ &s_2 \leq -\frac{(1-\eta)rF_{zb}}{K_{D_2}} \end{aligned} \tag{5.31}$$

We can ensure the area of the region of stability is small by ensuring larger gain K_{D_2} and a smaller radius r.

5.1.2 **Implementation of Adaptive Nonlinear Controller**

We choose the unknown vector a and the matrix Y as follows

$$a = \begin{bmatrix} m_Q \\ \eta m_Q r \\ J_y \end{bmatrix} \tag{5.32}$$

$$Y = \begin{bmatrix} \ddot{z}_r + g & 0 & 0 \\ 0 & \ddot{z}_r \cos\theta + \ddot{\theta}_r r \sin^2\theta - \dot{\theta}\dot{z}_r \dfrac{\sin\theta}{2} + \dot{\theta}\dot{\theta}_r r \dfrac{\sin(2\theta)}{2} + g\cos\theta & 0 \end{bmatrix} \tag{5.33}$$

Note that we have clubbed terms in a and also introduced J_y as unknown.

The initial condition a_0 for the integral in Equation-(5.22) is a nominal expected value of a. Based on the knowledge of real system, we numerically ensure that

$$a_{\min} \le a \le a_{\max} \tag{5.34}$$

which in turn ensures the adaptive controller is robust to noise. The Adaptive gain Γ is tuned with experimental trials.

Further, we stated that the desired push force is given by Equation-(5.3). Using the estimated parameters, we can then compute the estimate of the push force as

$$\hat{F}_{push} \approx \frac{\hat{a}_2}{r} g \tan\theta \tag{5.35}$$

5.1.3 **Simulation**

In this section, we present the simulation results of the adaptive control scheme. The plot in Figure 5.2 shows the tracking performance when the adaptive controller is not active. That is, the adaptive update given by Equation-(5.22) is not performed. This leads to a large tracking error when the mass m_Q is changed.

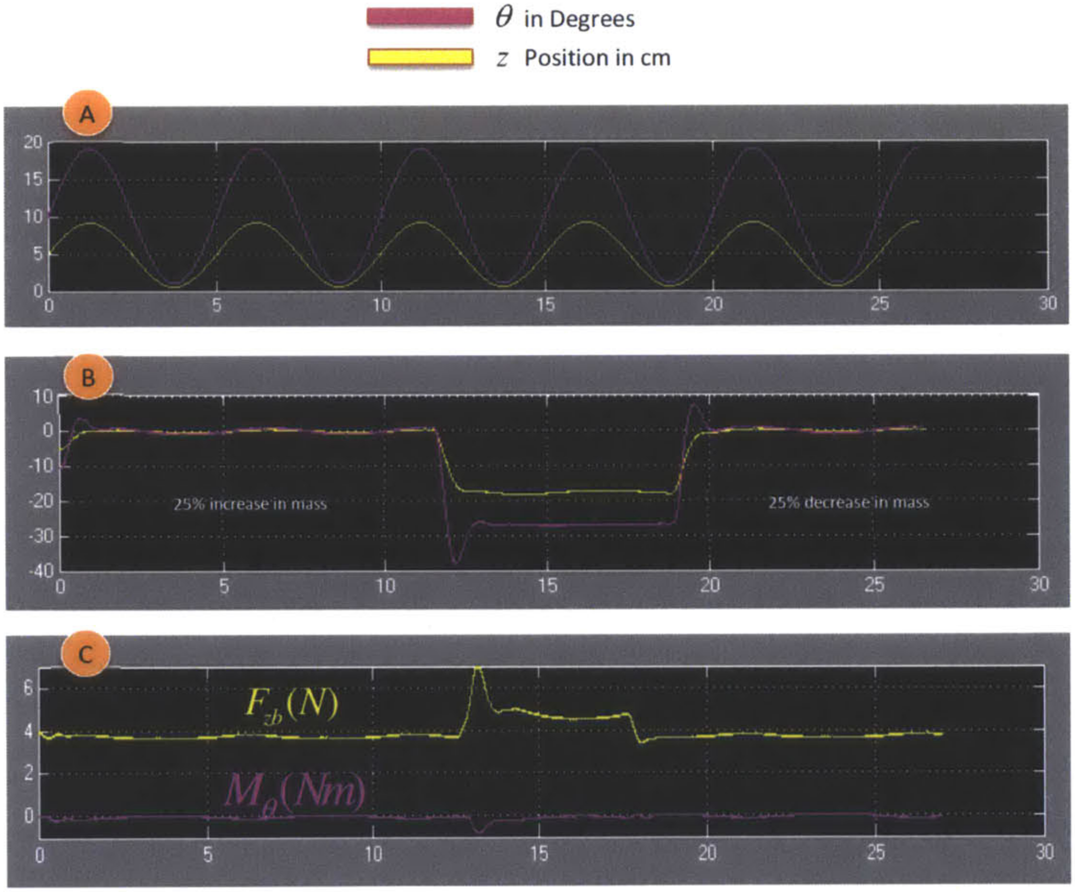

Figure 5.2: Tracking performance when only the PD controller is active. (A) Shows the reference command $\tilde{X}_{des}(t)$. (B) Shows the tracking error when the mass of the system m_Q changes by 25%. (C) Shows the corresponding control inputs.

In contrast, the plot in Figure 5.3 shows the tracking performance when the adaptive controller is active. That is, Equation-(5.22) is used to update the parameters. When the mass m_Q is changed, the resulting tracking error is significantly lower than what we see in Figure 5.2.

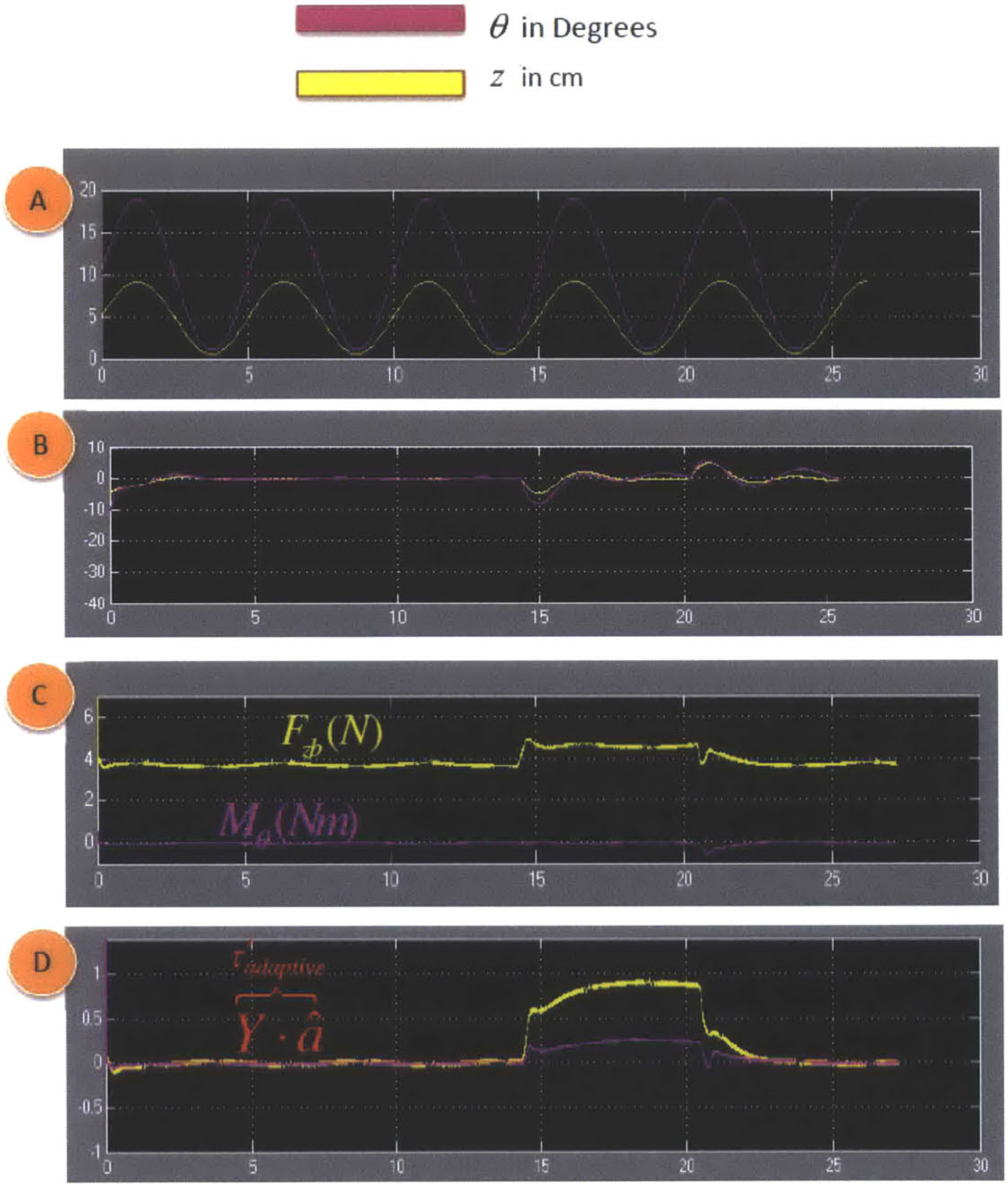

Figure 5.3: Tracking performance using Adaptive Nonlinear Controller. (A) Shows the reference command. (B) Shows the tracking error when the mass of the system changes by 25%. The mass is increased at around t=14sec, and decreased at around t=22sec. (C) shows the net control inputs computed by the controller. (D) Shows the contribution of adaptive controller towards the control input. The adaptive controller is silent elsewhere, and starts to contribute only when the system parameters have changed.

5.1.4 Experimental Results

The Adaptive controller was implemented on a Quadrotor pushing an object. A hardware-in-the-loop setup was used to perform this experiment. The experimental setup is shown in Figure 5.4. Details of the A hardware-in-the-loop setup is covered in Chapter-2.

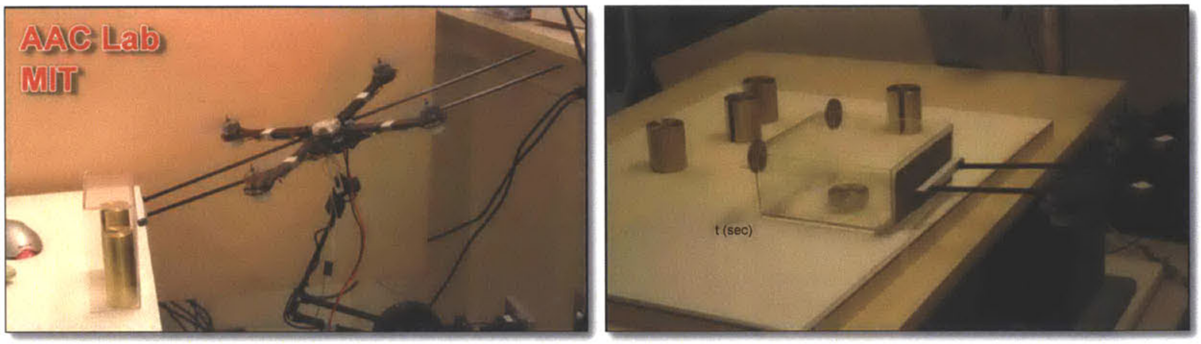

Figure 5.4: Hardware-in-the-loop Experimental Setup used to test the Adaptive Controller.

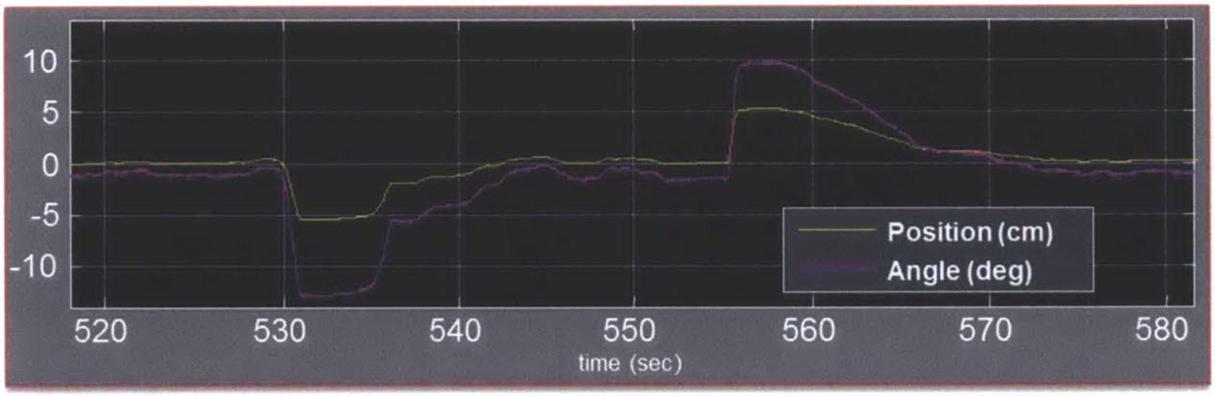

Figure 5.5: Plots shows the tracking error $z(t)-z_{des}(t)$ in centimeters and $\theta(t)-\theta_{des}(t)$ in degrees, for a reference input of $\theta_{des}(t)=20^{\circ}$, and $z(t)=r\sin\theta_{des}$.

As the experimental plot shows, the adaptive controller is able to handle uncertainties in the system and minimize the tracking error. When the Quadrotor was docked to the Object, as shown in Figure 5.4, additional mass was added to the Quadrotor, which lead to increase in m_Q. The increase causes the Quadrotor to dip, as seen in the plot at t=530sec. The adaptive control law ensured that the parameters are adjusted to the new values, and the tracking error is driven close to zero (at about t=540sec). When t=555sec, the added mass was removed and the Quadrotor now has the original mass. Again, the adaptive control law ensured the parameters are adjusted such that the tracking error approaches zero.

It can be noted that for a change in 25% of the mass, the time taken by the adaptive controller is roughly about 10sec. This of course is a function of adaptive gain Γ.

The rate of adaptation was adequate for practical uses. The main drawback of using Adaptive is its sensitivity to delays. The experiments were performed in a HIL setup, where the delays are well under the time step of the control loop. The control loop itself was abound 500Hz. In the actual flight systems, where delays are unavoidable, adaptive controller application becomes more challenging.

5.2 Manipulation in the Presence of Kinematic Uncertainty

The previous section proposed an adaptive controller to deal with uncertainty in the parameters. The specific task that we address is one that involves the manipulation with unknown kinematic constraints, such as the opening of a door. In order to accommodate the kinematic uncertainty, the path of least resistance approach is used. The proposed methods are validated through experimental demonstrations of a Quadrotor opening a door with an unknown hinge location, and numerical studies of tracking an arbitrary trajectory with kinematic constraints.

Consider the problem of manipulating an object that is kinematically constrained. We consider manipulating a rigid object on a 2D surface. That is, the Object is free to move in $x-y$ plane and rotate about $z-axis$. Further, the Object is kinematically constrained such that it has only one degree of freedom. Examples include opening of a door and motion of a cart on curved tracks. The key feature is that the Object can be made to move along its constrained path using a one dimensional force. Since the Quadrotor is capable of generating such a one-dimensional force, these tasks, as we shall show in this section, can be carried out in a straightforward manner. We will focus mainly on the door-opening task.

Consider the Quadrotor manipulating a door as shown in Figure 5.6. The goal is to control the end-effector of the Quadrotor to be positioned at a designated dock point on the door P_{dock} (measured in the world frame $W=[\hat{x}_w,\hat{y}_w,\hat{z}_w]^T$) and to apply a desired amount of force $F_{push,des}$ on the door. As the door moves and turns in the $x-y$ plane, the Quadrotor has to continuously re-align to ensure the force is applied normal to the door, and thus avoid end-effector slipping. We consider a local coordinate frame $[x_D,y_D,z_D]^T$ at P_{dock} that rotates and moves as the door rotates. The door has a vertical axis of rotation, hence z_D is aligned with $\hat{z}_w$. x_D is perpendicular to the surface of the door. Let Ω be the door rotation angle measured in world frame of reference. When $\Omega=0$, x_D is along $\hat{x}_w$.

5.2.1 **Path of Least Resistance Approach**

The approach that is used in order to carry out object manipulation is based on the Path of Least Resistance proposed in [14]. This approach consists of the following two steps: (i) we first fly the Quadrotor using the free-flight controller and ensure the end-effector is at P_{dock}. (ii) Once docked, we switch to the $\tilde{X}_{manip}$ controller described in Chapter-4.2.2, which allows us to apply a desired amount of push force on the door. As the door rotates, the coordinate frame $[x_D, y_D, z_D]^T$ rotates and therefore the Quadrotor has to yaw in order to ensure that the applied force is normal to the surface of the door. As the process continues, the door continues to open. Unlike in Chapter-4, where the Quadrotor was confined to move in a straight line, here the Quadrotor has to turn to account for the turning of the door. Therefore, we redesign the controller for X_{free} to accommodate turning.

We note that we do not use any information pertaining to the kinematics of the door, that is, we do not consider the hinge location or the radius of turning. We show that this approach works despite unknown kinematic constraints. The same approach as above can be used to carry out other manipulation tasks such as the motion of a cart on tracks with track geometry unknown, as we will show in the simulation studies.

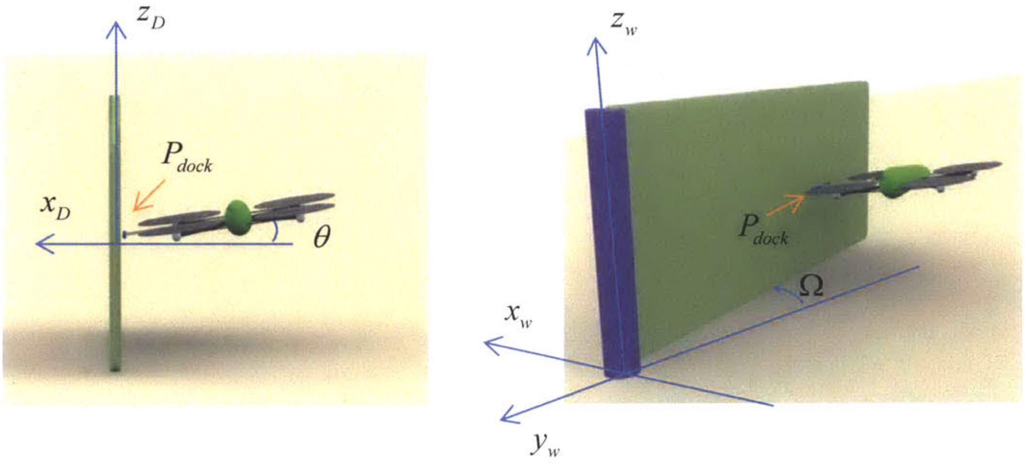

Figure 5.6: Quadrotor end-effector is docked to the door at P_{dock}. By applying force, the Quadrotor pushes the door by to rotated it by an angle Ω. X_{free} of the Quadrotor is controlled to ensure the applied force is always normal to the door, which means the Quadrotor yaws as the door rotates. Figure on the left shows the profile view, while the figure on the right shows the perspective view.

We will now design a suitable controller for X_{free} tracking, that is, a controller will ensure the end-effector is tracked to P_{dock} as the door rotates. See Figure 5.6 for illustration. We present the solution as a set of three tasks.

Task 1 – Ensuring $P_E \rightarrow P_{dock}$

First, we define a surface normal vector N_D passing through P_{dock} (we observe $N_D = -x_D$). We easily can compute N_D from the door angle Ω. We then consider the perpendicular distance d_{NQ} from the center of the Quadrotor P_Q to the line along N_D emanating from P_{dock}. It can be shown that d_{NQ} is given by

$$d_{NQ} = N_{D,x}\left(P_{dock,y} - P_{Q,y}\right) - N_{D,y}\left(P_{dock,x} - P_{Q,x}\right) \tag{5.36}$$

Note that the surface normal N_D is a unit vector.

We define another term z_{err} as

$$z_{err} = P_{dock,z} - z_E \tag{5.37}$$

which can also be written as

$$z_{err} = P_{dock,z} - \left(z_Q - r\sin\theta\right) \tag{5.38}$$

Task 2 – Alignment of Quadrotor with the Door

We define α_{err} as the difference between the door angle Ω and the Quadrotor yaw angle ψ.

$$\alpha_{err} = \Omega - \psi \tag{5.39}$$

Noting that in the local frame of reference ψ is coincident with α_{err} and replace $\psi - \psi_{des}$ with $\dot{\alpha}_{err}$ in Equation-(4.7). As $\alpha_{err} \rightarrow 0$, we can ensure the Quadrotor is pushing normal to the door.

Task 3 – Applying a Desired Push Force

Given a desired push force $F_{push,des}$, we compute θ_{des} using (4.19). The controller in Section-4.2 will ensure $z_{err} \to 0, \theta \to \theta_{des}$ and $F_{push} \to F_{push,des}$.

We can now define the control objective for all of the tasks as

$$\begin{aligned} &d_{NQ} \to 0 \\ &z_{err} \to 0 \\ &\alpha_{err} \to 0 \\ &F_{push} \to F_{push,des} \end{aligned} \tag{5.40}$$

We start with the controller in Equation-(4.9). Since in the local frame of reference, y_Q = d_{NQ}. This allows us to choose

$$\phi_{des} = \left(K_D\left(\dot{d}_{NQ}\right) + K_P\left(d_{NQ}\right)\right) \tag{5.41}$$

We ensure $\alpha_{err} \to 0$ by letting $\psi_{des} = \Omega$. We now have all the necessary ingredients to execute the controller in Equation-(4.7) which will ensure $\alpha_{err} \to 0, \phi \to \phi_{des}$ and hence $d_{NQ} \to 0$.

It should be noted that in order to implement the controllers in Equation-(4.7), it is necessary to measure the door angle Ω as it is required to compute the surface normal N_D and α_{err}. In the next task, we propose a method which circumvents the measurement of Ω by estimating it through the measured door dock position P_{dock} .

Note: The main motivation in choosing to quantities: $d_{NQ}, z_{err}, \alpha_{err}$ is that, they can be measured using Quadrotor onboard sensors. This of course requires necessary instrumentation of the UAV.

Task 4 – Estimation of Ω

In this section, we propose an approach to estimate Ω. This is carried out using a differential set of measurements of $P_{dock} = \begin{bmatrix} P_{dock,x} & P_{dock,y} & P_{dock,z} \end{bmatrix}^T$ given by $P_{dock}\left(t_1\right)$ and $P_{dock}\left(t_2\right)$ measured at t_1 and t_2 respectively, with $t_2 > t_1$.

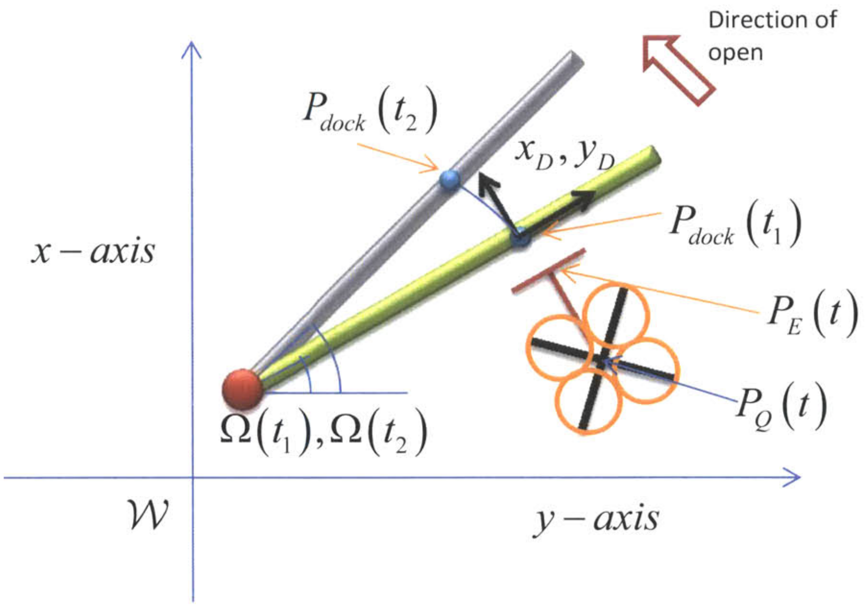

Figure 5.7: Top view of door opening manipulation showing the docking positions and differential measurement of the door dock positions.

The estimate $\hat{\Omega}$ is computed using simple geometry (see Figure 5.7 for illustration) as

$$\hat{\Omega} = \tan^{-1}\left(\frac{P_{dock}(t_1)_y - P_{dock}(t_2)_y}{P_{dock}(t_1)_x - P_{dock}(t_2)_x}\right) + \frac{\pi}{2} \tag{5.42}$$

In order to make the system robust to noise, we ensure that

$$\left\| P_{dock}(t_1) - P_{dock}(t_2) \right\| > \Delta \tag{5.43}$$

where Δ is a small quantity with units of distance. The above algorithm can be summarized as follows.

1. Measure $P_{dock}(t_1)$ at t_1.

2. Measure $P_{dock}(t_2)$ at t_1 such that Equation-(5.43) is satisfied for some Δ.

3. Compute $\tilde{\Omega}$ using Equation-(5.42)

4. Computer $\tilde{\alpha}_{err} = \hat{\Omega} - \psi$

The choice of Δ is a tradeoff between noise and speed of estimation. For instance, a larger Δ would make estimation robust to noise but will introduce lag in estimation, while a smaller Δ would reduce lag at the expense of introducing noise. A lag in the estimation has a direct influence on the acceptable turning radius of the door. That is, a misalignment

$$\delta\hat{\Omega} = \frac{\Delta}{r_{door}} \tag{5.44}$$

can result, where r_{door} is the radius of turning (that is distance between the door hinge and P_{dock}). Depending on the friction properties of the door and end-effector, this misalignment can potentially lead to sliding and instability. A factor of safety thus puts a limit on maximum permissible $\delta\hat{\Omega}$ and hence we can evaluate the upper limit on Δ with the knowledge of minimum expected radius of turning r_{door}. Lower limit on Δ is essentially depicted by the noise characteristics of position measurement.

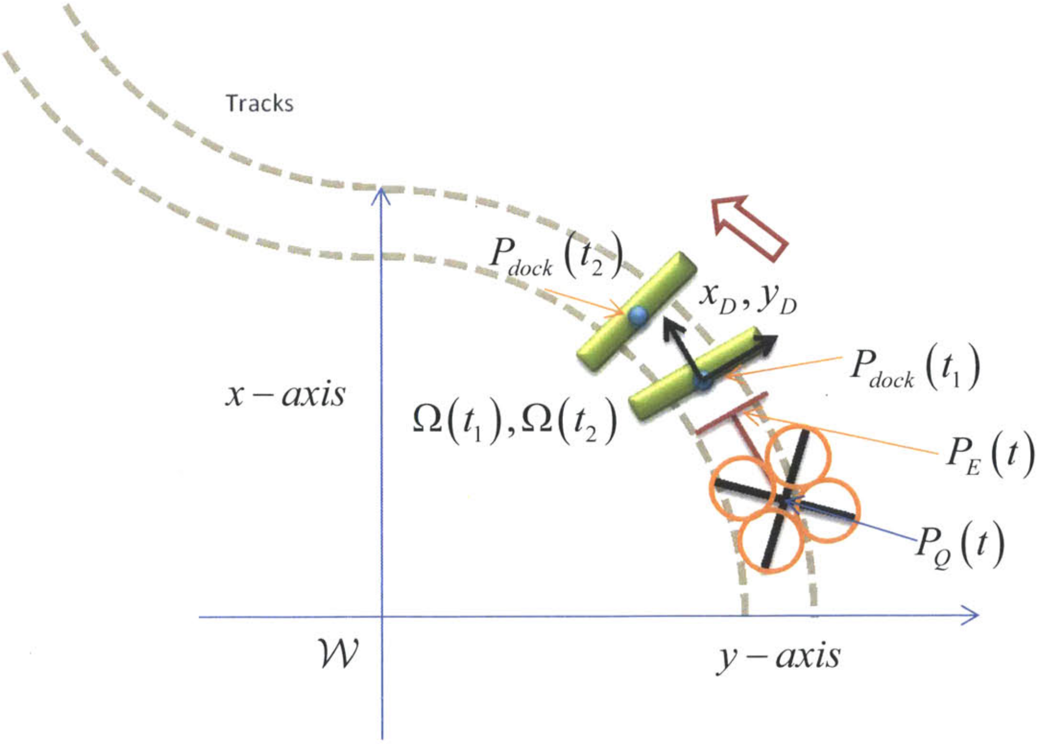

Figure 5.8: A general case of kinematically constrained Object manipulation on a 2D surface.

A similar procedure can be used for manipulation of an Object that is constrained to move on an arbitrary trajectory, as illustrated in Figure 5.8. As before, one can take a differential measurement of Object dock positions and use it to estimate the direction of motion. The UAV can then be aligned such that the push force is applied normal to the direction of motion of the Object. Essentially, the same procedure as for opening a door can be used here, by accommodating the fact that the radius of curvature and the virtual center of rotation change over the duration of manipulation.

We validate our proposed control scheme and manipulation strategy through simulation. We then validate the door opening task using an experimental setup.

5.2.2 **Simulation Results**

We use STeVE in order to perform realistic simulation. The key parameters for the system are

- Mass of the Quadrotor $m_Q = 440 grams$
- Moment of Inertia $J_x = J_y = J_z/2 = 0.001 kgm^2$
- Mass of the door = $4kg$, End-effector length $r = 0.35m$
- Door turning radius (assumed to be unknown) = $\approx 0.8m$

The plot shown in Figure 5.9 shows the simulation result. The top plot shows the door opening simulation using a measured door angle Ω, while the bottom plot shows the simulation result using an estimated door angle $\tilde{\Omega}$. The plots also show the Quadrotor yaw angle ψ which is regulated to track the door angle.

The plot in Figure 5.10 shows the result of manipulation of Object with an arbitrary kinematic constraint. The radius of curvature varies along the trajectory and the assumption is that at any point on the trajectory, the radius of curvature satisfies Equation-(5.44). The red-trace shows the trajectory of the End-effector P_E .

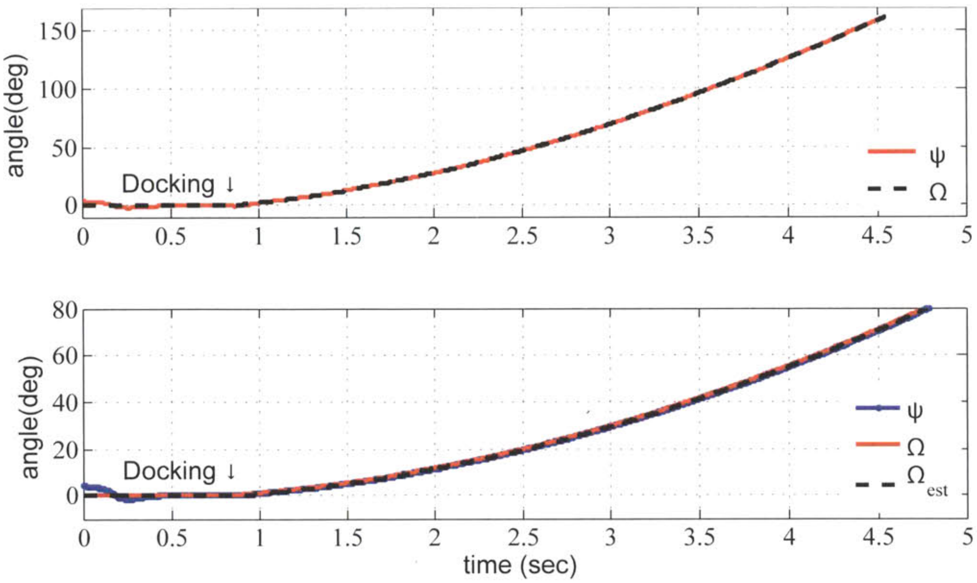

Figure 5.9: Door opening simulation. In the plot on top, the door angle Ω is measured while in the bottom plot, the estimated door $\tilde{\Omega}$ angle is used.

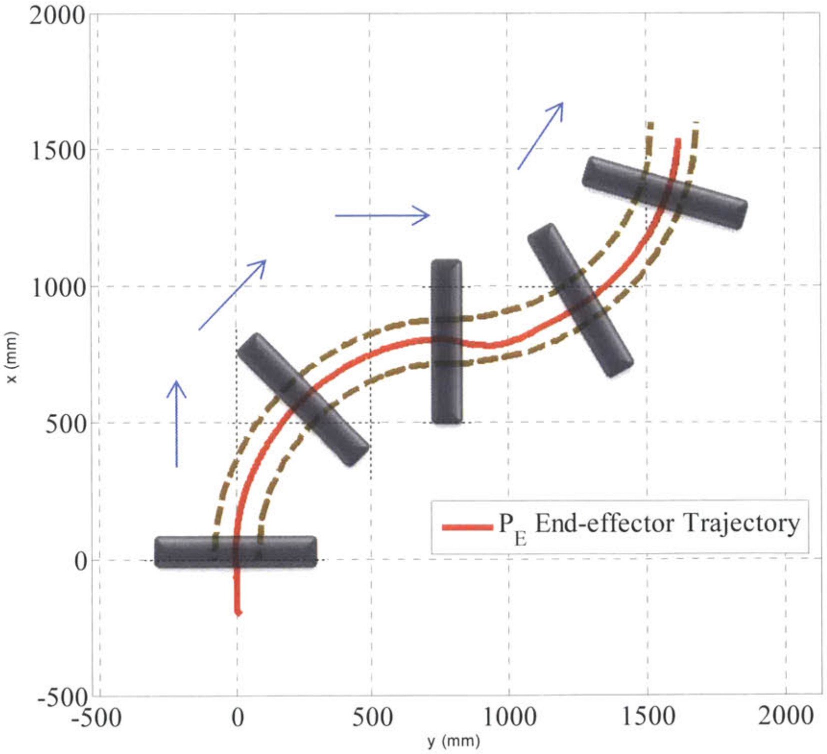

Figure 5.10: Manipulation of object constrained to move on an arbitrary trajectory (brown dotted lines). The red-trace shows the End-effector trajectory. The Object is overlaid on top of its trajectory. A disturbance was introduced to the Quadrotor when the Object was at y=900mm, x=750mm, which caused it the end-effector P_E to deviate from dock position P_{dock}.

5.2.3 **Experimental Results**

We conduct two experiments to demonstrate door opening task using the setup shown in Figure 5.11. In the first experiment the door hinge is located on the left hand side as shown in the figure. In the second experiment, the door hinge is shifted to the right hand side. In both cases, the Quadrotor is unaware of the location of the hinge.

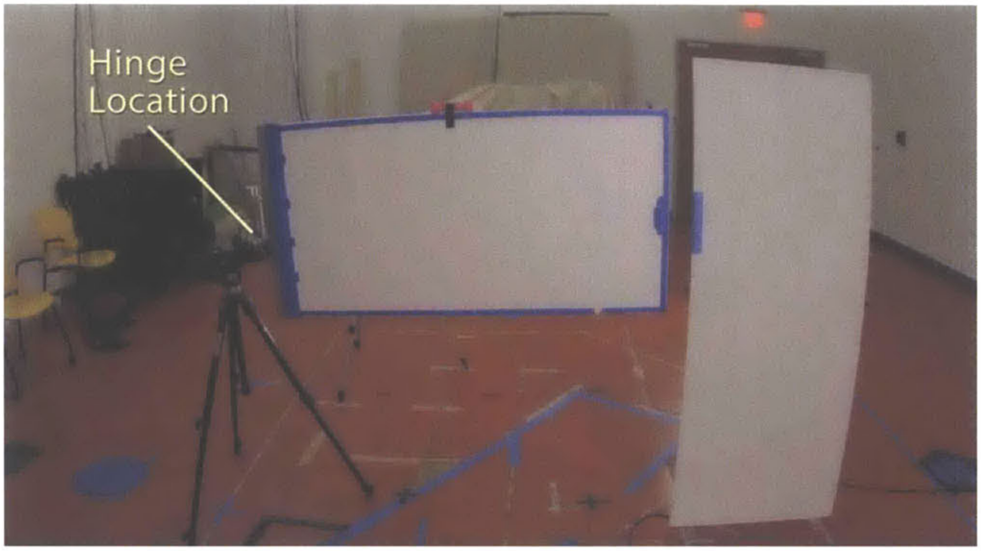

Figure 5.11: Door with hinge located on the left side. The distortion in the image is due to use of wide-angle lens.

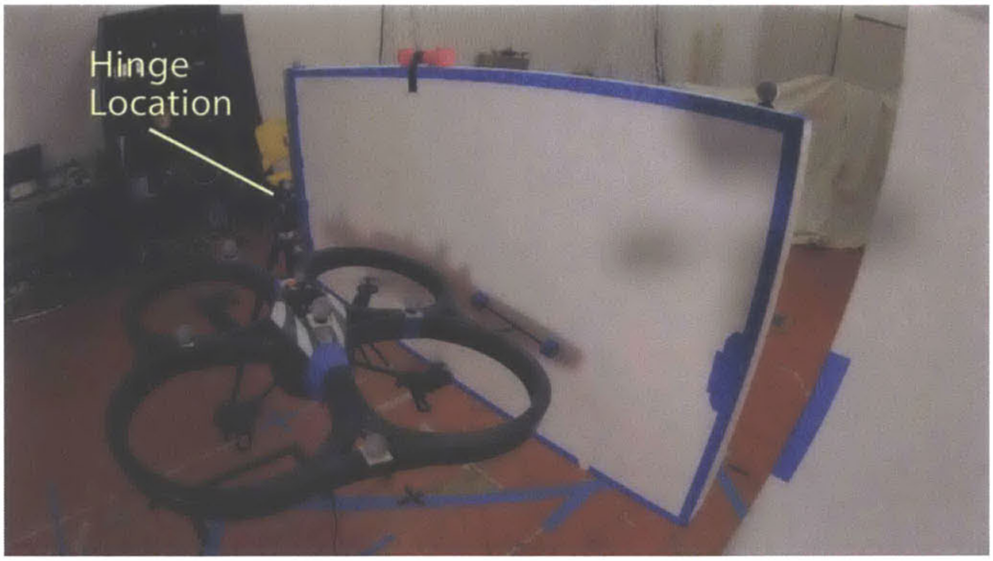

Figure 5.12: Another view of the experiment showing Quadrotor along with the end-effector docked to the door.

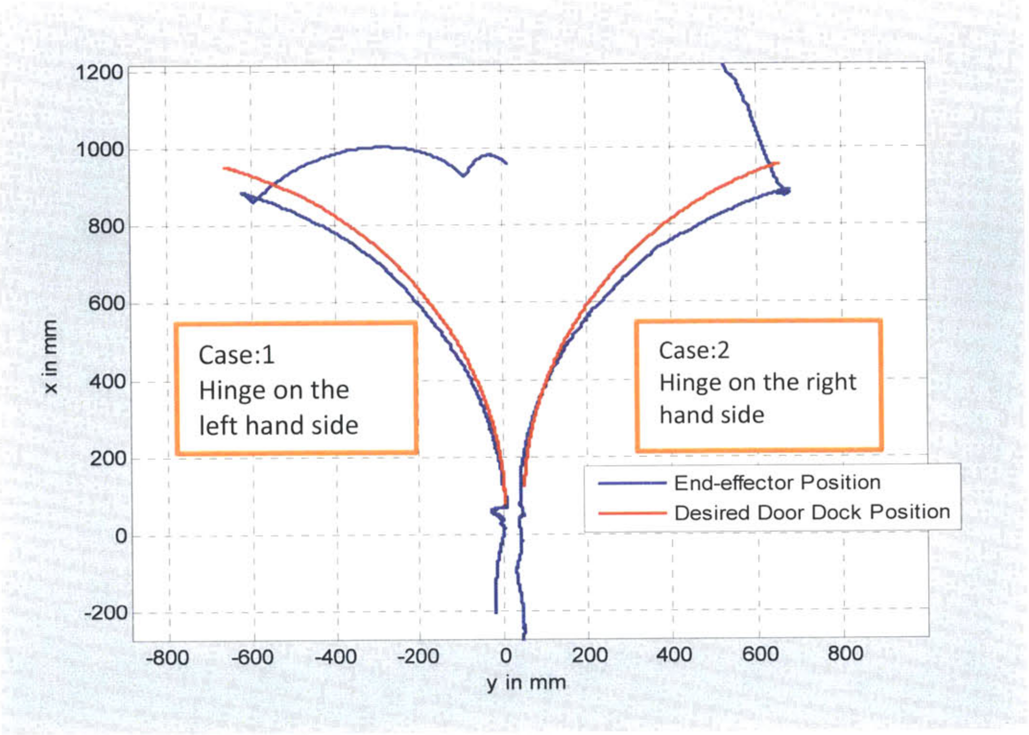

Figure 5.13: Experimental result of Quadrotor opening the door. Case-1: The hinge is located on the left hand side, as shown in Figure 5.11. Case-2: The hinge is located on the right hand side.

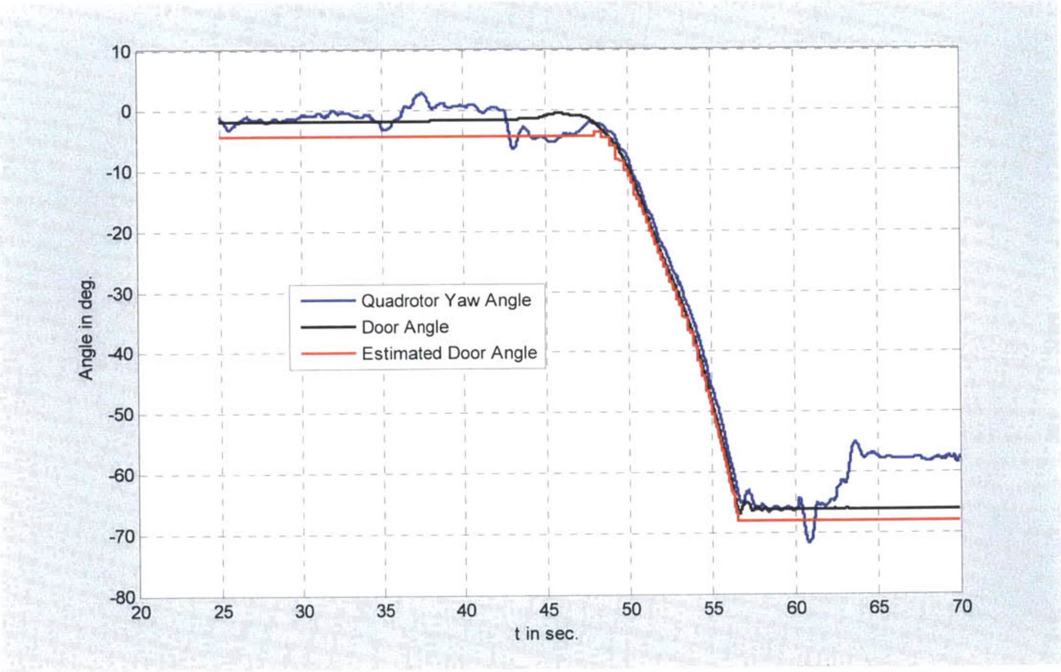

Figure 5.14: Plot of yaw angle of Quadrotor ψ, the estimated door angle $\tilde{\Omega}$ and, the actual door angle Ω. The docking occurred at around t=45sec. The Quadrotor undocked at around 57 sec.

Figure 5.12 shows another view of the Quadrotor pushing the door with the hinge on the left hand side. The result of manipulation is shown in Figure 5.13. The red trace correspond the dock position P_{dock} and the blue trace corresponds to the Quadrotor end-effector position. In the next experiment, the hinge is shifted to right hand side and the Quadrotor successfully pushes open the door despite the change in hinge location. The resulting trajectories are plotted in Figure 5.13 along with the previous result. Figure 5.14 shows the time plot of this manipulation.

5.3 Summary

In this chapter, we presented two manipulation schemes that accommodate uncertainty in the system. In one case the mass parameter was assumed to be unknown, and in the second case the kinematic constraint is assumed to be unknown. In both the cases, the approach is validated through simulation and experiments. Quadrotor opening a door is demonstrated.

In the next chapter, we will present a novel flexible Quadrotor design denoted as ParaFlex. Unlike the rigid Quadrotor we used so far in this thesis, ParaFlex due to the compliant structure is robust to accidental collision.

6 Design and Control of a Flexible Quadrotor

In the preceding chapters we used a Quadrotor which is rigid by construction. When the Quadrotor was subjected to aggressive docking, we ensured stability by active control method. While contact was intentional in the manipulation case, making it robust to aggressive docking renders the UAV robust to accidental contacts. This chapter presents a novel design for a Quadrotor UAV that is allowed to flex when subject to collisions. We propose a flexible UAV Design designated as ParaFlex.

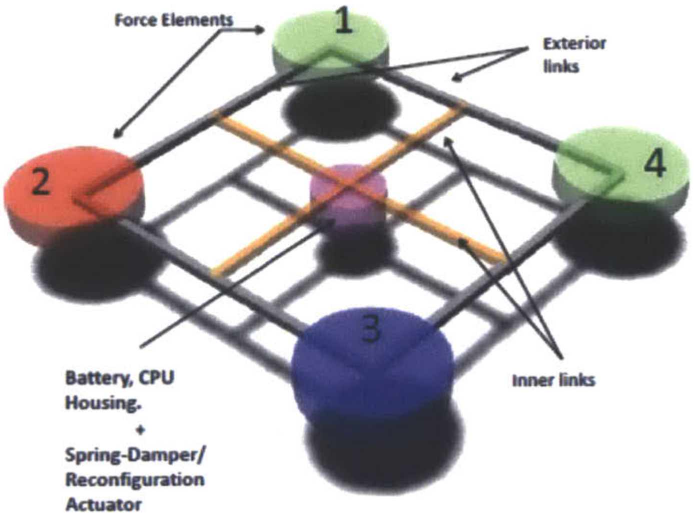

Figure 6.1: ParaFlex Quadrotor UAV

ParaFlex design consists of four link mechanisms with revolute spring-joints at corners. When a diagonal force is applied, the frame deforms. Because of the spring and dampers at the revolute joints, the original shape is restored with decaying oscillations. ParaFlex Quadrotor UAV is shown in Figure 6.1, has four propellers that are protected within the shrouds. When the Quadrotor collides, the collision forces are transferred to the body through the Shroud. Figure 6.2 shows the scenario of ParaFlex colliding with a wall. We note that the component of the force perpendicular to the diagonal cannot deform ParaFlex. Our assumption is that external forces and impacts occur mostly along the diagonal. This is reasonable, because ParaFlex can be made to fly along the diagonal, which can be its heading direction, and hence likely to receive impacts along the diagonal.

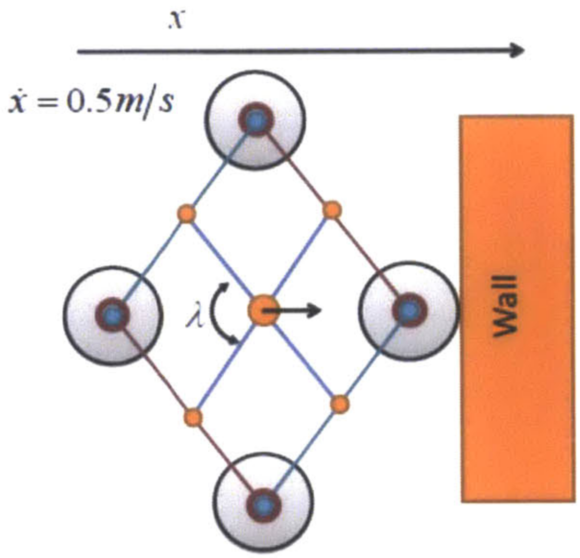

Figure 6.2: ParaFlex colliding with a wall. Collision force changes the deformation angle λ .

6.1 Benefits of ParaFlex over a Rigid Quadrotor

In comparison to active control method proposed to handle aggressive docking in Chapter-4, the main advantage here is that ParaFlex can bring mechanical robustness to collision. During a high velocity collision, the forces appearing on the body are usually impulsive. Because of high bandwidth of the impulsive forces, it is difficult to deal with it using a band limited controller. Moreover, the impact forces can also damage the UAV frame which intron may lead to crash. The mechanical flexibility provides inherent "softness" to the collision reducing the collision force bandwidth. As a result, the low bandwidth controller may have a better chance of stabilizing the UAV during a collision.

Another benefit of introducing flexibility is that the kinetic energy can be stored inside the spring and released after the impact. This permits the ParaFlex to sprint away from the obstacle after the impact. Further, flexibility permits of estimation of impact forces that can be used for improving control performance. With the knowledge of the model, impact force can be estimated which will then permit designing of collision response algorithms.

6.2 Objectives

Our goal in this chapter is to demonstrate the benefits of incorporating flexibility. In order to show the benefits, we focus on three main objectives as follows

1. We first show that when a ParaFlex collides with a wall, significantly less forces appears on the frame in comparison to the rigid Quadrotor. We assume both ParaFlex and the rigid Quadrotor are running the same free-flight controller and undergo a similar collision course.
2. We then show that presence of flexible joint will minimally compromise free-flight performance. That is, when ParaFlex is not colliding, its flight performance closely matches to that of a rigid Quadrotor. We will show that a linear controller designed for a Quadrotor can easily be employed for ParaFlex.
3. We propose use of a nonlinear controller based on the contact model. We use the control design proposed in Chapter-4.4.3. The goal is to combine the robustness from the controller and from the flexible structure.

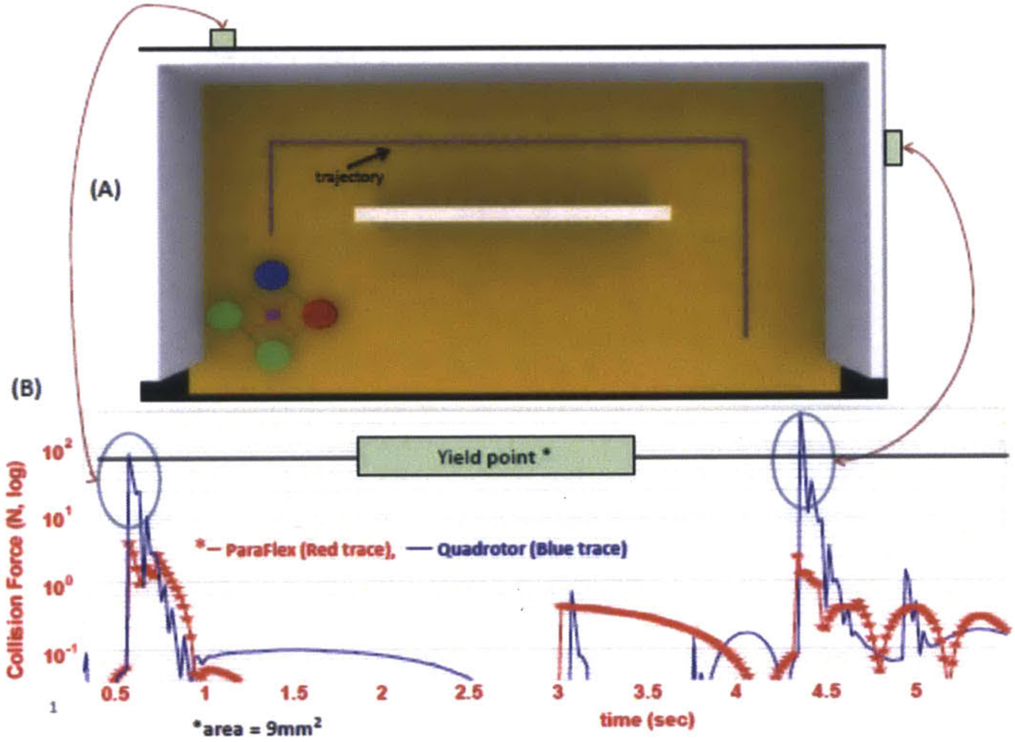

Figure 6.3: Simulation of ParaFlex navigating through a tight hallway. Top view shows the ParaFlex and its path. The plot below shows the forces appearing on body. Red-trace is the forces appearing on the ParaFlex and the Blue-trace is the force appearing on a rigid frame equivalent Quadrotor.

6.2.1 **Objective - (1)**

To demonstrate the first objective, that is, to show the impact force reduction capability, we conduct simulation studies. We then present experimental result of a collision course using a fabricated ParaFlex.

Figure 6.3 shows simulation of ParaFlex colliding with the wall as it navigates through a tight space. The numerical simulation was performed in Nastran [24]. Two simulation experiments were conducted. In one case, ParaFlex was made to traverse the trajectory shown in the figure. Because of the high speed, the ParaFlex collides into the walls. The Points of impact are shown in green boxes. The corresponding force on the Shroud is plotted. A log scale is used for the y-axis in order to accommodate large swings. In the second case, an equivalent rigid Quadrotor was used. The force appearing on the shroud in the case of a rigid Quadrotor is plotted in blue trace. As expected, the results suggest that ParaFlex incurred far less collision forces in comparison to the rigid Quadrotor. In order to conduct experiments, two version of ParaFlex were fabricated. The details are presented in the following section.

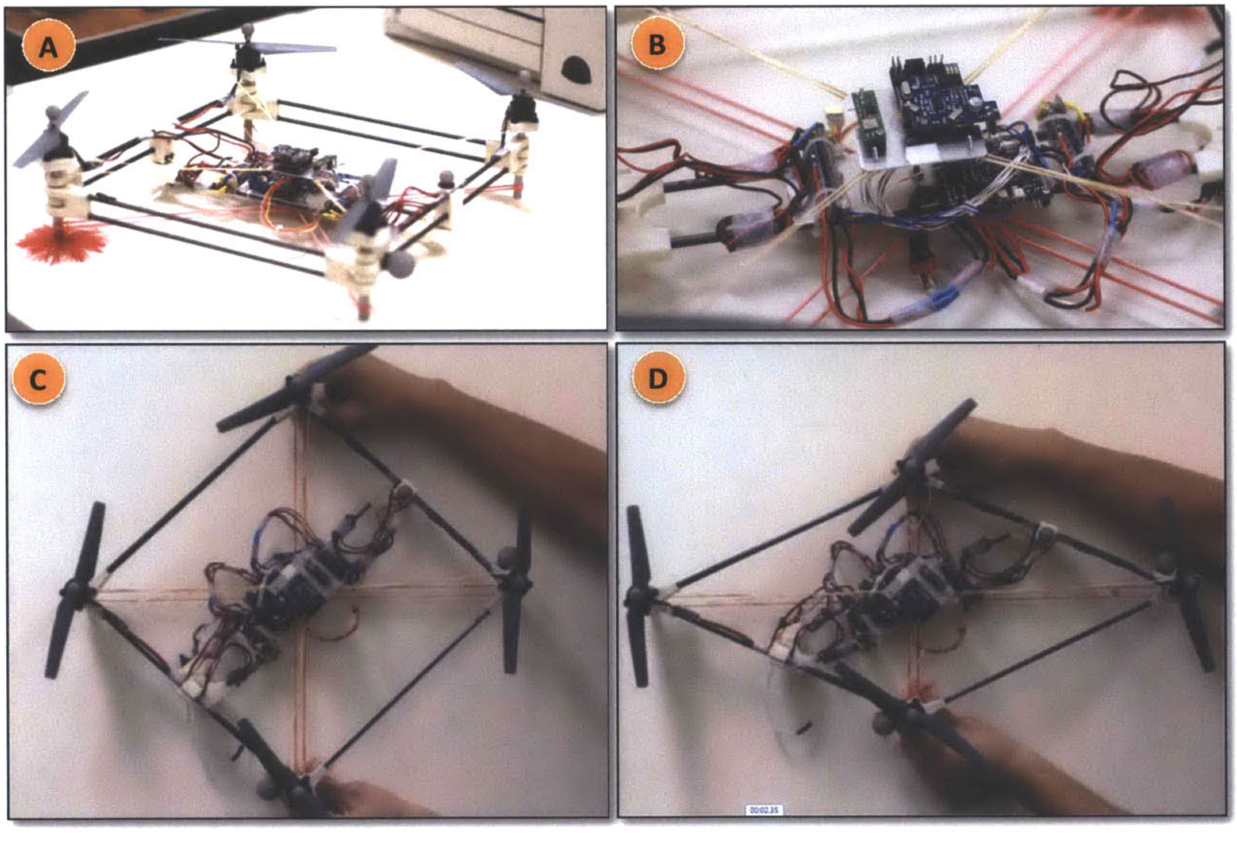

Figure 6.4: Version 1 of ParaFlex. (A) Shows the complete view of ParaFlex. (B) Shows the onboard computer made by X-UFO. (C) Shows the ParaFlex in relaxed state and (D) shows the ParaFlex in deformed state.

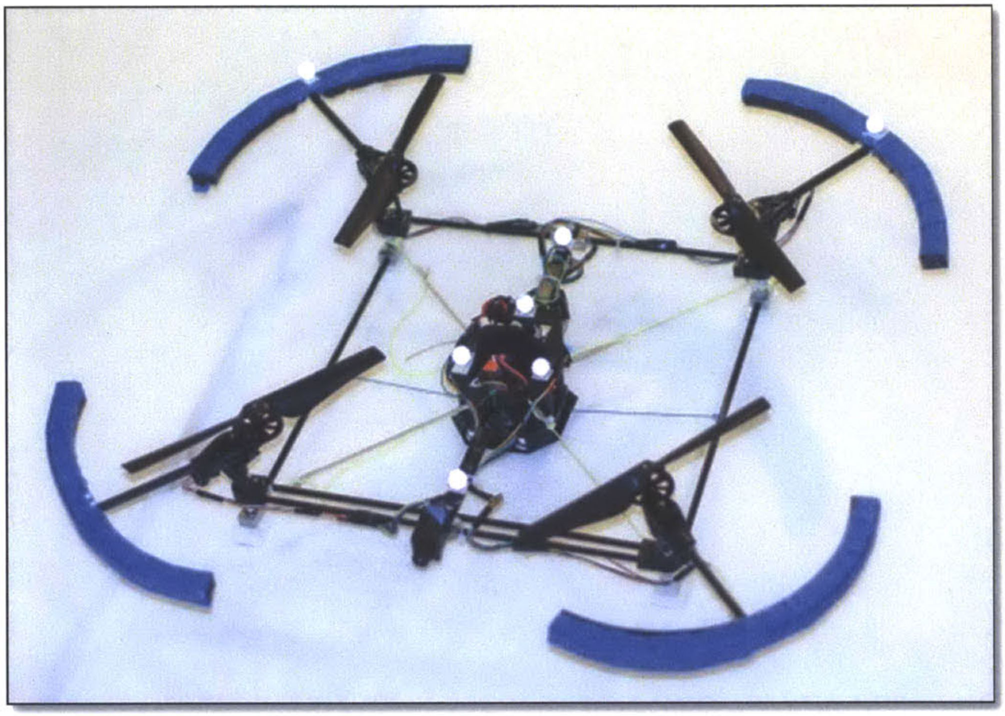

Figure 6.5: Version 2 of ParaFlex showing the protective shrouds around the propellers.

Figure 6.4 shows the first version of fabricated ParaFlex. It consists of four main components: The frame made of carbon fiber tubes, the revolute joints fabricated through rapid prototyping, the central processing unit and the actuators. The spring/damper is made using elastic bands that connected the diagonally opposite corners. Figure 6.5 shows second version of ParaFlex that includes four shrouds and is much lighter than the first version. Both the versions of ParaFlex sustained up to 8 minutes of flight time with one fully charged battery. The next section presents the collision experiment with ParaFlex.

Using the fabricated ParaFlex, a collision experiment was conducted as follows. The ParaFlex was made to collide with a static wall as shown in the inset of Figure 6.6. A Force-Torque sensor embedded in the wall measured the impact forces. The resulting force is shown in red-trace in Figure 6.6. The collision occurred at t=13.75sec. The spikes indicate repeated collision after which at t=13.9sec, a continuous force appeared due to sustained contact point. The key aspect to note is that the magnitude of the force is of the order of 60N.

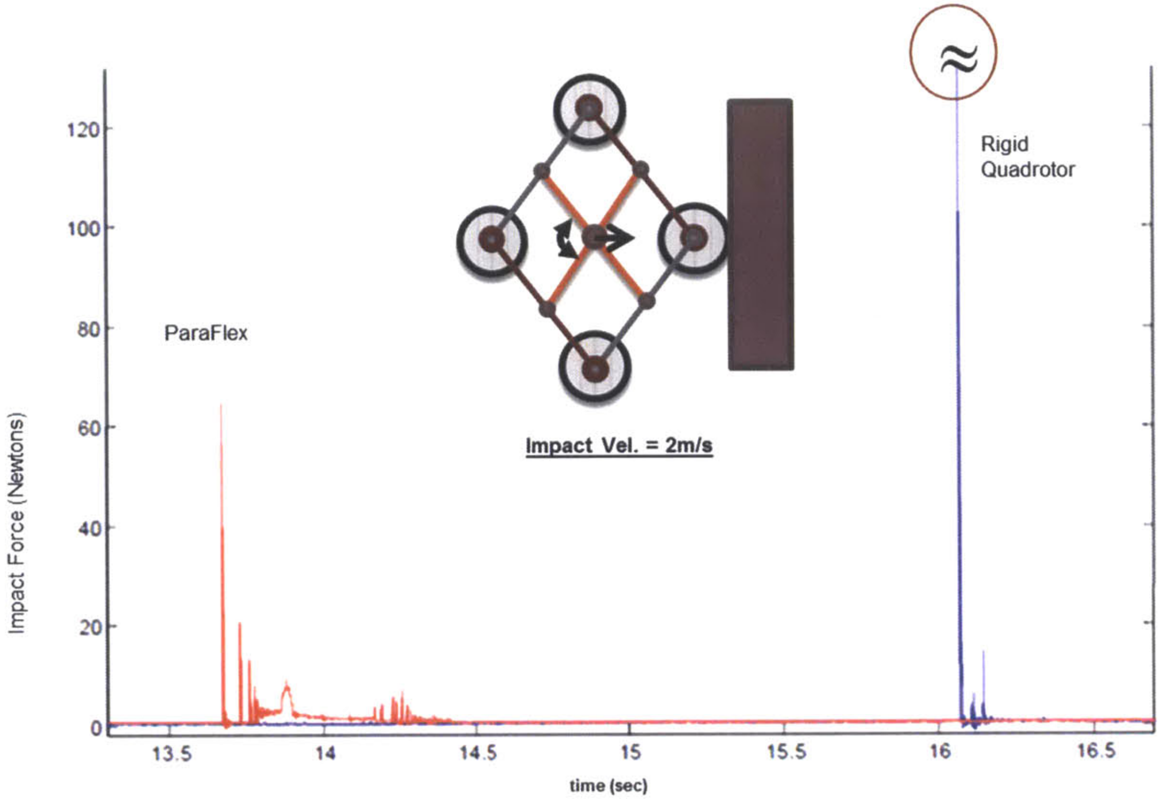

Figure 6.6: Experimental result of ParaFlex colliding with a wall (Red-trace). Blue trace corresponds to a rigid Quadrotor colliding with the wall with same impact velocity.

A similar experiment was conducted using a rigid Quadrotor. The resulting plot is shown with blue-trace in Figure 6.6. The collision occurred at t=16.1sec. The impact force was beyond the scale of the Force-Torque sensor and was estimated to be beyond 180N. The experiment resulted in a fracture of the corner joint, suggesting significantly higher impact force. The experiments clearly demonstrate the force reduction capability of ParaFlex.

6.2.2 **Objective – (2)**

We will now focus on the second objective mentioned in Section-6.2. That is, we show that the included spring and damper will not compromise the nominal flight performance. In other words, through analysis and simulation we study the impact of spring stiffness and damping coefficients on nominal flight performance.

We note that ParaFlex has 7 degrees of freedom. 6 degrees of freedom are due to rigid body motion and the 7th degree comes from the flexible mode. We derive the linearized dynamics of a Quadrotor assuming small angle of deformation λ. The approximate rigid body dynamics can be derived as

$$\begin{aligned}
\ddot{x}_Q &\approx \theta g + \frac{F_{ext}}{m_Q} \\
\ddot{y}_Q &\approx -\phi g \\
\ddot{z}_Q &\approx -g + \cos\theta\cos\phi\frac{F_{zb}}{m_Q} \\
\ddot{\phi} &\approx \frac{L}{L_0}\frac{M_\phi}{J_x} \\
\ddot{\theta} &\approx \frac{L}{L_0}\frac{M_\theta}{J_y} \\
\ddot{\psi} &\approx \frac{M_\psi}{J_z}
\end{aligned} \tag{6.1}$$

where, F_{ext} is the externally applied impact force, which is long the x-axis. L is the diagonal length of the ParaFlex and L_0 is the nominal relaxed length. The approximate dynamics of λ can be derived as

$$\ddot{\lambda} + \dot{\lambda}\frac{2K_D}{m_A L^2} + \left(\lambda - \frac{\pi}{2}\right)\frac{2K_S}{m_A L^2} = -\frac{M_\psi}{m_A L^2} + \frac{F_{ext}}{\sqrt{2}m_A L} \tag{6.2}$$

where m_A is the mass of the actuator and the shroud. According to Equation-(6.2), even when there is no external impact force, that is when $F_{ext} = 0$, the λ dynamics is excited by the yaw control input M_ψ. In other words, when the ParaFlex is commanded to Yaw, the yaw input causes an undesirable deformation in the structure. The extent of deformation depends on the parameters of the system.

A linear controller is implemented based on the dynamics in Equation-(6.1). The control design for such a system is discussed in Chapter-4.1. The plot in Figure 6.7 shows the performance of Yaw for a square wave reference input. The first plot shows a series of plots for $\psi(t)$ with varying spring stiffness K_S . When the spring stiffness is high ($K_S = 10$), $\psi(t)$ closely follows the commanded yaw $\psi_{cmd}(t)$. When the spring stiffness is reduced, yaw tracking error increases. The corresponding variation in the deformation angle λ can be seen in the second plot of Figure 6.7. The third plot shows the control input M_{ψ} . It is interesting to note that when the spring stiff is low, the control input energy is higher. This is simply due to larger tracking error in $\psi(t)$.

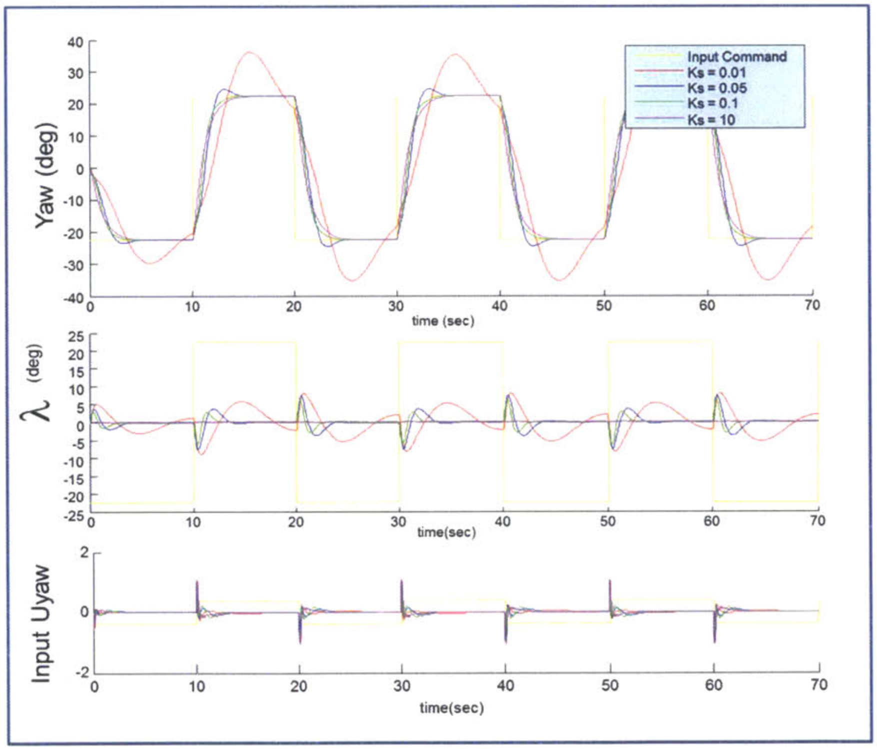

Figure 6.7: ψ and $\lambda-\frac{\pi}{2}$ tracking performance during free-flight.

The spring stiffness K_S is designed as follows. We first assume a maximum anticipated external impulsive for F_{ext} . We then design K_S (and K_D) such that the dynamics in Equation-(6.2) will result in a maximum swing of $\|\lambda\| < \lambda_{MAX}$. This is necessary to ensure the deformation is within this practical limits. We these designed K_S, K_D we perform simulation studies to estimate the maximum tracking error $|\psi - \psi_{cmd}|_{MAX}$. This permits design of input trajectory $\psi_{cmd}(t)$ that will ensure tracking error is within the tolerances.

6.2.3 **Objective – (3)**

When the ParaFlex collides with the wall, the impact forces deform the flexible joints. If the angle of impact (pitch angle) is such that $\theta > 0$, then the resulting impact will cause further increase in the pitch angle, that is the ParaFlex may flip over and slam into the obstacle. Let θ_{t+}, θ_{t-} be the angles just after and just before impact respectively. The extent of this transient depends on the impact velocity, mass of the Quadrotor, and the angle itself, and of course the control gains. This process is similar to that of docking we presented in in Chapter-4.4.3, wherein the proposed controller improved the robustness to aggressive docking.

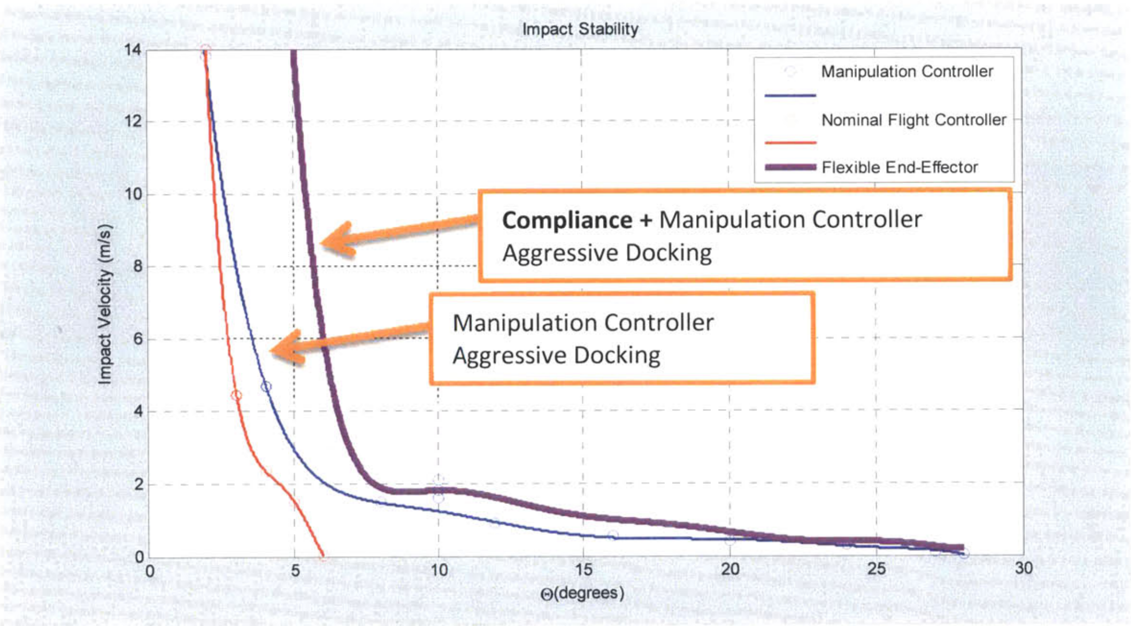

Figure 6.8: Performance of nonlinear controller on flexible Quadrotor. The red trace is the collision performance of a rigid Quadrotor using a baseline free-flight controller. Blue trace is the collision performance of a rigid Quadrotor with the controller presented in Chapter-4.4.3. The Magenta trace is the collision performance of a flexible Quadrotor using the controller of Chapter-4.4.3. The effective area covered under the curve is indicative of robustness.

We conducted simulation studies to use such a controller on the flexible UAV, which resulted in significant improvement in robustness. Plot in Figure 6.8 show the result of performing collision test with various pitch angle θ and impact velocities. It can be noted that a Flexible UAV with nonlinear our controller has a significantly improved robustness to collision.

6.3 **Summary**

In this chapter, we presented a novel flexible UAV design called ParaFlex. Through analysis, simulation and experiments we demonstrated the advantages of such a design. We showed that ParaFlex is robust to collisions and its nominal free- flight performance is close to that of a rigid Quadrotor. We proposed a nonlinear controller that improved collision robustness.

7 Conclusions and Future Work

The thesis proposed a novel UAV based manipulation scheme to apply controlled amount of forces to an Object in a stable manner. The thesis presented single and multi UAV, multi-degree of manipulation schemes. In order to achieve the mentioned goal, the thesis proposed several design of passive, light-weight end-effector that facilitated stable docking and efficient manipulation of Objects. To validate the proposed manipulation schemes, a Simulation, Test and Validation Environment (STeVE) is developed. The thesis presented the components and their interconnections within STeVE and how they permitted fast and efficient validation.

The underlying dynamics during manipulation is modeled by considering both the free-flight and contact cases. In both cases, the underlying state is divided into those pertain to degrees of freedom that govern manipulation, and those that do not. These are in turn used to ensure manipulation, and contact stability. Both single and two UAV-based manipulation tasks are considered. In all cases, contact stability conditions are derived and procedures to generate reference commands that ensure contact stability are presented. Simulation results are reported to support the validity of the models and model-based manipulation strategies. Flight tests with single and two Quadrotors are presented that demonstrate successful and stable UAV-based manipulation.

The thesis proposed a strategy for manipulation of kinematically constrained Object on a 2D surface. The proposed method used path of least resistance approach to manipulate object even when the kinematic constrains are unknown. By estimating the direction of motion of the Object through differential measurements, the Quadrotor UAV is commanded to apply a pushing force that not only ensures contact stability, but also results in successful manipulation. When the system parameters such as the mass are unknown, the thesis experimentally demonstrated the performance of adaptive nonlinear tracking controller.

The thesis proposed a novel flexible UAV design called ParaFlex. Through simulation and experiments, the advantages of ParaFlex over a rigid Quadrotor were demonstrated.

7.1 Future Work

The UAV based manipulation approach presented in this thesis is the first step in the direction. Manipulation by means of docking has many advantages over the other methods as described in chapter 1. In this section, we will present new of the next steps towards UAV based manipulation.

7.1.1 Multi Degree of Freedom Manipulation

The thesis considered a single degree of freedom manipulation. Using the same approach, multi degree manipulation scheme can easily be implemented. For instance, an Object on a 2D surface can be manipulated in all three degree of freedom, which is, pushing front-back, left-right and rotation.

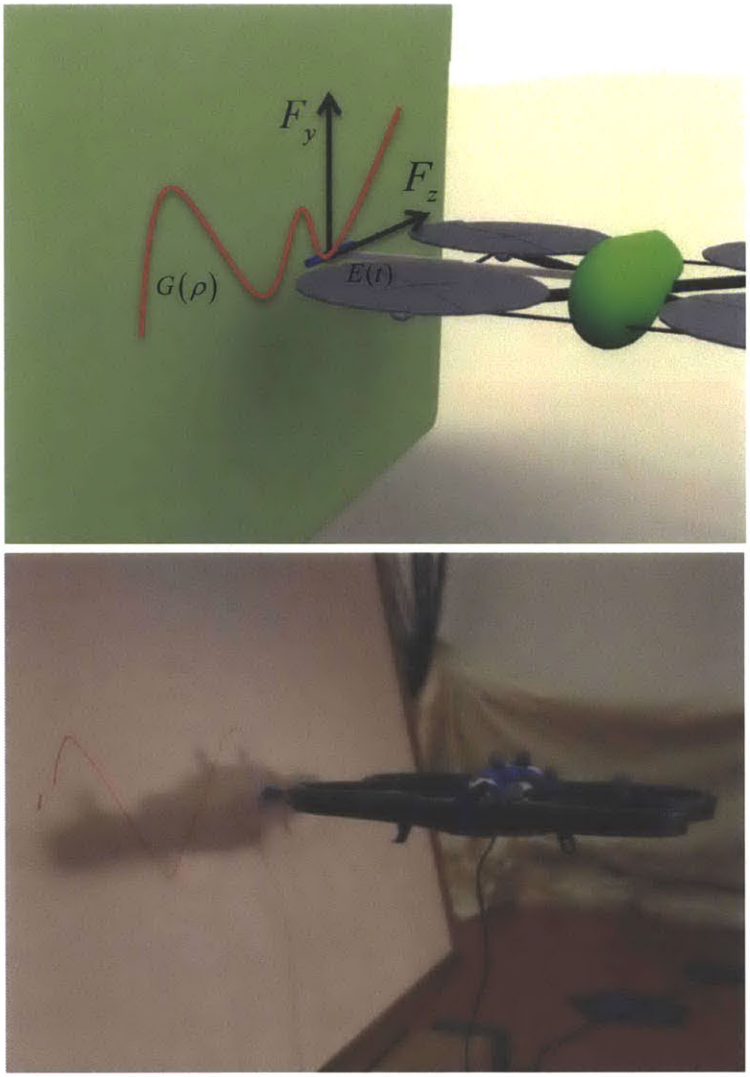

Figure 7.1: Example of Quadrotor scribing on a board.

Another example is shown in Figure 7.1, where a Quadrotor UAV is performing 2D manipulation on a surface. (Although this experiment was conducted during the thesis, the details of the approach are not presented in this thesis).

7.1.2 **Assistive Manipulation**

Since UAVs have limited air lifting payload capacity, larger conventional robotic systems can be used to air lift an Object. Using the approach proposed in this thesis, smaller UAVs can assist the larger robotic systems to perform small scale manipulation. One example is where a large crane can lift a heavy Object using a cable, and the UAV perform sideways manipulation to precisely maneuver the Object.

7.1.3 **Manipulation in GPS Denied Environments**

All of the experimental validation presented in this thesis used full state feedback. That is, the position and orientation of the Quadrotors were assumed to be known (As presented in Chapter 2, Flight Test Environment). This state information, however, may not be available in outdoor environments. The manipulation methods presented in this thesis can be suitably modified to incorporate onboard local sensing. One advantage of manipulation by docking is that, precise docking location is not necessary in most cases. Moreover, many of the state information (like angles) can be sensed using onboard sensors. Robust and practical use of the manipulation scheme presented in thesis requires development of state estimation methods in order to implement the control algorithms.

7.1.4 **Human in the Loop Manipulation**

Instead of having the computer perform the complete control task, a hybrid control approach can be developed that combines the human operator skills with controller's performance to achieve an overall improvement in manipulation performance. This essentially supplements the state information that is limited in GPS denied environments.

7.1.5 **Force Feedback**

The manipulation scheme proposed in this thesis can be thought of as a remote teleported robot. Data from the UAV can be sent back to the human operator as force feedback that permits the human operator to perform efficient manipulation. For instance, impedance control approach can be used to determine the force feedback. This of course requires suitable force feedback device on the operator end.

8 Bibliography

1. Parrot, A.R.D. *AR Drone*. 2012; Available from: http://ardrone.parrot.com/parrot-ar-drone/usa/.
2. Diy, D. *Arducopter*. 2012; Available from: http://diydrones.com.
3. Wikipedia. *Quadrotor*. 2012; Available from: http://en.wikipedia.org/wiki/Quadrotor.
4. Bouabdallah, S. and R. Siegwart. *Full control of a quadrotor*. in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2007.
5. Bouabdallah, S. and R. Siegwart. *Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor*. in *International Conference on Robotics and Automation*. 2005.
6. Hoffmann, G.M., et al., *Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment*, in *2007 AIAA Guidance, Navigation, and Control Conference and Exhibit; Hilton Head, SC; USA; 20-23 Aug. 2007*. 2007.
7. Pines, D.J. and F. Bohorquez, *Challenges Facing Future Micro-Air-Vehicle Development*. Journal of Aircraft, 2006. **43**(2): p. 290-305.
8. Siciliano, B. and O. Khatib, *Springer handbook of robotics*. 2008, Berlin: Springer. lx, 1611 p.
9. Valavanis, K.P. and M. Kontitsis, *A Historical Perspective on Unmanned Aerial Vehicles*. Advances in Unmanned Aerial Vehicles, 2007: p. 15-46.
10. Valavanis, K.P., G.J. Vachtsevanos, and P.J. Antsaklis, *Conclusions and the Road Ahead*. Advances in Unmanned Aerial Vehicles, 2007: p. 533-543.
11. Valavanis, K. and SpringerLink (Online service), *Advances in unmanned aerial vehicles state of the art and the road to autonomy*, in *International series on intelligent systems, control and automation--science and engineering v 33*. 2007, Springer: Dordrecht.
12. Mellinger, D., et al. *Design, Modeling, Estimation and Control for Aerial Grasping and Manipulation*. in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011.
13. Michael, N., J. Fink, and V. Kumar, *Cooperative manipulation and transportation with aerial robots*. Autonomous Robots, 2011. **30**(1): p. 73-86.
14. Pounds, P., R. Mahony, and P. Corke, *Modelling and control of a large quadrotor robot*. Control Engineering Practice, 2010. **18**(7): p. 691-699.
15. Pounds, P.E.I., D.R. Bersak, and A.M. Dollar. *Grasping From the Air: Hovering Capture and Load Stability*. in *IEEE International Conference on Robotics and Automation*. 2011.
16. Muller, M., S. Lupashin, and R. D'Andrea. *Quadrocopter ball juggling*. in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. 2011.
17. D'Andrea, R. *Distributed Flight Array*. Available from: http://www.idsc.ethz.ch/Research_DAndrea/DFA.
18. D'Andrea, R. *Flying Machine Enabled Construction*. Available from: http://www.idsc.ethz.ch/Research_DAndrea/fmec.
19. Mellinger, D. and V. Kumar. *Minimum Snap Trajectory Generation and Control for Quadrotors*. in *IEEE International Conference on Robotics and Automation*. 2011.
20. Mellinger, D., N. Michael, and V. Kumar, *Trajectory generation and control for precise aggressive maneuvers with quadrotors*. International Journal of Robotics Research, 2012.
21. Michael, N., et al., *The GRASP Multiple Micro UAV Testbed*. IEEE Robotics \& Automation Magazine, 2010. **17**(3): p. 56-65.
22. Mathworks. *MATLAB Simulink*. 2012; Available from: http://www.mathworks.com/products/simulink/.
23. Nvida. *PhysX -- Realtime Physics Engine Middleware*. 2012; Available from: http://en.wikipedia.org/wiki/PhysX.

24. MSC. *MSC Visual Nastran Desktop Student Edition*. Available from: http://www.mscsoftware.com/

25. PhantomHaptic. *Phantom Haptic Device -- Premium 1.5A. 6DoF Motion Sense, 3DoF Force Feedback*. 2012; Available from: www.sensable.com.

26. Vicon. *Vicon MX*. 2012; Available from: http://www.vicon.com/products/viconmx.html.

27. Amir, M.Y. and V. Abbass. *Modeling of quadrotor helicopter dynamics*. in *International Conference on Smart Manufacturing Application*. 2008.

28. Das, A., K. Subbarao, and F. Lewis, *Dynamic inversion with zero-dynamics stabilisation for quadrotor control.* IET Control Theory Applications, 2009. **3**(3): p. 303-314.

29. Huang, H., et al. *Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering*. in *IEEE International Conference on Robotics and Automation*. 2009.

30. Lavretsky, E. and N. Hovakimyan, *Stable adaptation in the presence of actuator constraints with flight control applications.* Journal of guidance, control, and dynamics, 2000. **30**(2): p. 337--345.

31. Lee, D., H.J. Kim, and S. Sastry, *Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter.* International Journal of Control, Automation and Systems, 2009. **7**: p. 419-428.

32. Madani, T. and A. Benallegue. *Adaptive Control via Backstepping Technique and Neural Networks of a Quadrotor Helicopter*. in *17th World Congress The International Federation of Automatic Control*. 2008.

33. Mokhtari, A. and A. Benallegue. *Dynamic feedback controller of Euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle*. in *International Conference on Robotics and Automation*. 2004.

34. Yushu, Y. and D. Xilun, *A Quadrotor Test Bench for Six Degree of Freedom Flight.* Journal of Intelligent \& Robotic Systems, 2012. **DOI 10.1007/s10846-012-9680-y**: p. 1-16.

35. Huang, M., et al. *Adaptive tracking control of underactuated quadrotor unmanned aerial vehicles via backstepping*. in *American Control Conference*. 2010.

36. Pounds, P., et al. *Towards Dynamically Favourable Quad-Rotor Aerial Robots*. in *Australasian Conference on Robotics and Automation*. 2004.

37. Slotine, J.J.E. and W. Li, *Applied nonlinear control*. 1991, Englewood Cliffs, N.J.: Prentice Hall. xv, 459 p.

38. Narendra, K.S., I.H. Khalifa, and A.M. Annaswamy, *Error Models for Stable Hybrid Adaptive Systems.* Ieee Transactions on Automatic Control, 1985. **30**(4): p. 339-347.

39. Ducard, G. and R. D'Andrea. *Autonomous quadrotor flight using a vision system and accommodating frames misalignment*. in *Industrial Embedded Systems*. 2009.

40. Mian, A.A., M.I. Ahmad, and D. Wang. *Backstepping based PID Control Strategy for an Underactuated Aerial Robot*. in *World Congress The International Federation of Automatic Control*. 2008.

41. Adigbli, P., et al. *Nonlinear Attitude and Position Control of a Micro Quadrotor using Sliding Mode and Backstepping Techniques*. in *3rd US-European Competition and Workshop on Micro Air Vehicle Systems \& European Micro Air Vehicle Conference and Flight Competition*. 2007.

42. Arimoto, S. and F. Miyazaki, *Stability and robustness of PID feedback control for robot manipulators of sensory capability*, in *Robotics Research*. 1984, MIT Press, Cambridge, Massachussetts. p. 783--799.