Manipulators on Micro Aerial Vehicles (MAVs)

Thesis

submitted in partial fulfillment of the requirements
for the degree of

Bachelor of Technology

by

Harleen Hanspal

Roll No: 13EE01026

under the guidance of

Dr. N.B. Puhan

Electrical Engineering

School of Electrical Sciences

Indian Institute of Technology Bhubaneswar

Bhubaneswar - 751013.

2017

Declaration

I certify that:

The work contained in this thesis is original and has been done ourselves under the general supervision of my supervisor and professors.

The work has not been submitted to any other institute for any degree or diploma.

I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

Whenever we have used material (data, theoretical analysis and text) from other sources, we have given due credit to them by citing them in the text of the thesis and giving their detials in the references.

Harleen

13EE01026

Abstract

Historically, UAV research has been focused on avoiding interaction with the environment. This fact is mostly due to poor payload capabilities available in micro UAVs in addition to the possibility of crashing and causing injury. So far, UAVs have been used mostly in surveillance and search and rescue missions. However, the ability for air vehicles to manipulate a target or carry objects they encounter could greatly expand the types of missions achievable by unmanned systems. Flying robots with dexterous arms could lead to transformative applications in near-Earth cluttered and clustered environments. Such applications could be infrastructure inspection and repair, agricultural care and possibly even construction and assembly. However, aerial grasping and manipulation still remains largely underdeveloped.

Therefore, in this project, we work towards making a manipulator platform on an aerial vehicle with the objective being to place a tool on an (Revolute Revolute Revolute (RRR)) movable platform on top of a quadcopter and make it exert a given force-time curve on a surface. Final objective is to make this a complete product easily usable by people.

Acknowledgments

I express my sincere gratitude towards my guide **Dr. N.B. Puhan** for his constant help, encouragement, inspiration and the immense confidence that he showed in me throughout the project work. Without his invaluable guidance and the support he gave me, this work would never have been a successful one.

I must thank Dr. Satyanarayan Panigrahi, Assistant Professor, School of Mechanical Sciences, IIT Bhubaneswar, for constantly guiding me and brainstorming with me, giving ideas and solutions whenever I was stuck and teaching me the right approach to working about any problem.

I thank Dr. Ranjith Mohan, Associate professor, Aerospace Department, IIT Madras without whom I would have never ventured into the field of Aerial Vehicles and Professor Jayanta Pal, School of Electrical Sciences, IIT Bhubaneswar, without whose encouragement and valuable guidance, I could not have sustained my zeal for Controls and Dynamics. I also thank all the faculty at School of Electrical Sciences for reviewing my presentations and giving their valuable feedback and perspective time and again. I also thank the above people for letting me use their laboratory resources whenever I needed.

I would also like to thank my friends, especially Rahul Kumar Kumawat, final year student, School of Mechanical Sciences, IIT Bhubaneswar, who helped me with their ideas and suggestions whenever I got stuck, were there to make videos or was simply scared to experiment with my crazy quad.

Harleen Hanspal

IIT Bhubaneswar

April 30, 2017

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Historically, UAV research has been focused on avoiding interaction with the environment, majorly due to poor payload capabilities of micro UAVs, leading to heavy battery discharge rates, and more alarmingly the possibility of crashing and causing injury. Therefore, so far UAVs have been used mostly in surveillance, search and rescue missions. However, the ability for air vehicles to manipulate a target or carry objects they encounter could greatly expand the types of missions achievable by unmanned systems. Flying robots with dexterous arms could lead to transformative applications in near-Earth cluttered and clustered environments. Such applications could be infrastructure inspection and repair, agricultural care, in military and possibly even construction and assembly. However, aerial grasping and manipulation still remains largely underdeveloped.

Therefore, in my final year project, I work towards developing a manipulator platform on an aerial vehicle with the objective being to mount any tool on an (Revolute Revolute Revolute (RRR) spatial manipulator) placed on a movable platform like quadcopter in our case and make it exert a given force-time curve on a surface. Final objective is to make this as a complete product easily usable by a user. There have been similar works by some individual researchers and in labs like MIT Adaptive Control lab where they have used quadcopters for force generation tasks but those are very general tasks i.e. there target of interaction is not localized and it does not require precise force exertion on that target. While in what we are trying to achieve, stability and a good level of precision both have to be achieved simultaneously i.e. the reaction forces from the effort of force exertion have to be countered while maintaining the precise position and attitude

of the quad, which is a challenge.
In a nutshell, this B.Tech. project is more focused on the particular application of UAVs to do precise force intensive contact tasks. The tasks may be in indoor or outdoor environments but the environment is assumed to be known and localized rather than unknown, therefore even if the environment is congested and cluttered, dynamic trajectory planning is not the major concern here. We focus more on the modeling and control of the aerial vehicle.

1.1 Motivation

When the aerial vehicles these days are so accessible and undoubtedly useful, one questions what next? Ideas come when you compare their usabilities with those of the ground mobile vehicles and one can clearly ask why to use them just as a portable personal satellite i.e. just for surveillance/scouting purposes or for picking and placing services but extend their capabilities to interact in a more tactile manner with their environment. Also since I have always been interested in exploring and working in the field of controls and dynamics, I found this project a good opportunity to hone myself in every aspect of this field since this project essentially is just a design and control problem. Researchers are still facing some challenges in the control field because the quadrotor is a highly nonlinear, multivariable system and since it has six Degrees of Freedom (DOF) but only four actuators, it is an underactuated system. They are very difficult to control due to the nonlinear coupling between the actuators and the degrees of freedom. Although the most common and used flight control algorithms are linear flight controllers, these controllers can only perform when the quadrotor is flying around hover, they suffer from a huge performance degradation and are prone to failure when the quadrotor leaves the nominal range about the set operating point or performs aggressive maneuvers.

1.2 Thesis Significance and Scope

The contributions of this work therefore are deriving an accurate and detailed mathematical model of the quadrotor UAV along with the manipulator mounted on it, developing and testing some linear and nonlinear control algorithms for quadcopter and applying those on both the derived mathematical model in computer based simulations and on a self-built quadcopter. We also draw comparison between these control algorithms in terms of their dynamic performance and their ability to stabilize the system under the effect of possible disturbances which are inevitable in our application as the quad interacts with various targets in the environment like wall, windows, etc. or when it is doing work involving impulse forces like sudden discharge of fluid or bullets (potential applications), etc.

The research put in for this thesis gave us great insight and knowledge into the area of Mobile Base Manipulators which are used in area of aircrafts and naval bases missile launches where the mobile base is the ship or the aircraft and its stability is maintained in spite of the external forces or the impulses from the missile firings, etc. Possible applications in the industry:

- infrastructure inspection and repair
- agricultural care
- for various applications in military
- construction and assembly
- chores like drilling, painting, cleaning, etc.
- ventilation ducts, sewage cleaning, etc.

The main advantage of this project over a regular conventional ground robot or manipulator is when one cannot find a base for keeping/fixing the manipulator to work on.

1.3 Thesis Outline and Overview

The project was divided into 2 parts, first focusing on the quadcopter assembly, stability and navigation and second the manipulator (robotic arm) design and study. The final step being the integration of these two parts. The goal outline is as follows: **The quadcopter, with the manipulator mounted on top,should fly to the position of the target and the manipulator then should exert the specified amount of time-varying force on the target by adjusting its pose.** Therefore, systematically dividing the goal into first, a trajectory following problem (but with extra payload load unsymmetrical about its Center of Gravity (CoG) and then, a pose (orientation) maintaining problem amidst large reaction force, moment and vibrations.

1.3.1 Manipulator

A robotic arm is a type of mechanical arm, usually programmable, with similar functions to a human arm; the arm may be the sum total of the mechanism or may be part of a more complex robot. The links of such a manipulator are connected by joints allowing either rotational motion (such as in an articulated robot) or translational (linear) displacement. The links of the manipulator can be considered to form a kinematic chain. The terminus of the kinematic chain of the manipulator is called the end effector and it is analogous to the human hand. The end effector, or robotic hand, can be designed to perform any desired task such as welding, gripping, spinning etc. In some circumstances, close emulation of the human hand is desired, as in robots designed to conduct bomb disarmament and disposal.

1.3.2 Aerial Vehicle

Micro aerial vehicles, in contrast to the vehicles which can weigh a lot, which are very big and fly high above the Earth's surface, are much smaller and capable of navigating in indoor environments and three-dimensional unstructured settings. They come in many different geometries and designs like fixed wing vehicles, flapping wing vehicles and rotor crafts like helicopters, quadrotors or hex rotors. // A quadcopter is a multirotor helicopter that is propelled by four, generally vertically oriented, rotors or propellers. The speed of each rotor is varied independently to achieve control of quadcopter. Reasons for choosing to work on quadcopter:

- At a small size, quadcopters are cheap, safe, highly maneuverable (capable of attaining speeds of up to 2000deg/sec) and durable.
- The mechanical design and control are both are very simple.
- The small size of the blades is advantageous because the kinetic energy possessed by them is less, hence reducing their ability to cause damage. This makes the vehicles safe for close interaction.

However increasing blade size increases their momentum. This means that change in blade speed take longer time, which negatively impacts the control. At the same time, increasing blade size improves efficiency as it takes less energy to generate thrust by

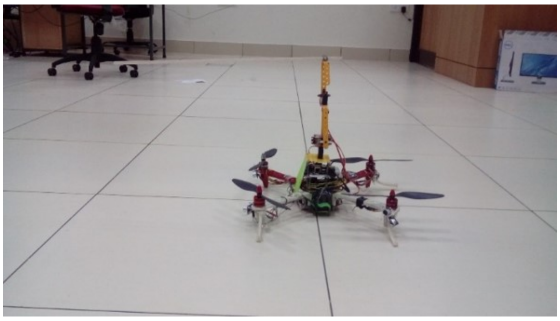


Figure 1.1: The Prototype developed at DIC lab.

moving a large mass of air at a slow speed than by moving a small of air at a high speed. Therefore increasing the efficiency comes at the cost control.

Chapter 2

Literature Survey

2.1 Previous work

Although work has been done in this area with ground-based vehicles, little work has been done in aerial vehicles where arm or manipulator motions may lead to decreased stability and control problems. There have been a few successful attempts in aerial grasping using a 1-DOF (degree of freedom) grasper or gripper. Some groups have introduced gimbals, suspended payload, force sensors, or brushes attached to quadrotors or duct-fan vehicles where the manipulator is used for contact inspection. The AIRobots consortium is developing service robots for use in hazardous or unreachable locations. In recent years, robots that rely on dynamic control techniques for movement or manipulation have become increasingly prevalent. Robotic systems of this type involve challenging control problems due in part to the fact that a robots base and its manipulators may be dynamically coupled. For dynamically balancing robots, robots with flexible bases, or humanoids, base reactions created by the vehicles manipulator may be significant enough to warrant active reduction or compensation to maintain stability, which we also verified from our initial experiments done fr this project. Additionally, others have analyzed arm recovery motions to reduce the impact on a dynamically stable base vehicle. It is apparent that compensation of reactionary forces caused during manipulation or manipulator movement is critical to robust control of an aerial vehicle and that is something which has been tried to achieve in this project as well. There have been works by some individual researchers and in labs like MIT Adaptive Control lab where they have used quadcopters for force generation tasks but those are very general tasks i.e. not localized or requiring precise force exertion on a surface.

2.2 Direction of Present Work

In this work, a novel approach for coordinating the onboard manipulator and a quadcopter is taken up such that they together apply time dependent force on a target with accuracy and precision. Stability and a good level of precision both have to be achieved simultaneously i.e. the reaction forces from the effort of force exertion have to be countered while maintaining the precise position and attitude of the quadcopter.

Chapter 3

Quadcopter

In this chapter, the kinematics and dynamics models of a quadrotor will be derived based on a Newton-Euler formalism with the following assumptions:

- The structure is rigid and symmetrical.
- The Center of Gravity (CoG) of the quadrotor coincides with the body frame origin.
- The propellers are rigid.
- Thrust and drag are proportional to the square of propeller's speed.

The aerodynamic and gyroscopic effects acting on the quadrotor body and dynamics of the quadrotor motors are not considered here, since they are not found to have a significant effect on the quadcopter in our experiments in near-earth environments and also in most literature. In the end we derive a state space model for the quadrotor system that will be used to test our controllers. During further simulation study, the assumptions of the symmetricity and CoG coinciding with the body frame origin have been relaxed because they are not practically possible in our application.

3.1 Kinematic Model

Figure 3-1 shows the Earth reference frame with N, E and D axes and the body frame with x, y and z axes. The Earth frame is an inertial frame

$x_e d$ on a specific place at ground level as its name implies, it uses the N-E-D notation where the axes point to the North, East and Downwards respectively. Another frame, the body frame is at the center of the quadrotor body, with its x-axis pointing towards propeller 1, y-axis pointing towards propeller 2 and the z-axis is pointing to the ground.

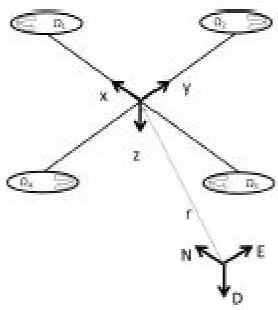


Figure 3.1: Figure 3-1: Quadrotor Reference Frames

The distance between the Earth frame and the body frame describes the absolute position of the center of mass of the quadrotor $r = [xyz]^T$. The rotation R from the body frame to the inertial frame describes the orientation of the quadrotor. The orientation of the quadrotor is described using roll, pitch and yaw angles ($\phi\theta and \chi$) representing rotations about the X, Y and Z-axes respectively. Assuming the order of rotation to be roll (ϕ), pitch (θ) then yaw (χ), the rotation matrix R which is derived based on the sequence of principle rotations is:

$$R = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\theta c\psi & c\phi s\theta s\psi - s\theta c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

where c and s denote cos and sin respectively.
To acquire information about the angular velocity of the quadrotor, typically an on-board Inertial Measurement Unit (IMU, a combination of an accelerometer, gyroscoe and magnetometer) is used which will in turn give the velocity in the body coordinate frame. To relate the Euler rates $\eta = [\dot{\phi}\dot{\theta}\dot{\chi}]^T$ that are measured in the inertial frame and angular body rates $\omega = [pqr]^T$, a transformation is needed as follows:
$\omega = R_r\dot{\eta}$
where

$$R_r = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}$$

3.2 Dynamics Model

The motion of the quadrotor can be divided into two subsystems; rotational subsystem (roll, pitch and yaw) and translational subsystem (altitude and x and y position).

3.2.1 Rotational Equations of Motion

The rotational equations of motion are derived in the body frame using the Newton-Euler method:

$$\mathrm{J}\dot{\omega} + \omega \text{ x } \mathrm{J}\omega + \omega \text{ x } [0\ 0\ \mathrm{J}_r\Omega_r]^T = \mathrm{M}_B$$

where:

J Quadrotor's diagonal inertia Matrix

$\dot{\omega}$ Angular body rates

M_G Gyroscopic moments due to rotors' inertia

M_B Moments acting on the quadrotor in the body frame

J_r rotors' inertia

r rotors' relative speed $\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$

$$M_B - M_d = \begin{bmatrix} lK_f(-\Omega_2^2 + \Omega_4^2) \\ lK_f(+\Omega_1^2 - \Omega_3^2) \\ K_M(-\Omega_2^2 + \Omega_1^2 - \Omega_4^2 + \Omega_3^2) \end{bmatrix} \quad (3.1)$$

where l is the moment arm, which is the distance between the axis of rotation of each rotor to the origin of the body reference frame which should coincide with the center of the quadrotor and M_d is the drag moment. Using which and putting it in the above relation, we get:

$$I \begin{bmatrix} \ddot{phi} \\ \ddot{theta} \\ \ddot{sci} \end{bmatrix} + \begin{bmatrix} \dot{phi} \\ \dot{theta} \\ \dot{sci} \end{bmatrix} * I \begin{bmatrix} \ddot{phi} \\ \ddot{theta} \\ \ddot{sci} \end{bmatrix} + \begin{bmatrix} \ddot{phi} \\ \ddot{theta} \\ \ddot{sci} \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ J_r \omega_r \end{bmatrix} = M_B - M_d \tag{3.2}$$

3.2.2 Translational Equations of Motion

The translation equations of motion for the quadrotor are based on Newton's second law and they are derived in the Earth inertial frame.

$$M\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ -K_f(+\Omega_2^2 + \Omega_1^2 + \Omega_4^2 + \Omega_3^2) \end{bmatrix} - F_d(Dragforce) \tag{3.3}$$

The second matrix is multiplied by the rotation matrix R to transform the thrust forces of the rotors from the body frame to the inertial frame, so that the equation can be applied in any orientation of the quadrotor.

3.2.3 State Space Modeling

Defining the state vector of the quadcopter to be:

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \end{bmatrix}^T$$

which is mapped to the degrees of freedom of the quadcopter in the following manner:

$$X = \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & z & \dot{z} & x & \dot{x} & y & \dot{y} \end{bmatrix}^T$$

Control Input Vector

$$U = \begin{bmatrix} U_1 & U_2 & U_3 & U_4 \end{bmatrix} \tag{3.4}$$

where

$U_1 = K_f(+\Omega_2^2 + \Omega_1^2 + \Omega_4^2 + \Omega_3^2)$

$U_2 = lK_f(-\Omega_2^2 + \Omega_4^2)$

$U_3 = lK_f(+\Omega_1^2 - \Omega_3^2)$

$U_4 = K_M(-\Omega_2^2 + \Omega_1^2 - \Omega_4^2 + \Omega_3^2)$

Using which we can find the expressions for the motor rpms in terms of the control vector elements, as below:

$$
\begin{aligned}
\Omega_1 &= \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_3 + \frac{1}{4K_M}U_4} \\
\Omega_2 &= \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_2 - \frac{1}{4K_M}U_4} \\
\Omega_3 &= \sqrt{\frac{1}{4K_f}U_1 - \frac{1}{2K_f}U_3 + \frac{1}{4K_M}U_4} \\
\Omega_4 &= \sqrt{\frac{1}{4K_f}U_1 + \frac{1}{2K_f}U_2 - \frac{1}{4K_M}U_4}
\end{aligned}
$$

Rotational Equations of Motion

$$
\begin{aligned}
\ddot{\phi} &= \frac{l}{I_{xx}}U_2 - \frac{J_r}{I_{xx}}\dot{\theta}\Omega_r + \frac{I_{yy}}{I_{xx}}\dot{\psi}\dot{\theta} - \frac{I_{zz}}{I_{xx}}\dot{\theta}\dot{\psi} \\
\ddot{\theta} &= \frac{l}{I_{yy}}U_3 - \frac{J_r}{I_{yy}}\dot{\phi}\Omega_r + \frac{I_{zz}}{I_{yy}}\dot{\phi}\dot{\psi} - \frac{I_{xx}}{I_{yy}}\dot{\psi}\dot{\phi} \\
\ddot{\psi} &= \frac{1}{I_{zz}}U_4 + \frac{I_{xx}}{I_{zz}}\dot{\theta}\dot{\phi} - \frac{I_{yy}}{I_{zz}}\dot{\phi}\dot{\theta}
\end{aligned}
$$

With the choice of the control input vector U, it is clear that the rotational subsystem is fully-actuated, it is only dependent on the rotational state variables $x_1 to x_6$ that correspond to$\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}$respectively.

Translational Equations of Motion

$$
\begin{aligned}
\ddot{x} &= \frac{-U_1}{m}(\sin x_1 \sin x_5 + \cos x_1 \cos x_5 \sin x_3) \\
\ddot{y} &= \frac{-U_1}{m}(\cos x_1 \sin x_5 \sin x_3 - \cos x_5 \sin x_1) \\
\ddot{z} &= g - \frac{U_1}{m}(\cos x_1 \cos x_3)
\end{aligned}
$$

It is clear here that the translational subsystem is underactuated as it dependant on both the translational state variables and the rotational ones.

Using both the rotational and translational equations of motion the complete mathematical model of the quadrotor can be written in a state space representation. The thesis by Helba Elkosy [1], among others has been used to understand the above dynamics aspects.

3.3 Simulation Study

There are several control techniques that can be used to control a quadrotor varying between the classical linear Proportional-Integral-Derivative (PID) or Proportional Derivative (PD) controller to more complex nonlinear schemes as Backstepping or Sliding-mode

controllers, some of which have been tried on our testbed. The flight control systems can be classified into four main categories which are: linear fight control systems, non-linear flight control systems, hybrid and learning-based flight control systems as in [1]. It was found that the most common control technique used is the PID or PD controller. Although it is a linear controller used for the nonlinear multivariable quadrotor system, it was proven successful in many literature.
For this project, the whole flight has been simulated and tested in 2 modes:

- Mode 1- the conventional trajectory control with PID controller.

- Mode 2- similar to attitude control with $Z_des = [x, \theta, z, \psi]$ tracking and where the originality of our work lies.
  We try and compare 3 controllers with modified dynamic model i.e. one involving reaction forces and vibrations. Also in this mode, quad has to constantly tilt to a big angle, approximately. 30-40 degrees, which necessitates the use of the nonlinear controller rather than a linear controller like PID or if PID is used the operating point has to be updated. The second option is easier and produces better results also. For verifying the first option, controllers like SMC and Backstepping are simulated and tried.

3.3.1 Trajectory generation

Mode 1:

For trajectory generation, the quadrotor along a straight line between each pair of consecutive waypoints using a constant velocity of 0.5 m/s (minimum distance trajectory assuming a 1st order dynamical system, therefore meaning that the systems velocities can be controlled as desired) which is working well for Mode 1. Effort for a better smoother trajectory generation method like minimum jerk trajectory and cubic spline was made by the method below. This will also be required if the quadrotor is moving while working since in that case using just the minimum distance (velocity) trajectory planning will not work at all such as in the application of using the quadcopter to write on the board or in case of complex trajectories. For finding $x(t)$:

- Finding an $x(t)$ that minimises the functional i.e. the Langrange function.

- Solving to find the Euler Langrange equation which is a necessary condition satisfied by the optimal function irrespective of what function is taken in the functional.

Putting $5nm$ boundary conditions and solving for coefficients $(c_1, c_2, c_3, c_4, c_5)$ as in $x(t) = c_1 + c_2x + c_3x^2 + c_4x^3 + c_5x^4$ in case of minimum jerk trajectory, by simple matrix multiplication where n is the number of trajectory waypoints and m depends on what kind of functional one is using i.e. m is 5 in the above case.

For trajectory tracking, the 'r' desired corresponding to the desired trajectory is desired from a Proportional plus derivative controller.

$$e_i = (r_{iT} - r_i)$$

For error to go exponentially to zero:

$$\ddot{r_{iT}} + k_d(\dot{r_{iD}} - \dot{r}_i) + k_p(r_{iD} - r_i) = 0$$

The mathematical deriviation of the above comes from trying to decrease the order of convergence of the error in say, x exponentially down to zero, from which this relation comes:

$$u(t) = \ddot{x}_{des}(t) + K_v\dot{e(t)} + K_pe(t)$$

Mode 2:

Changed model dynamics for Mode 2:

$$M\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + R(\begin{bmatrix} 0 \\ 0 \\ -F \end{bmatrix} - F_d - \begin{bmatrix} 0 \\ Fsin(\theta) \\ 0 \end{bmatrix}) \tag{3.5}$$

where F is the net thrust by the quadcopter

$$F = K_f(+\Omega_2^2 + \Omega_1^2 + \Omega_4^2 + \Omega_3^2) \tag{3.6}$$

Also M_B gets modified to include:

$$M_B - M_d = \begin{bmatrix} lK_f(-\Omega_2^2 + \Omega_4^2) \\ lK_f(+\Omega_1^2 - \Omega_3^2) \\ K_M(-\Omega_2^2 + \Omega_1^2 - \Omega_4^2 + \Omega_3^2) \end{bmatrix} + \begin{bmatrix} Fsin(\theta) * (d_1cos(\theta) - d_2sin(\phi)) \\ 0 \\ Fsin(\theta) * d_2cos(\phi) \end{bmatrix} \tag{3.7}$$

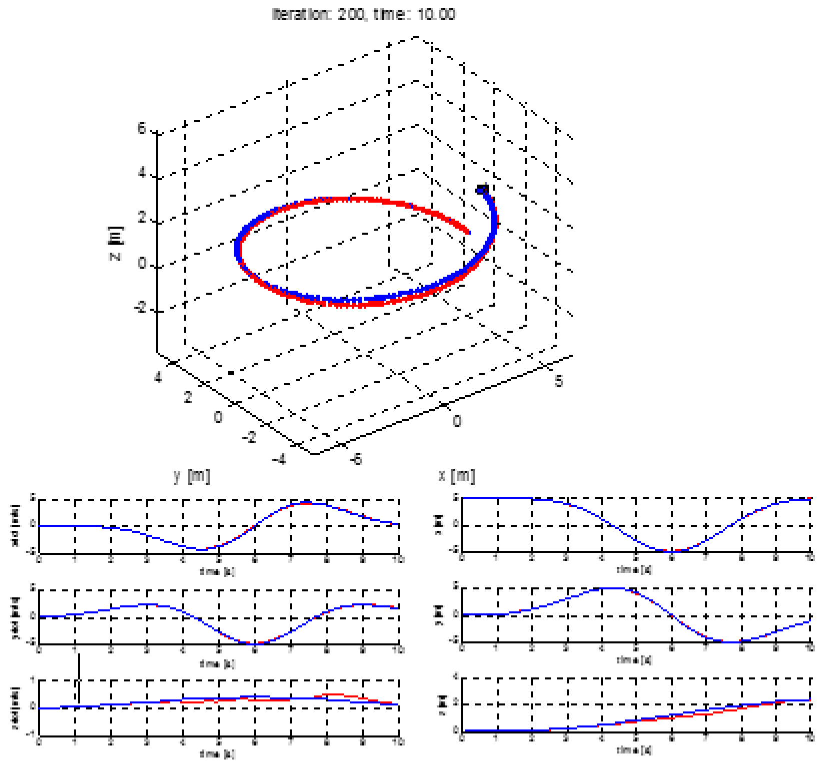


Figure 3.2: Trajectory following of quad in 3 dimensions, simulated using GRASP testbed. Error tolerances: position-0.05m angles-5Bachelor of Technology

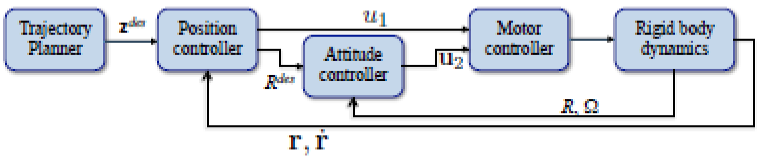


Figure 3.3: Controller loops- Inner running at 100Hz for pose control; Outer running at 50Hz for position control

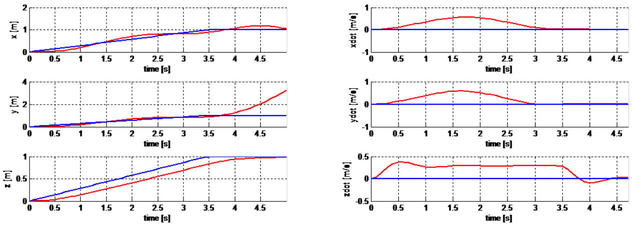


Figure 3.4: PID simulation results in UPenn GRASP labs testbed in both mode 1 (till 3.5 secs and mode 2 afterwards) without target

The d_1 and d_2 are as per the distances of the tool bit position with respect to the CoG of the quadcopter plus manipulator assembly and is a function of ($\phi and \psi$). This force to be added in the modified model can also be found from the reverse dynamics analysis of the manipulated (explained in the next chapter). Also a noise vector is added to simulate the vibrations. The controller was able to keep the quad stable amidst noise of magnitude only up to 5-10 % of the quadcopters net thrust at any point.

3.3.2 Controller Design

The controller loops associated with quadcopter are as in any standard quadcopter system: Simulation for studying the various controllers for these loops is made in Matlab using the University of Pennsylvania GRASP labs quadcopter testbed. Quadcopter flying without any reaction force. Quadcopter flying to the target and applying force on the target (supposedly a wall). To simulate the action of it applying a force on the wall, we include a force component acting on the quadcopter in the y-axis, which I suppose is perpendicular to the wall.

There are 2 PD controllers for mode 1 (simple trajectory following) and mode 2. Different set of PD values should be used for both because the same set of values gives good results for either mode 1 or mode 2.

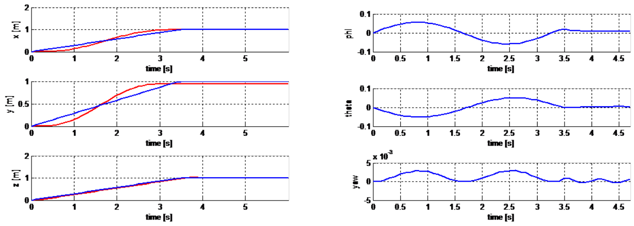
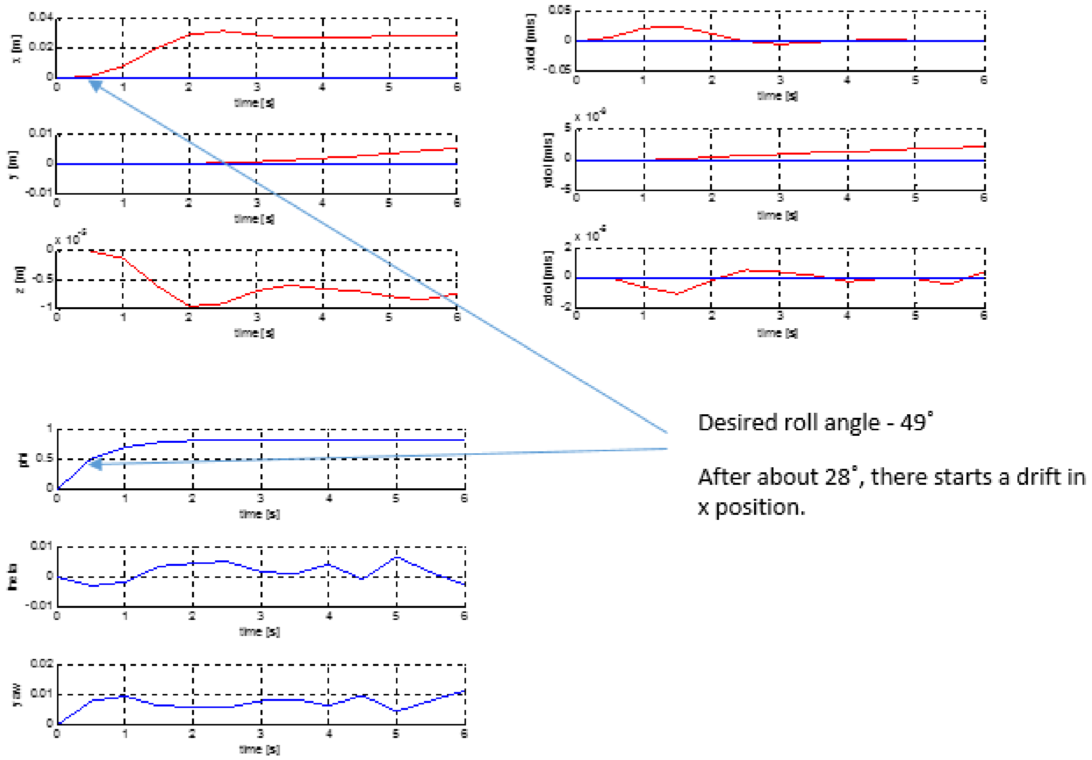


Figure 3.5: PID simulation results in UPenn GRASP labs testbed in both mode 1 (till 3.5 secs and mode 2 afterwards) on the target



As mentioned in [1], the PD controller operates well only in the linear near hover region, after which there is a drift and a steady state error observed in the x position though it is being controlled by the PD controller. On making it the PID controller, the steady state error decreases but the overshoot is still observed. The PID values used for this were based on [2] and [3]. Therefore, as suggested in paper [1], Sliding Mode Control and Backstepping Control is used. SMC is tried only for comparison, but Backstepping is better and works well in both the paper and also in our simulation. Also the need for robust controllers is felt because of the model imperfection due to onboard manipulator movements.

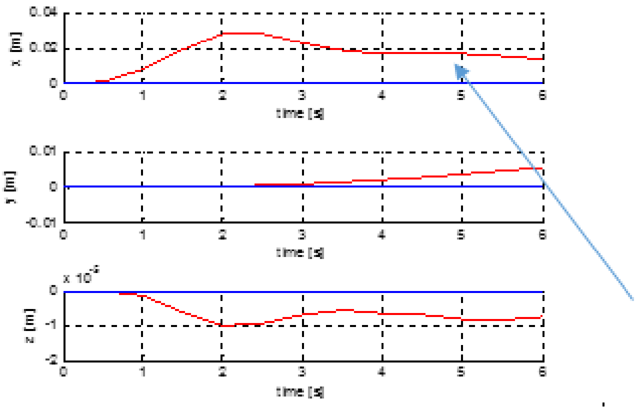


Figure 3.6: PD and PID controller results on tilting the quad.

3.3.3 Backstepping Controller

Backstepping is a recursive control algorithm that works by designing intermediate control laws for some of the state variables. These state variables are called virtual controls for the system. Unlike other control algorithms that tend to linearize nonlinear systems such as the feedback linearization algorithm, backstepping does not work to cancel the nonlinearities in the system. This leads to more flexible designs since some of the nonlinear terms can contribute to the stability of the system. An example of such terms that add to the stability of the system are state variables taking the form of negative terms with odd powers (e.g. x3), they provide damping for large values of x

3.3.4 Sliding Mode Controller

A SMC is a type of Variable Structure Control (VSC). It uses a high speed switching control law to force the state trajectories to follow a specified, user defined surface in the states space and to maintain the state trajectories on this surface. The control law for a SMC consists of two parts as per Equation; a corrective control part and an equivalent control part. The corrective control function is to compensate any variations of the state trajectories from the sliding surface in order to reach it. The equivalent control on the other hand, makes sure the time derivative of the surface is maintained to zero, so that the state trajectories would stay on the sliding surface.

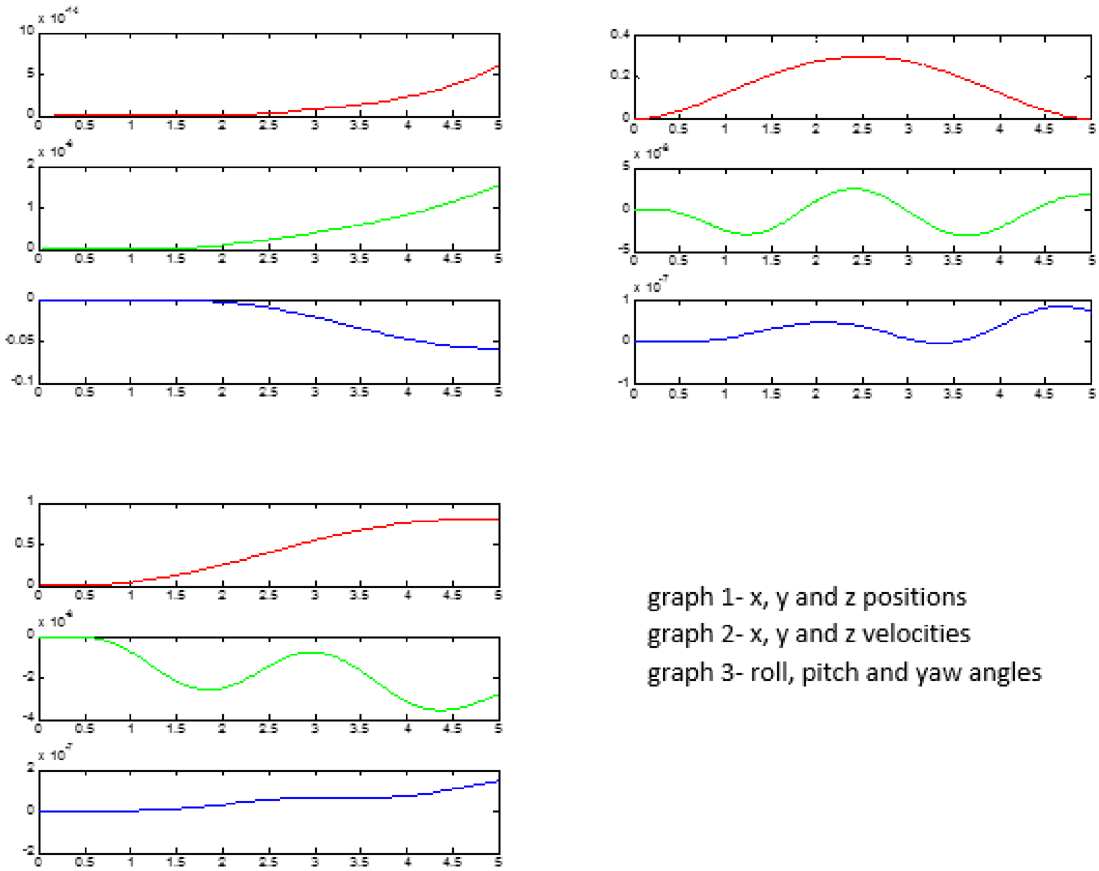


Figure 3.7: Backstepping controller results using on tilting the quad

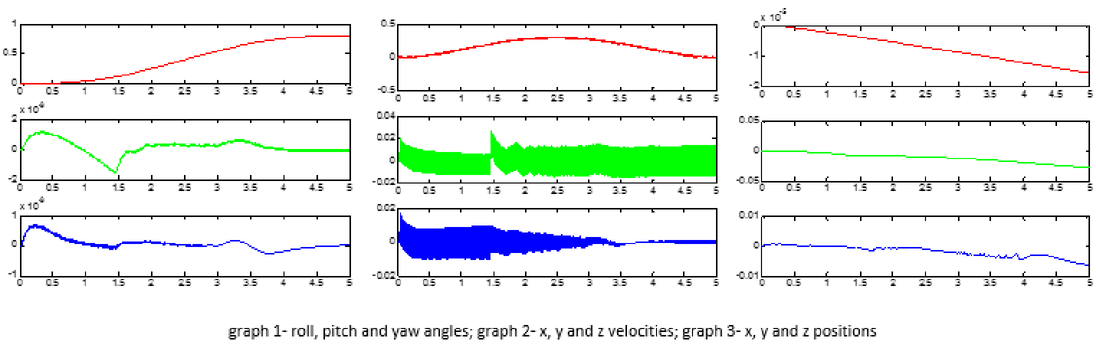


Figure 3.8: 3R spatial manipulator with spherical wrist (PUMA560)

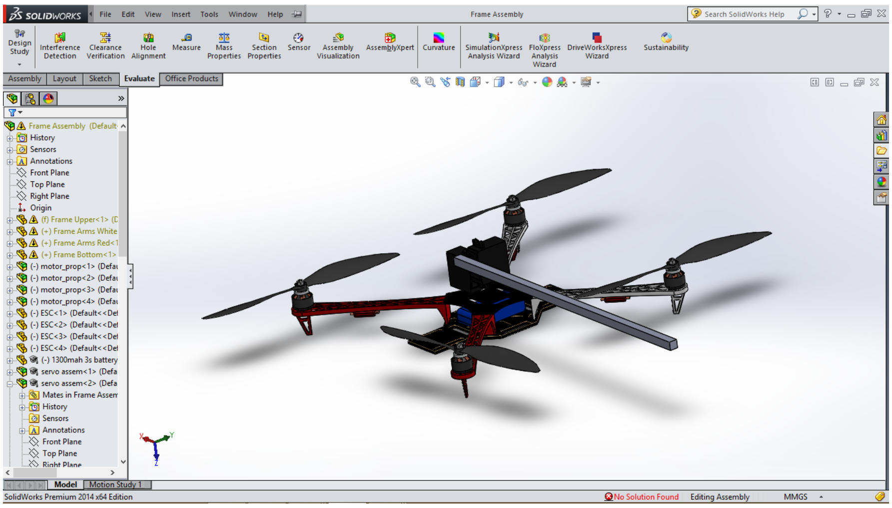


Figure 3.9: Solidworks assembly of the quad with the initial manipulator designed

3.3.5 Controller discussion and comparison

All reach the desired angle of 0.8, the drift in other controlled variables i.e. x, z and yaw over the entire angular range of 48 is least in backstepping control while the noise immunity is maximum in PD controller. Backstepping control seems very sensitive to both disturbances and parameter value changes in even the second place after decimal.

3.3.6 SolidWorks Modelling

The modeling of the quadcopter was done in Solidworks and using the Measure functionality in it, its Inertia Matrices were calculated.

Chapter 4

Manipulator

The serial RRR spatial manipulator is one of the most trivial of all manipulators. For that reason, it is frequently used as a demonstration example in many texts in robot kinematics, e.g. , (Tsai, 1999; Craig, 2003). Through these texts, the majority of necessary work for workspace determination has been presented. Therefore only a brief summary of the required details will be presented here. The RRR serial planar architecture is depicted in Figure 10. It consists of three links and three actuated revolute joints. The analysis of this 3 R spatial manipulator was done using Robotics toolbox in Matlab by Peter Corke.

4.0.1 Manipulator Kinematics

Kinematics is the study of motion without regard to the forces which cause it. Within kinematics one studies the position, velocity and acceleration, and all higher order derivatives of the position variables. The kinematics of manipulators involves the study of the geometric and time based properties of the motion, and in particular how the various links move with respect to one another and with time. Typical robots are serial-link manipulators comprising a set of bodies, called links, in a chain, connected by joints1. Each joint has one degree of freedom, either translational or rotational. For a manipulator with n joints numbered from 1 to n, there are n 1 links, numbered from 0 to n. Link 0 is the base of the manipulator, generally fixed, and link n carries the end-effector. Joint i connects links i and i_1. To facilitate describing the location of each link we affix a coordinate frame to it frame i is attached to link i. Denavit and Hartenberg proposed a matrix method of systematically assigning coordinate systems to each link of an artic-

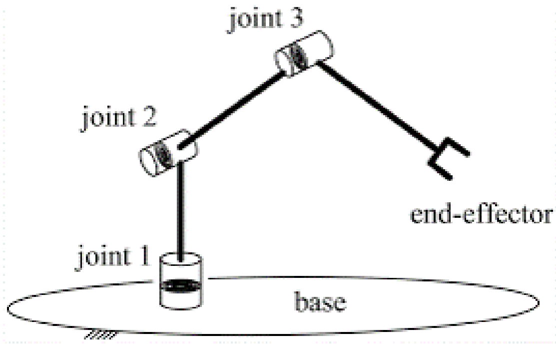


Figure 4.1: It can be clearly seen that inspite of the movement of the base of the manipulator, the manipilator end-effector remains at the same position as desired.

ulated chain. The axis of revolute joint i is aligned with zi 1. The xi 1 axis is directed along the normal from zi 1 to zi and for intersecting axes is parallel to zi 1 zi. The link and joint parameters may be summarized as: link length ai the offset distance between the zi 1 and zi axes along the xi axis; link twist *i* the angle from the zi 1 axis to the zi axis about the xi axis; link offset *di* the distance from the origin of frame i 1 to the xi axis along the zi 1 axis; joint angle *i* the angle between the xi 1 and xi axes about the zi 1 axis. For a revolute axis *i* is the joint variable and di is constant, while for a prismatic joint di is variable, and i is constant. In many of the formulations that follow we use generalized coordinates, *qi*, where *qi* is: *i* for a revolute joint *di* for a prismatic joint The Denavit-Hartenberg (DH) representation results in a 4x4 homogeneous transformation matrix: representing each links coordinate frame with respect to the previous links coordinate system; that is where 0Ti is the homogeneous transformation describing the pose of coordinate frame i with respect to the world coordinate system 0.
Robotica toolbox is an easy to use toolbox compatible with Matlab that can be used to easily specify the different robotic arm configurations, compute their workspace boundaries, the dynamics involved and visualise the trajectories.

Forward and inverse kinematics

For an n-axis rigid-link manipulator, the forward kinematic solution gives the coordinate frame, or pose, of the last link. Of more use in manipulator path planning is the inverse kinematic solution which gives the joint angles required to reach the specified end-effector position.

This forward and inverse kinematics of the manipulator was simulated in Matlab

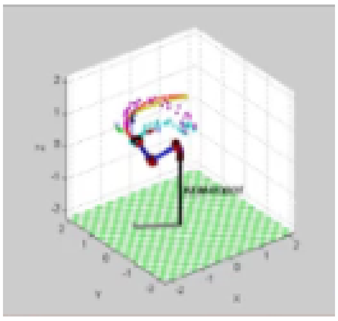


Figure 4.2: 3R manipulator workspace

using the Robotica toolbox functions like fkine, ikine and Links, etc. The code was written for a standard PUMA-560 manipulator, since it is a 6 Axis arm with 3 axis making up a spherical wrist, similar to the 3R spatial manipulator we intended to put on the quadcopter sans the spherical wrist (whose role, if present should be to keep the toolbit perpendicular to the wall/target at all instants). Its workspace volume is doughnut shaped and hence sufficiently large.

4.0.2 Manipulator rigid-body dynamics

Manipulator dynamics is concerned with the equations of motion, the way in which the manipulator moves in response to torques applied by the actuators, or external forces. The equations of motion for an n-axis manipulator are given by

$$Q = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) + Gq \tag{4.1}$$

where q is the vector of generalized joint coordinates describing the pose of the manipulator $\dot{q}$ is the vector of joint velocities; $\dot{\dot{q}}$ is the vector of joint accelerations M is the symmetric joint-space inertia matrix, or manipulator inertia tensor C describes Coriolis and centripetal effects Centripetal torques are proportional to $\dot{q_i}^2$, while the Coriolis torques are proportional to $\dot{q_1 q_j}$ F describes viscous and Coulomb friction and is not generally considered part of the rigid body dynamics G is the gravity loading Q is the vector of generalized forces associated with the generalized coordinates q. This manipulator dynamics was simulated in Matlab using the Robotica toolbox functions like $Serial_Link :: rne$, $Serial_Link :: accel$, etc.

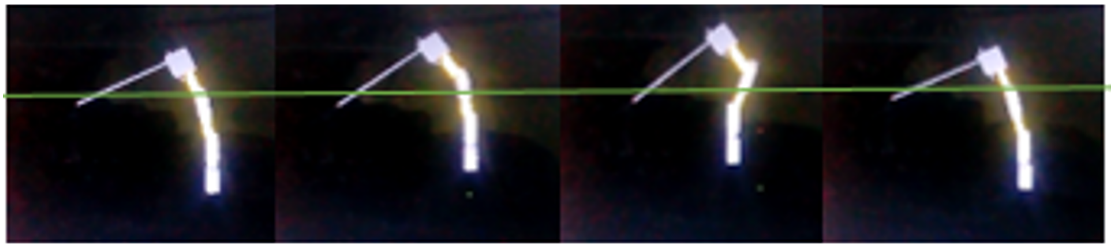


Figure 4.3: SMC controller results on tilting the quad

4.0.3 Simulation Study

After the design of the manipulator and its Denavit-Hartenberg (DH) parameters which depended on many factors such as:

- the level of precision our quadcopter is maintaining
- the amount of payload that can be added on the quadcopter
- the maximum torques the servo motors can hold and subsequently the current we can provide to all the 3 motors being used

The quadcopter was simulated in a handcoded robotics simulator in Processing IDE by importing the Manipulator arm components like Servo motors, motor brackets, toolbit, links etc. made in Solid Works. This simulation helped in visually confirming that the inverse kinematics done for the manipulator was indeed correct for maintaining the end-effector/toolbit position at exactly the same point/target location irrespective of the base (quadcopter) disturbances or position drifts/variations.

4.0.4 Implementation

Thereafter the manipulator was made using 3 servo motors on an easily attachable add-on platform for the quadcopter, balsa wood and mechanics kit pieces. This was then added on top of the quadcopter. One of the novel contributions and challenges of this project is also the fact that the manipulator has been added on top of the quadcopter, analogous to an inverted pendulum problem, and not below it, analogous to a simple pendulum, making it a harder problem but giving the product the added functionality of being able to exert force.

4.1 Setup Integration

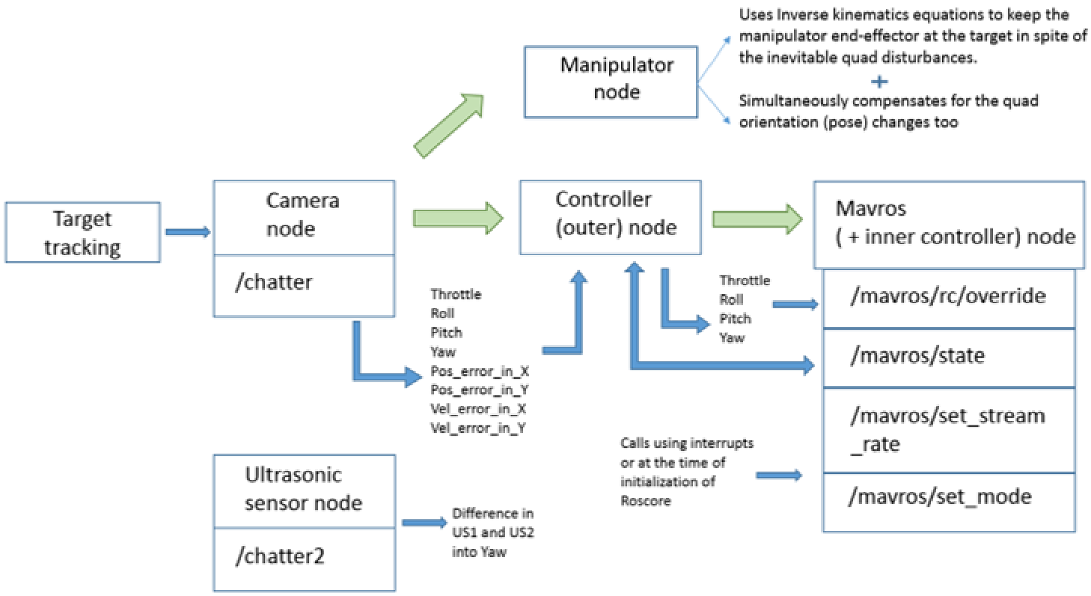
After both the main sections of the project were almost prepared, the final task remained to put them together and coordinate their working so that the quadcopter flies to, stays at and does some work on the target and the manipulator compensates for quadcopter's unwanted movements during this interaction. Since this involved several different modules, we used ROS (Robotics Operating System) for this. The whole system has to stand-alone work on the quadcopter and work autonomous using vision feedback, hence an onboard computer had to be used, here our choice for the same being RaspberryPi 3.

4.1.1 RaspberryPi 3

The Raspberry Pi 3 is the third generation Raspberry Pi. A Raspberry Pi is a general-purpose computer, usually with a Linux operating system, and the ability to run multiple programs. It is more complicated to use than an Arduino and basically can do everything that a full-fledged computer can do. RaspberryPi 3 is the latest edition that comes equipped with 802.11n Wireless LAN. Therefore, it was used so that the external Wifi adapters were not needed.

4.1.2 Robotics Operating System (ROS)

ROS (Robot Operating System) is a BSD-licensed system for controlling robotic components from a PC. A ROS system is comprised of a number of independent nodes, each of which communicates with the other nodes using a publish/subscribe messaging model. For example, a particular sensors driver might be implemented as a node, which publishes sensor data in a stream of messages. These messages could be consumed by any number of other nodes, including filters, loggers, and also higher-level systems such as guidance, pathfinding, etc. ROS do not have to be on the same system (multiple computers) or even of the same architecture! One can have an Arduino publishing messages, a laptop subscribing to them, and an Android phone driving motors. This makes ROS really flexible and adaptable to the needs of the user. ROS is also open source, maintained by many people. The ROS network for this system is as follows:



ROS Nodes:

- Camera node - For the outer control loop that is giving the position feedback, we are using a single Logitech camera. The camera is properly calibrated i.e. its external and internal parameters are known and using the algorithm given in the paper [3], Robust Pose Estimation from a Planar Target (2006) by Gerald Schweighofer and Axel Pinz, we determine the pose of a planar target.

- Ultrasonic sensor node - redundant data, only for Extended Kalman Filtering

- Controller node - The node combining the reference signals being sent by the vision feedback to the Controller dumped on the Ardupilot.

- APM node (MavROS) - This package provides communication driver for various autopilots with MAVLink communication protocol.

- Manipulator node - The 3 motors being controlled directly by the Raspberry Pi GPIO pins.

- EKF node - The Extended kalman Filter ROS node is used to combine the data coming from the different sensors and amidst the sensor, process and measurement noise, provide the best state estimation. Our system matrix was a 12 by 12 matrix for Kalman filtering and standard noise covariances were taken.

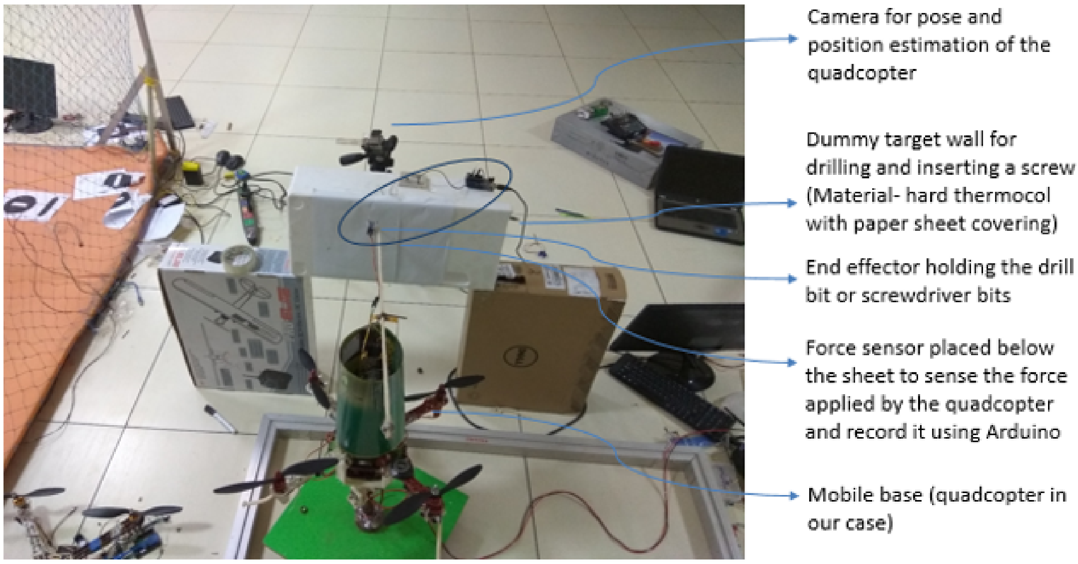


Figure 4.4: Experimental setup in lab

4.1.3 Controller board

For the hardware implementation, we shifted to using the Ardupilot 2.5 board which is nothing but the ATMega 2560 board with IMU (Accelerometer, Gyroscope and Magnetometer) along with external GPU capability integrated. Also this is the last version of the APM series which has open source code available and allows for the controller code to be changed and uploaded. The internal clock setting also allow the constructed ROS network to get data from the IMU for the inner control loop at 100Hz.

Chapter 5

Results and Relations

5.1 Results

The quadcopter tracking algorithm used here, is very robust but in case of very fast, drastic or sudden movement of the camera, the algorithms at times take much longer to converge and compute the translation and rotation quaternion matrices. Also at times the algorithm fails to detect very slow transition between nearly identical frames continuously. Both the above reasons lead to a progressive delay between the control input to be given to the quadcopter. Since the computed control matrix is being sent to the Ardupilot remotely through Wifi, there is another varying delay in the system depending on the Wifi signal strength, which many a times makes the quad response sluggish. But the controller gave good results in tracking the visual cues, especially even while a heavy (250 gms) payload. Therefore, validating the quadcopter has significant power available to exert on the target after attaining a certain height. The different modules i.e. the Outer Controller module, Mavros (or the Inner Control module), the camera module and the Manipulator module are communicating over ROS.

5.1.1 Relations

The quadcopter was made to apply force on a target:

- first at a constant angle of nearly $25 degree$ and varying throttle values i.e. 25, 50 75 and 100 percent of the maximum throttle.
- second at a constant throttle of 50 percent with varying angles of inclination

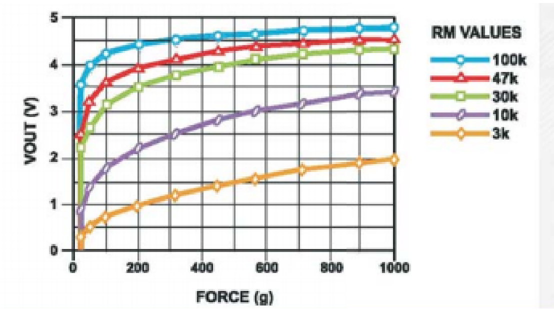


Figure 5.1: Force recorded on the target

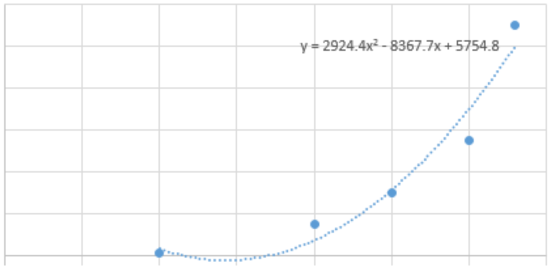


Figure 5.2: Force recorded on the target versus the throttle

towards the target.

The force exerted by the quadcopter was measured by a force sensor placed on the wall behind a protective covering. The analog voltage readings from the sensor were mapped to the force using the following relation available on its datasheet: The results obtained between the Force versus the Throttle were of parabolic shape as below: This analysis was done to make lookup table and validate the already known relations that are: $Fsin(\theta) = f$ and F being proportional to the rpm square therefore throttle square. Therefore, the results prove that the readings in accordance with the theoretical relations. In the second case however, the force experienced was found to vary relatively weakly with the inclination angle. Though theses experiments need to be repeated and better regression analysis needs to be done to find the relations between the force and angle of inclination versus the throttle.

These set of experiments were carried out on a 6 Degree of Freedom stand. It was also found that after around $30 - 35 degree$ of inclination, the quadcopter was unable to sustain itself and hover. Therefore, the maximum inclination that we can provide is around 30 since $Fcos(\theta)$ component has to balance quadcopter's weight.

During the interaction, the precision in the position of the quadcopter was around +-30-

40 cms while the end effector position inaccuracy was reduced from this by about +-10 cms which though not as good as desired but is still satisfactory as a proof of concept of such compensation. The final precision at the target therefore, being around +-20 cms (worst case scenario).

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, it has been shown that we can successfully do somewhat precise force involving jobs using a manipulator put on a mobile base like a quadcopter. We complement the quadcopter abilities with the manipulator abilities to tackle the inherent instability issues arising with underactuated systems like aerial vehicles. Also the add-on manipulator is controlled using the same single camera vision feedback and proprioceptive sensors which are already used for the autonomous navigation of the quadcopter so no new or extra sensors are being used for this greatly useful added functionality.

6.2 Issues and Future Scope

Since ten months or even a year is not enough to make the product being proposed here so robust that it can be taken out into the market to be used by the customers, therefore there is still a lot of scope for improvement and further research in this project.
A lot of hindrances in this project were due to:

- unavailability of high power density batteries
- servo motor jittering
- unsymmetrical quadcopter configuration
- lack of multi-camera equipped lab for localizing the various room objects.

- lack of light weight, high torque micro servo motor and lightweight manipulator links.

So, more efforts need to be put to march past these problems and make this product more robust. Also as per the literature, some controllers like the H-infinity controller, etc. seem to offer better control than PID or Backstepping in situations of unknown disturbances and unmodelled parameters like in our case but due to lack of time and expertise, they could not be tried and tested on the quad. Given that the setup is ready in this project, this research can be easily worked upon in future.

Bibliography

[1] Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches- A Thesis Submitted by Heba talla Mohamed Nabil ElKholy.

[2] S. Mukhopadhyay, PID equivalent of optimal regulator, Electronics Letters, vol. 14, no. 25, pp. 821822, 1978.

[3] Designing a System for Mobile Manipulation from an Unmanned Aerial Vehicle by Christopher M. Korpela, Todd W. Danko and Paul Y. Oh, Drexel Autonomous Systems Lab, Drexel University, Philadelphia, PA

[4] Mobile Manipulating Unmanned Aerial Vehicle (MM-UAV): Towards Aerial Manipulators by Matko Orsag; Journal of Intelligent and Robotics Systems; January 2013, Volume 69, Issue1

[5] Data Driven, Force Based Interaction for Quadrotors, by Christopher D. McKinnon, MS thesis report

[6] Control of a Multirotor Outdoor Aerial Manipulator by G. Heredia, A.E. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J.A. Acosta and A. Oller

[7] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, The GRASP Multiple Micro-UAV Testbed," IEEE Robotics and Automation Magazine, 2010.

[8] A. Pak and P. Paroubek, Twitter as a corpus for sentiment analysis and opinion mining, in Proceedings of the Seventh International Conference on Language Resources and Evaluation, 2010, pp. 13201326

[9] Control of a Multirotor Outdoor Aerial Manipulator by G. Heredia, A.E. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J.A. Acosta and A. Ollero. in

Technologies for Practical Robot Applications (TePRA), 2011 IEEE Conference on April 2011.

[10] "Control of a Multirotor OutdoorAerial Manipulator", 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems,Sept. 2014 by G. Heredia, A.E. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J.A. Acosta and A. Ollero.

[11] Robot Modeling and Control by Mark W. Spong, Seth Hutchinson, and M. Vidyasagar