“ Design, Optimisation, and Control of a UAV with 2 DOF Robotic Arm for Agriculture Harvesting ”

Abstract: Agriculture is one of the main sources of income in India. Sourcing skilled labor for the agricultural sector has become a tedious job in today’s world. Conventional methods of harvesting areca nuts or any other similar fruits point out the need for an automated process. This report describes an Unmanned Aerial Vehicle(UAV) with 2 degrees of freedom (DOF) robotic arm for agriculture harvesting. Here the Unmanned Aerial vehicle is a quadrotor with a robotic arm attached to a pruning device to cut down branches. In this study, the robot is modeled using Solidworks, then a mathematical model of a quadrotor dynamics is derived using Newton’s and Euler laws. A linearized version of the model is obtained, and therefore a linear controller, Proportional Integral Derivative (PID) controller, and also Linear Quadratic Regulator(LQR) is designed in Matlab Simulink.

TABLE OF CONTENTS:

INTRODUCTION:

UAVs for precision agriculture or the forest industry are a major research field and have booming development in recent years by addressing remote sensing challenges such as forest yield and health monitoring of crops. Motivated by the advent of aerial manipulation research, this study proposes pruning of Areca Nut branches with an Unmanned aerial vehicle. This would replace the conventional robots such as tree climbing robots that are labor-intensive and time-consuming. This aerial manipulation design feature is motivated by unique application-related challenges. A 2 Degree of Freedom(DOF) manipulator is longitudinally offset from the aircraft center to avoid rotor wake disturbances to branches and augment end effector reach, facilitating pruning operations from above. Cutting branches are achieved with a rotating cutter that is attached to a prismatic joint which helps in cutting down the branch.

Aerial vehicle dynamics are modeled and characterized using existing methods. An accurate simulation of free-flight scenarios is controlled and this project comes under environment interactive aerial manipulation. Once the nonlinear equations are linearized it is controlled by the PID controller in Simulink or Linear Quadratic Regulator. The 2R planar robotic arm is modeled in Solidworks and it is controlled using a PI controller in Simulink. This aerial manipulation system can be further developed to automate the Areca Nut industry by pruning the branches and harvesting Areca Nut and other similar fruits.

The study of the kinematics and dynamics helps to understand the physics of the quadrotor and its behavior. Together with modeling, the control algorithm is very important to be determined for stability. The whole system can be tested using a Matlab Simulink program with sim mechanics that provides a 3 D graphic output.

LITERATURE REVIEW:

The state of the art in Vertical Take-off and Landing (VTOL) Unmanned Aerial Vehicle (UAV) has seen many developments. There are many commercially available platforms from parrot bebop2 to Draganflyer and DJI matrice series which is quite famous. A UAV can now be able to interact with a rigid structure or an object, these were possible by aerial manipulation and one of the proposed technique was based on Gain Scheduled Proportional Integral Derivative algorithm [1] and this system consist of a 2DOF robotic arm which is attached to a parrot bebop-2 drone. The gain schedule technique has an approach to control nonlinear systems using a set of linear controllers providing adequate control responses to various operational points of the system.

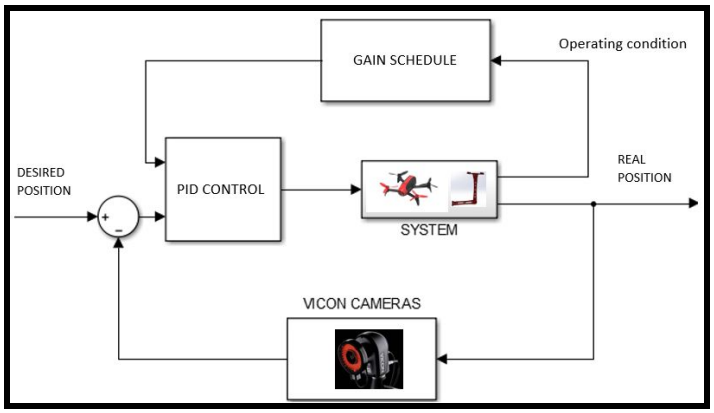


Figure: Parrot Drone with 2dof robotic arm

The field of aerial manipulation can be split into three sub-classifications: load carrying, surface manipulation, and multi-degree of freedom manipulation. They have their unique design and dynamic couplings between aircraft, manipulator, and environment, resulting in different modeling and control approaches. Load-carrying UAV [2] considers the addition of payload mass and the resulting change in COM. The key difference between this and other aerial manipulation categories is that manipulators and environmental interaction dynamics are considered negligible.

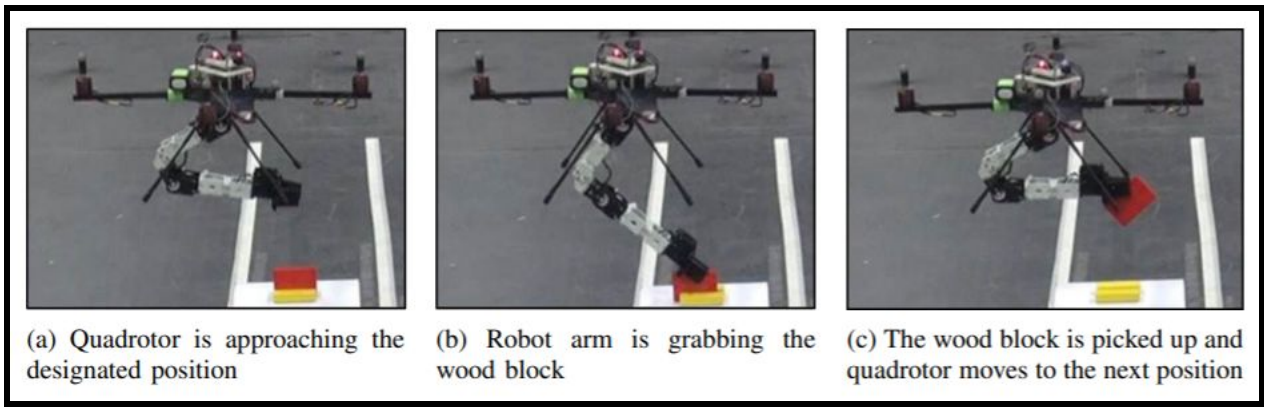


Figure: Quadrotor with 2dof arm using adaptive sliding mode controller

Aerial grasping studies to utilize a fixed gripper, relying on the DOFs of the UAV to track the desired object. One of the advances in aerial grasping with a 2 DOF robotic arm considers a quadrotor and robotic arm as a combined system and an adaptive sliding mode controller [3] is developed after designing the kinematic and dynamics of the system. The quadrotor picks up and delivers the object precisely to its location, another similar system was implemented using a hexacopter[4]. This hexacopter was also controlled using adaptive sliding mode control for both position and velocity. This was fused with a deep neural network for object detection and the three-dimensional coordinates were achieved relative to the quadrotor from the stereo camera.

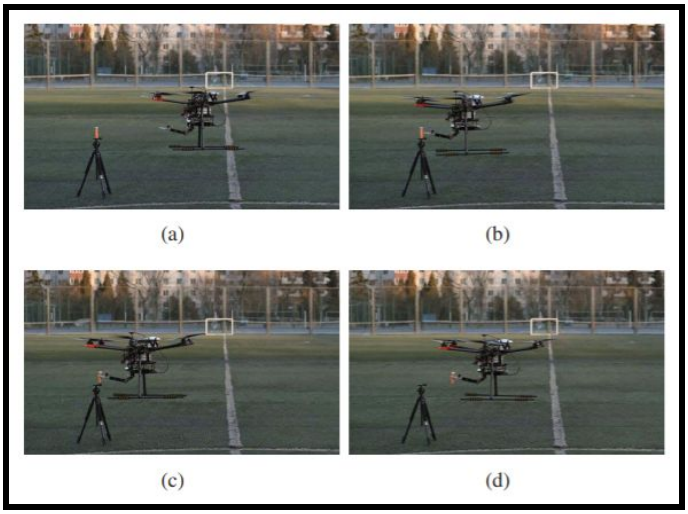


Figure: The process of grasping an object from Hexacopter

A 4 DOF planar dexterous manipulator attached to a Micro aerial vehicle that performs aerial manipulation [5] and also a novel approach combines a visual processing scheme for object detection, tracking, and also for surface interactions with the environment. The manipulator is also known as CARMA (Compact AeRial MAnipulator) and has a stereo camera attached to the end effector. Apart from surface interactions, there is research going into physical interaction with the environment that is to physical Canopy Samplings from a UAV [6] that has a 6 DOF manipulator to cut canopy samples in a forest. It was to develop an aerial manipulator UAV for autonomous canopy sampling.

Figure: Hexacopter with 4dof manipulator

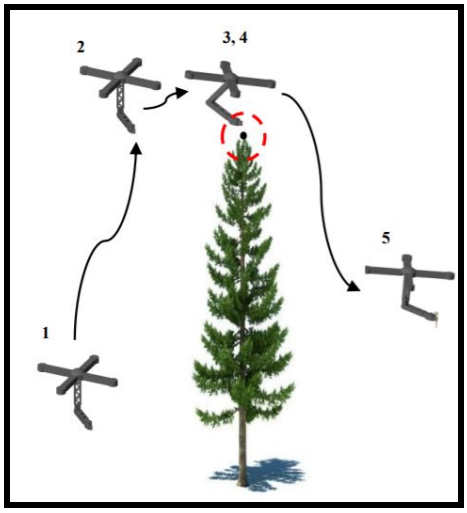


Figure: Conceptual sketch of canopy sampling scenario

Autonomous Unmanned Aerial manipulators are useful in mobile 3D grasping applications and their challenges in indoor positioning and target detection. One of the novels oriented object detection techniques is Rotation squeezedet. A DJI hexacopter frame which is equipped with Nvidia Jetson TX2 and intel real sense camera and a 3DOF robotic arm which grasps is used as an end effector. An approach towards enabling the Unmanned Aerial Manipulators [7] to grasp oriented objects and also detecting and localizing objects in the UAM perspective.

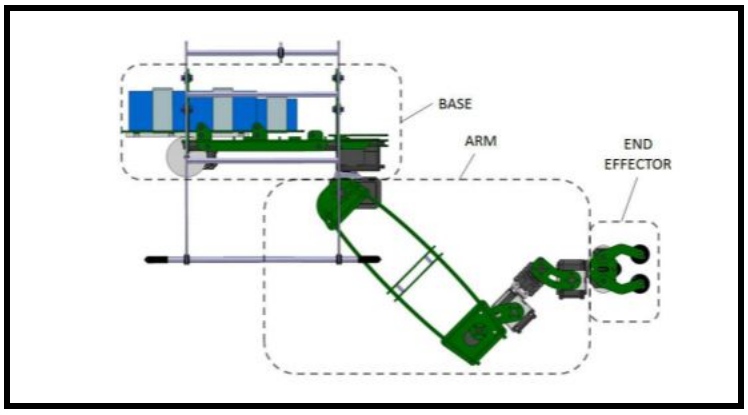


Figure: Manipulator component for a battery compensation system

A multilayer control system [8] was proposed which consisted of basic PID control actions for the quadrotor position and altitude and the three layers are Battery movement-compensation where the displacement compensation system consists of a counterweight(battery) that is to be moved on a linear slide during the manipulator operation to keep the center of gravity in control. The battery movement only achieves gravity compensation and another is the arm static compensation and the last layer consists of the low-level roll-pitch-yaw controller, an angular rate controller, and saturation of forces and torques. This control scheme was proven to be very effective.

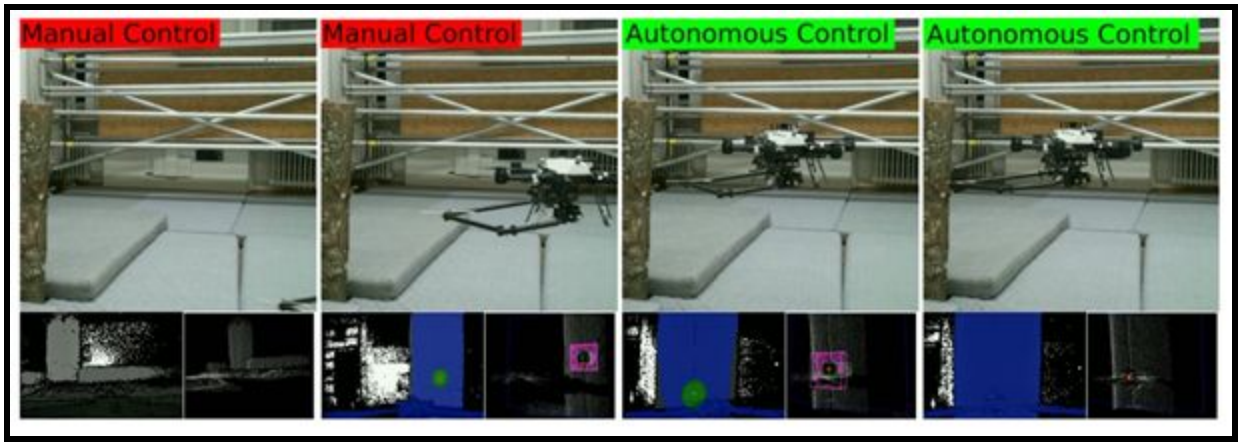


Figure: MAV with a dexterous manipulator

An aerial robotic platform remote tree cavity inspection, based on a hexacopter Micro-Aerial vehicle (MAV) equipped with a dexterous manipulator. This work [9] focuses on two key enabling technologies, namely a vision-based cavity detection system and strategies for high-level control of the MAV and manipulator. This study [10] proposes a developed mechanical system to achieve stable impact absorption without bouncing away from the interacting environment. Their work proposes a new direction towards aerial manipulators that are capable of performing highly dynamic physical interaction tasks.

TIMELINE:

JUNE–

- Literature survey of all the aerial manipulation techniques and tasks in depth.
- Reviewed numerous research papers on the control of aerial manipulators that interact with the environment.

JULY–

- Designed 2 degrees of freedom manipulator.
- Designed a suitable gripper to hold the branch.
- Designed a prismatic joint that is attached to a rotary saw cutter to cut the branch.
- Designed a quadrotor attached to the robotic arm and simulated using Solidworks motion analysis.

AUGUST–

- Aerial Vehicle Dynamics and Mathematical modeling of a Quadrotor.
- Tuned and tested the PID controller for a quadrotor in Simulink using simmechanics.
- Tuned and tested the LQR control of UAV in Simulink and MatLab code.

SEPTEMBER–

- Tuned and tested the PID controller using the vehicle dynamics.
- Inverse Kinematics of 2R planar robotic arm.
- Controlled the robotic arm in Simulink using simmechanics multibody.

AERIAL MANIPULATION:

The aerial manipulator systems are modeled as a multi-rigid-body system. Equations of motion are derived using conventional methods such as Newton-Euler and Euler Lagrange. The quadrotor has four rotors which are directed upwards and are equally placed in a square formation from the center of mass. There are different controllers to control this nonlinear system. Either these nonlinear equations can be linearised by applying controllers like PID/LQR/back-stepping control or directly applying nonlinear H∞ control would stabilize the system. The control methods require accurate information from the position and attitude measurements performed with a gyroscope, an accelerometer, and other measuring devices such as GPS and sonar and laser sensors.

The first aim is to study the mathematical model of the quadrotor dynamics and the proper control methods for stabilization can be executed. The quadrotor has six degrees of freedom but only four control inputs can be given to control the quadrotor. Since Euler-Lagrange formalism is more complex and difficult to implement, in this study Newton-Euler formalism is derived and further control methods are developed.

QUADROTOR DYNAMICS:

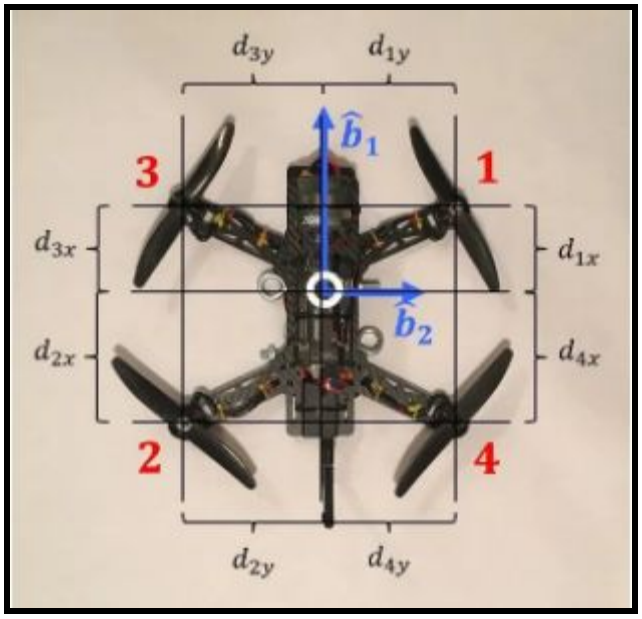
The six degrees of freedom of quadrotor gives the position and the orientation of the aerial vehicle in three dimensions. These dynamic quotations simply represent the change in orientation and position over time. The assumptions are that the force of gravity is constant and the center of mass is equal to the center of gravity and the vehicle is a rigid body.
The quadrotor has 6 degrees of freedom which are described by the two reference frames that are the earth inertial frames and the body-fixed frame. The illustration is given above. The dynamic model is derived using the Newton-Euler formalism.
The variables of equations of motions define the position and orientation of the vehicle in space. Using the right-hand rule and old convention referring X, Y, and Z to North, East, and Down respectively, we can derive the equations of dynamics.

Linear Velocity Variables	
V^b	Linear velocity in body frame
u	Longitudinal velocity
v	Lateral velocity
w	Normal velocity

Rotational velocity variables	
W^b	Rotational velocity in body frame
p	Roll rate
q	Pitch rate
r	Yaw rate
Forces	
F_X	Force in the X direction
F_Y	Force in the Y direction
F_Z	Force in the Z direction
Euler angles	
Φ	Roll angle
θ	Pitch angle
Ψ	Yaw angle
Moments or Torques	
L	A rotational moment along the x-axis
M	A rotational moment along the y-axis
N	A rotational moment along the z-axis

External Forces and Moments:

The thrust produced by the quadrotor acts perpendicular to the vehicle and moment acts at the center of gravity of the vehicle. The following figure shows the distance of propellers from the center of gravity of the vehicle. These distances are later used to calculate the moment around the center of gravity of the vehicle.



The following equations represent the thrust equation of the vehicle

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -F_1 - F_2 - F_3 - F_4 \end{bmatrix}$$

Moments are a simple product of forces and distance from the center of gravity around all three axes of the body frame. The equation shows the moment along the X-axis of the vehicle.

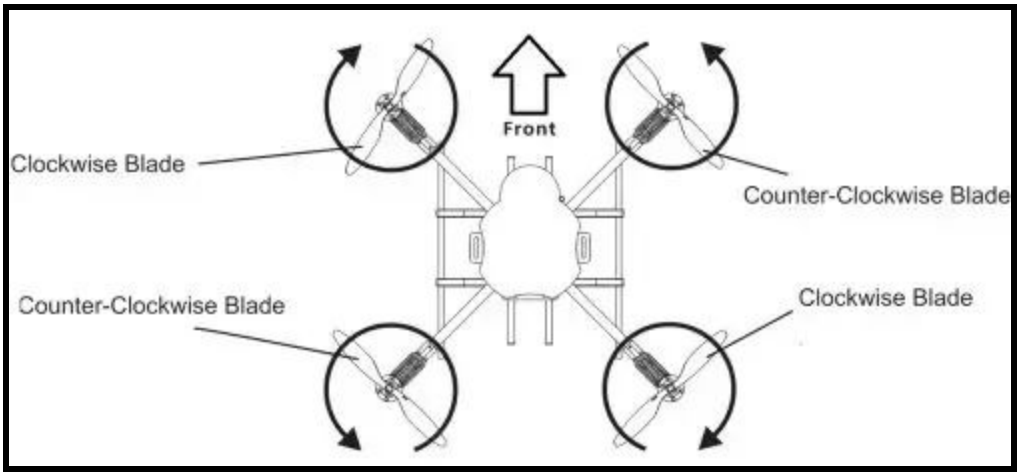
$$L = F_1 d_{1y} - F_2 d_{2y} - F_3 d_{3y} + F_4 d_{4y}$$

The equation shows the moment along the Y-axis of the vehicle.

$$M = -F_1 d_{1x} + F_2 d_{2x} - F_3 d_{3x} + F_4 d_{4x}$$

Yaw is produced in quadcopter by the rotation of the propellers. Two propellers rotate clockwise and the other two rotate counter-clockwise to balance the yaw moment. Equation 14 shows how to calculate the yaw moment of the quadcopter.

$$N = -T(F_1 d_{1x} d_{1y}) + -T(F_2 d_{2x} d_{2y}) + T(F_3 d_{3x} d_{3y}) + T(F_4 d_{4x} d_{4y})$$



The equation of gravity is simplified as

$$F_g^b = \begin{bmatrix} -mg\,sin(\theta) \\ mg\,sin(\phi)\,cos(\theta) \\ mgcos(mg\,cos(\phi)\,cos(\theta)) \end{bmatrix}$$

Moment of inertia:

For an asymmetrical body, the moment of inertia on opposite sides of the vehicle cancel each other. We have assumed our vehicle is symmetrical, So our inertia matrix will be simplified as follows

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

Linear Acceleration and Motion:

Using the Coriolis theorem, we can convert inertial acceleration into a rotating body frame as below.

$$\dot{V}_b = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix}^b + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix}$$

Then by putting forces and acceleration in Newton's second law F=ma we get linear motion

$$\begin{bmatrix} -mg\,sin(\theta) \\ mg\,sin(\phi)\,cos(\theta) \\ -F_1 - F_2 - F_3 - F_4 mgcos(mg\,cos(\phi)\,cos(\theta)) \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix}$$

Rotational Acceleration:

Using the Coriolis theorem as above, here angular velocity is considered instead of linear velocity.

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{q} \end{bmatrix}^b + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{p} I_{xx} \\ \dot{q} I_{yy} \\ \dot{q} I_{zz} \end{bmatrix} + \begin{bmatrix} -I_{yy} qr + I_{zz} qr \\ I_{xx} pr - I_{zz} pr \\ -I_{xx} pq + I_{yy} pq \end{bmatrix}$$

Angular Velocity and Euler angles:

The angular velocities p,q, and r are changing over time and a gyro will give us these values and the Euler angles are Φ, θ, and Ψ. Angular velocity is the rate of change of angles with the body axis, while Euler angles are rotation in their frame of reference. Using the coordinate transformation angular velocity can be represented as an Euler angle derivative.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & s(\phi) \\ 0 & -s(\phi) & c(\phi) \end{bmatrix} \begin{bmatrix} c(\theta) & 0 & -s(\theta) \\ 0 & 1 & 0 \\ s(\theta) & 0 & c(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & s(\phi) \\ 0 & -s(\phi) & c(\phi) \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} -\dot{\psi}\, s(\theta) \\ \dot{\psi}\, c(\theta)\, s(\phi) \\ \dot{\psi}\, c(\phi)\, c(\theta) \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta}\, c(\phi) \\ \dot{\theta}\, s(\phi) \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

$$p = \dot{\phi} - \dot{\psi}\, sin(\theta)$$

$$q = \dot{\theta} cos\phi + \dot{\psi}\, cos(\theta)\, sin(\phi)$$

$$r = \dot{\psi} cos\phi\, cos(\theta) + \dot{\theta} sin(\phi)$$

Inertial Coordinate:

To calculate the position of the vehicle in an inertial frame, we simply use the Euler angle Transformation matrix to convert body frame velocities to an inertial coordinate position.

$$\begin{bmatrix} \dot{x}^E \\ \dot{y}^E \\ -\dot{z}^E \end{bmatrix} = \begin{bmatrix} cos(\theta)\, cos(\Psi) & -cos(\phi)\, sin(\Psi) + cos(\Psi)\, sin(\theta)\, sin(\phi) & sin(\Psi)\, sin(\phi) + cos(\Psi)\, cos(\phi)\, sin(\theta) \\ cos(\theta)\, sin(\Psi) & cos(\Psi)\, cos(\phi) + sin(\theta)\, sin(\phi)\, sin(\Psi) & -sin(\phi)\, cos(\Psi) + cos(\phi)\, sin(\theta)\, sin(\Psi) \\ -sin(\theta) & cos(\theta)\, sin(\phi) & cos(\theta)\, cos(\phi) \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b$$

\

The non-linear equations are derived after applying simple mathematical operations to the above model and then are rewritten as below

These nonlinear equations represent the dynamic behavior of the quadrotor, but still need to represent the equations including motors rpm, propellers thrust, the mass of quadrotor, torque function, and inertial variable values.

$$\dot{u} = -gsin(\theta) + rv - qw$$

$$\dot{v} = -gsin\phi cos(\theta) - ru + pw$$

$$\dot{w} = \frac{1}{m}(-Fz) + gcos\phi cos(\theta) - qu - pv$$

$$\dot{p} = \frac{1}{I_{xx}}\left(L + (I_{yy} - I_{zz})qr\right)$$

$$\dot{q} = \frac{1}{I_{yy}}(M + (I_{zz} - I_{xx})pr)$$

$$\dot{r} = \frac{1}{I_{zz}}\left(N + (I_{xx} - I_{yy})pq\right)$$

$$\dot{\phi} = (p + (qsin\phi - rcos\phi)tan\theta)$$

$$\dot{\theta} = qcos\phi - rsin\phi$$

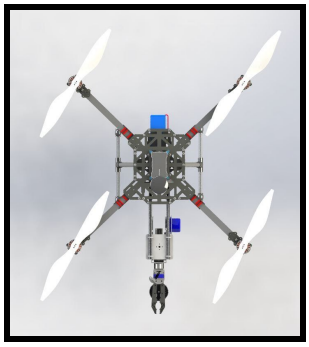
$$\dot{\Psi} = (qsin\phi - rcos\phi)sec\theta$$

$$\dot{x}^F = (qsin\phi - rcos\phi)sec\theta$$

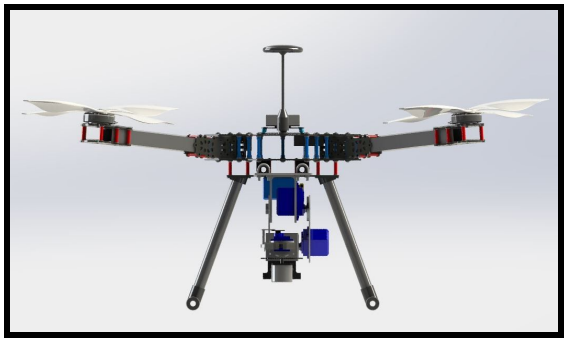
CAD DESIGN:

The parts of the design were modeled in fusion 360 and later assembled in Solidworks. All the weights were considered and the center of mass was in the center by placing the battery at the back. The quadrotor is a Tarot X4 750mm radius and the motors are tarot motors.

The following cad models are when the robotic arm is in rest position and it's near the quadrotor.

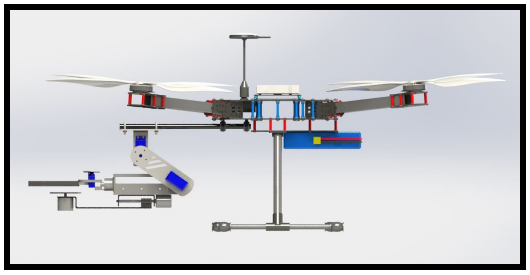


Top view



Front view

Isometric view



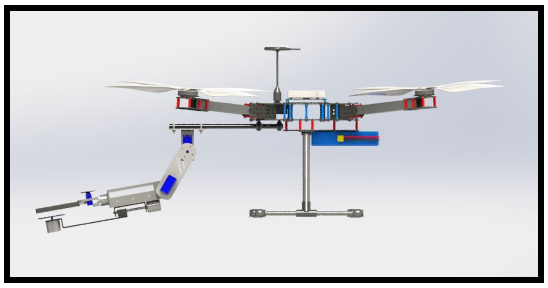
Side view

The below cad models are when the robotic arm has gripped the branch.

Top view

Front view

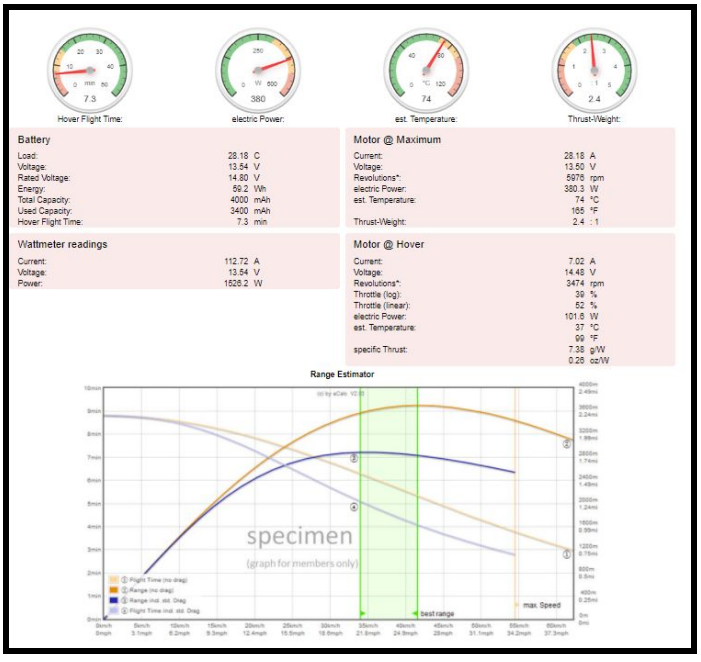
Isometric view



side view

Table of the system

Parameter	Value
Quadrotor mass(m)	4 kg
Arm length(l)	75 cm
Thrust Factor(b)	6.66×10^{-5}
Drag Factor(d)	1.642×10^{-4}
Ixx	0.155 kg-m^2
Iyy	0.183 kg-m^2
Izz	0.066 kg-m^2
Jr	824×10^{-6} kg-m^2



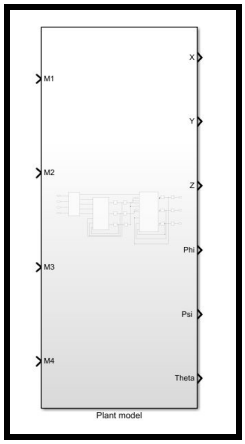
The motion analysis of the cad model has been executed and the simulation is in the drive link.

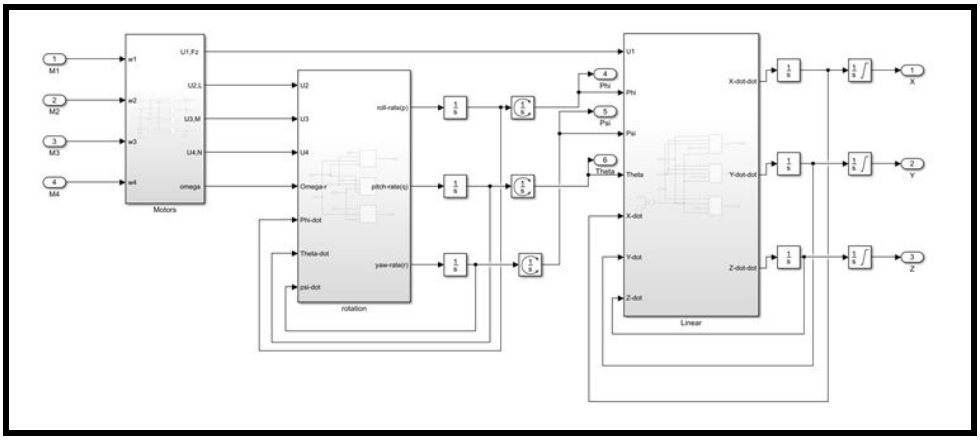
CONTROLLER:

PID Controller -

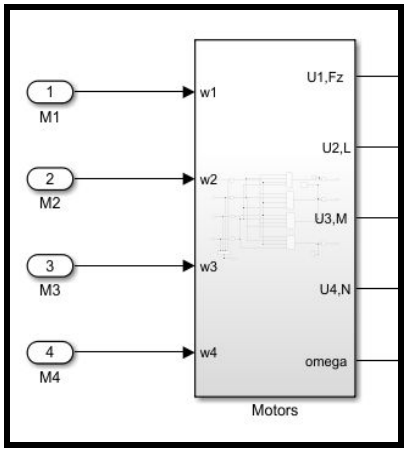
A simple plant model of the vehicle can be built in Simulink depending on the non-linear equations. A subsystem for the plant modeling is created with x, y, h or z, and Ψ as outputs and M1, M2, M3, and M4 as inputs. Different parameters are represented using several subsystems in the plant model subsystem depending on their equations. The aim is to make a model consisting of all the dynamic equations of the vehicle and then build a controller for the model.

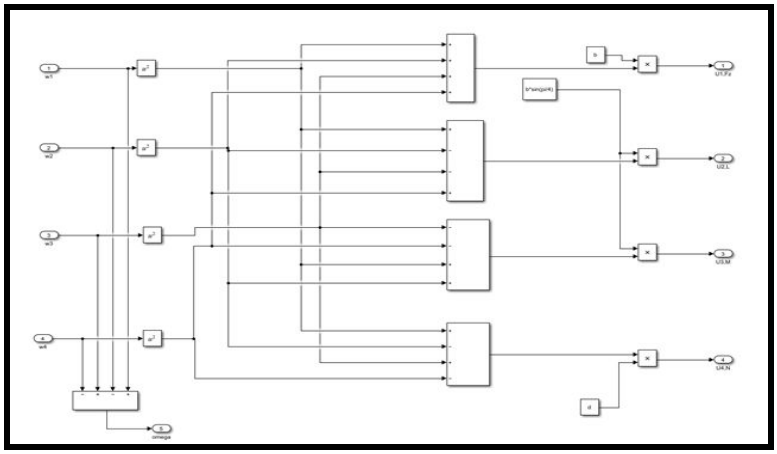
The plant model subsystem consists of several subsystems that are given in the figure below.





The *orientation and motors* are a subsystem under the plant model subsystem that depends on the moment equation for the specific orientation of the vehicle. These equations are for the X orientation of the vehicle. The Orientation and Motors subsystem looks as follows on opening. This subsystem takes the motor commands and converts them to moments to control the orientation and altitude of the vehicle. Omega is the total angular velocity of the quadcopter; the propellers rotate in the opposite direction to balance the total angular velocity.

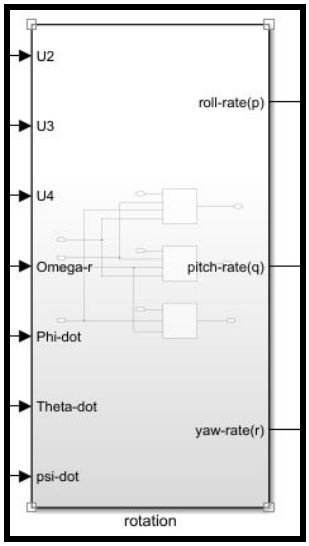


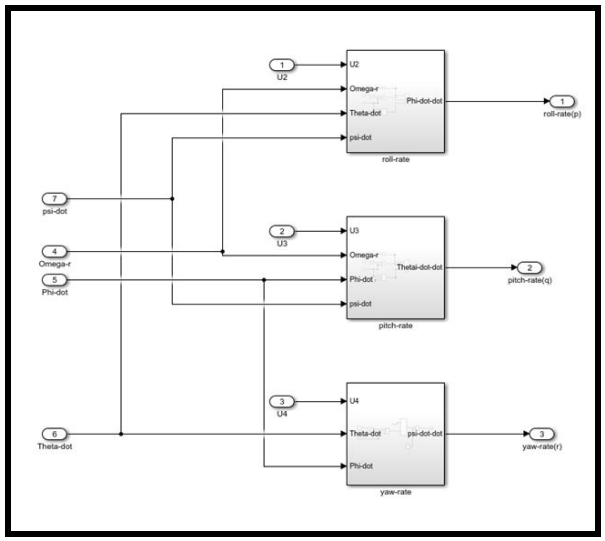


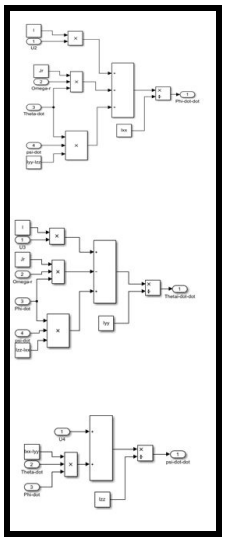
Rotational Dynamics:

This subsystem is based on equations 33, 34, and 35 in which motors thrust is converted into angular rates. It takes U2, U3, U4, and omega from the motors subsystem and outputs p, q, and r. While U1 is fed straight to the next subsystem, as U1 is the total thrust of all motors in the Z direction.

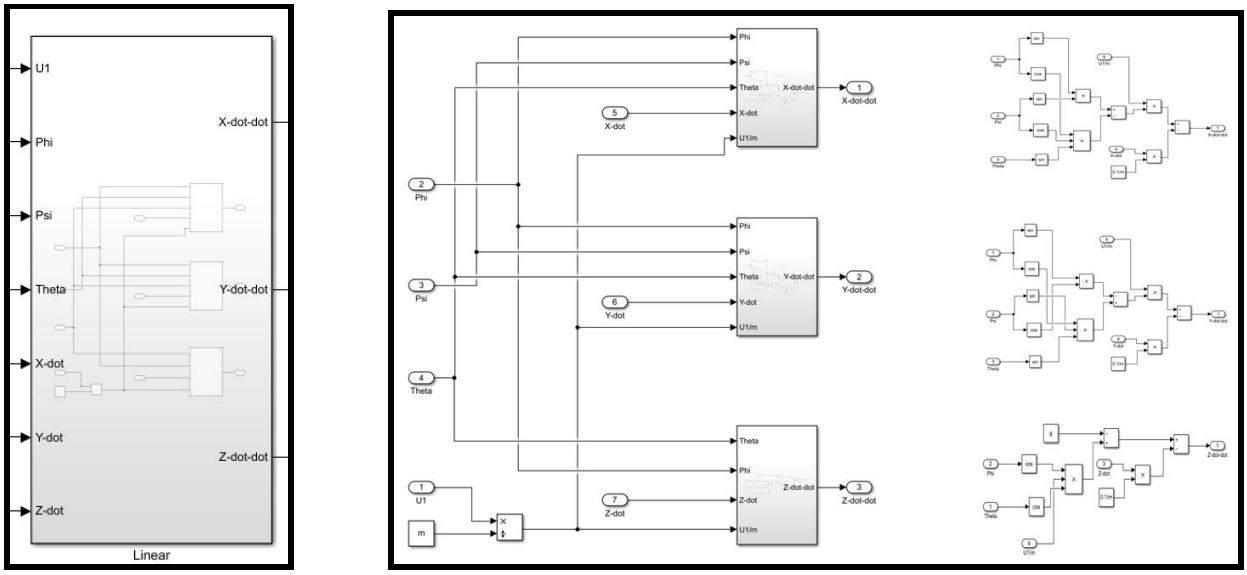
Angular acceleration subsystem has three more subsystems to control roll, pitch, and yaw acceleration separately, containing the equations. Subsystems can be opened to see the equations and their arrangement.



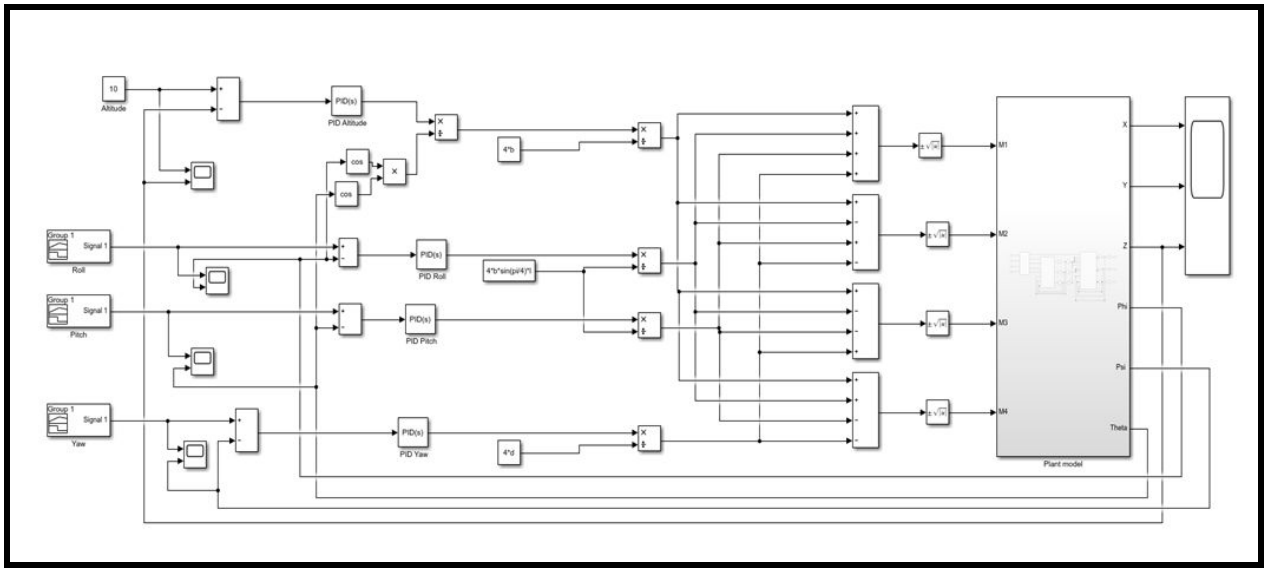




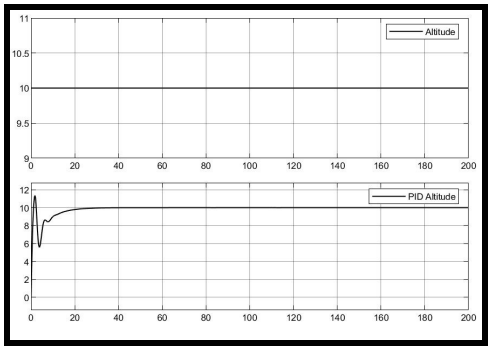
The *Linear Dynamics system* also has three subsystems for x, y, and z acceleration separately, containing equations for each.



There are many control strategies to control a quadrotor, one of which is the Proportional-Integral-Derivative controller. It has three control terms and the error signal that is defined as the difference between the setpoint value and the actual controller output at any particular instance is given as the input signal to the plant. The proportional term increases/decreases the error by multiplying with a proportionality constant. The derivative term is used to estimate the controller's future response based on the error concerning time. The damping factor can be changed by tuning the derivative constant such that it has a smooth response. The integral term sums up all the past values, and the integral coefficient is used for assigning the weight to the integral term to obtain the desired output response in the shortest possible time. Here the attitude control of the quadrotor is implemented and the PID controller is built by using simple gains or the PID block available in Matlab Simulink.



As one can observe from the above figure altitude, pitch, roll, and yaw have PID controllers that are tuned and then those control signals are given to the plant model, which consists of all the dynamic equations. The below results are achieved after tuning the PID controller.



RESULTS:

This plot shows altitude of 10m as setpoint the following results have been achieved. The PID controller for attitude was tuned and had parameters Kp=10 ;Ki=1.5 ;Kd=4.7 ; and similarly for PD for Pitch Kp=0.04 ;Kd=0.6 ; PD for Roll Kp=0.001 ;Kd=0.13 ; PD for Yaw Kp=0.001 ;Kd=0.6 ; .

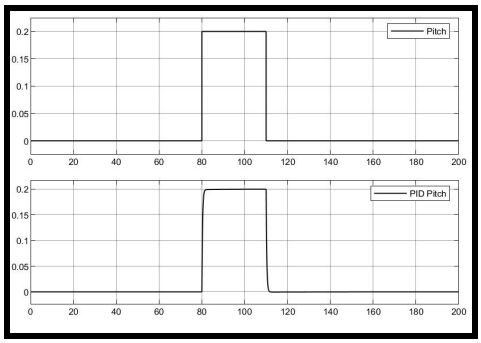


Figure: PD plot for Pitch

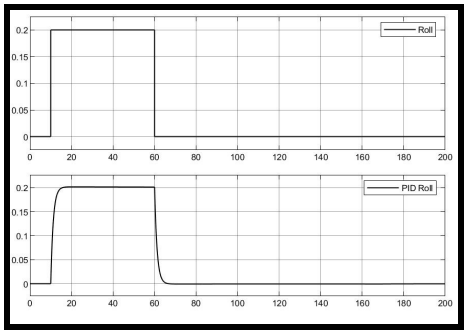


Figure: PD plot for Roll

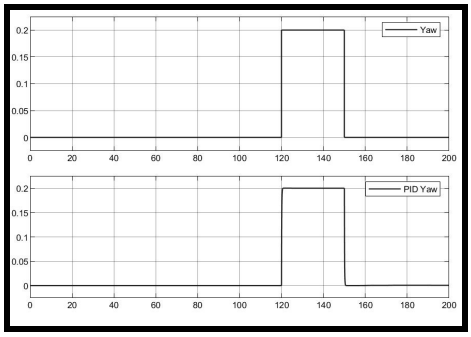


Figure: PD plot for Yaw

LQR controller:

The torque applied on the vehicle's body along an axis is the difference between the torque generated by each propeller on the other axis.

$u2=b*(\omega_4^2-\omega_2^2)$

$u3=b*(\omega_1^2-\omega_3^2)$

$u4=k*(\omega_1^2-\omega_2^2+\omega_3^2-\omega_4^2)$

The rotor inertia is $Jr=Jp-J_mr$

The full dynamic model of quadrotor can be shown in equations below

$$\ddot{x} = (cos\varphi sin\theta cos\psi + sin\varphi sin\psi)\frac{F}{m}$$

$$\ddot{y} = (cos\varphi sin\theta sin\psi - sin\varphi cos\psi)\frac{F}{m}$$

$$\ddot{z} = -g + (cos\varphi cos\theta)\frac{F}{m}$$

$$\dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{Jr}{I_{xx}} q\omega + \frac{u_2}{I_{xx}}$$

$$\dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}} pr - \frac{Jr}{I_{yy}} p\omega + \frac{u_3}{I_{yy}}$$

$$\dot{r} = \frac{I_{xx} - I_{yy}}{I_{zz}} pq + \frac{u_4}{I_{zz}}$$

Where Jr is the rotor inertia and p,q, and r represents the angular velocities concerning roll, pitch, and yaw motions respectively. The LQR optimization problem requires a linear state-space system as the A and B matrices along with the Q and R optimal control matrices are necessary for computing the full state feedback K matrix. The state-space modeling of the quadrotor state-space system consists of 12 state variables, four input variables, and four output variables. The state variables represent the absolute quadrotor orientation in space consisting of linear and rotational coordinates and their respective velocities. The input variables consist of the four quadrotor motions, namely thrust, roll, pitch, and yaw motion. The output consists of the required state variables for stability analysis of the quadrotor, which are the vertical displacement (z), roll (φ), pitch (θ), and yaw (ψ) angle displacements.

$$\dot{X} = AX + BU$$

$$Y = CX + DU$$

$$X^T = [x \quad y \quad z \quad \varphi \quad \theta \quad \psi \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad p \quad q \quad r]$$

$$U^T = [u_1 \quad u_2 \quad u_3 \quad u_4]$$

$$Y^T = [z \quad \varphi \quad \theta \quad \psi]$$

The LQR optimization problem requires a linearized state-space model of the system. The cost function which needs to be optimized is given by

$$J = \int (X^T QX + U^T RU)dt$$

After tuning the matrices according to the desired output matrix, the Q and R matrices are used to solve the Algebraic Riccati Equation (ARE) to compute the full state feedback matrix, which is essentially the LQR controller.

$$A^T S + SA - SBR^{-1}B^T S + Q = 0$$

The S matrix obtained from the ARE is used to calculate the Full state feedback gain matrix K using the equation

$$K = R^{-1}B^T S$$

The final feedback control relation is then calculated as

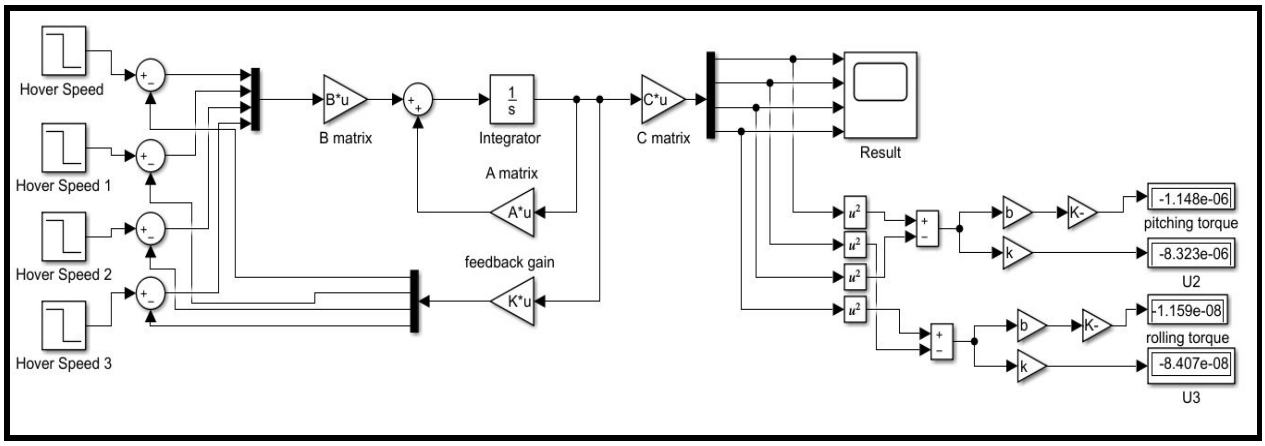
$$U = -K * X$$

LQR design of quadrotor for hovering state.
This controller design is for hovering state of quadrotor so F=Fg, omega=sqrt((mg)/16b), where m is the mass of the quadrotor and b is the thrust coefficient. The Q and R matrix are tuned accordingly to get a stabilized output. The control signal and torque are obtained using quadrotor coefficients. The MatLab code for the following is given below

```
b=6.66e-5;   %thrust coefficient
k=1.642e-4; %drag coefficient
m = 4;          % mass of quadcopter
Ixx =  155237782e-9 ;
Iyy =  183039510e-9 ;
Izz =   066119823e-9 ;
A = [ 0 0 0 1 0 0 0  0 0 0 0 0;
      0 0 0 0 1 0 0  0 0 0 0 0;
      0 0 0 0 0 1 0  0 0 0 0 0;
      0 0 0 0 0 0 0  2 0 0 0 0;
      0 0 0 0 0 0 0 -2 0 0 0 0 0;
      0 0 0 0 0 0 0  0 0 0 0 0;
      0 0 0 0 0 0 0  0 0 1 0 0;
      0 0 0 0 0 0 0  0 0 0 1 0;
      0 0 0 0 0 0 0  0 0 0 0 1;
      0 0 0 0 0 0 0  0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0];
B = [ 0 0 0 0;
      0 0 0 0;
      0 0 0 0;
      0 0 0 0;
      0 0 0 0;
     1/4 0 0 0;
```

```
        0 0 0 0;
        0 0 0 0;
        0 0 0 0;
        0 1/Ixx 0 0;
        0 0 1/Iyy 0;
        0 0 0 1/Izz];
C =  [1 0 0 0 0 0 0 0 0 0 0 0;
        0 1 0 0 0 0 0 0 0 0 0 0;
        0 0 1 0 0 0 0 0 0 0 0 0;
        0 0 0 1 0 0 0 0 0 0 0 0];
D =  [0 0 0 0;
        0 0 0 0;
        0 0 0 0;
        0 0 0 0];
%X_dot = A*X +B*u;
Q = diag([500000 500000 5000 500000 1 1 1 1 1 1 1 1]);
R = diag([0.01 0.01 0.01 0.01]);
sys = ss(A,B,C,D);
control=ctrb(sys);
check = rank(control);
K = lqr(A,B,Q,R);
eig(A-B*K);
```

And the following is the Simulink representation



RESULTS:

The above model shows the LQR design in Simulink using the state-space model of the hover quadrotor. The hover speed is constant at 260 and was calculated to hover at that speed of the propeller. The A, B, C matrices were as given above in the Matlab code and after getting the angular velocities of each propeller, it was passed to a square function, and then pitching torque and rolling torque along with the U_2 and U_3 control signals are known. The scope of the result was achieved by tuning the Q and R terms to get an optimal controller to control the UAV.

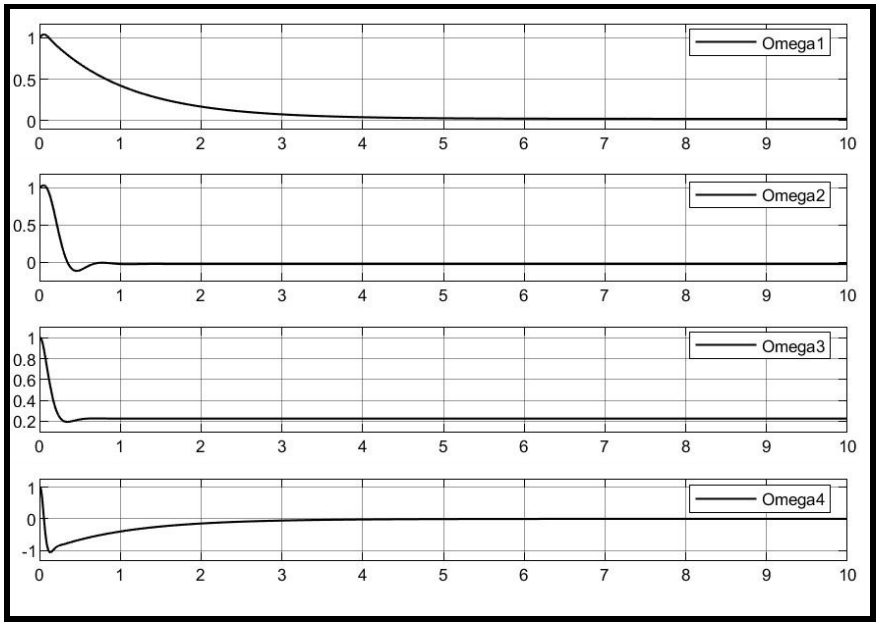


Figure: Plot of various Omega values

ROBOTIC ARM:

The 2 DOF robotic arm has 2 revolute joints which are planar and also a prismatic joint that has an end effector which is a rotary cutter. This rotary cutter will cut the branch once it grips the branch. The robotic manipulator is attached with an offset of 230mm to the center of the quadrotor as there will be less disturbance caused by the wind. Since this has a gripper attached at the end of the 2R planar arm, the inverse kinematics of a 2dof manipulator can be applied and then a PI controller to control the robotic arm at that position.

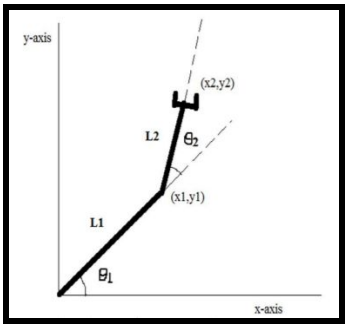


Figure: A 2dof manipulator

The inverse kinematics means to find the joint angles for the particular position (X_2, Y_2) and is given by:

$$cos\theta_2 = \frac{X_2^2+Y_2^2-L_1^2-L_2^2}{2L_1L_2}$$

$$tan\theta_1 = \frac{Y_2(L_2cos\theta_2+L_1)-X_2L_2sin\theta_2}{X_2(L_2cos\theta_2+L_1)+Y_2L_2sin\theta_2}$$

This is then modeled in Simulink and applied to the robotic arm to get the end-effector position in Matlab.

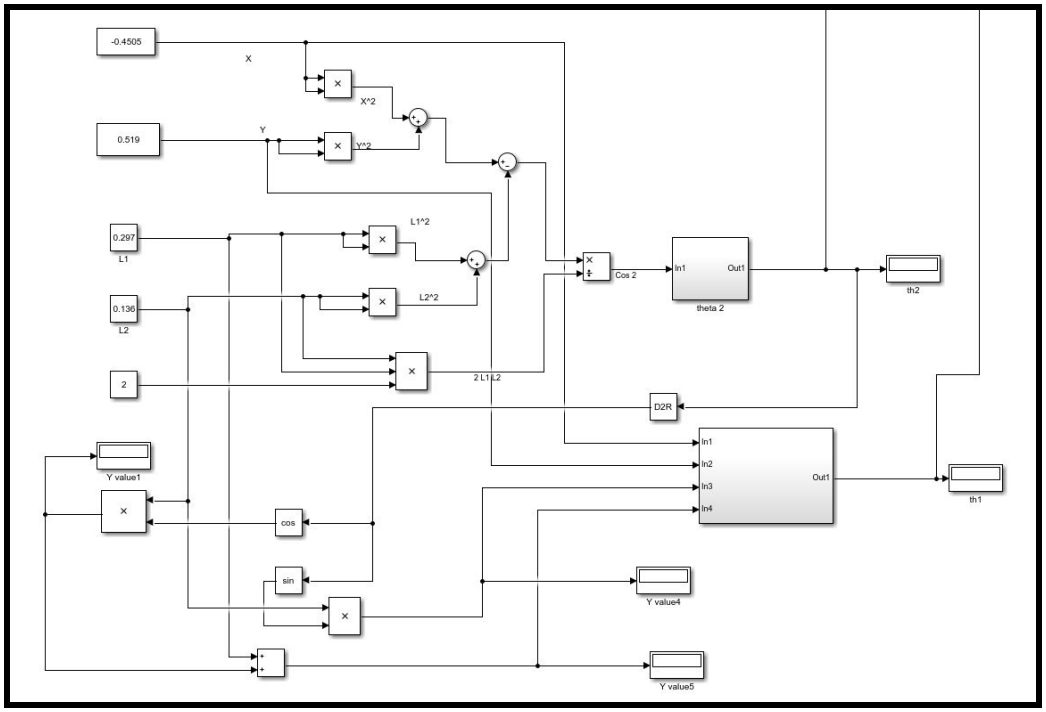


Figure: the positions for the end effector is given and inverse kinematics is modeled

Then the robotic arm dynamics is imported from Solidworks using simmechanics and a PI controller for each joint is given in the figure below

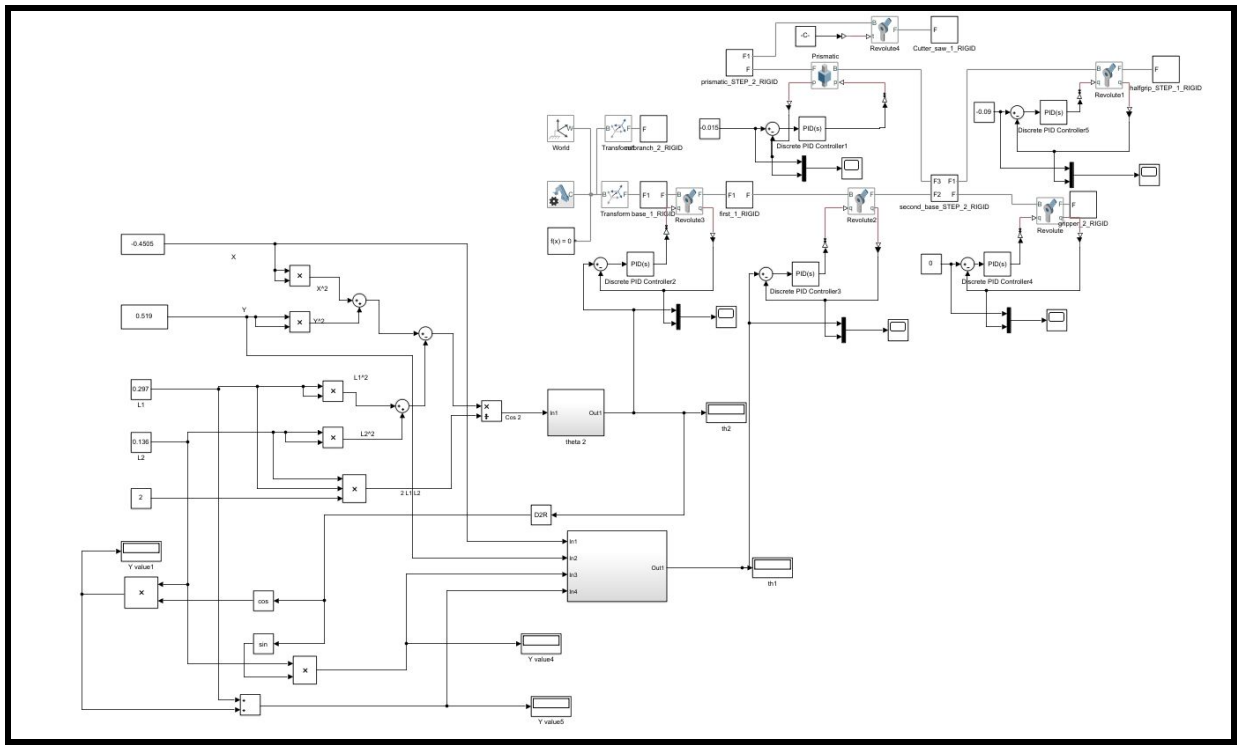


Figure: The Simulink model of the dynamics of the robotic arm with the controller

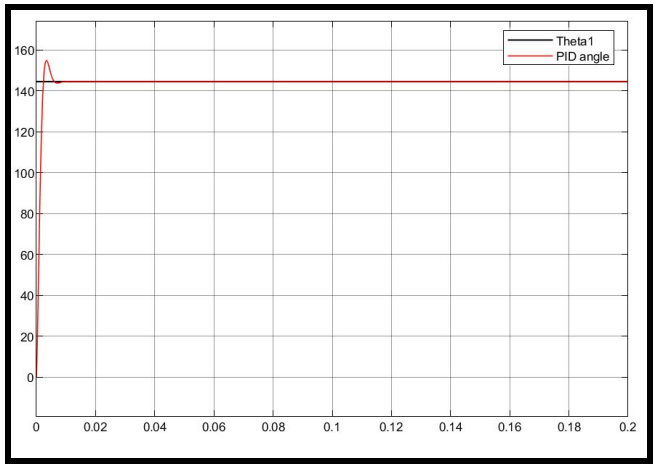


Figure: PI plot of robotic arm angle1 (Kp=1.5; Ki=1091)

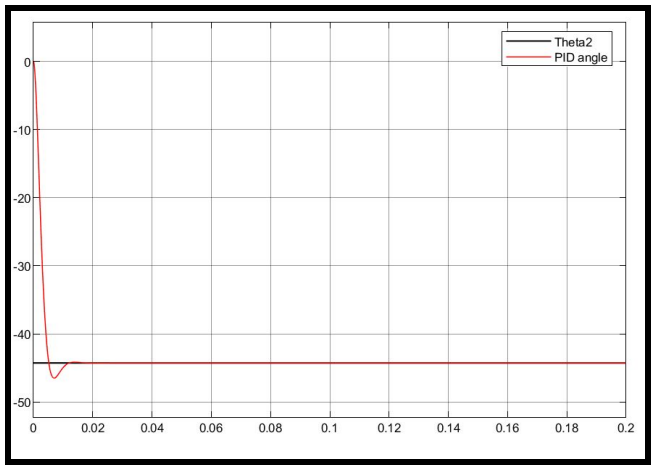


Figure: PI plot of robotic arm angle2 (Kp=0.37;Ki=447.2)

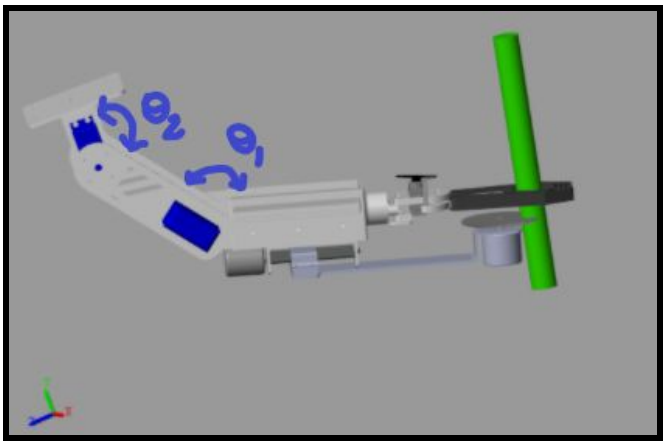


Figure: Simulation of the robotic arm

CONCLUSION & FUTURE SCOPE:

In this study, Unmanned Aerial Manipulation has been thoroughly researched and a 2 DOF robotic arm was modeled with a prismatic joint that is attached to the rotary cutter mainly for agriculture harvesting. This robotic arm can be used to cut branches of areca nut trees to harvest areca nuts while the quadrotor is in a stable position. The quadrotor can be controlled using PID/LQR and the robotic arm is PI controlled. The design of the robotic arm was done in Solidworks and fusion360 and various controllers like PID/LQR of quadrotors have been studied for these parameters, which will be useful for future research in the area of agricultural harvesting using aerial vehicles. The robotic arm is attached at an offset from the quadrotor which will be helpful when cutting the branch so that there are minimal disturbances due to the wind. Autonomous aerial vehicles with manipulators can help farmers to harvest vegetables and fruits which are at a high altitude and where sourcing labor is difficult.

ACKNOWLEDGEMENT:

I would like to thank Dr. Abhra Roy Chowdhury from Robotics Innovations lab IISc for constant guidance and support during the work.

REFERENCES:

[1] Luna, Aaron & Vega, Israel & Martinez-Carranza, Jose. (2018). Gain-Scheduling and PID Control for an Autonomous Aerial Vehicle with a Robotic Arm. 1-6. 10.1109/CCRA.2018.8588145.

[2] I. H. B. Pizetta, A. S. Brandão and M. Sarcinelli-Filho, "Control and Obstacle Avoidance for an UAV Carrying a Load in Forestal Environments," 2018 International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX, 2018, pp. 62-67, doi: 10.1109/ICUAS.2018.8453399.

[3] S. Kim, S. Choi and H. J. Kim, "Aerial manipulation using a quadrotor with a two DOF robotic arm," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, 2013, pp. 4990-4995, doi: 10.1109/IROS.2013.6697077.

[4] R. Jiao, M. Dong, W. Chou, H. Yu and H. Yu, "Autonomous Aerial Manipulation Using a Hexacopter Equipped with a Robotic Arm," 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 2018, pp. 1502-1507, doi: 10.1109/ROBIO.2018.8664845.

[5] C. Kanellakis, M. Terreran, D. Kominiak and G. Nikolakopoulos, "On vision enabled aerial manipulation for multirotors," 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, 2017, pp. 1-7, doi: 10.1109/ETFA.2017.8247653.

[6] J. R. Kutia, K. A. Stol and W. Xu, "Canopy sampling using an aerial manipulator: A preliminary study," 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, 2015, pp. 477-484, doi: 10.1109/ICUAS.2015.7152326.

[7] Lin, S.; Wang, J.; Peng, R.; Yang, W. Development of an Autonomous Unmanned Aerial Manipulator Based on a Real-Time Oriented-Object Detection Method. *Sensors* 2019, *19*, 2396.

[8] F. Ruggiero et al., "A multilayer control for multirotor UAVs equipped with a servo robot arm," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 2015, pp. 4014-4020, doi: 10.1109/ICRA.2015.7139760.

[9] K. Steich, M. Kamel, P. Beardsley, M. K. Obrist, R. Siegwart and T. Lachat, "Tree cavity inspection using aerial robots," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, 2016, pp. 4856-4862, doi: 10.1109/IROS.2016.7759713.

[10] T. Bartelds, A. Capra, S. Hamaza, S. Stramigioli and M. Fumagalli, "Compliant Aerial Manipulators: Toward a New Generation of Aerial Robotic Workers," in IEEE Robotics and Automation Letters, vol. 1, no. 1, pp. 477-483, Jan. 2016, doi: 10.1109/LRA.2016.2519948.