Project Report: Design and fabrication of a test-bed to simulate vision-based autonomous landing of a UAV on a ship deck

Ritwik Shankar*, Abhishek†

I. Nomenclature

R	=	Radius of the stewart platform top plate
H	=	Radius of the stewart platform bottom plate
L	=	Length of the revolute joint
Z	=	Heave value considered from ground point
l_a	=	Deflection of the aft actuator from ground point
l_f	=	Deflection of the forward actuator from ground point
l_l	=	Deflection of the lateral actuator from ground point
θ	=	Pitch deflection of the platform along y axis
ϕ	=	Roll deflection of the platform along x axis
a_x	=	Raw acceleration data along x axis from MPU6050
a_y	=	Raw acceleration data along y axis from MPU6050
a_z	=	Raw acceleration data along z axis from MPU6050
θ^{accel}	=	Pitch deflection from acceleration data
ϕ^{accel}	=	Roll deflection from acceleration data
w_x	=	Angular velocity along x axis
w_y	=	Angular velocity along y axis
w_z	=	Angular velocity along z axis
α	=	Complimentary filter value

II. Introduction

Landing a rotorcraft on the deck of a moving ship represents a significant challenge due to the unique and demanding conditions of the maritime environment. This challenge is increasingly important as maritime operations expand the use of unmanned aerial systems (UAS) for essential tasks such as surveillance, supply delivery, reconnaissance, and search-and-rescue missions. A safe and precise landing on a ship deck is essential to complete these missions effectively, especially in situations where immediate or frequent landings and takeoffs are necessary. However, ocean swells, unpredictable winds, and limited deck space create an inherently unstable and dynamic landing platform.

The movement of the ship—pitching, rolling, and heaving in response to sea conditions—compounds the complexity of achieving a successful landing. In human-piloted operations, the pilot must time their descent to match the ship's motion closely, an approach that relies on extensive experience, precise judgment, and rapid adaptability. The margin for error is extremely small, as the confined space on the deck allows little room for adjusting in case of misalignment or turbulence.

For autonomous systems, the challenge is even greater, as they lack the immediate intuition and adaptability of human pilots. To safely and efficiently achieve ship deck landings, UAS must be equipped with advanced control systems capable of anticipating and adjusting to the ship's movements in real-time. This requires not only sophisticated control algorithms but also robust perception systems that provide accurate, up-to-date information on the ship's position and orientation. Autonomous, vision-based solutions offer promising potential, enabling precise estimation of deck motion and real-time adjustments to improve landing accuracy, even in adverse conditions.

Developing test bed systems capable of accurately simulating the complex motions of a ship deck—such as pitching, rolling, heaving, and swaying—often involves significant costs and technical challenges. Commercial solutions for these motion simulation platforms can easily exceed 4 million INR (40 Lakhs), making them prohibitively expensive for many

*Under-Graduate Student, Aerospace Engineering, Indian Institute of Technology Kanpur, Uttar Pradesh, India, 208016
†Professor, Aerospace Engineering, Indian Institute of Technology Kanpur, Uttar Pradesh, India, 208016

research and development teams. Consequently, there is a strong need for a more affordable, yet reliable, system that can reproduce these dynamic motions without compromising accuracy or performance.

Creating a cost-effective and simplified test platform would allow for broader accessibility, enabling more research institutions, organizations, and industries to test and refine UAS landing algorithms under realistic conditions. An affordable design could stimulate further innovations in autonomous landing systems by providing a practical setup for iterative testing, data collection, and improvement of control algorithms. Additionally, a streamlined, user-friendly setup would be easier to operate and maintain, reducing the technical expertise and time required for routine tests and adjustments.

A. Problem Statement

The continuous pitching, rolling, and heaving of a ship's deck, driven by ocean conditions and environmental factors, introduces significant instability and unpredictability. These movements reduce the margin for error, requiring any landing approach to be highly precise. The turbulent airflow around the ship, coupled with deck movement, creates further complications for maintaining stable flight. The problem is particularly complex for autonomous systems, which must calculate descent, adjust for deck motion, and compensate for instability in real time. To address this challenge, our approach utilizes vision-based platform estimation and an algorithm designed for precision landing on a dynamic surface. Testing of the algorithm leverages a custom-built Stewart platform capable of simulating the ship deck's 3-DOF motion, including roll, pitch, and heave, to replicate real-world conditions for controlled trials.

B. Literature Review

The autonomous landing of Unmanned Aerial Vehicles (UAV), capable of Vertically Takeoff and Landing (VTOL), is a widely researched topic with many control algorithms using different types of sensors such as IR-LEDs, LIDARs, RTK-GPS, and vision.

In vision-based methods for autonomous landing, the research started with identifying static geometrical patterns like T, H, circular, rectangular or a combination of them like in fiducial markers. In [1],"T" shape was detected by using adaptive threshold, whereas [2, 3] detected "H" shape. Using concentric circles, pose estimation is done by measuring the distances between the circle in [4, 5]. Fiducial markers provide more robust and accurate pose estimation along with high detection speed and hence are widely used for autonomous landing algorithms. ARTag and ARToolkit Plus was studied in [6]. [7, 8] explored the landing task by using AprilTag and AprilTag 2 was used by [9].

The dynamics of a Stewart platform were explored through the examination of two sources ([10] & [11]). These sources were consulted to gain insight into the mechanics and functioning of the Stewart platform

III. Ship Deck Simulation Platform

A. Introduction to Stewart Platform

A *Gough-Stewart* platform was planned to replicate the scone ocean dataset. The anticipated versatility of the Stewart platform for stabilizing ships at sea is illustrated in Fig. 1.

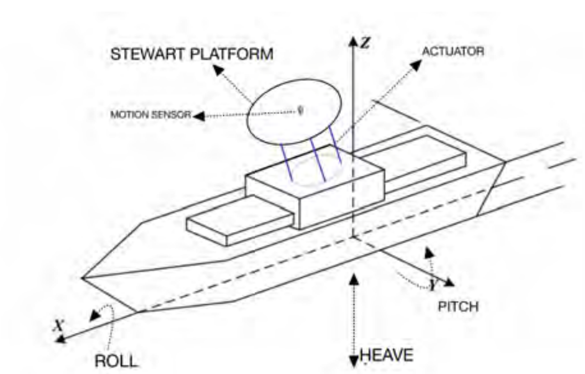
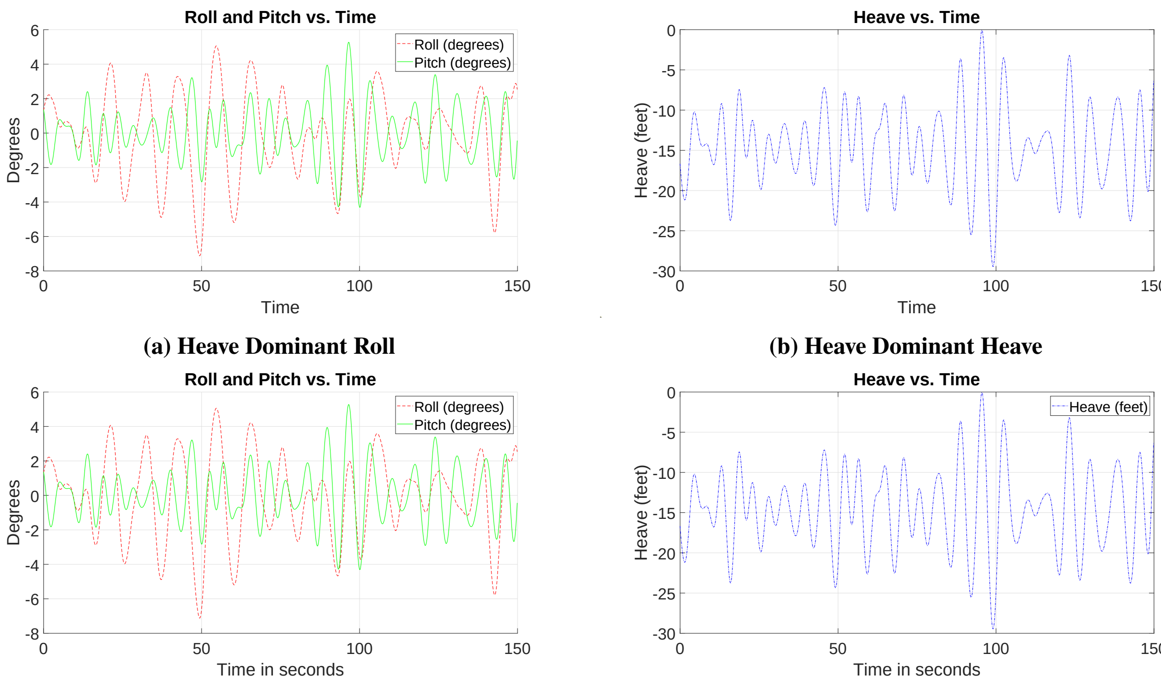


Fig. 1 The broad applicability of the Stewart platform for stabilizing ships at sea.

Rather than stabilizing the shipboard directly, in our experiment, experimentally taken ship data [12] was inputted into the platform to simulate ship conditions at sea state 4 [13]. The aim was to simulate roll(ϕ), pitch(θ), sway() and heave(Z) motions to test the landing algorithm, as these are the most significant motions captured in the scone dataset; The SCONE dataset [12], acquired through experimentation, was slated for simulation. This dataset comprises two distinct types of data: one predominantly influenced by roll and the other by heave. It is structured as a MATLAB variable containing 36,000 data points, with a time interval of 0.05 seconds between consecutive points. Our research focuses on testing our algorithm using both the roll-dominated and Heave-dominated subset of this dataset (‘SCONE_D3R_1‘ and ‘SCONE_D3H_1‘), where ‘Rot_X‘ represents roll, ‘Rot_Y‘ represents pitch, and Z represents heave. The deck is situated 15 feet (Z = -15 feet) above sea level (Z = 0). The data which is to be simulated is shown in Fig. 2.



(a) Heave Dominant Roll

(b) Heave Dominant Heave

(c) Roll Dominant Roll

(d) Roll Dominant Heave

Fig. 2 Comparison of Heave and Roll Dominant Parameters

B. Kinematics of the 3-legged platform

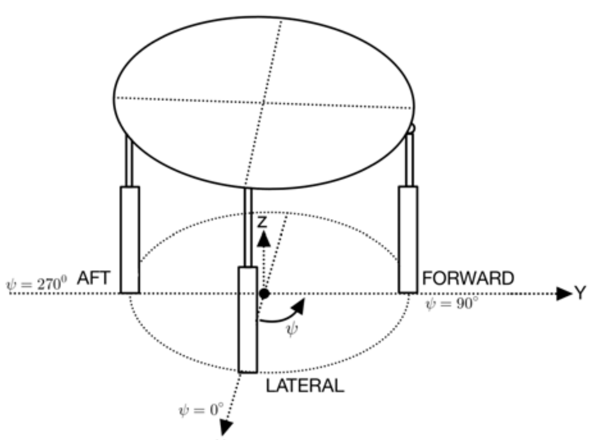


Fig. 3 Proposed design of the model

Given that actuators constituted the majority of the cost, reducing their number was crucial. Drawing inspiration from UH-60A swashplate [14], a 3-actuator model was chosen for the platform. Since the design drew inspiration from a swashplate, a similar naming convention was adopted for the three actuators: 'lateral', 'aft', and 'forward'. In this configuration (Fig. 3), the lateral actuator generates pitching movement, while the aft and forward actuators produce rolling movement.

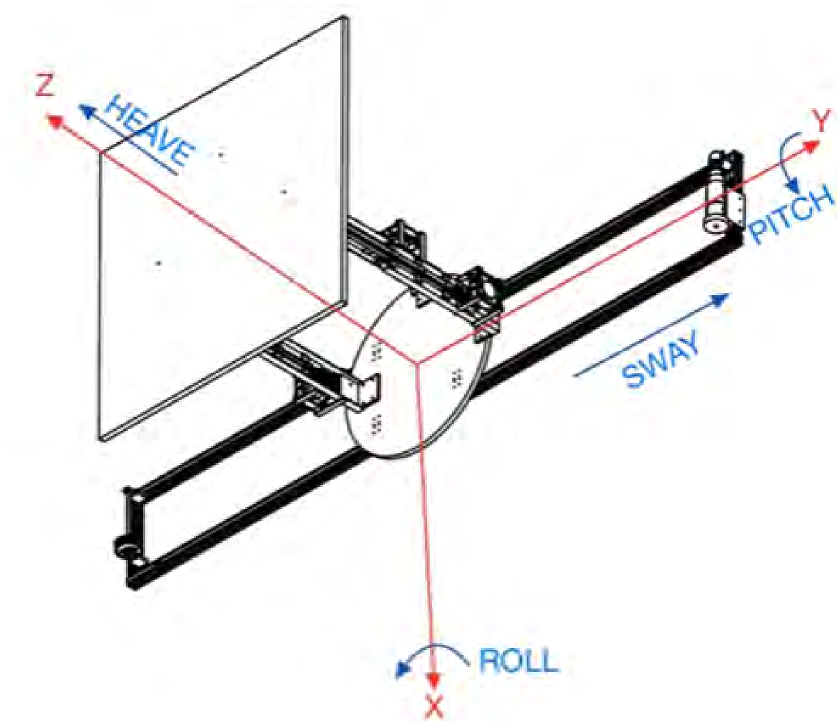


Fig. 4 Design of the Stewart Platform

The inverse kinematics (Fig.4) of the system were solved to determine the necessary movements (Eq. 1) of the actuators to achieve the desired roll, pitch, and heave motions. This process allowed for precise control over the platform's movements to accurately simulate the conditions needed for testing the landing algorithm.

$$
\begin{aligned}
l_a &= z - R\sin(\theta) - \sqrt{L^2 - (R(\cos(\theta) - \sin(\theta)\sin(\psi)) - H)^2} && \text{AFT} \\
l_l &= z + R\cos(\theta)\sin(\psi) - \sqrt{L^2 - (R\cos(\psi) - H)^2} && \text{LATERAL} \\
l_f &= z + R\sin(\theta) - \sqrt{L^2 - (-R(\cos(\theta) + \sin(\theta)\sin(\psi)) + H)^2} && \text{FORWARD}
\end{aligned}
\tag{1}
$$

C. Design of the platform

The decision was made to combine a hinge joint and a universal joint instead of using a three-degree-of-freedom ball joint to allow the top plate to rotate freely around the two desired axes while ensuring it can withstand significant loads. This resulted in adopting a linear actuator-hinge joint-universal joint setup, with the universal joint affixed to the top plates for all three actuators. Techincally this is known as a 2PRU-1PRS manipulator, where P is Prismatic join; R is revolute/hinge joint; U is universal joint and S is a spherical joint. Mechanical components were carefully chosen based on their functionality and cost considerations.

A customised linear actuator with an optical encoder to get deflection was chosen with a 30-inch stroke length to accommodate the needs of roll, pitch and heave adequately. The maximum linear speed is 20 cms per second with a load capacity of around 4Kg powered by a 24V SMPS box for power supply.

Keeping these configurations in mind, a CAD model was created for precision plasma cut for the base plate of the platform so that the actuators can be fixed on the base plate (Mild steel) with precision.The sway mechanism was introduced using a conveyor mechanism and a DC motor, incorporating limit switches and an encoder to prevent overshooting and get estimates of the sway, respectively.

Fig. 5 Platform's final Configuration

To verify the kinematics and dynamics of the platform, [15] was used to create a Matlab simulation, which would be useful to check for any structural limitations of the platform.

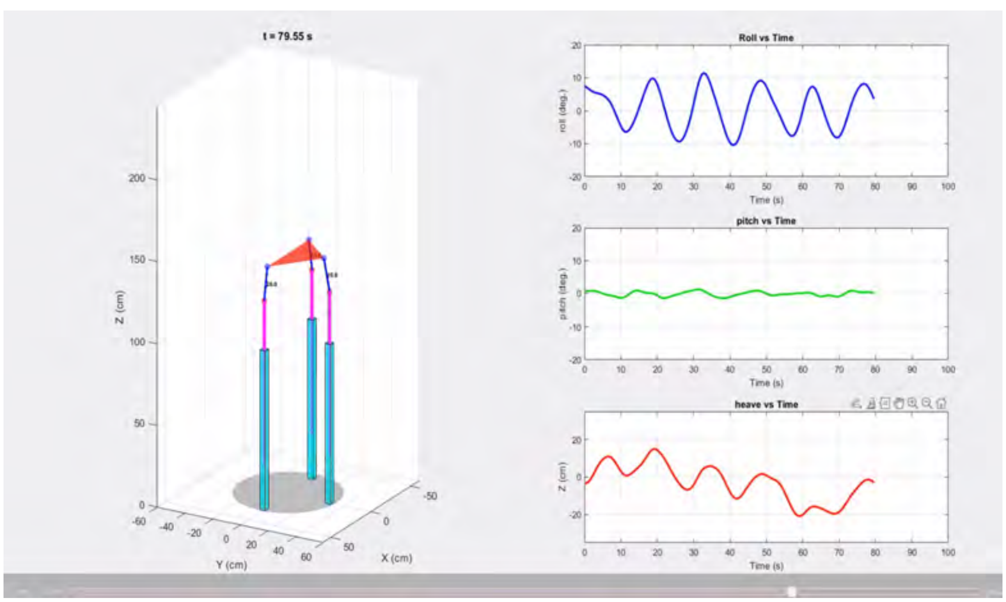


Fig. 6 Stewart platform Matlab validation

D. State estimation & Controller design

The MPU6050 IMU sensor was utilized to capture raw acceleration and gyroscope data from the platform.It was attached at the centre of the top plate of the platform. The accelerometer and gyroscope biases are calculated as zero offsets by keeping the IMU stationary and flat on level ground. For the scale factor of the accelerometer, we solve the system of linear equations for three known orientations. As described in ([16]) the euler angles, roll(ϕ) and pitch(θ) from acceleration readings are calculated as follows:

$$\phi^{accel} = tan^{-1}\left(\frac{a_y}{a_z}\right) \tag{2}$$

$$\theta^{accel} = tan^{-1}\left(\frac{-a_x}{\sqrt{(a_y)^2 + (a_z)^2}}\right) \tag{3}$$

Before discussing the calculation of the final euler angles, it is important to understand the context and objectives of our data collection methods.

- Gyroscope data furnishes rapid response and high-frequency details regarding orientation changes, albeit prone to drift over time.
- Accelerometer data delivers precise insights into gravity's direction but is vulnerable to noise and inaccuracies in an actuated system, particularly during motion.

Employing a blending factor of 0.9, the complementary filter ([17]) combines the gyroscopic and acceleration data to compute the final pitch and roll values. At each iteration, a new roll(Eq.4) and pitch(Eq.5) are determined, factoring in the difference in time (dt) between consecutive iterations.

$$\phi = \alpha \cdot (\phi + w_x \cdot dt) + (1 - \alpha) \cdot \phi^{accel} \tag{4}$$

$$\theta = \alpha \cdot (\theta + w_y \cdot dt) + (1 - \alpha) \cdot \theta^{accel} \tag{5}$$

The Heave calculation for the platform is determined by pose estimation using an ArUco marker and a webcam. The control architecture of the full system is given in Fig. 7

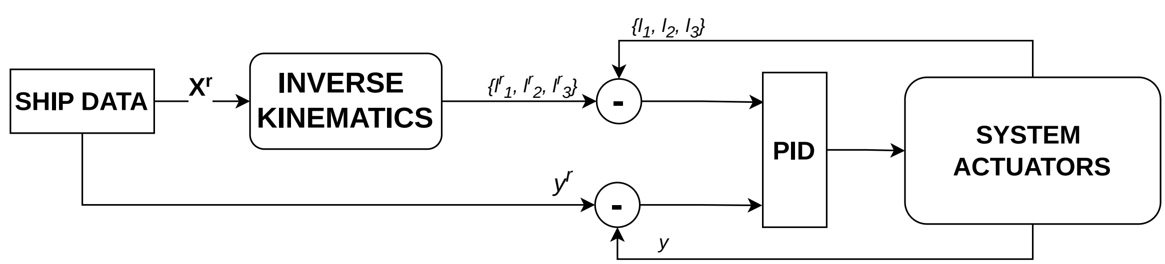


Fig. 7 Control architecture of the system

E. Hardware Implementation

The platform's electronics include an MPU650 sensor, which collects raw accelerometer and gyroscopic data and transmits it to an Arduino Due microcontroller based on the 32-bit Atmel SAM3X8E ARM Cortex-M3 CPU. This microcontroller performs inverse kinematics calculations and controls the linear actuators to achieve the desired lengths using the Cytron Dual Channel Enhanced 10Amp DC Motor Driver 30A Peak MDD10A, which features an H-bridge and is powered by a 24 Volt 100 Amps SMPS board. An STM-32 Master control unit takes the data from the quadrature encoders of the 3 actuators and sends the control input as PWM to the motor controllers. An SD card provides the microcontroller with reference ship sea state data and logs the resulting platform data. The master MCU, via Bluetooth, sends required motions to the Sway slave MCU. The overall hardware architecture of the platform is detailed in Figure 8.

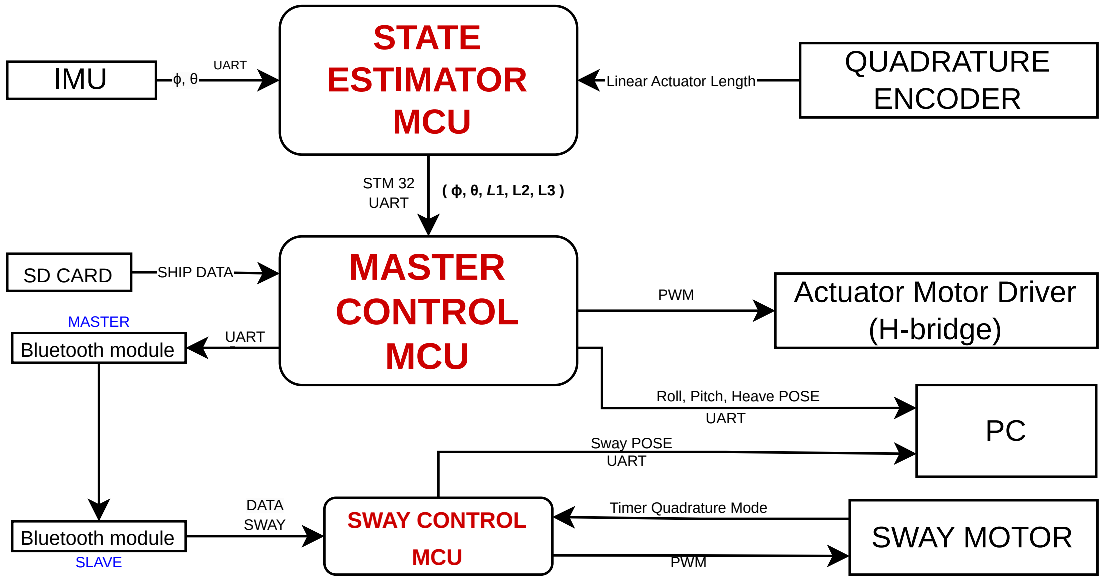


Fig. 8 Hardware architecture of the system

F. Results

Fig. 9 Mechanical assembly

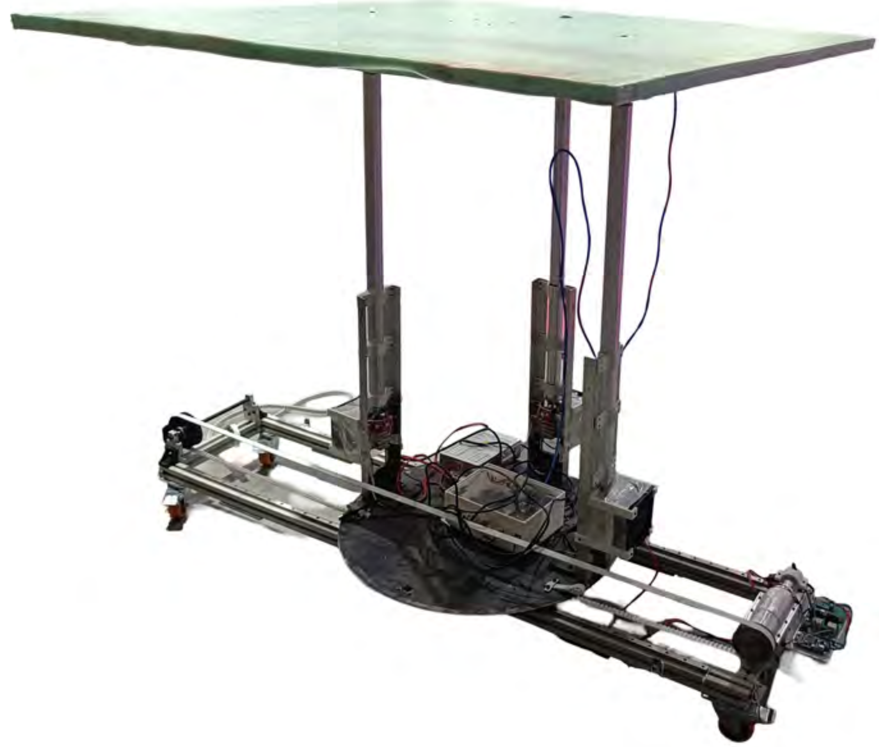


Fig. 10 Final Stewart Platform

The base plate was constructed and actuators were assembled (Fig. 9), with a top plate made of XPS foam board attached to the three actuators, chosen for its lightweight properties and minimal bending, which helps reduce vibrational noise affecting the IMU sensor. A Qualisys motion capture system was set up, and five super-spherical markers were placed on the platform for accurate motion tracking. The final setup is illustrated in Fig. 10.
To evaluate the platform's functionality, the Scone dataset was loaded onto the platform. The tracked values from the IMU, ArUco marker, and motion capture system were compared against the reference values for the roll(Fig. 11). Additionally, the sinusoidal data tracked by the ArUco marker and motion capture system was compared to the reference values for heave(Fig. 12).

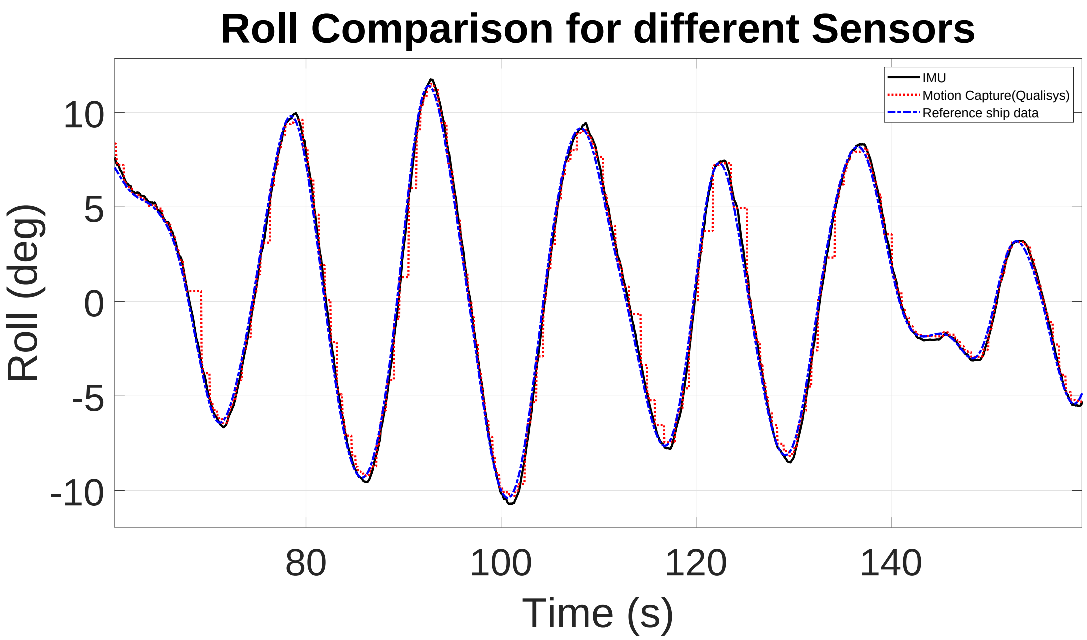


Fig. 11 Roll data performance of the platform

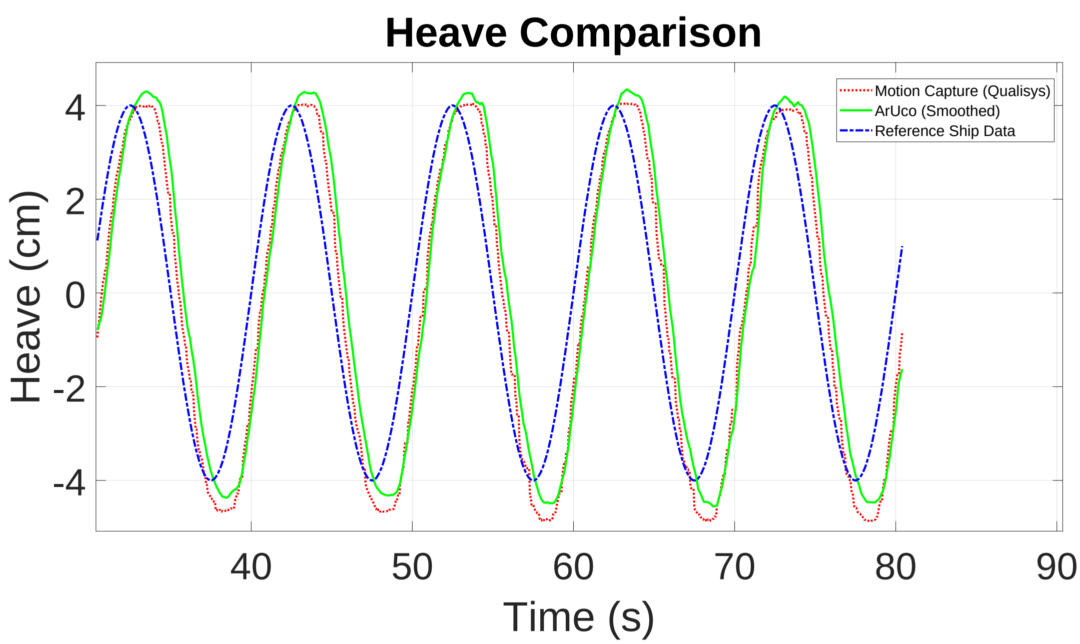


Fig. 12 Heave data performance of the platform

IV. Vision and Landing algorithm

The overall system consists of two components: the Unmanned Aerial vehicle in the air and the landing platform on the ground. The UAV consists of all the electronics, actuators and sensors, where as the ground system consists of Deck simulation platform with the visual markers pasted on the top of it. The UAV used for the experiments is a self-assembled quad-rotor in the standard X frame and running PX4 autopilot, see Fig.15.

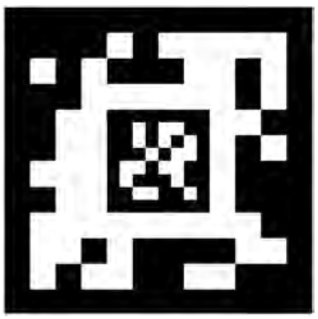


Fig. 13 Fractal marker composed of two internal markers [18]

Fig. 14 Detected all visible internal markers of the fractal marker [18]

Fig. 15 UAV used for the experiments

A. Landing algorithm

Two algorithms are operating in parallel within this module. One is the Landing algorithm, and the other is a Horizontal (X, Y) motion controller. This horizontal controller handles the trajectory planner to guide the UAV to hover above the X and Y locations of the landing zone. This is handled by the PX4 position controller. Currently, I am giving relative pose as position error for the PX4 position controller. The landing algorithm takes care of the descent phase of the UAV. It handles whether the UAV is ready to perform descent or whether it should hover depending upon perceived relative angle conditions and error in sway position. If a UAV is descending, it lowers its altitude by generating a way-point with its Z coordinate reduced by 0.02m and tracking X and Y. As the flight controller module runs at approximately 50Hz, reducing height by 0.02m results in a smooth execution.The UAV will descend if roll and pitch are both less than 5 degrees, otherwise it will hover to get to that position. The autonomous landing protocol is described in Algorithm 1:

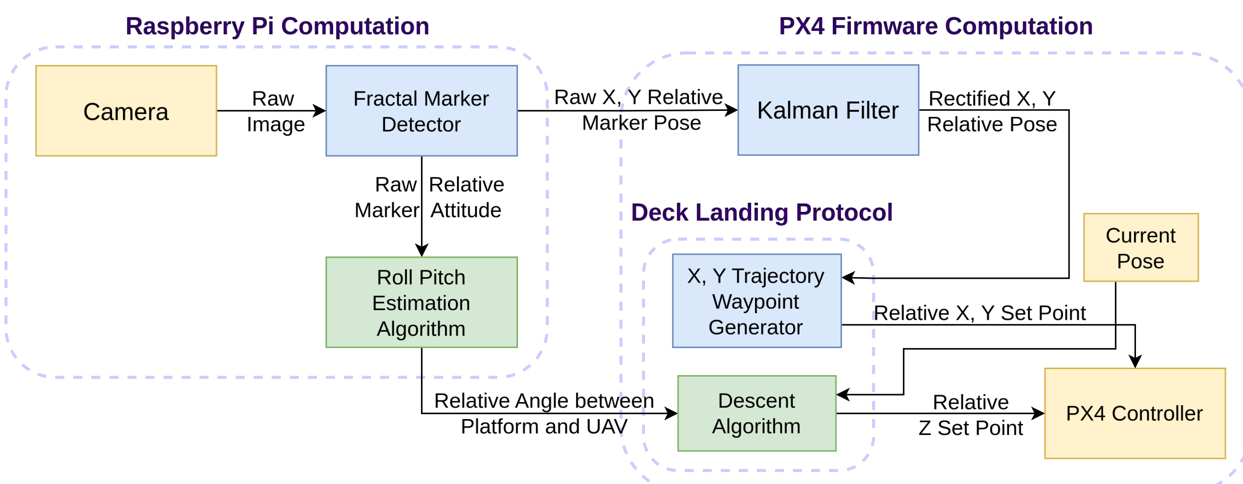


Fig. 16 Autonomous vision-based landing pipeline

Algorithm 1 Landing Algorithm

Require: $Relative\_Angle \leftarrow Updated$

Ensure: UAV hovers above the landing platform

$Set\_Pose \leftarrow 0, 0, 0$
$Curr\_Pose \leftarrow 0, 0, 0$
while $Landed \leftarrow False$ **do**
    $Curr\_Pose \leftarrow UAV.Current\_Pose$
    $Set\_Pose \leftarrow Curr\_Pose$
    if $Relative\_Angle \leq abs(Threshold)$ **&&** $abs(Error\_sway) < Threshold$ **then**
        $Set\_Pose.Z \leftarrow Curr\_Pose.Z - 0.02$
    end if
    Publish(Set_Pose) to UAV Controller
end while

B. Gazebo simulation

A gazebo simulation was set up with ROS1 and GAZEBO-SITL(software in the loop) using PX4's MAVROSE PROTOCOL. Iris drone was set up as the test bed for my algorithm, and the necessary changes were done in both the modules of PX4, and a ROS1 package was created, which would give vision the prediction and computer vision data by using a Raspberry Pi. I added a 6DOF platform in my simulation from GITHUB, which was my simulating platform in gazebo. The platform was initially RL-based, but I converted the IK from RL to the general IK of the platform, which was able to provide 1.2m of heave and 20 degrees of roll and pitch. I added an external sway feature which would track my ship data.

C. Results

A comparison of the 3D position estimated by the fractal marker with the ground truth(given by the motion capture system) is shown in Fig.16. Here, we can see the drone's trajectory while landing on a stationary fractal ArUco marker. The Sway tracking results are shown in Fig. 17 . The mean error was 0.0216 m. The Algorithm has already been tested in a Gazebo environment with an iris drone. The tracking results after the landing algorithm was triggered are shown in Fig. 19 and in Fig. 20 . The snapshots from the landing procedure are shown in Fig. 21.

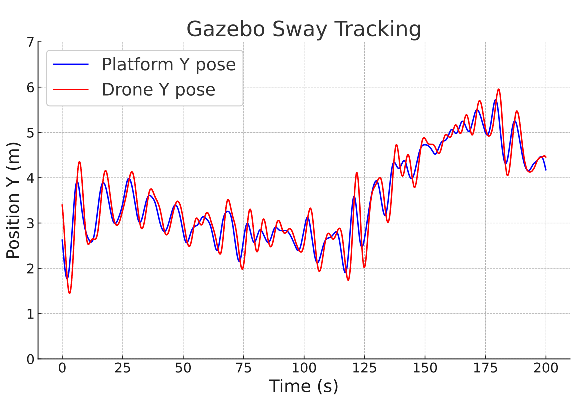


Fig. 17 Gazebo simulation of ship-deck sway tracking with 2.16 cm mean error

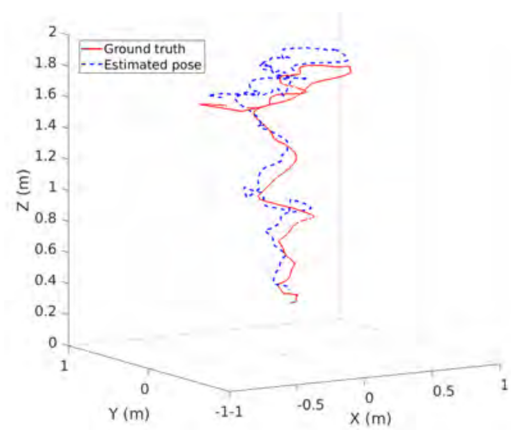


Fig. 18 3D position estimation along with ground truth

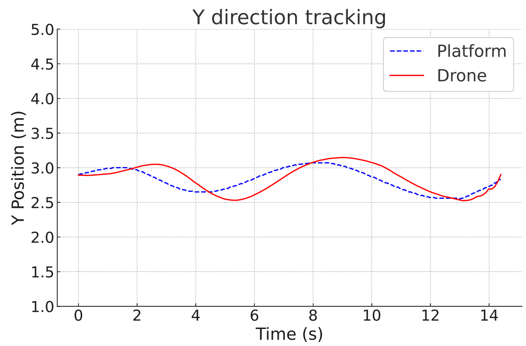


Fig. 19 Sway tracking in landing mode

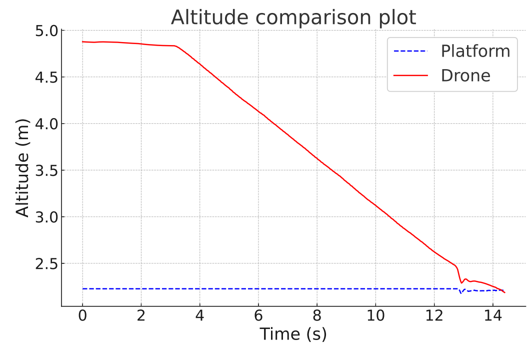
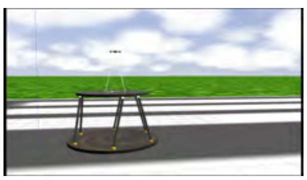
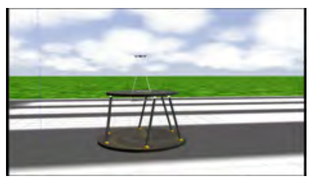


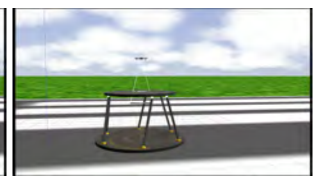
Fig. 20 Altitude comparison in landing mode



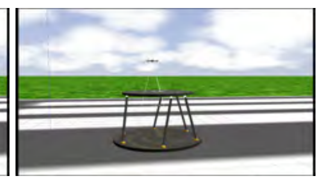
(a) Tracking snapshot 1



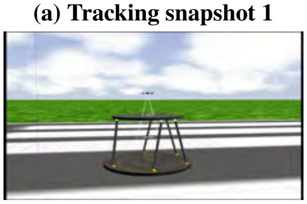
(b) Tracking snapshot 2



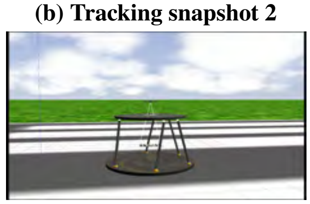
(c) Tracking snapshot 3



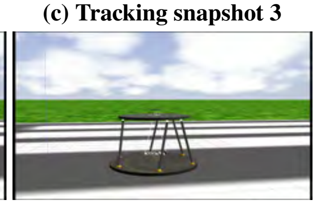
(d) Landing snapshot 4



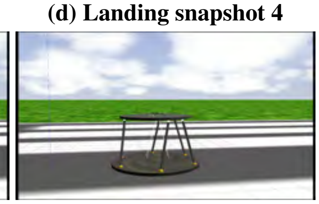
(e) Landing snapshot 5



(f) Landing snapshot 6



(g) Landing snapshot 7



(h) Disarmed snapshot 8

Fig. 21 Landing sequence snapshots

V. Conclusion

In this report, I have presented a validation experiment for my algorithm to land on a swaying, rolling and pitching platform. The algorithm for Heaving would require prediction using machine learning; that pipeline is almost finished, too. The real testing of both these algorithms is being tested on the given platform; these tests will finish by 22nd November This work has led to 2 conference papers and one potential patent(to be filled). The details of the conference papers are given below:

- *R. Shankar, C. Prachand, J. Singh, A. Abhishek, K.S. Venkatesh, "Vision-Based Landing of UAV on Simulated Ship Deck with Roll Pitch and Sway Motions", 2025 Vertical Flight Society, Forum 81* (abstract submitted)
- *C. Prachand, R. Rustagi, R. Shankar, J. Singh, A. Abhishek, K.S. Venkatesh, "Vision-Based Autonomous Ship Deck landing of an Unmanned Aerial Vehicle using Fractal ArUco markers", 2025 AIAA*(paper accepted)

For future work, as I mentioned, include real-time testing of the above-mentioned algorithms to land on roll, pitch and sway & to fully integrate the heave landing algorithm which includes a combination of Prediction of future heave by using a Gated Recurrent Unit network and using a QP solver to give optimal flight path for landing.

VI. Acknowledgments

I would sincerely like to thank my professor, Dr Abhishek (Professor, Dept. of Aerospace Engineering), for constantly guiding me, Mr Chiranjeev Pranchand (PHD, EE), who guided me throughout the work pertaining to the algorithm and finally, Mr Jitendra Singh (Sr. project officer) for guiding me throughout the platform problem. I would also like to thank the people of the Helicopter lab for constantly providing 3D-printed parts and logistical support.

References

[1] Xu, G., Zhang, Y., Ji, S., Cheng, Y., and Tian, Y., "Research on computer vision-based for UAV autonomous landing on a ship," *Pattern Recognition Letters*, Vol. 30, No. 6, 2009, p. 600–605. https://doi.org/10.1016/j.patrec.2008.12.011, URL http://dx.doi.org/10.1016/j.patrec.2008.12.011.

[2] Fucen, Z., Haiqing, S., and Hong, W., "The object recognition and adaptive threshold selection in the vision system for landing an Unmanned Aerial Vehicle," *2009 International Conference on Information and Automation*, IEEE, 2009. https://doi.org/10.1109/icinfa.2009.5204904, URL http://dx.doi.org/10.1109/icinfa.2009.5204904.

[3] Fan, Y., Haiqing, S., and Hong, W., "A vision-based algorithm for landing unmanned aerial vehicles," *2008 International Conference on Computer Science and Software Engineering*, Vol. 1, IEEE, 2008, pp. 993–996.

[4] Benini, A., Rutherford, M. J., and Valavanis, K. P., "Real-time, GPU-based pose estimation of a UAV for autonomous takeoff and landing," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 3463–3470.

[5] Lange, S., Sunderhauf, N., and Protzel, P., "A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments," *2009 international conference on advanced robotics*, IEEE, 2009, pp. 1–6.

[6] Fiala, M., "Comparing ARTag and ARToolkit plus fiducial marker systems," *IREE International Worksho on Haptic Audio Visual Environments and their Applications, 2005.*, IEEE, ???? https://doi.org/10.1109/have.2005.1545669, URL http://dx.doi.org/10.1109/have.2005.1545669.

[7] Li, Z., Chen, Y., Lu, H., Wu, H., and Cheng, L., "UAV Autonomous Landing Technology Based on AprilTags Vision Positioning Algorithm," *2019 Chinese Control Conference (CCC)*, IEEE, 2019. https://doi.org/10.23919/chicc.2019.8865757, URL http://dx.doi.org/10.23919/chicc.2019.8865757.

[8] Wang, Z., She, H., and Si, W., "Autonomous landing of multi-rotors UAV with monocular gimbaled camera on moving vehicle," *2017 13th IEEE International Conference on Control amp; Automation (ICCA)*, IEEE, 2017. https://doi.org/10.1109/icca.2017.8003095, URL http://dx.doi.org/10.1109/icca.2017.8003095.

[9] Wang, J., and Olson, E., "AprilTag 2: Efficient and robust fiducial detection," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016. https://doi.org/10.1109/iros.2016.7759617, URL http://dx.doi.org/10.1109/iros.2016.7759617.

[10] Staicu, S., "Dynamic analysis of the 3-3 Stewart platform," *UPB Scientific Bulletin D*, Vol. 71, No. 2, 2009, pp. 3–18.

[11] Abedinnasab, M. H., Yoon, Y.-J., and Zohoor, H., "Exploiting higher kinematic performance-using a 4-legged redundant PM rather than gough-stewart platforms," *Serial and Parallel Robot Manipulators-Kinematics, Dynamics, Control and Optimization*, Vol. 10, 2012, p. 32141.

[12] Schwartz, A., "Systematic Characterization of the Naval Environment (SCONE) - Standard Deck Motion Data for a Generic Surface Combatant," [Online]. Available: http://www.navalengineers.org/Symposia/PastSymposia/Launch-and-Recovery-2016/Program/Schwartz, 2015.

[13] Owens, E. H., *SEA CONDITIONSDouglas scale; Peterson scale; Sea stateSea conditions*, Springer US, New York, NY, 1984, pp. 722–722. https://doi.org/10.1007/0-387-30843-1_397, URL https://doi.org/10.1007/0-387-30843-1_397.

[14] Abhishek, A., Datta, A., and Chopra, I., "Prediction of UH-60A structural loads using multibody analysis and swashplate dynamics," *Journal of aircraft*, Vol. 46, No. 2, 2009, pp. 474–490.

[15] Herrero, S., Pinto, C., Altuzarra, O., and Diez, M., "Analysis of the 2 P RU-1 P RS 3DOF parallel manipulator: kinematics, singularities and dynamics," *Robotics and Computer-Integrated Manufacturing*, Vol. 51, 2018, pp. 63–72. https://doi.org/10.1016/j.rcim.2017.11.018.

[16] Kharolia, D., Singh, J., and Kothari, M., "Autopilot Design and Development for Multirotor UAVs," *AIAA SCITECH 2024 Forum*, 2024, p. 1698.

[17] Van de Maele, P.-J., "Reading a imu without kalman: The complementary filter," *Pieter-Jan. com Creaty in Automation y More*, Vol. 26, 2013.

[18] Romero-Ramire, F. J., Muñoz-Salinas, R., and Medina-Carnicer, R., "Fractal Markers: A New Approach for Long-Range Marker Pose Estimation Under Occlusion," *IEEE Access*, Vol. 7, 2019, pp. 169908–169919. https://doi.org/10.1109/ACCESS.2019.2951204.

Name: Ritwik Shankar
Roll Number: 210866
Department: Aerospace Engineering
Signature:

Guide's Name: Dr. Abhishek
Designation: Professor
Department: Aerospace Engineering
Signature: