Project Report:Design and fabrication of Stewart platform for Testing of UAV Ship Deck Landing

Abhishek*, Ritwik Shankar†, Jitendra Singh‡

I. Nomenclature

R	=	Radius of the stewart platform
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

A. Problem Statement

The challenge of landing a rotor-craft on the deck of a moving ship presents a complex problem that requires precise piloting skills and specialized training. The continuous pitching and rolling of the ship's deck, driven by ocean swells and winds, creates an inherently unstable landing platform that challenges the pilot's ability to time the touchdown precisely. The limited physical dimensions of the ship's deck provide a little margin for error, requiring the pilot to execute the landing manoeuvre with exceptional precision within the constrained spatial envelope. The airflow patterns around the ship's superstructure and the interaction with the surrounding marine environment can result in turbulent and unpredictable wind conditions, further complicating the pilot's control inputs.

Successful ship deck landings demand the ability to accurately anticipate the ship's motion and synchronize the aircraft's descent. This requires specialized skills, such as precise control inputs to compensate for platform instability, keen situational awareness to monitor the landing environment, and rapid decision-making to address unexpected events. In this paper we investigate a solution to these problems by using vision based estimation of the ship deck platform then a landing algorithm which achieves precision landing on a rolling and pitching platform. To solve this problem, a control algorithm was developed using position estimates from computer vision data. To test the algorithm, a Parallel manipulator was developed, which can accurately simulate ship motions such as roll, pitch and heave, giving a 3-DOF motion to the landing plate. This type of parallel manipulator is known as a Stewart Platform.

B. Literature Review

Ready-made Stewart Platforms were inquired about, but those that fit our requirements were priced above INR 40 Lakhs. Consequently, creating an in-house alternative to cater to our required ship motions became necessary. The

*Professor, Aerospace Engineering, Indian Institute of Technology Kanpur, Uttar Pradesh, India, 208016, AIAA Member
†Under-Graduate Student, Aerospace Engineering, Indian Institute of Technology Kanpur, Uttar Pradesh, India, 208016
‡Senior Project Executive Officer, Aerospace Engineering, Indian Institute of Technology Kanpur, Uttar Pradesh, India, 208016

dynamics of a Stewart platform were explored through the examination of two sources ([1] & [2]). 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.

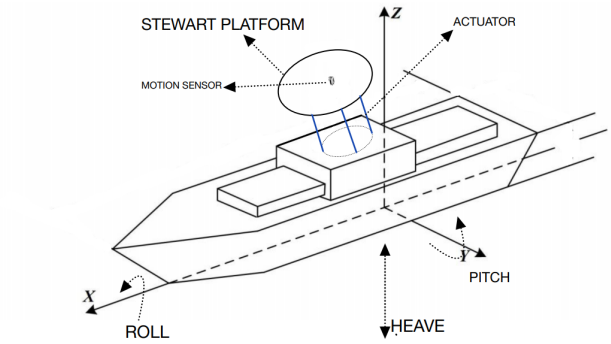
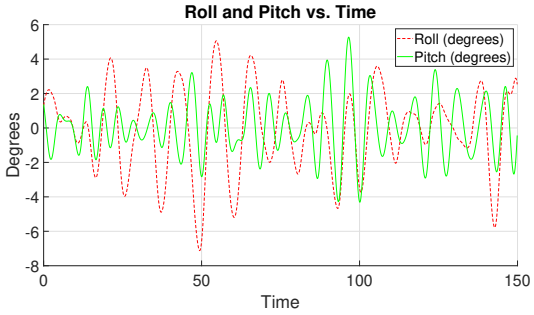


Fig. 1 The broad applicability of the Stewart platform for stabilizing ships at sea.

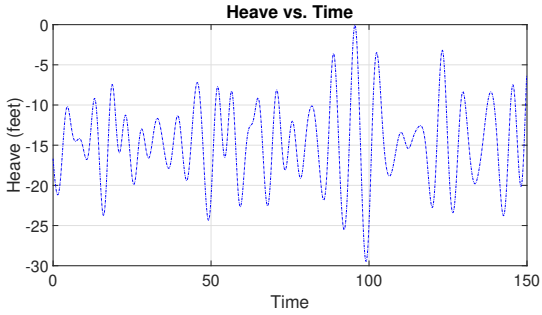
Rather than stabilizing the shipboard directly, in our experiment experimentally taken ship data [3] was inputted into the platform to simulate ship conditions at sea state 4 [4].

The aim was to simulate roll(ϕ),pitch(θ) and heave(Z) motions to test the landing algorithm, as these are the most significant motions captured in the scone dataset; other motions in the plane of the ocean can be mimicked by keeping the platform on a car, which can move in the two perpendicular directions on the ground and also rotate around the axis perpendicular to the plane. Existing market platforms, albeit with a higher frequency operating range, were priced at over 40 Lakhs INR. Hence, there was a significant demand for a cost-effective alternative to a Stewart platform capable of generating the desired motions at a lower frequency of approximately 10 Hertz.

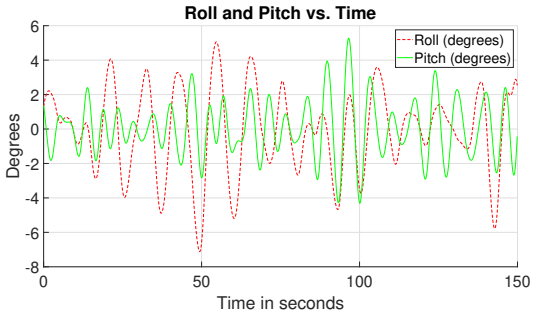
The SCONE dataset [3], acquired through experimentation, was slated for simulation. This dataset comprises two distinct types of data: one predominantly influenced by roll and the other by heave. It is structured as a MATLAB variable containing 36,000 data points, with a time interval of 0.05 seconds 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.



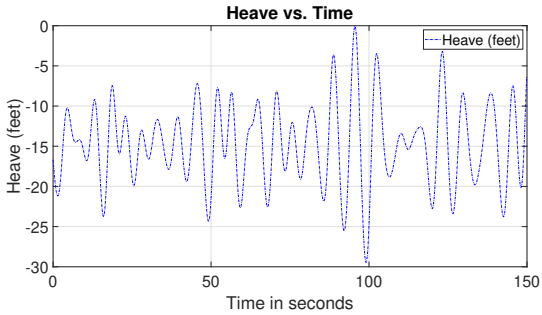
(a) Heave Dominant Roll



(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

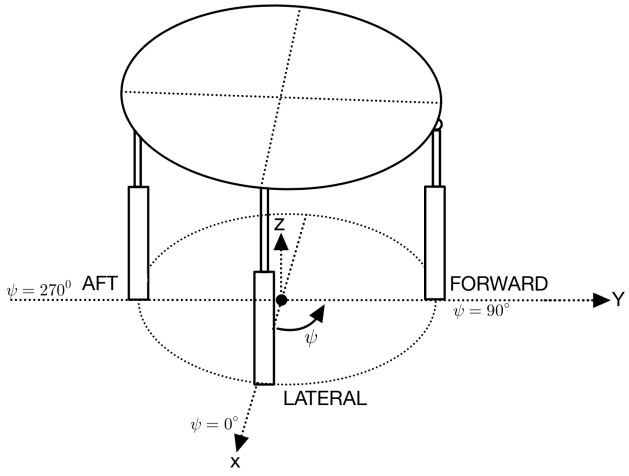


Fig. 3 Proposed design of the model

Given that actuators constituted the majority of the cost, reducing their number was crucial. Drawing inspiration from UH-60A swashplate [5], 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.

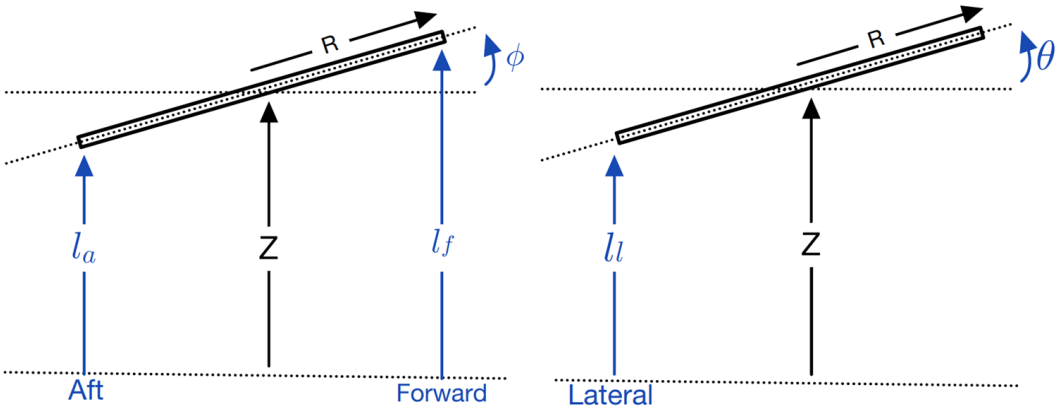


Fig. 4 Deflection diagram of the system

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.

$$\begin{aligned} l_a &= Z - R\sin(\phi) \quad \text{AFT} \\ l_f &= Z + R\sin(\phi) \quad \text{FORWARD} \\ l_l &= Z - R\sin(\theta) \quad \text{LATERAL} \end{aligned} \tag{1}$$

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. Mechanical components were carefully chosen based on their functionality and cost considerations.

A linear actuator with an optical encoder to get deflection was chosen with a 6-inch stroke length to accommodate the needs of roll, pitch and heave adequately. The linear speed was 2 inches per second with a load capacity of 36 pounds(16.32 Kg) and a backtracking limit of 200 pounds(90.71 Kg) powered by a 12V SMPS box for power supply. The distance of the three actuator cuts from the centre was 30 cm, which could give us a deflection of approximately 10 degrees in under a second.

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.

Fig. 5 Platform's final Configuration

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 ([6]) 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 ([7]) 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. 6

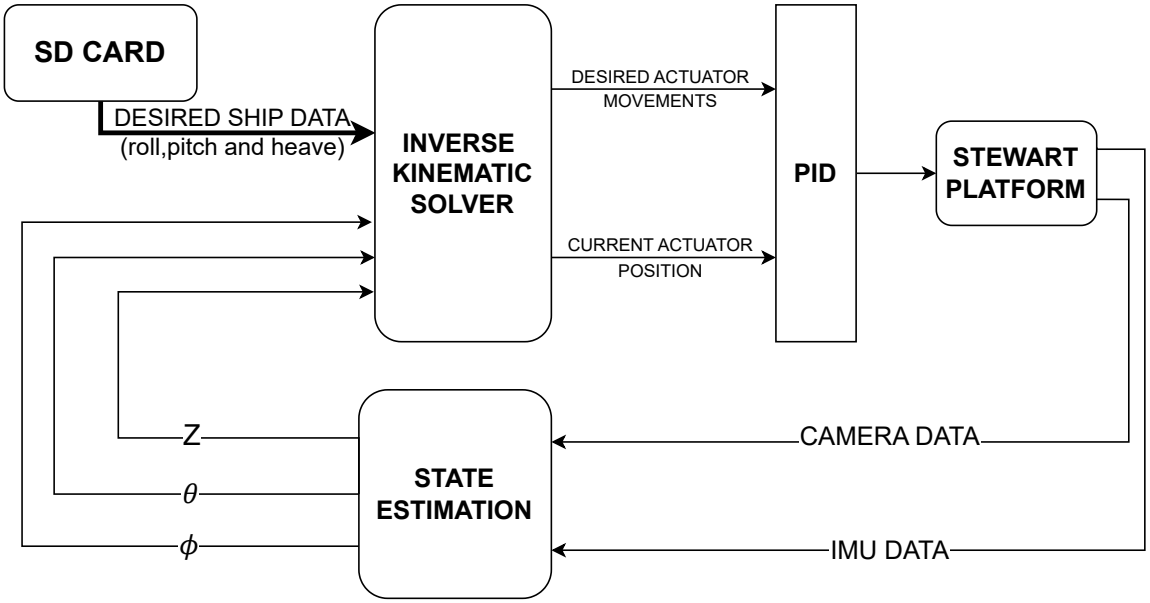


Fig. 6 Control architecture of the system

E. Hardware Implementation

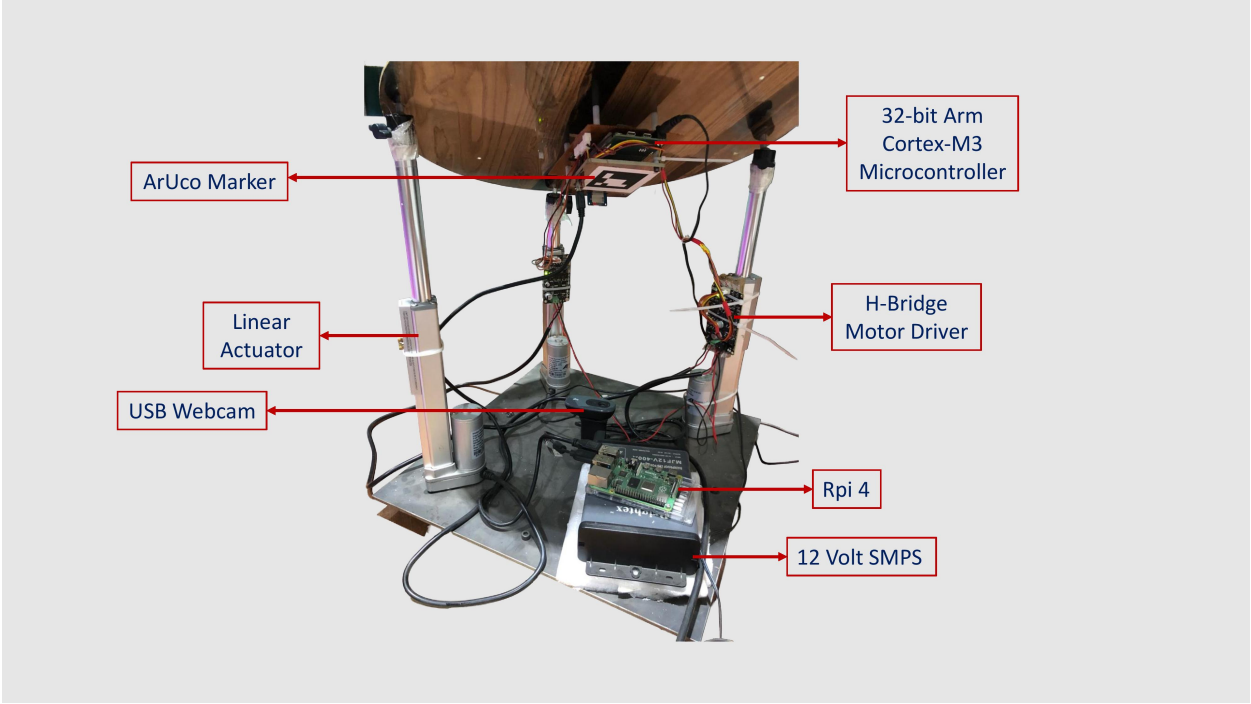


Fig. 7 Electronic components of the system

The platform's electronics, illustrated in Fig. 7, includes an MPU6050 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 12 Volt 100A SMPS board. A Quad-core Cortex-A72 (ARM v8) 64-bit Raspberry Pi 4 processes ArUco marker data from a USB webcam to determine the distance from the bottom plate to the board (Z, or heave) and receives Qualisys motion capture data via a master-slave link, with the Qualisys system as the master and the Raspberry Pi 4 as the slave. An SD card provides the microcontroller with reference ship sea state data and logs the resulting platform data. The overall hardware architecture of the platform is detailed in Figure 8.

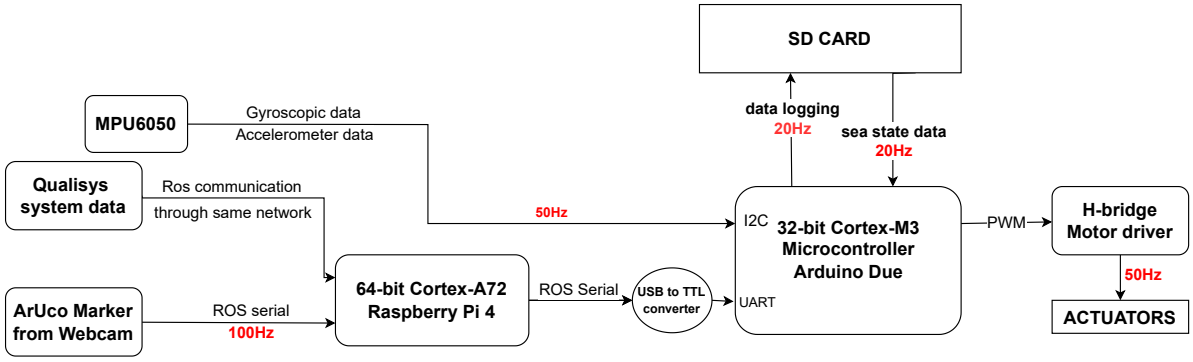


Fig. 8 Hardware architecture of the system

F. Results

Fig. 9 Stewart Platform 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).

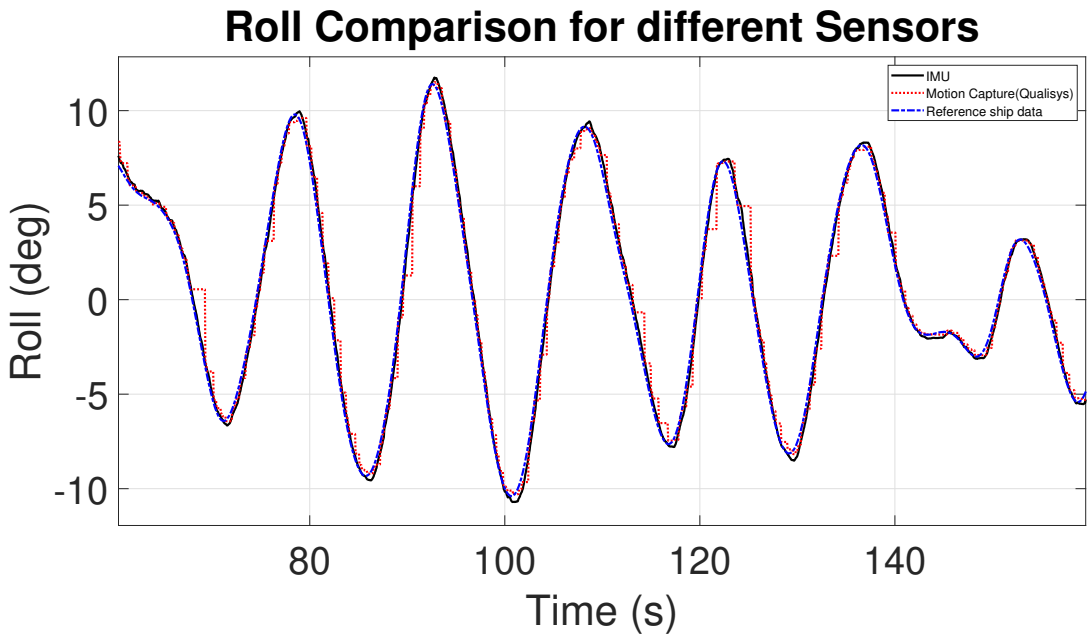


Fig. 11 Roll data performance of the platform

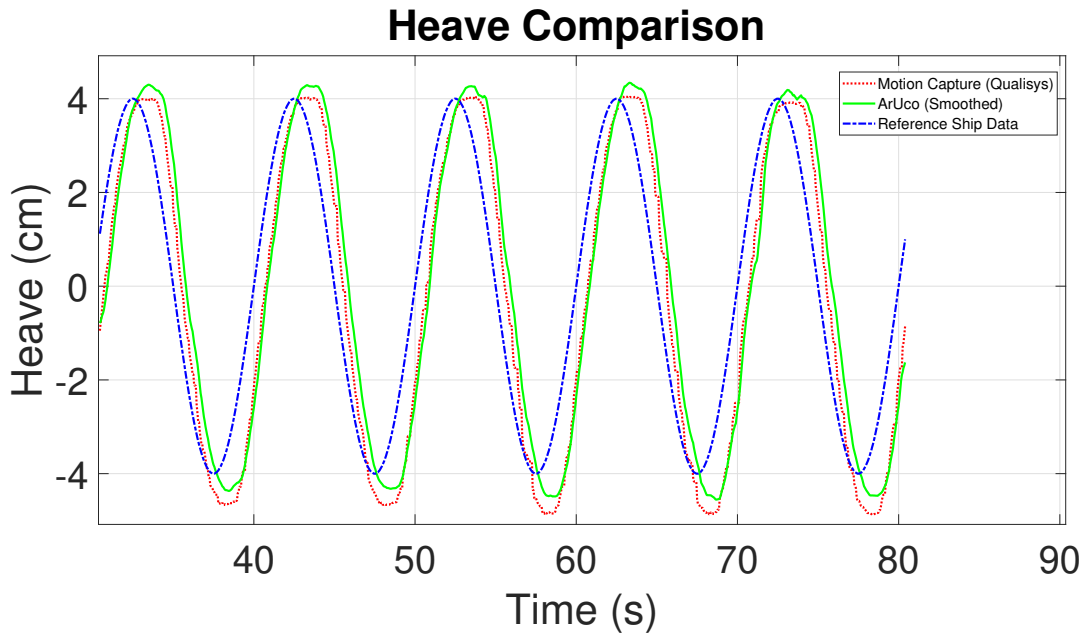


Fig. 12 Heave data performance of the platform

IV. Future Work

A. Linear Speed

The first iteration model was successfully able to cater for the needs of Roll/Pitch or Heave, but when all three motions were combined, it was not able to follow the data with the precision that we wanted.So, increasing the speed of the linear actuators was necessary. The faster motors, which is rated for 22 Volts, have been ordered, and the gear

assembly of the Linear actuator has been changed by using a custom 3D Printed Gear assembly and increasing the ratio from the initial 5:1 to 1:1.6, increasing the speed 8.33 times. We expect a further increase in linear speed after replacing the actuator motors. The motors have arrived and the final assembly is to be completed by 10th August.

B. Additional Degrees of Freedom

To test the algorithm for a more realistic condition, additional movements of $Sway$ are to be added as well,this will be achieved by **TO BE ADDED**

References

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

[2] **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.**

[3] **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.**

[4] **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.**

[5] **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.**

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

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