Project report

DESIGN AND FABRICATION OF A TEST-BED TO SIMULATE VISION-BASED AUTONOMOUS LANDING OF A UAV ON A SHIP DECK

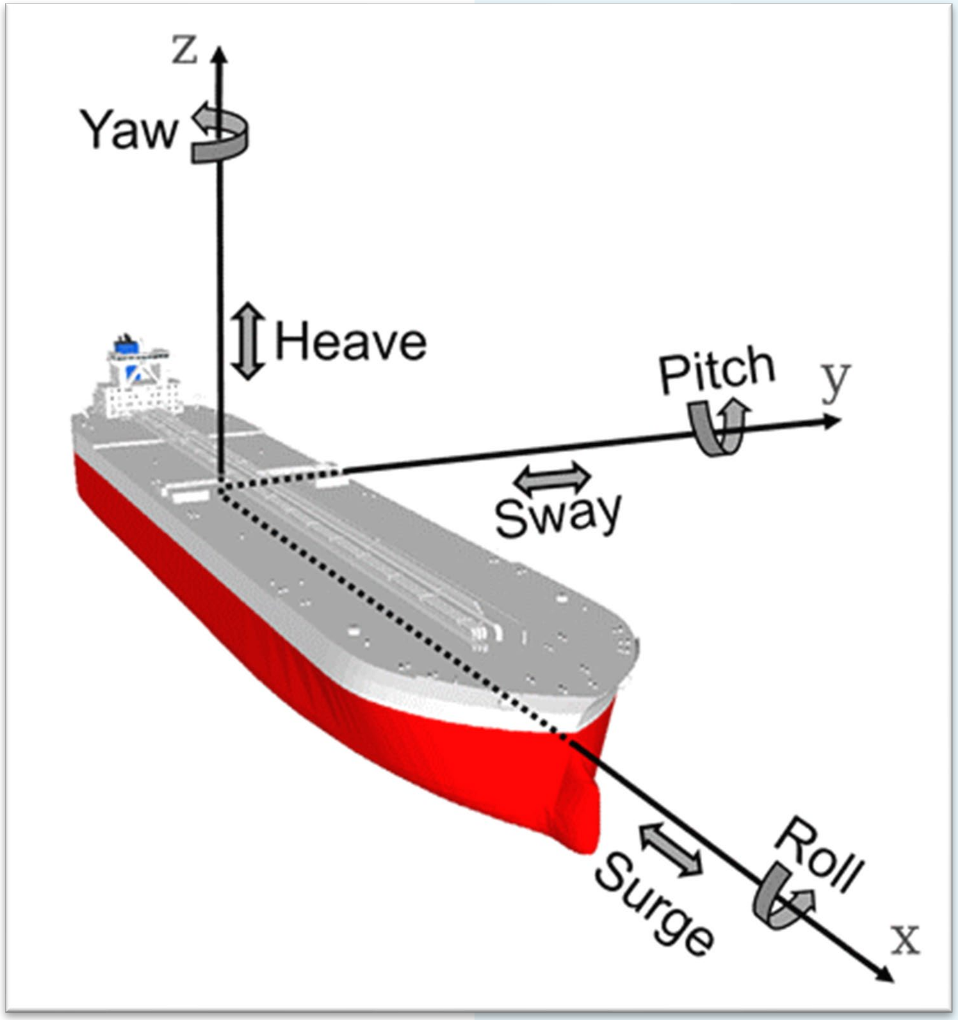
RITWIK SHANKAR

SENIOR UNDERGRADUATE IIT KANPUR

PROBLEM STATEMENT

SHIP DECK LANDING

- **Complex Challenge:**
Landing a rotor-craft on a moving ship requires precise piloting and specialized training.
- **Autonomy:**
Requires real-time pose(itions) estimates and a robust algorithm to land the aircraft precisely.
- **Proposed Algorithm:**
  Pose estimation using Computer vision
  Pose prediction using GRU



SHIP MOTIONS

SHIP-DECK EMULATOR

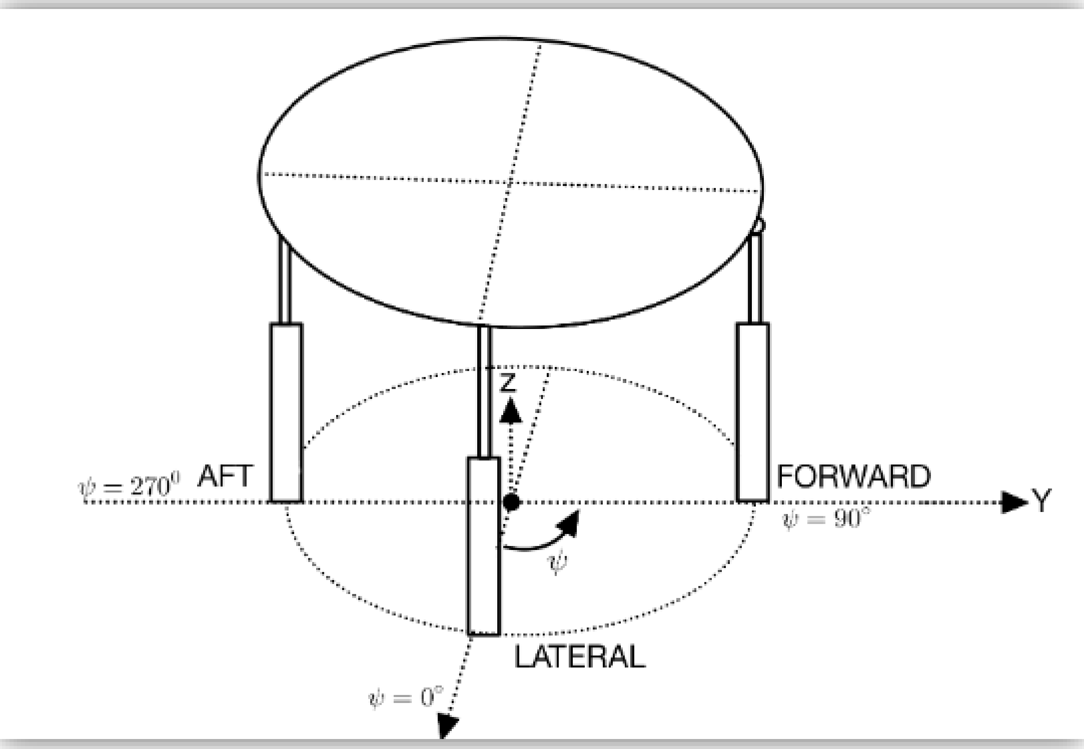
To simulate the ship-deck a Parallel manipulator will be used, this type of a parallel manipulator is called a Stewart Platform.

Its main features are :

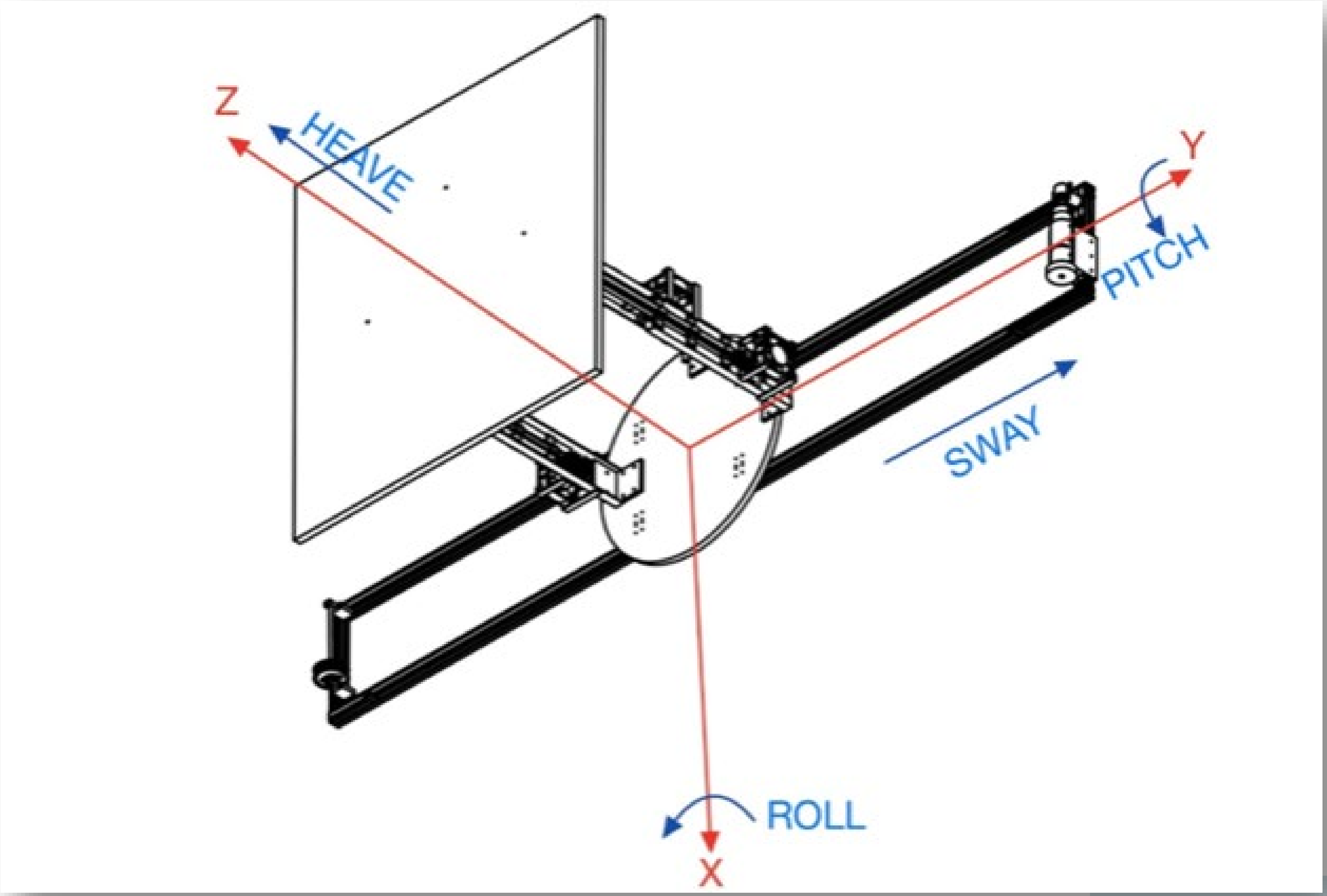
- 4 DOF Capabilities
- Up to ±30 ° movement capabilities
- Up to 80cm of Z movement
- Up to 1m of sway movement

SHIP-DECK EMULATOR

MODEL INSPIRED FROM UH-60 SWASHPLATE



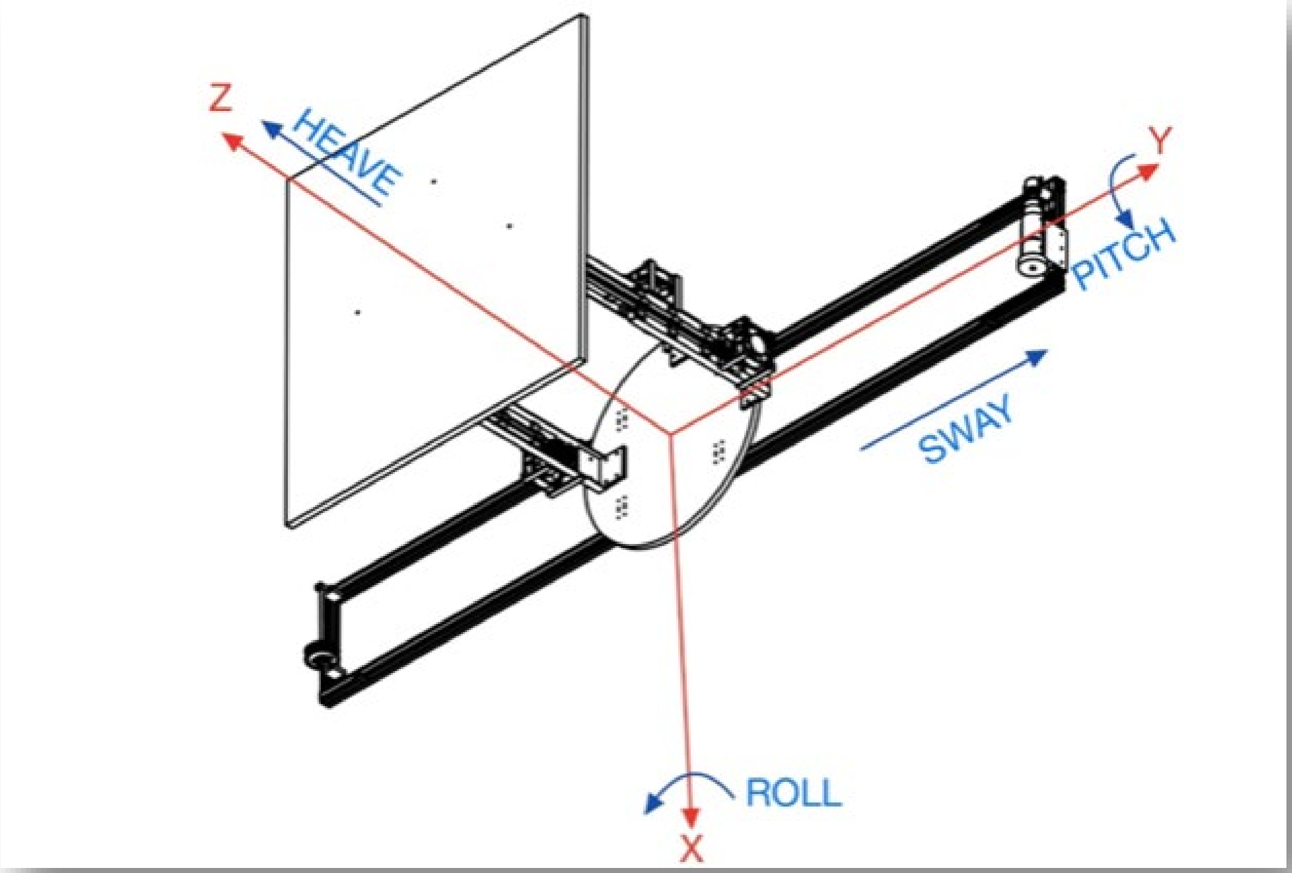
SHIP-DECK EMULATOR



SHIP-DECK SIMULATION

The features of this custom design are:

- Fully customized linear actuators to work at 20cm/s velocity
- Quadrature encoder was added to get accurate estimates of the actuator pose
- DC motor used with conveyer mechanism for sway motions
- The design is highly redundant because each component is replaceable



INVERSE KINEMATICS

$$l_a = z - R\sin(\theta) - \sqrt{L^2 - (R(\cos(\theta) - \sin(\theta)\sin(\psi)) - H)^2} \quad \text{AFT}$$

$$l_l = z + R\cos(\theta)\sin(\psi) - \sqrt{L^2 - (R\cos(\psi) - H)^2} \quad \text{LATERAL}$$

$$l_f = z + R\sin(\theta) - \sqrt{L^2 - (-R(\cos(\theta) + \sin(\theta)\sin(\psi)) + H)^2} \quad \text{FORWARD}$$

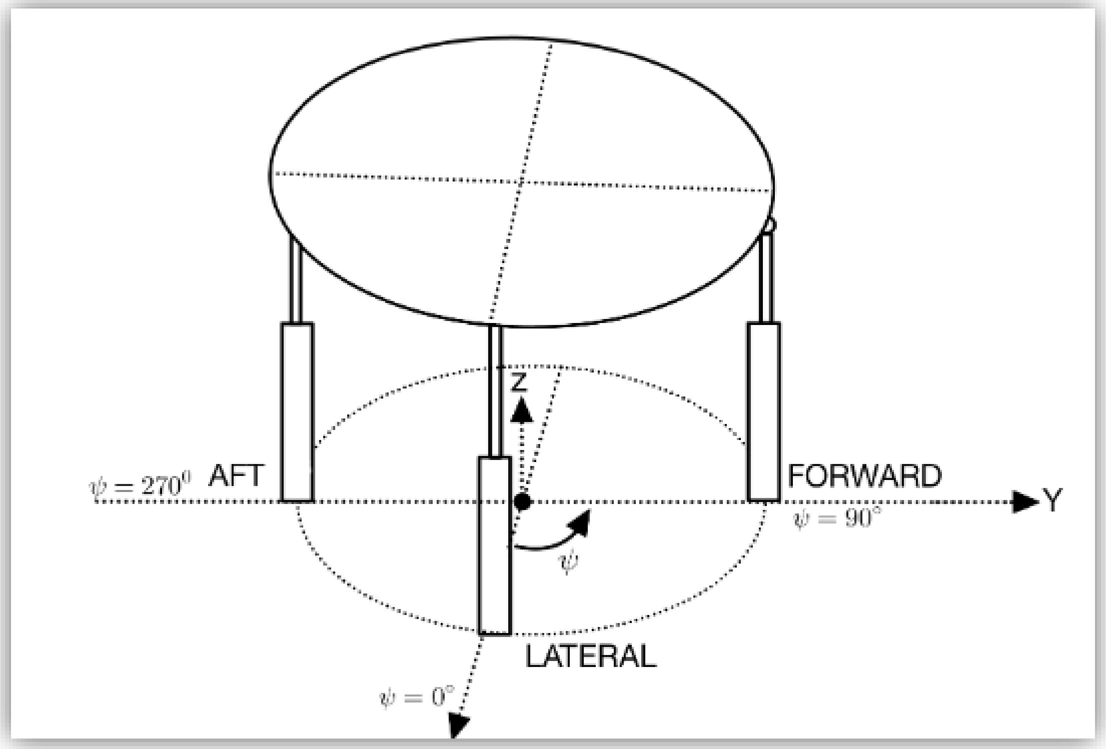
This good work is not mine, I got it from Abedinnasab, et al
“Exploiting higher kinematic performance-using a 4-legged redundant PMrather than gough-stewart platforms,”

INVERSE KINEMATICS

$$l_a = z - R\sin(\theta) - \sqrt{L^2 - (R(\cos(\theta) - \sin(\theta)\sin(\psi)) - H)^2} \quad \text{AFT}$$

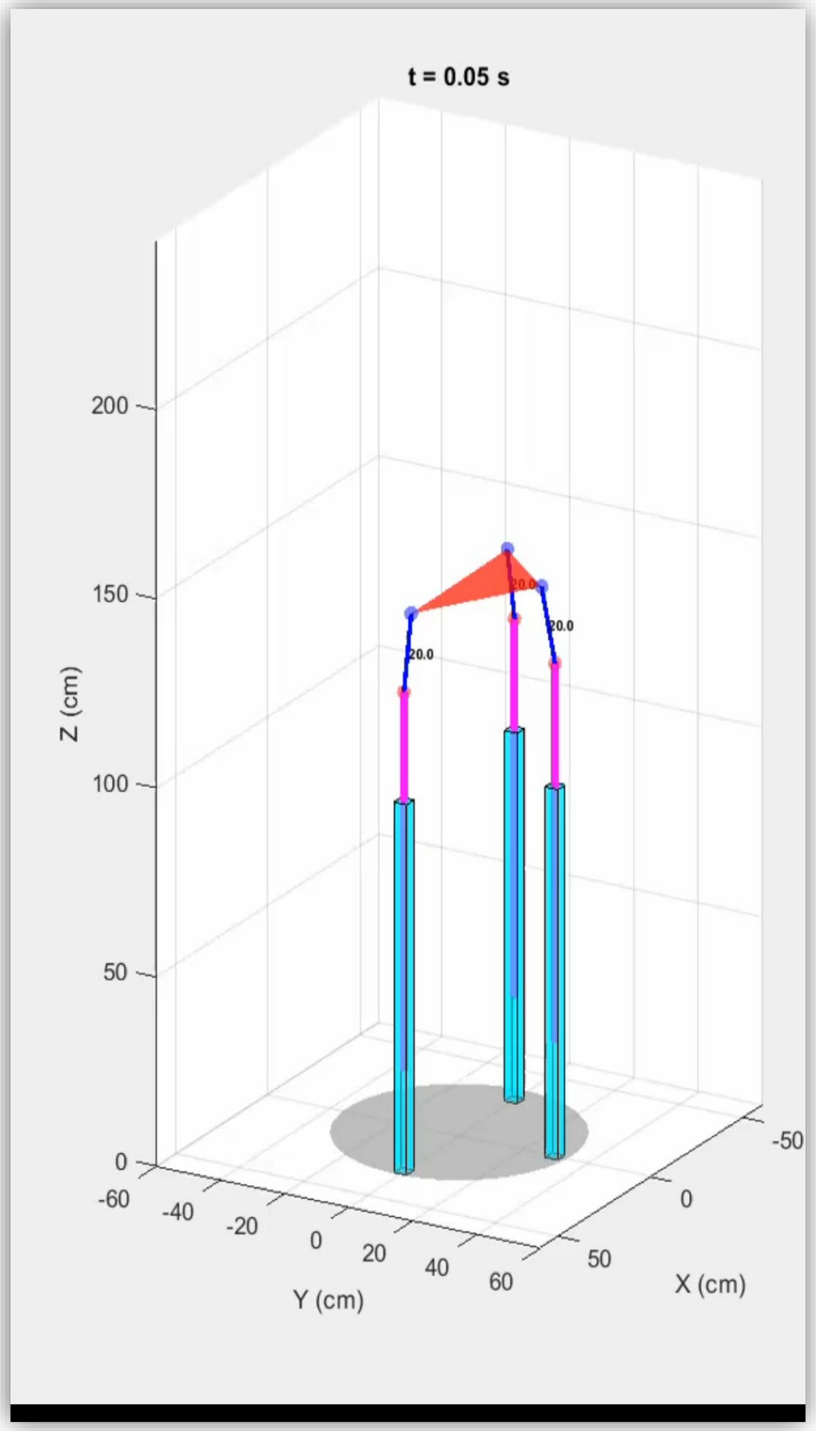
$$l_l = z + R\cos(\theta)\sin(\psi) - \sqrt{L^2 - (R\cos(\psi) - H)^2} \quad \text{LATERAL}$$

$$l_f = z + R\sin(\theta) - \sqrt{L^2 - (-R(\cos(\theta) + \sin(\theta)\sin(\psi)) + H)^2} \quad \text{FORWARD}$$

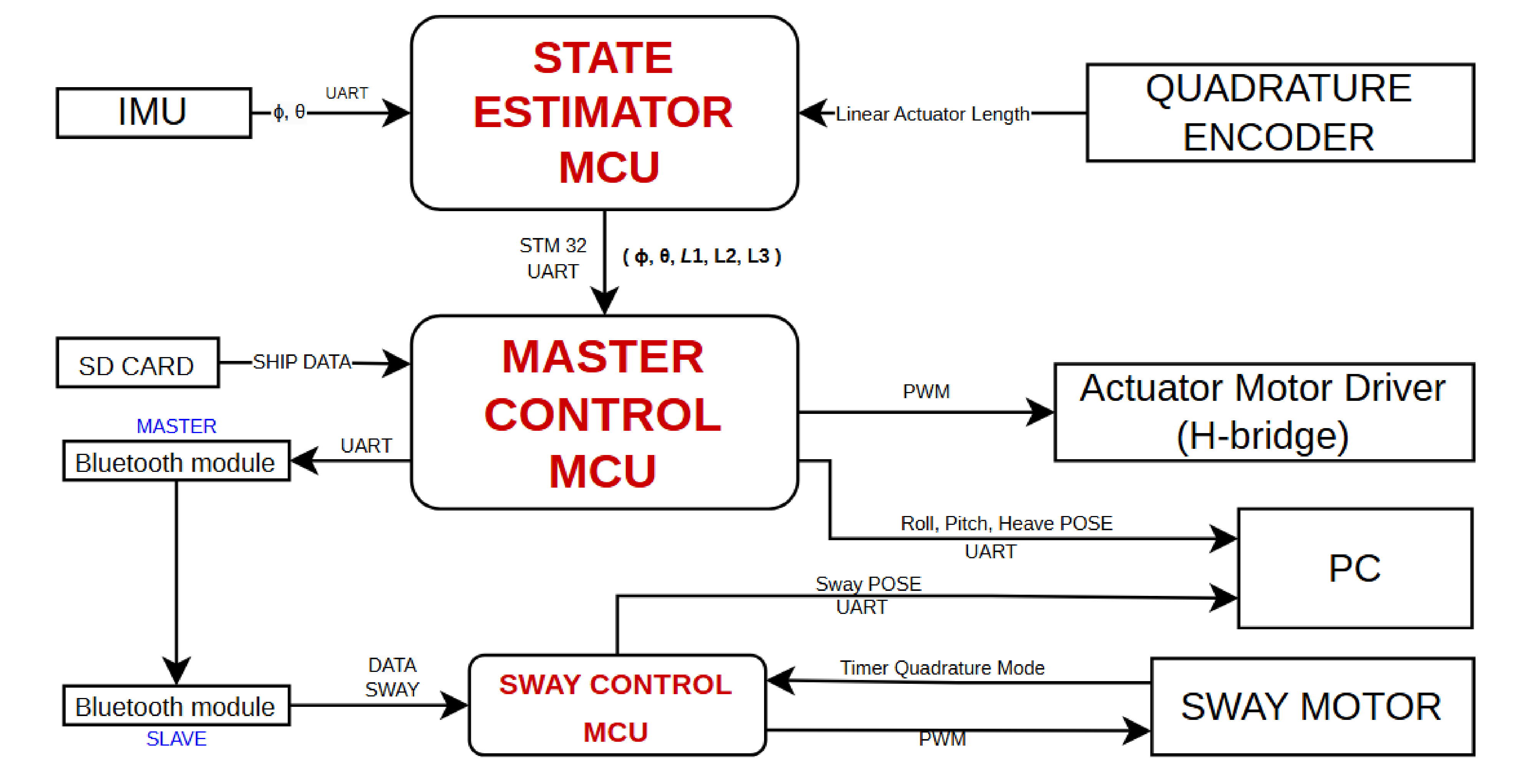


This good work is not mine, I got it from Abedinnasab, et al "Exploiting higher kinematic performance-using a 4-legged redundant PMrather than gough-stewart platforms,"

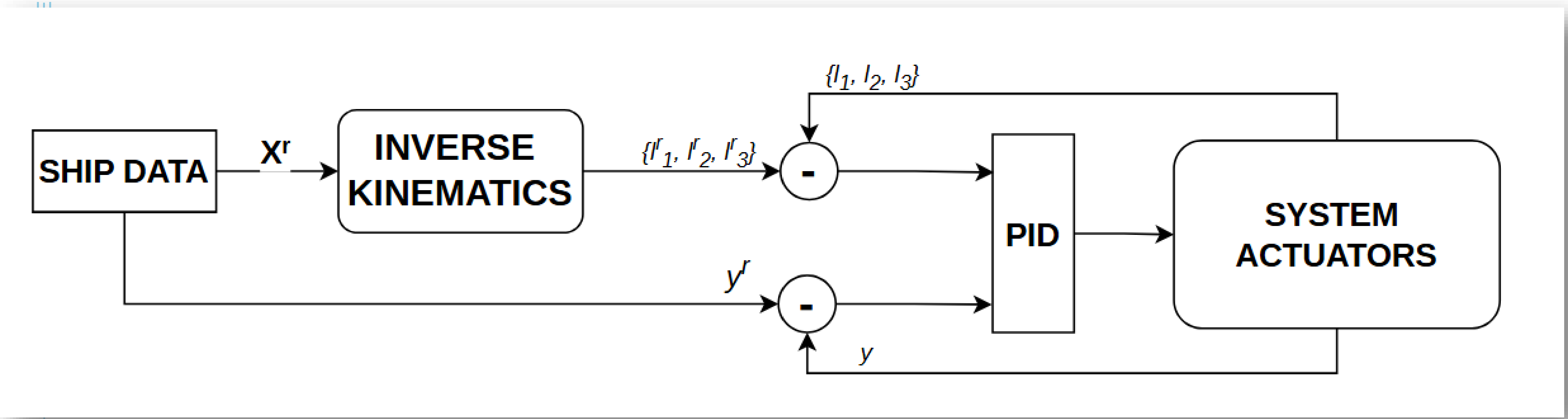
MATLAB SIMULATION



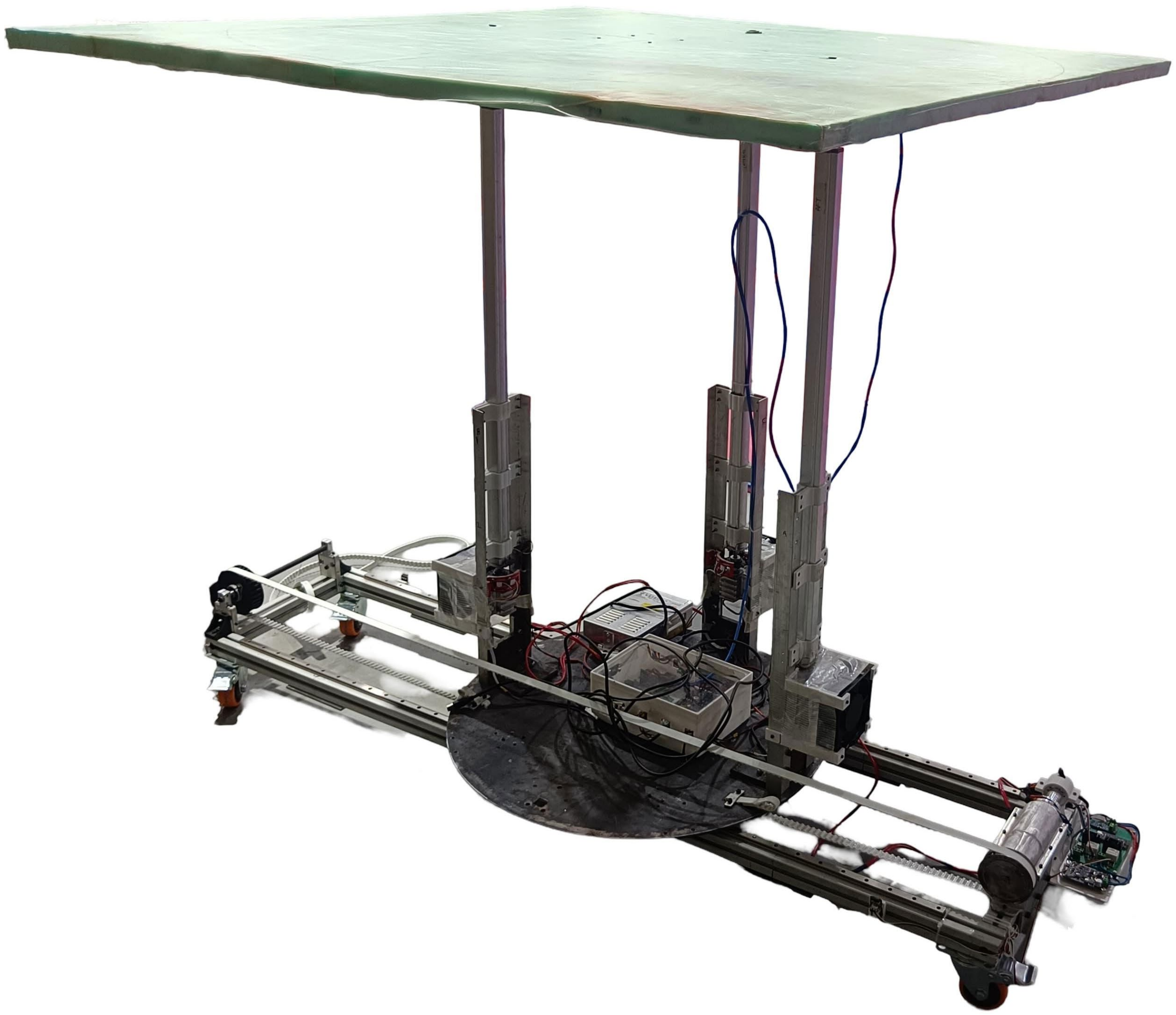
HARDWARE ARCHITECTURE

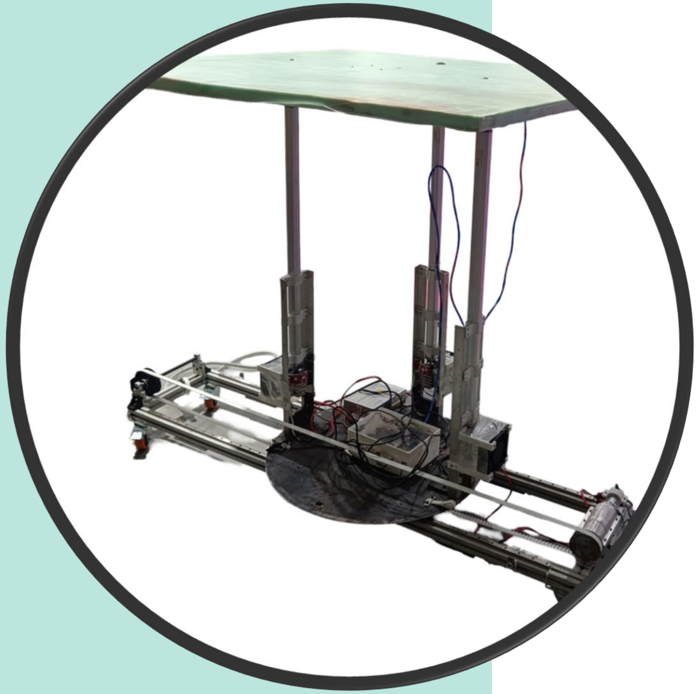


CONTROL ARCHITECTURE



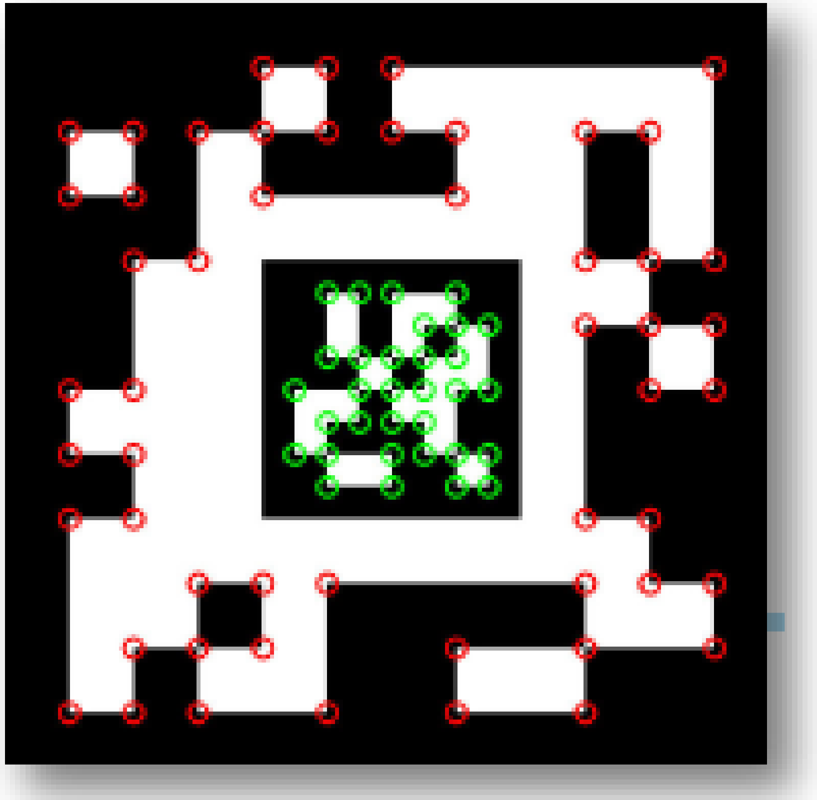
FINAL STEWART PLATFORM





FINAL STEWART PLATFORM

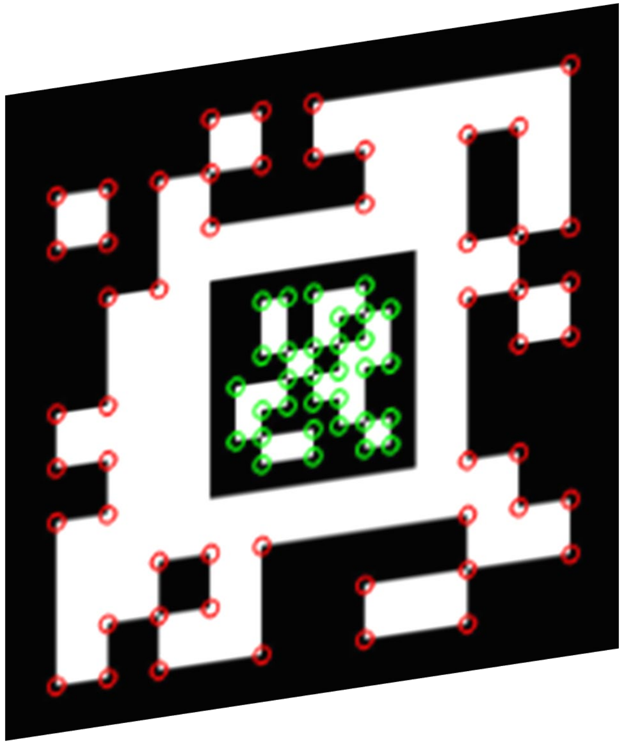
VISION AND LANDING ALGORITHM



ARUCO MARKER

TEST DRONE

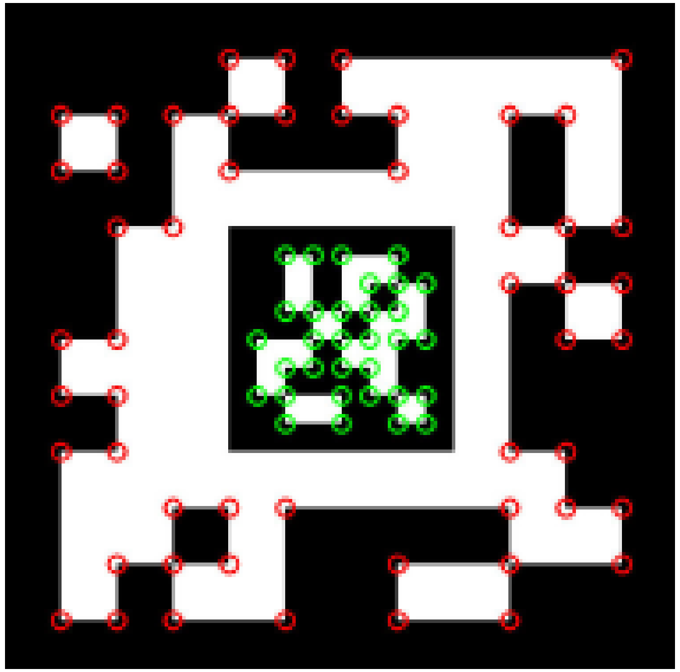
POSE ESTIMATION USING ARUCO



HOW DOES DETECTION WORK?

1. Edge detection Algorithms first detect the ArUCo marker and gets a warped image.
2. As the original ArUCo marker is already fed inside the algorithm so it rotates the acquired image to match the ideal position.

POSE ESTIMATION USING ARUCO



HOW DOES DETECTION WORK?

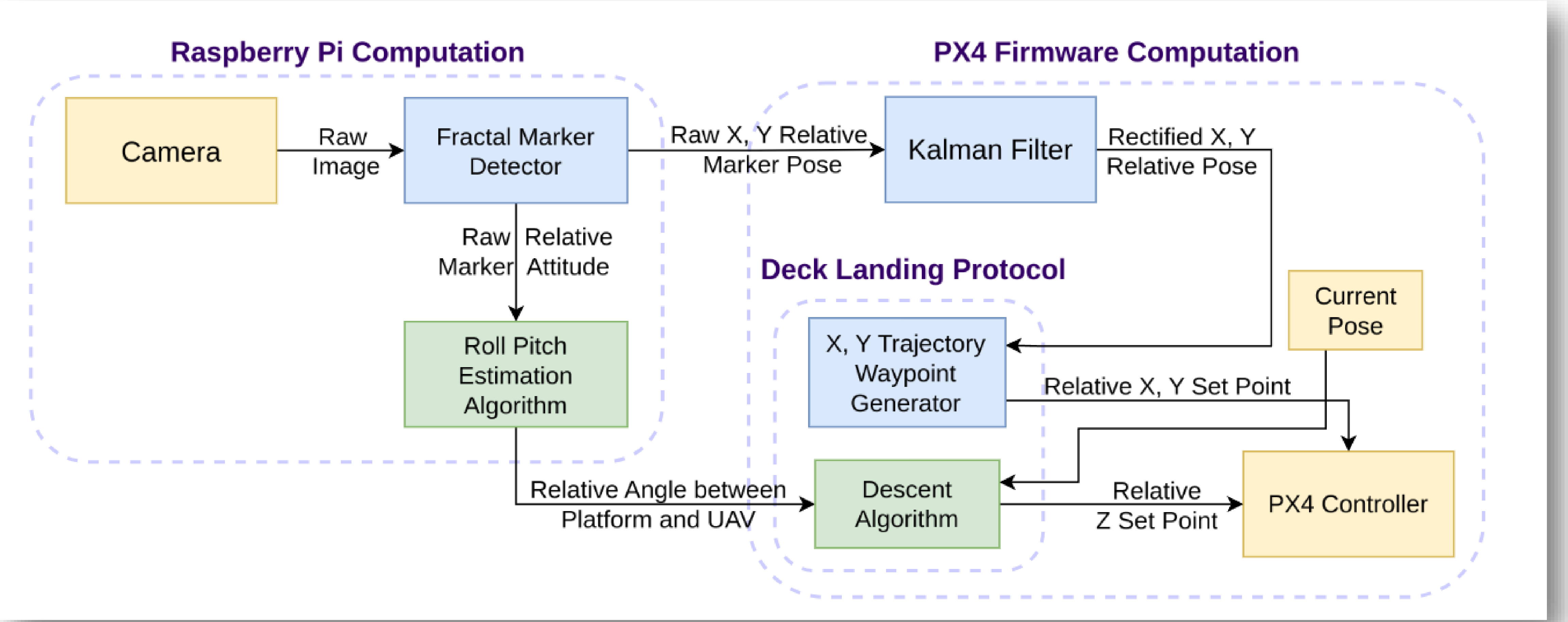
1. Edge detection Algorithms first detect the ArUCo marker and gets a warped image.
2. As the original ArUCo marker is already fed inside the algorithm so it rotates the acquired image to match the ideal position.

3. This rotation of the image gives us the Roll and Pitch of the UAV and the size of the edge gives the heave.

ONBOARD DETECTION

- The quadcopter has a onboard rasberry pi which uses OpenCV for ArUCo detection
- The pose estimates are sent as a ros topic at 100Hz

VISION AND LANDING ALGORITHM



VISION AND LANDING ALGORITHM

ROS

- PACKAGE: PRECISION_LAND_VISION
- ROSNODES
  - DETECTION_NODE
  - KALMAN_FILTER NODE
  - LANDING_NODE
- GAZEBO
  - PLATFORM MODEL
  - DRONE IRIS MODEL

PX4

- CUSTOM MODULE PREC_LAND_VISION
  - Takes in velocity setpoints from the Kalman filtered values
  - Applies a PID controller to these setpoints
  - Gives acceleration setpoints accordingly

VISION AND LANDING ALGORITHM

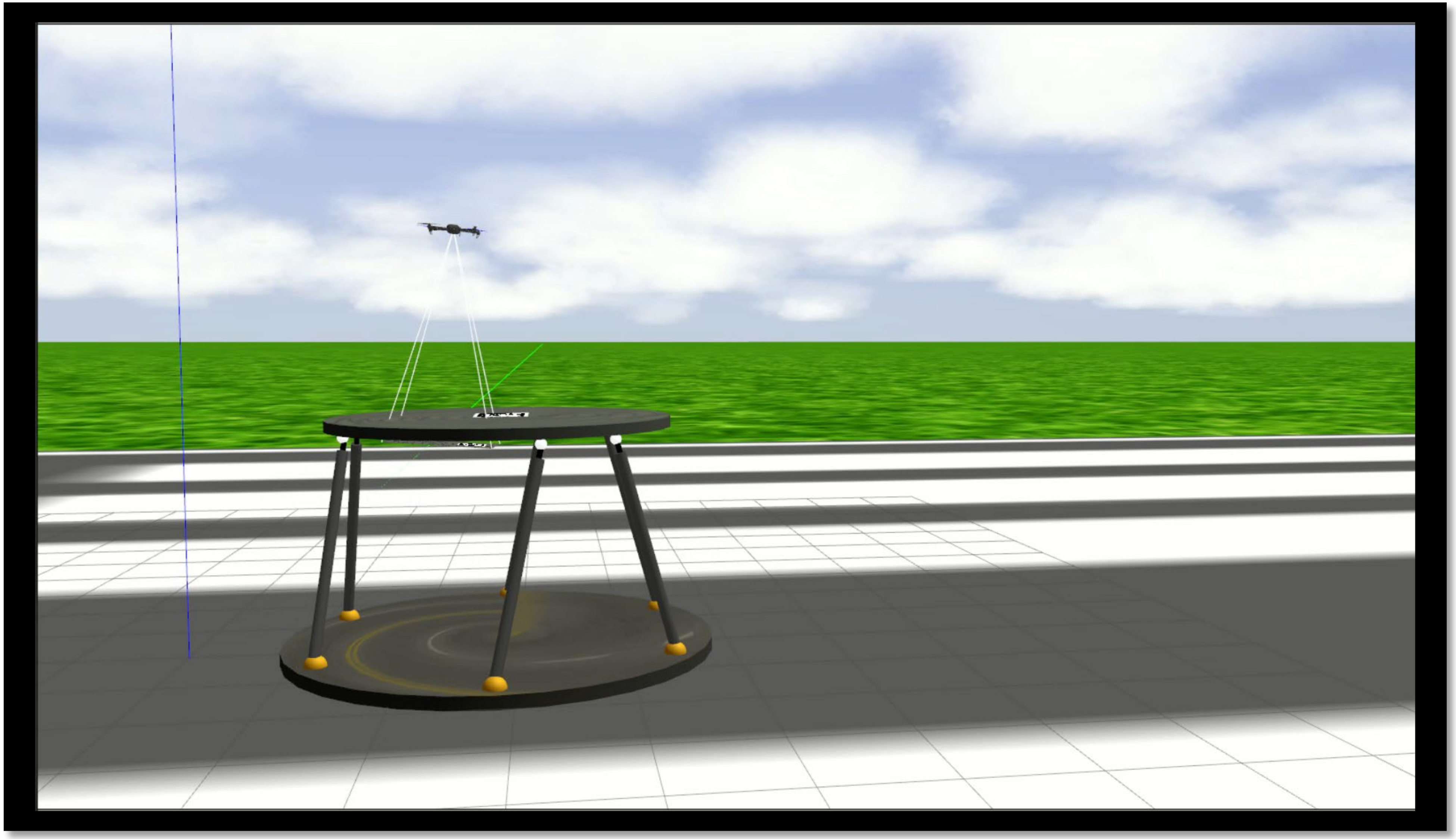
Algorithm 1 Landing Algorithm

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

VISION AND LANDING GAZEBO SIM



PUBLICATIONS FROM THIS UGP

Vertical Flight Society

AIAA
SHAPING THE FUTURE OF AEROSPACE

- 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 SCITECH FORUM.

THANK YOU

THANK YOU

Q/A