TEAM E: CRITICAL DESIGN REVIEW

DOCK-IN-PIECE

"FLYING IS SIMPLE! NOT HITTING THE ~~GROUND~~ DOCK IS HARD"

-Fortune Cookie

Rushat Gupta Chadha **Keerthana Manivannan** **Bishwamoy Sinha Roy**

Aishanou Osha Rait **Paul M. Calhoun**

OVERVIEW

- Project Description
- Use Case
- System-level Requirements
- Functional Architecture
- Cyber-physical Architecture
- Current System Status
- Project Management
- Conclusions

PROJECT DESCRIPTION

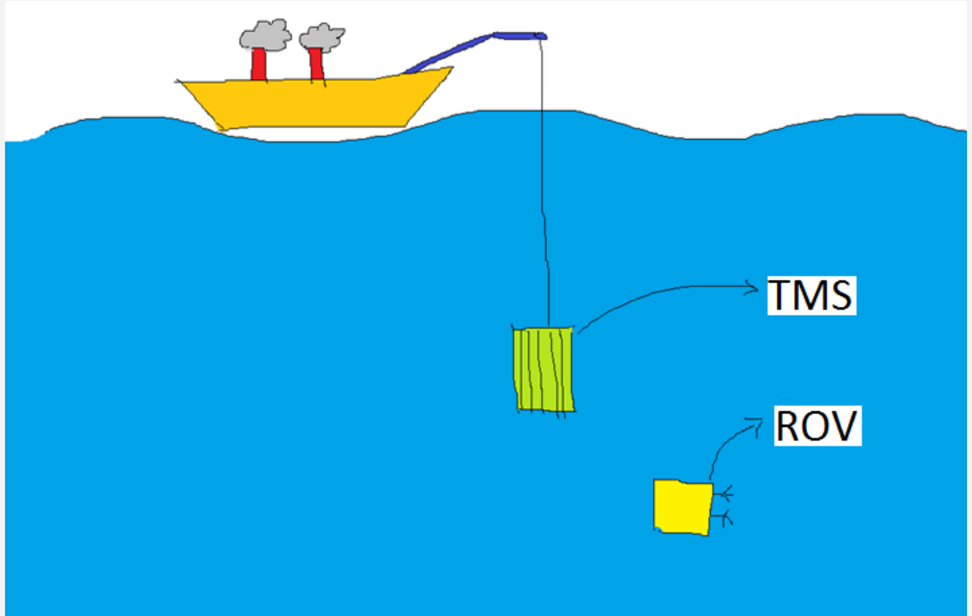
PRESENTED BY: RUSHAT

PROJECT DESCRIPTION

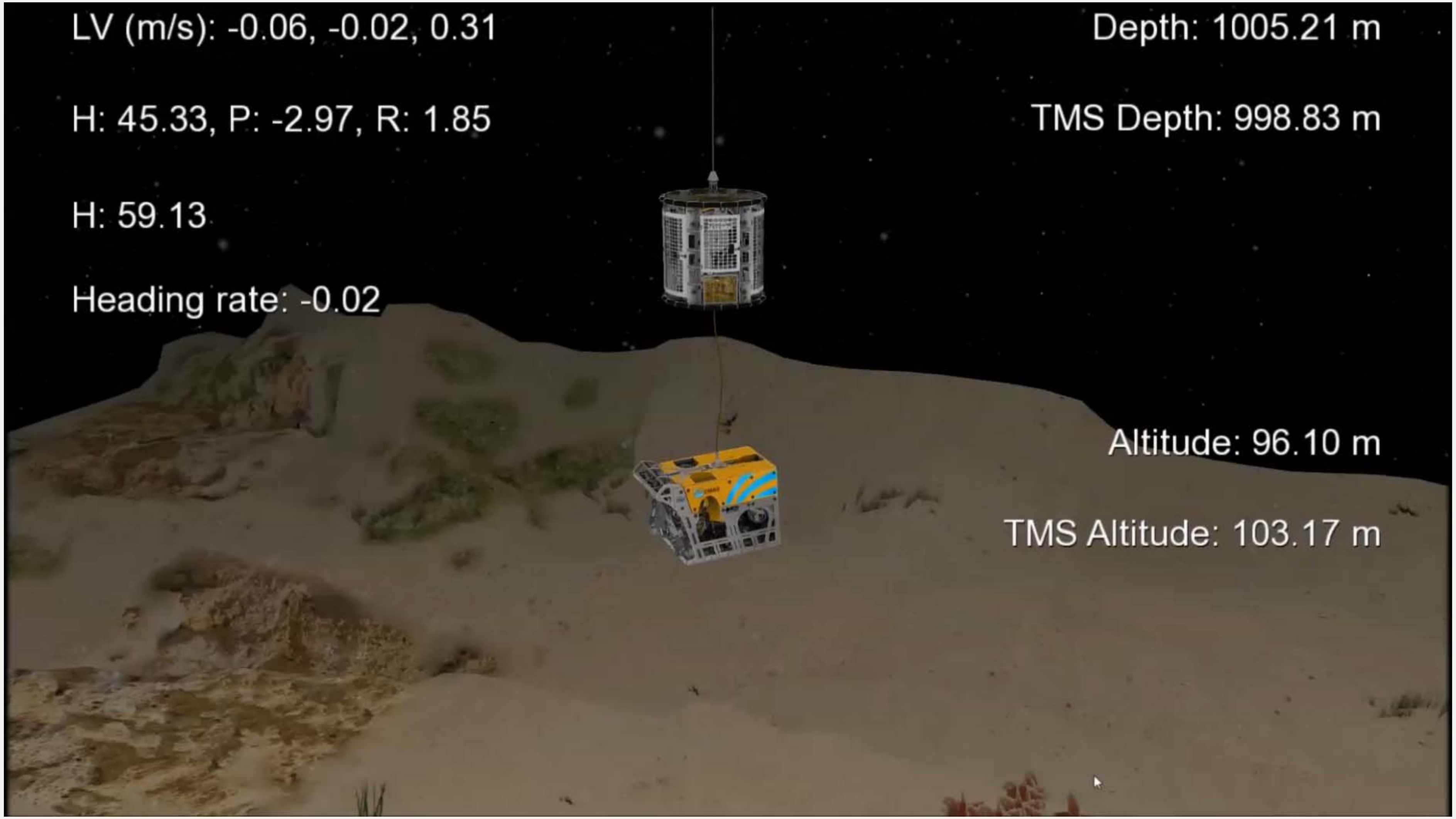
- Aim to provide aerial solution to a real world subsea problem at Schilling Robotics
- Main challenge to dock the ROV to the moving TMS
- Solution Scope– Autonomous docking

ANALOGY

Tether Management System -> Docking Platform
Remote Operated Vehicle -> Quadcopter
Underwater -> GPS degraded



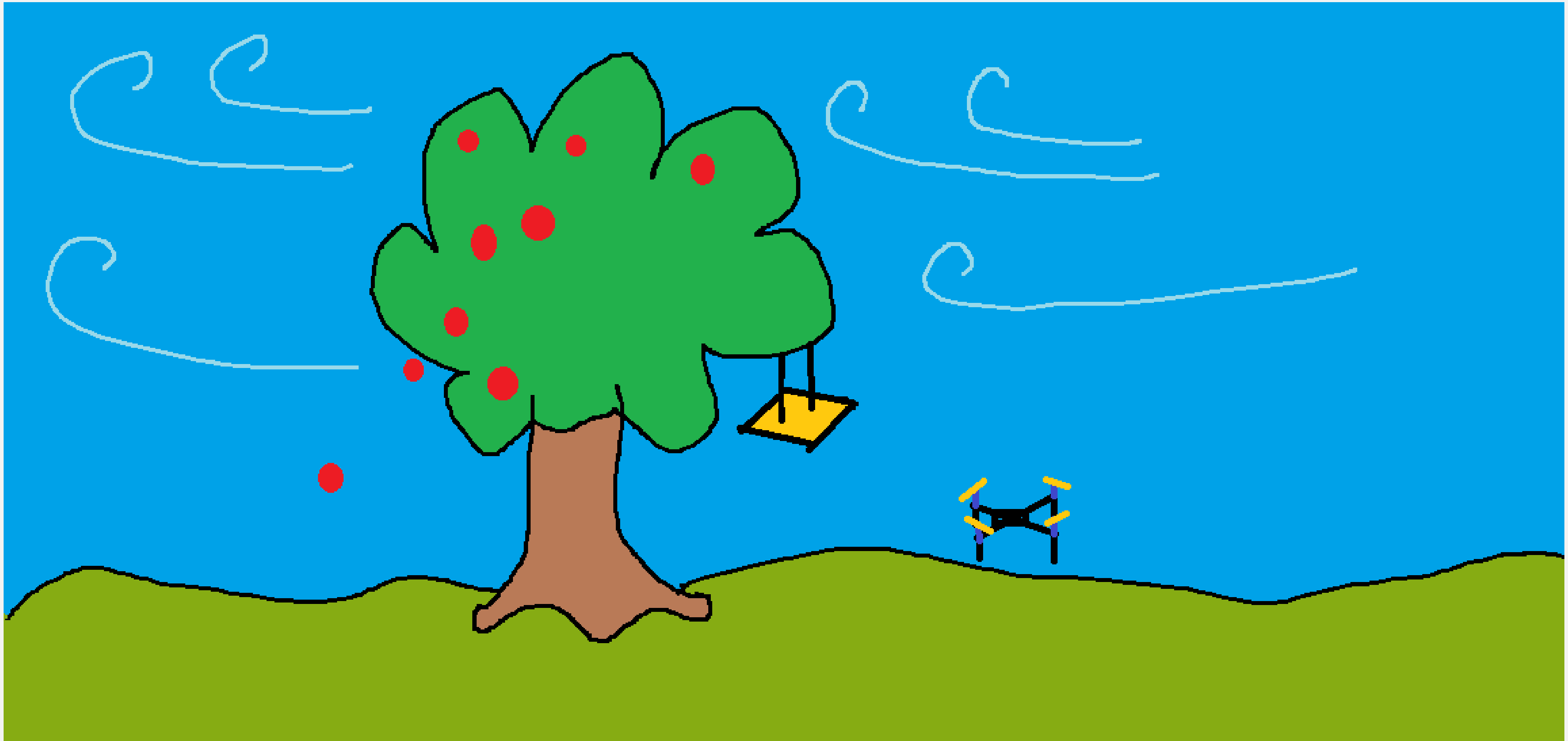
PROJECT DESCRIPTION - VIDEO



USE CASE

PRESENTED BY: PAUL

- Schilling employee finds a retrofit kit in a technology fair
- Thinks it might solve to their ROV-TMS problem



- Mounts the kit on the platform and quadcopter – Docks satisfactorily
- Doesn't dock on a windy day – Declares unsafe to dock

- Successful underwater tests
- Schilling decides to place bulk order to CMU team

SYSTEM LEVEL REQUIREMENTS

PRESENTED BY: KEERTHANA

MANDATORY FUNCTIONAL REQUIREMENTS

- The system shall –
  - MF1. Have two major components: quadcopter and a moving docking platform
  - MF2. Detect and communicate when docking is not possible
- The docking platform shall –
  - MF1.1 Be moving until the quadcopter has been docked
  - MF1.2 Withstand the weight of the quadcopter once it has been docked
- The quadcopter shall –
  - MF2.1 Localize itself w.r.t. the docking platform
  - MF2.2 Plan a path to the docking platform
  - MF2.3 Navigate to the platform
  - MF2.4 Dock to the platform without any collision

LEGEND
Blue: Addressed in Fall

MANDATORY NON FUNCTIONAL REQUIREMENTS

- The system shall:
  - MNF1. Function in a GPS degraded environment
  - MNF2. Provide a user interface with DOCK option and current status
- The quadcopter shall:
  - MNF2.1 Have a payload capacity of > 500g

MANDATORY PERFORMANCE REQUIREMENTS

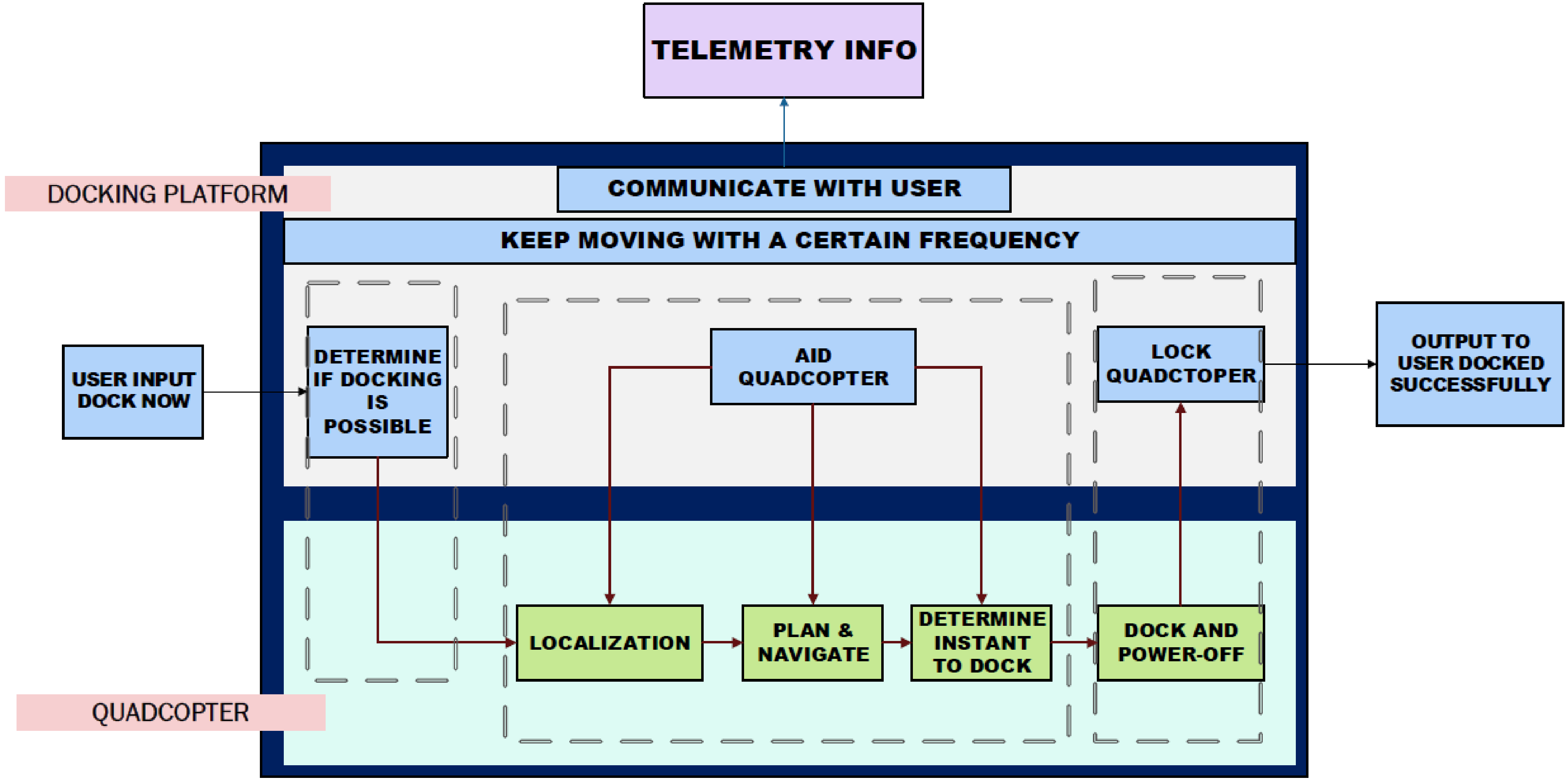
- The docking platform will-
  - MP1.1. Have 1 degree of freedom along Z-direction
  - MP1.2. Oscillate in harmonic motion with dominant frequency < 0.3Hz
  - MP1.3. Have oscillations' span ±200mm
  - MP1.4. Have a locking mechanism which supports weight of 5kg
- The quadcopter will –
  - MP2.1. Localize w.r.t. platform within +/-50 cm (previously 50mm) accuracy
    - Significant drift indoors
    - Change in design of locking mechanism
  - MP2.2. Navigate to the platform within 10 minutes
  - MP2.3. Dock to the platform autonomously and without colliding within 10 minutes

LEGEND
Red: Modified

FUNCTIONAL ARCHITECTURE

PRESENTED BY: KEERTHANA

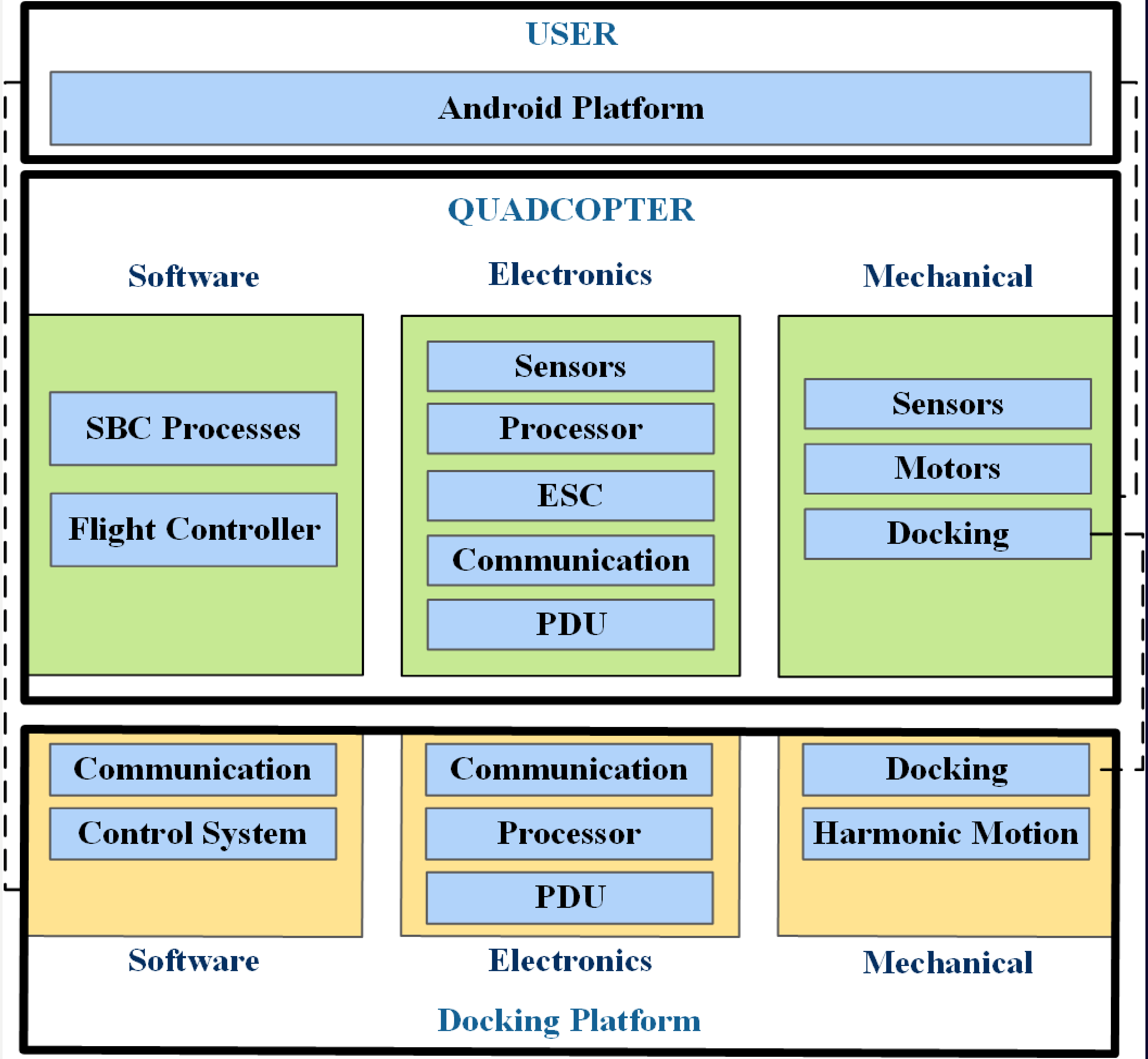
FUNCTIONAL ARCHITECTURE



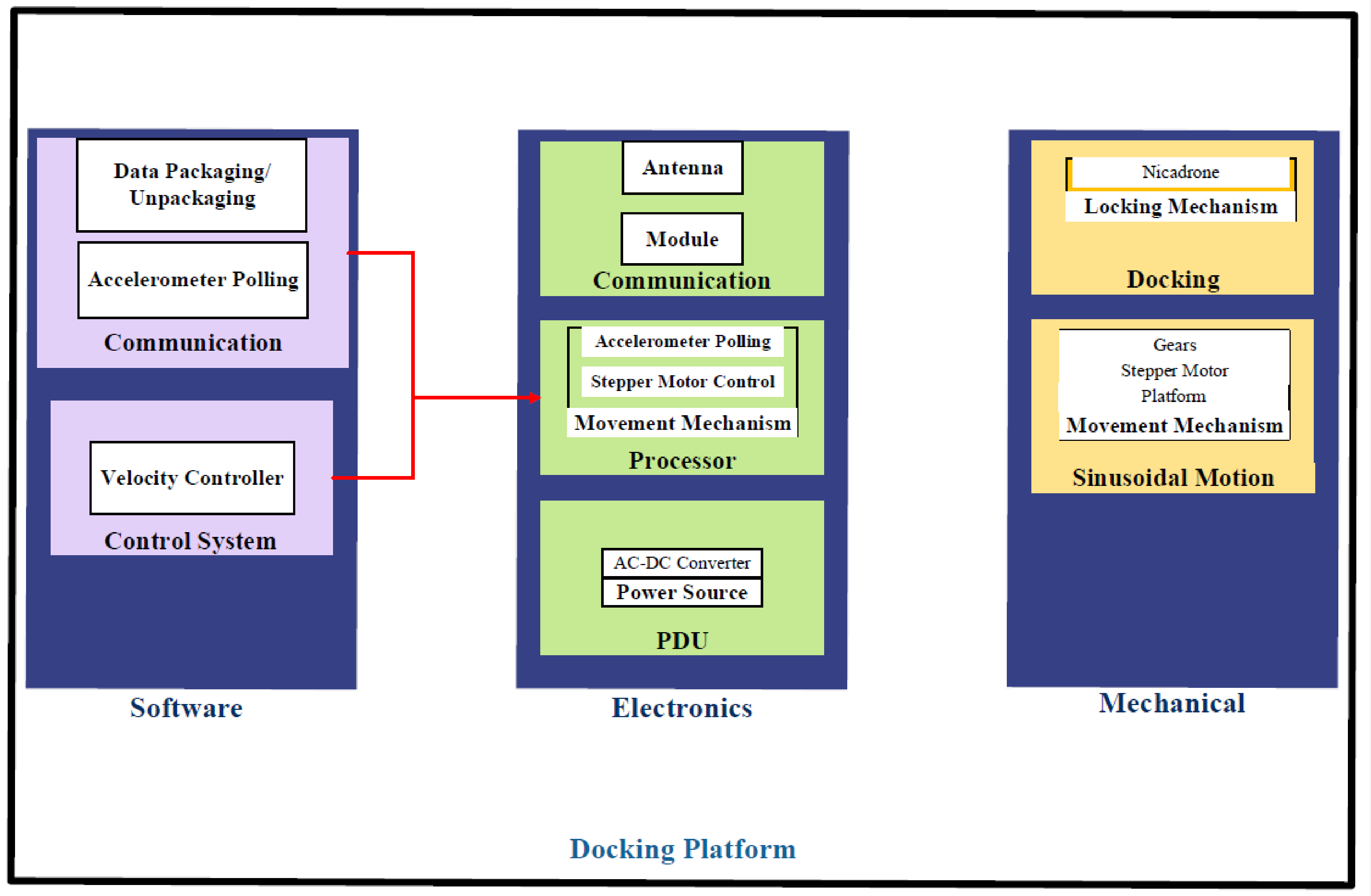
CYBER-PHYSICAL ARCHITECTURE (CPA)

PRESENTED BY: RUSHAT AND ROY

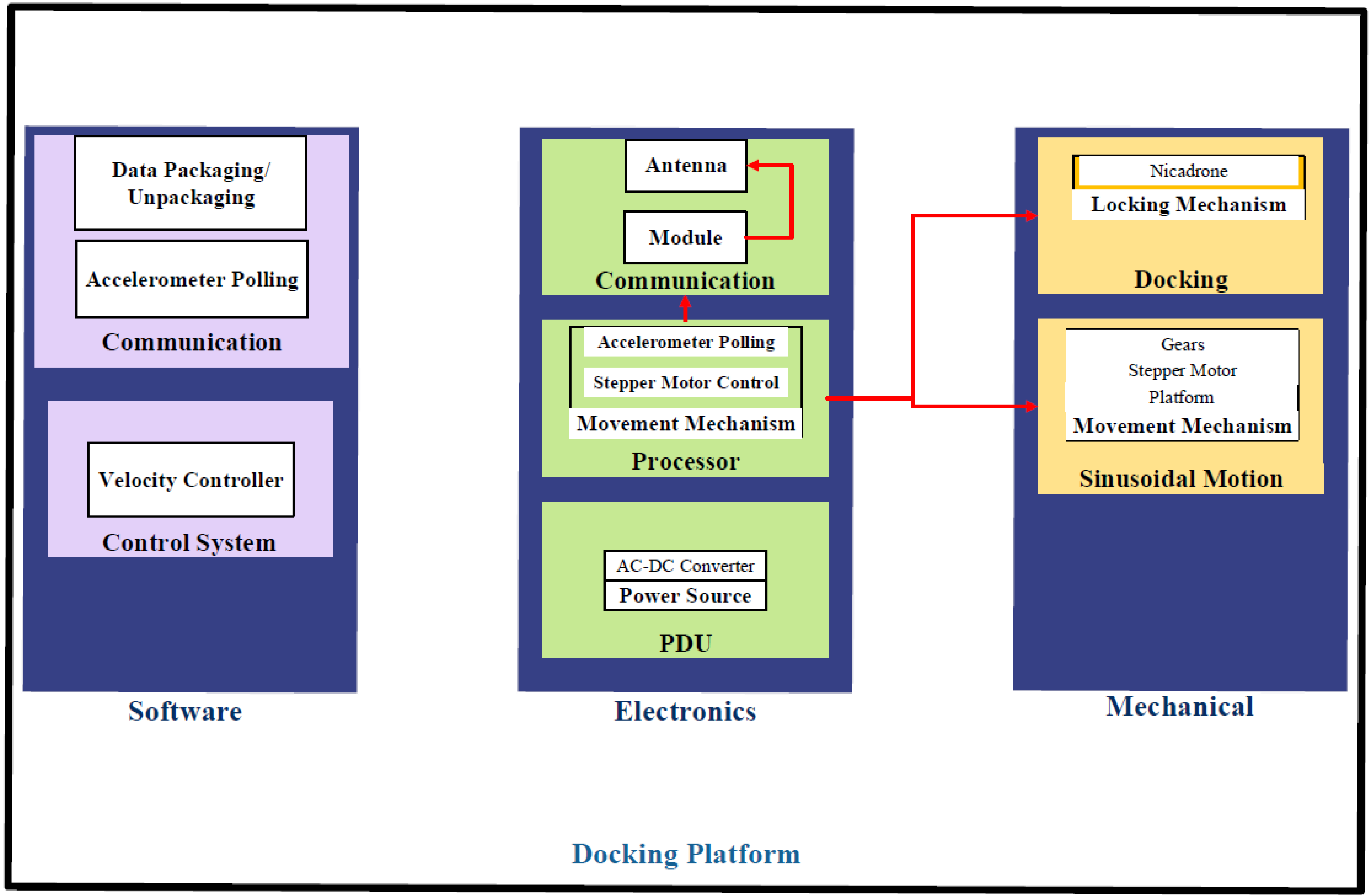
OVERVIEW


USER
Android Platform
QUADCOPTER
Software
Electronics
Mechanical
SBC Processes
Flight Controller
Sensors
Processor
ESC
Communication
PDU
Sensors
Motors
Docking
Communication
Control System
Communication
Processor
PDU
Docking
Harmonic Motion
Software
Electronics
Mechanical
Docking Platform

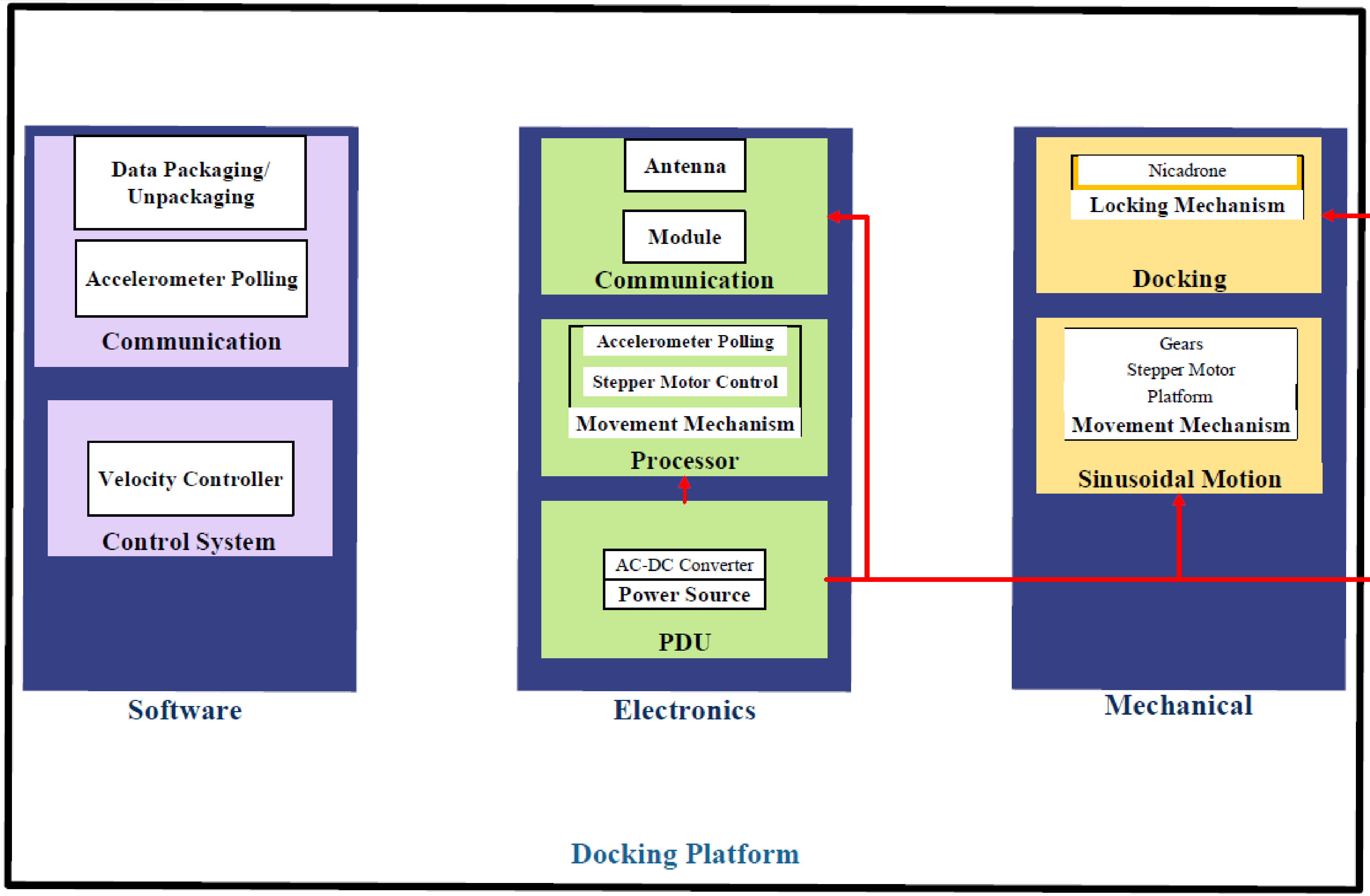
CPA DOCK- CODE



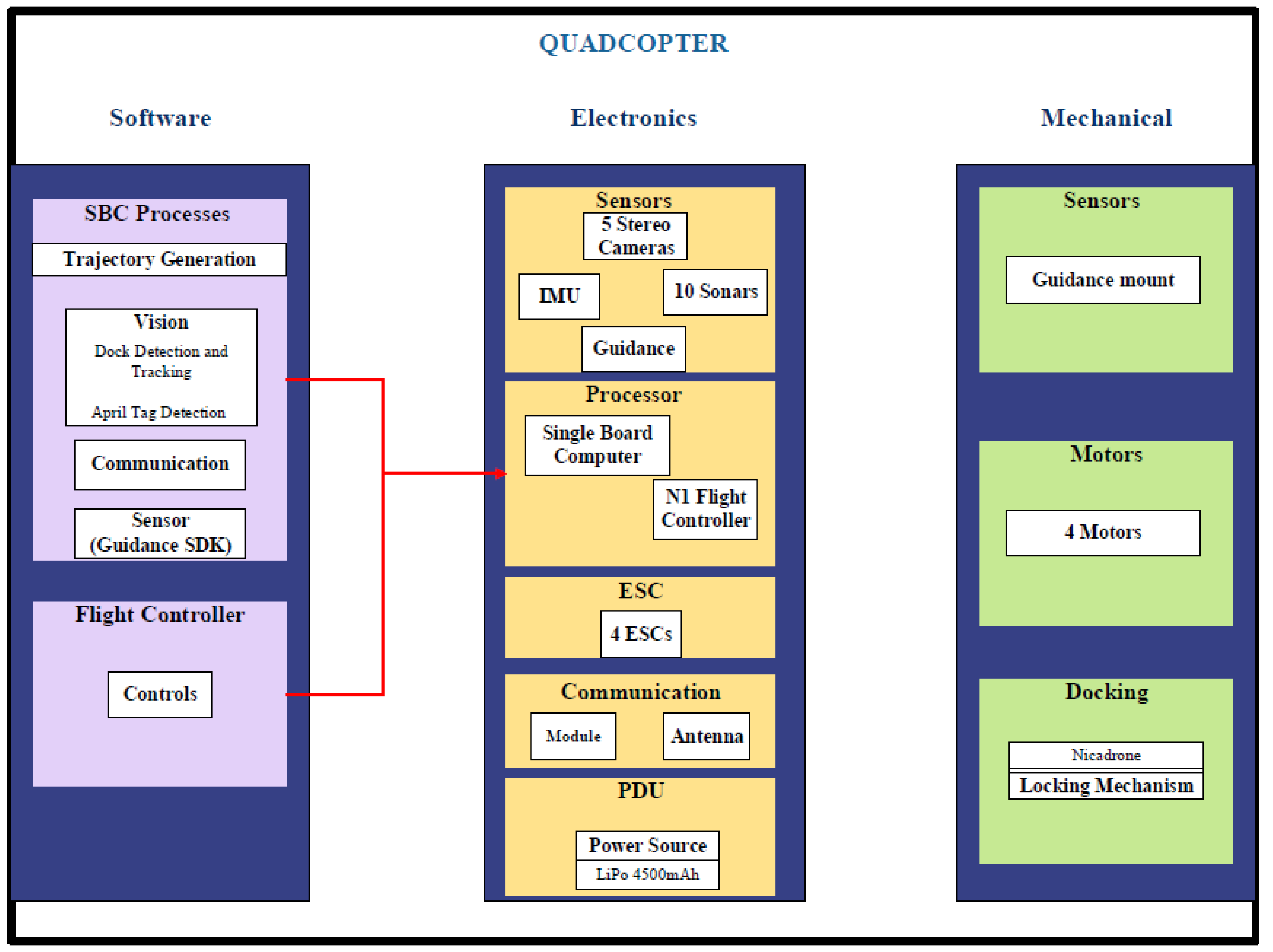
CPA DOCK- DATA



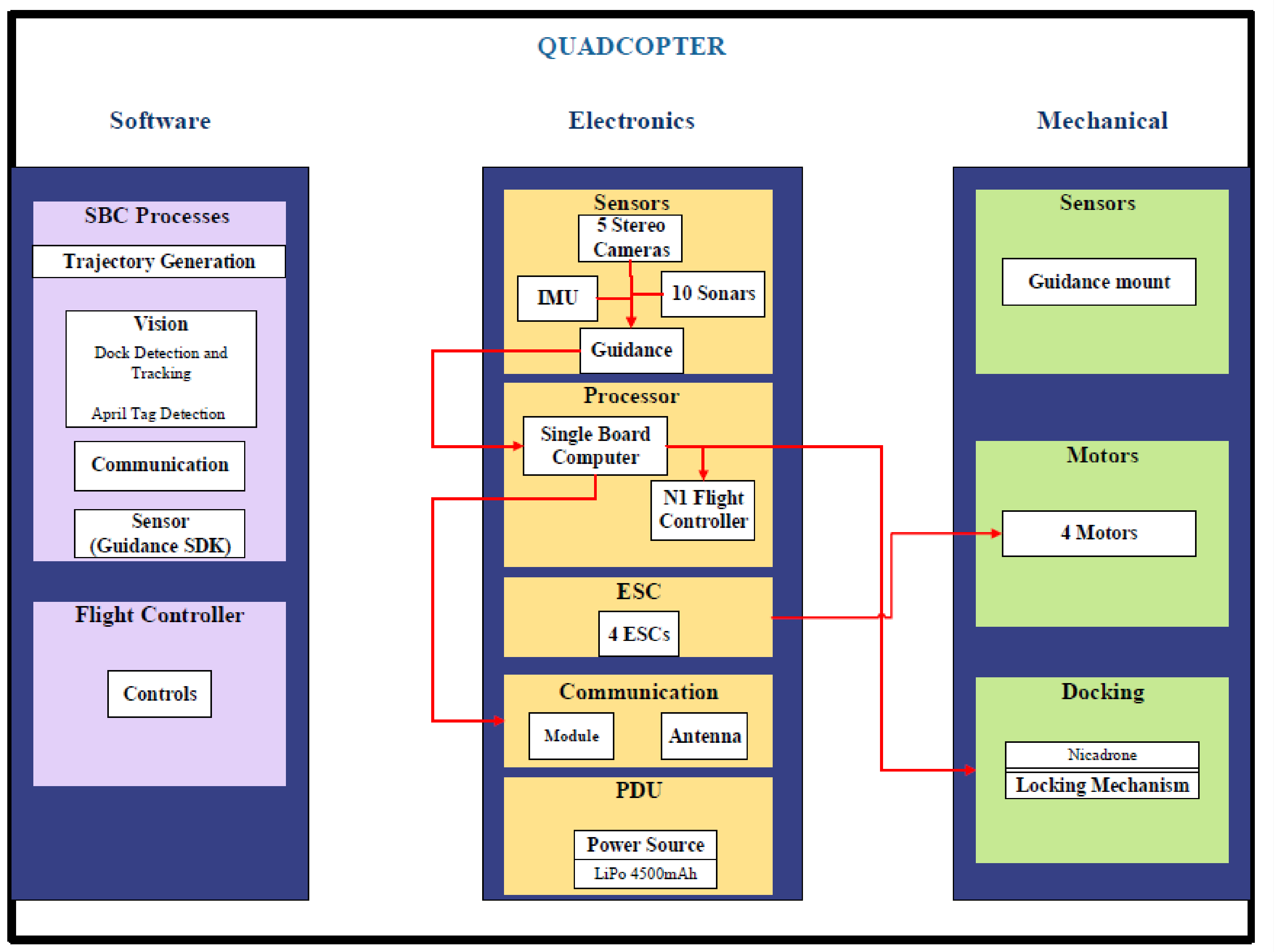
CPA DOCK- POWER



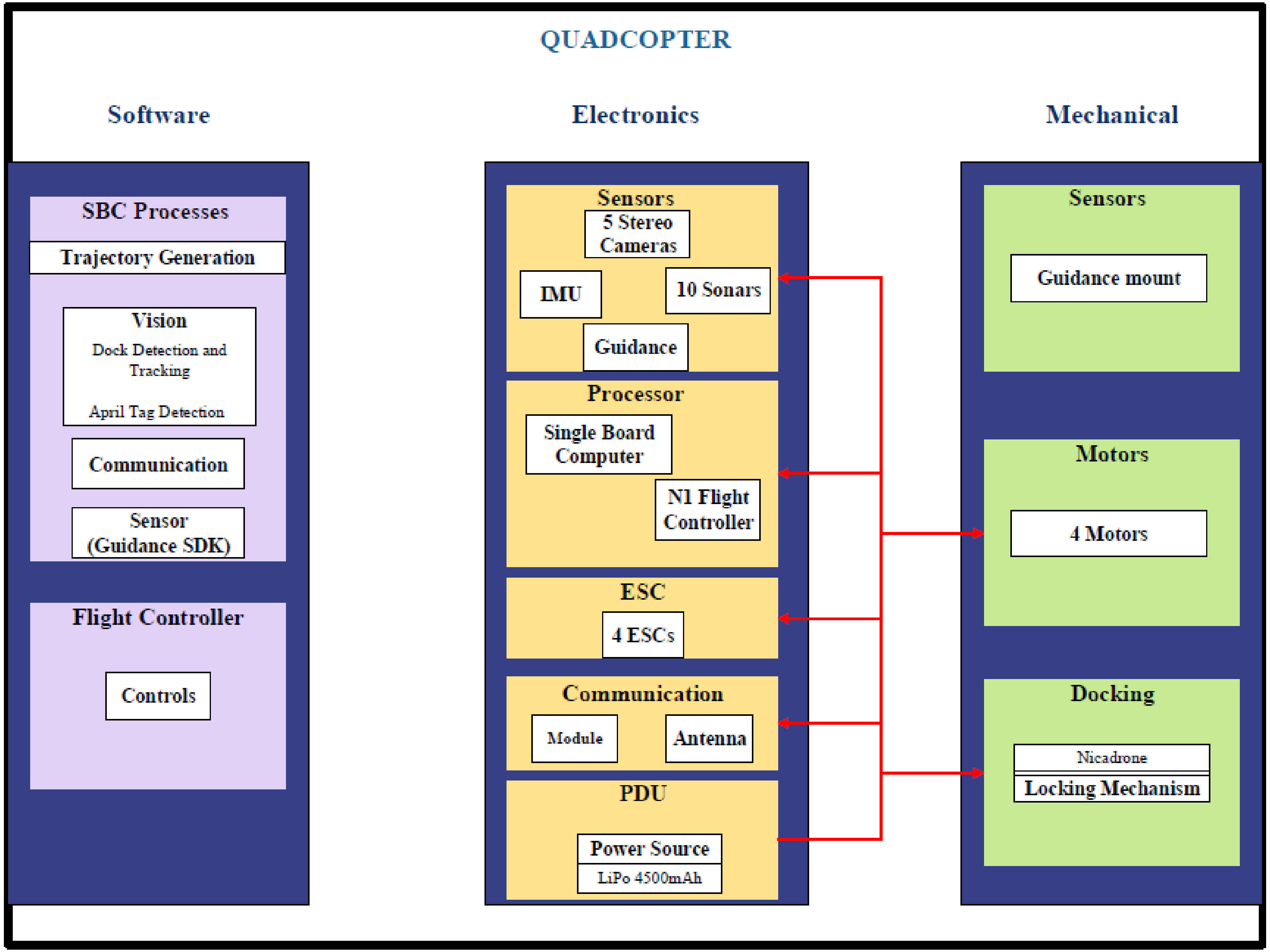
CPA QUADCOPTER - CODE



CPA QUADCOPTER - DATA



CPA QUADCOPTER - POWER



CURRENT SYSTEM STATUS

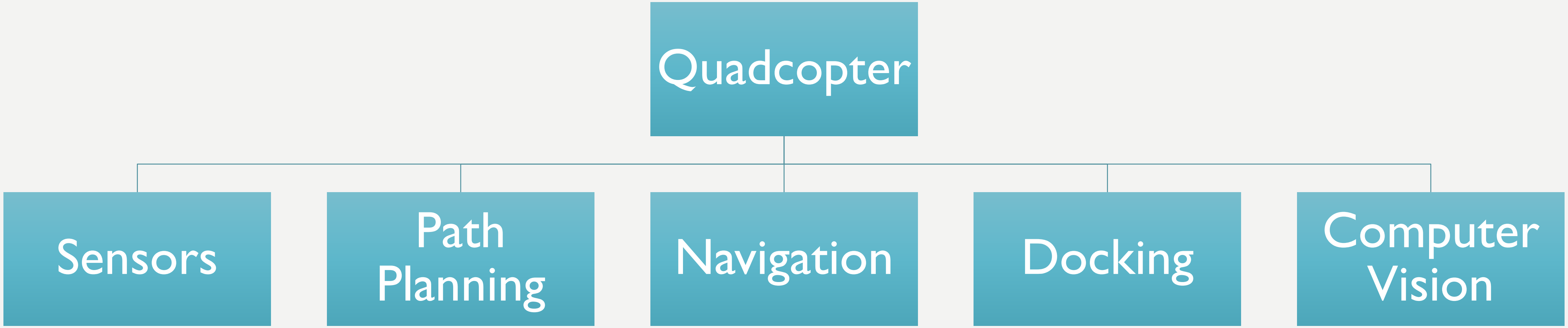
PRESENTED BY: ROY, RUSHAT, AISHANOU

TARGETED REQUIREMENTS

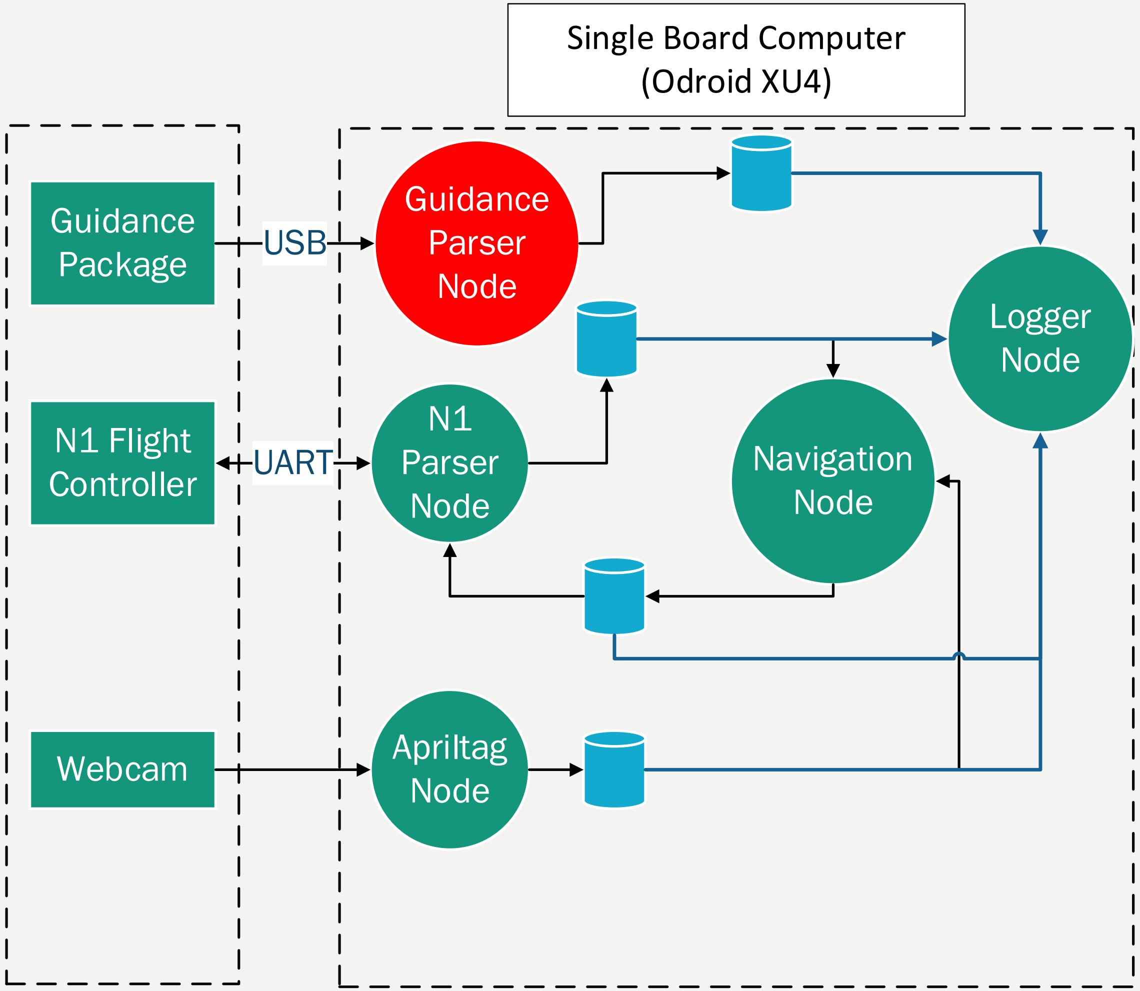
The Platform will:	**The Quadcopter will:**
Have 1 degree of freedom - Z-direction	Localize w.r.t. platform within 50cm accuracy
Oscillate at in harmonic motion with dominant frequency < 0.3 Hz	Navigate to the platform within 10 minutes
Have oscillations' amplitude <= ±200mm	Dock to the platform autonomously and safely within 10 minutes
Have a locking mechanism which supports weight of at least 5 kg	Get pose w.r.t. the platform within 3 m

LEGEND
Green: Completed in Fall
Blue: Addressed in Fall

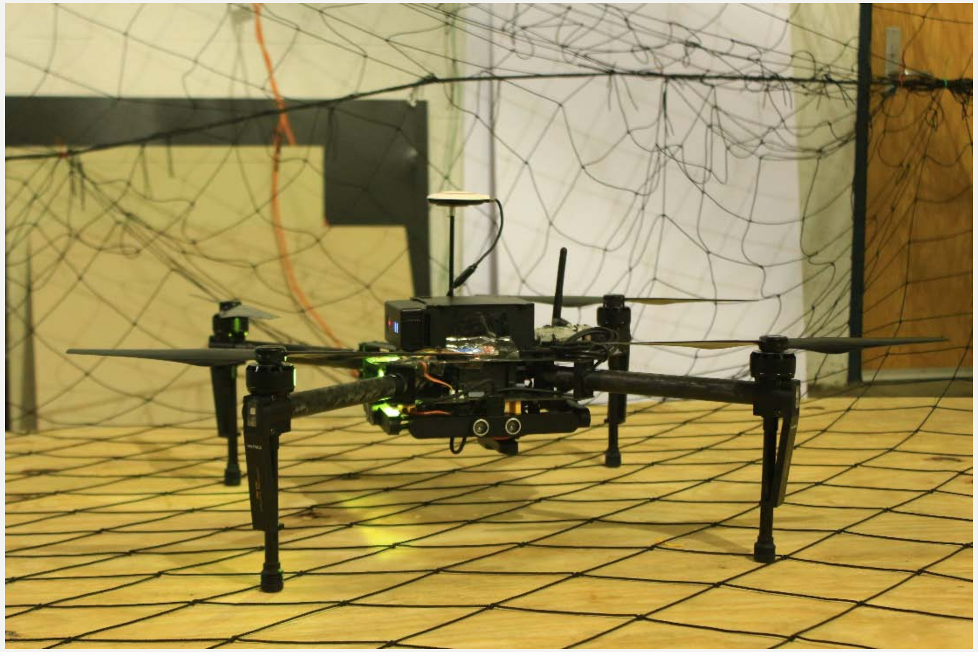
QUADCOPTER - OVERVIEW



QUADCOPTER OVERALL SYSTEM DESCRIPTION

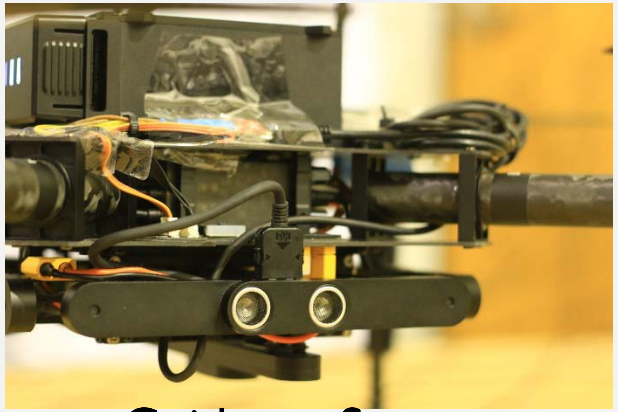


RED – Not Done
GREEN - Completed

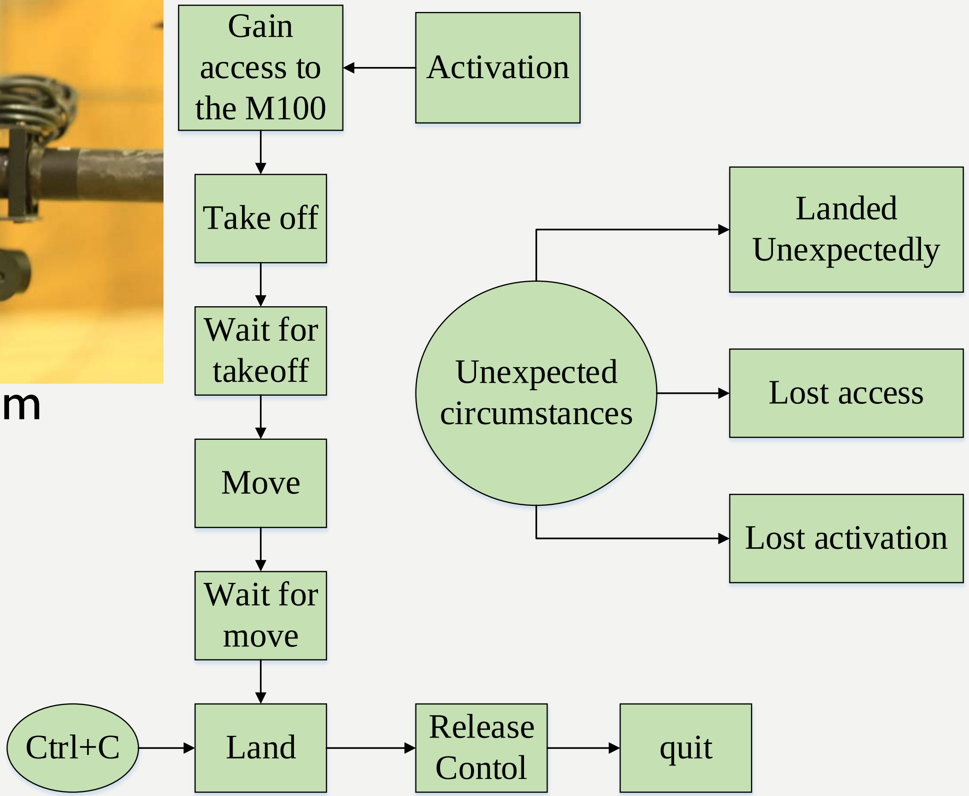


QUADCOPTER SUBSYSTEM DESCRIPTION - STATUS

- Complete
  - Stable Hover
  - Manual Safety Override
  - Logging Data
  - AprilTag Detection
  - Computer Vision Integrated Point A to B in Simulation
- To-do
  - Point A to B in reality
  - Localization with respect to Dock
  - Stabilizing under Dock
  - Rising up to meet the docking platform

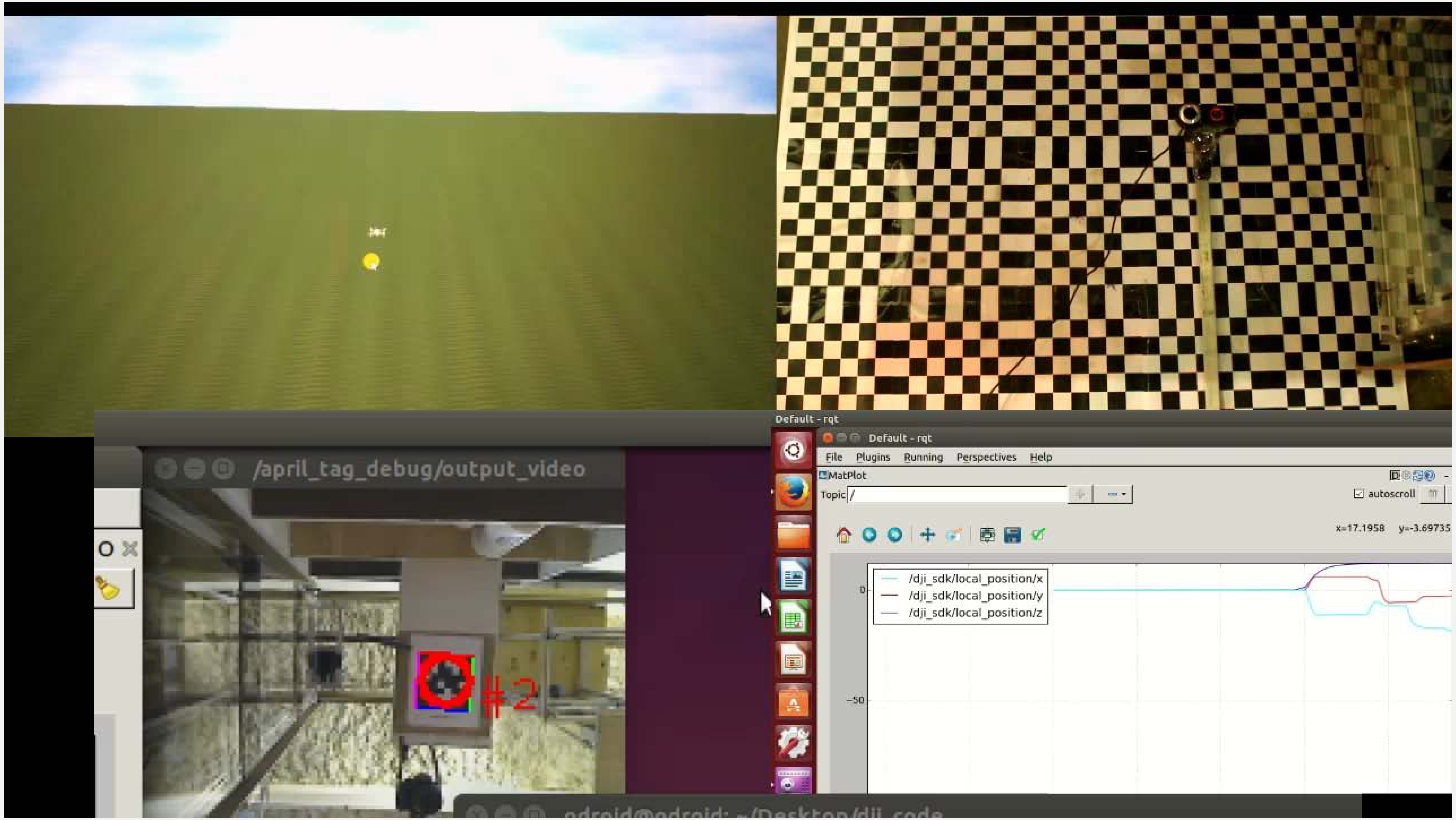


Guidance System

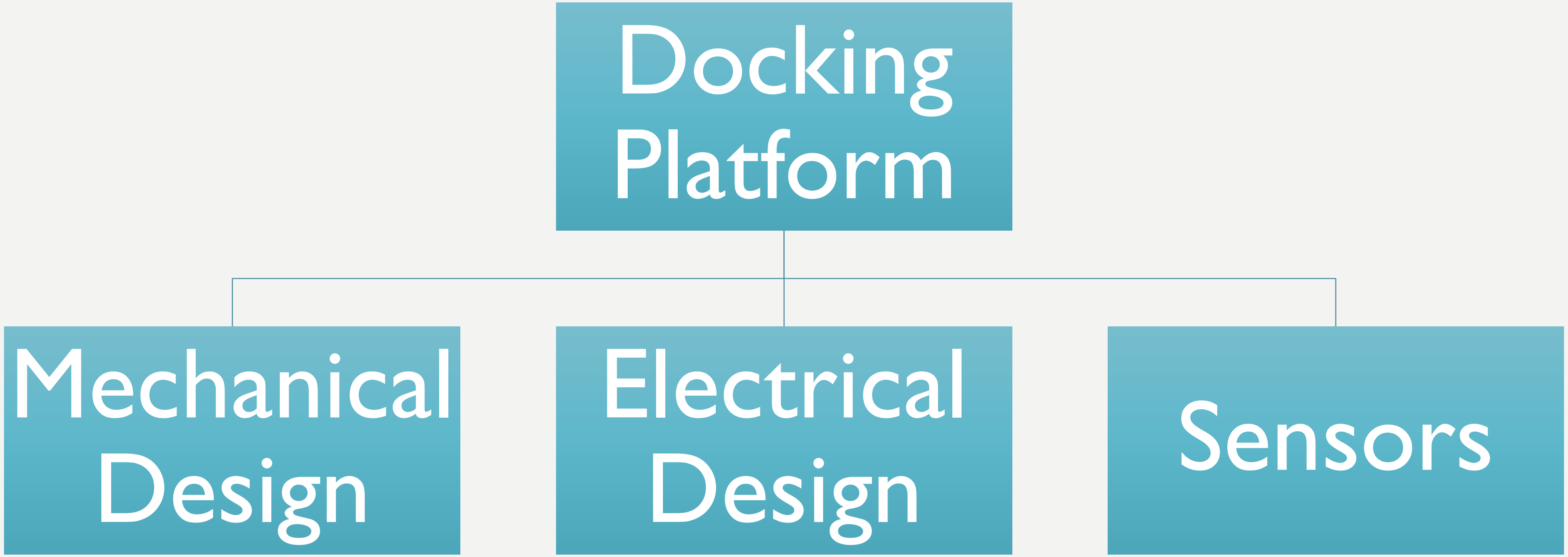


Current Quadcopter State Machine

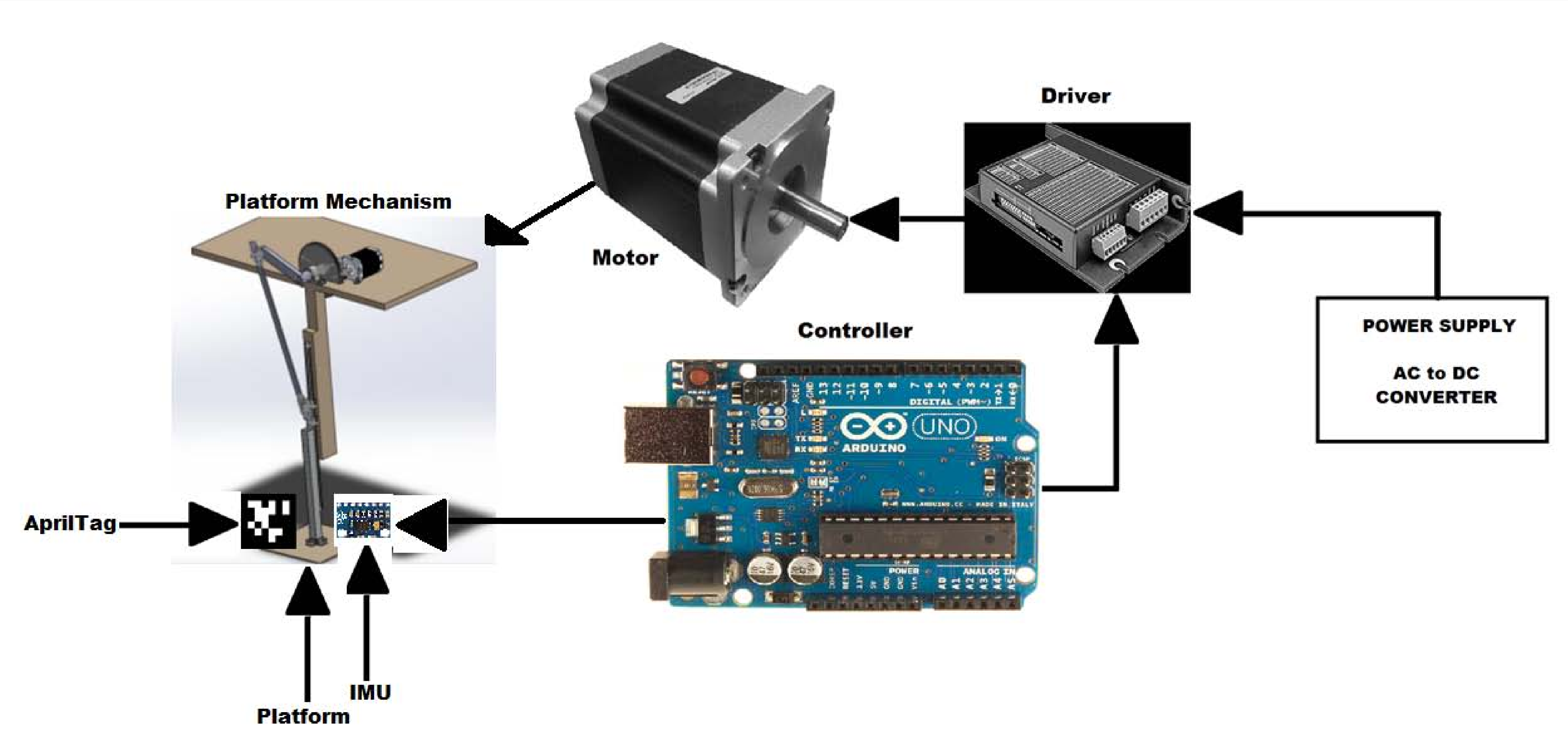
QUADCOPTER SUBSYSTEM DESCRIPTION - VIDEO



DOCKING PLATFORM - OVERVIEW



DOCK OVERALL SYSTEM DESCRIPTION



DOCK SUBSYSTEM DESCRIPTION - STATUS

- Complete
  - Dock Fabrication
  - Motor Control
  - Sensor Polling

To-do

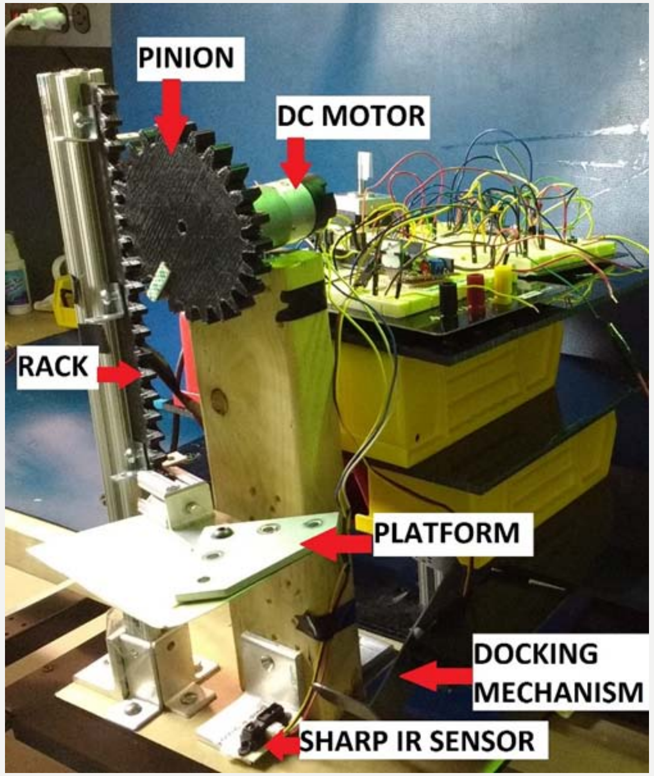
- Attach dock to a suitable fixture
- Reduce error in motor control
- Platform (metal sheet)
- Relay information to Quadcopter

MODELLING & ANALYSIS – DOCKING PLATFORM

- Modelling Mechanical Design in SolidWorks
- Mechanical Analysis by prototyping
- Functional Analysis via IMU data

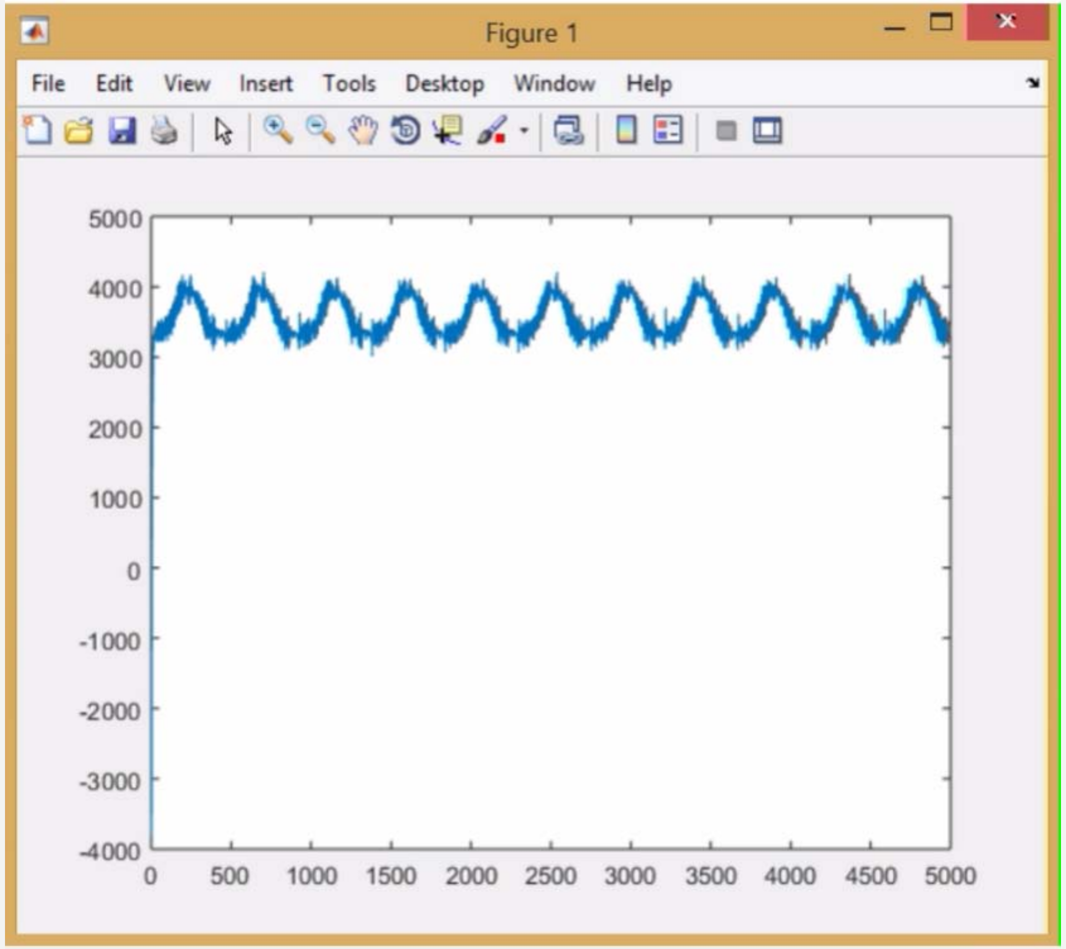
MODELLING & ANALYSIS – DOCKING PLATFORM

- Modelling Mechanical Design in SolidWorks
- Mechanical Analysis by prototyping
- Functional Analysis via IMU data



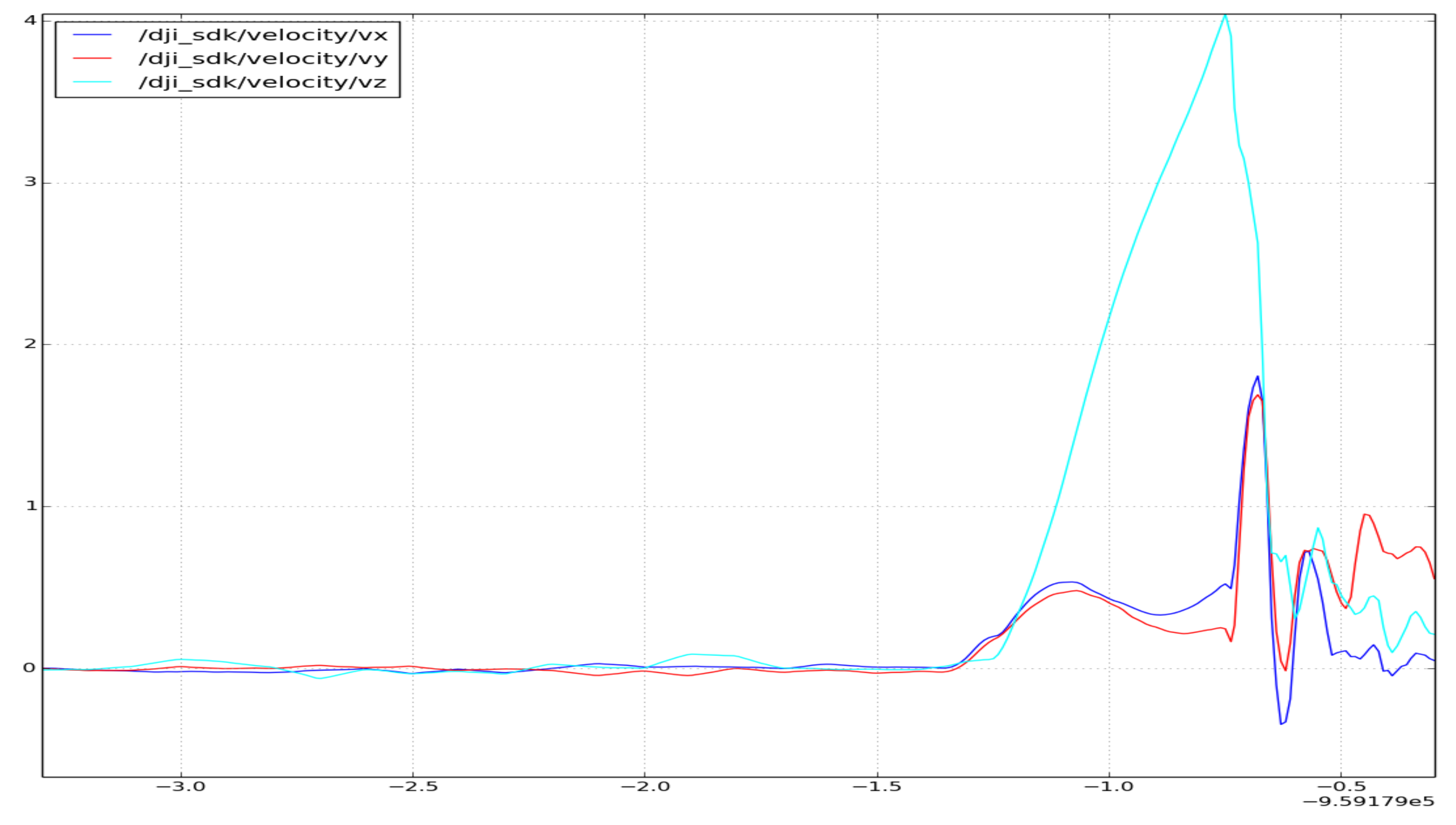
MODELLING & ANALYSIS – DOCKING PLATFORM

- Modelling Mechanical Design in SolidWorks
- Mechanical Analysis by prototyping
- Functional Analysis via IMU data



MODELLING & ANALYSIS – QUADCOPTER

- Dynamics modelled in DJI Simulation (Hardware in the Loop)
- Analysis run on log-files collected and stored on Odroid



Velocity Data from the Crash

TESTING

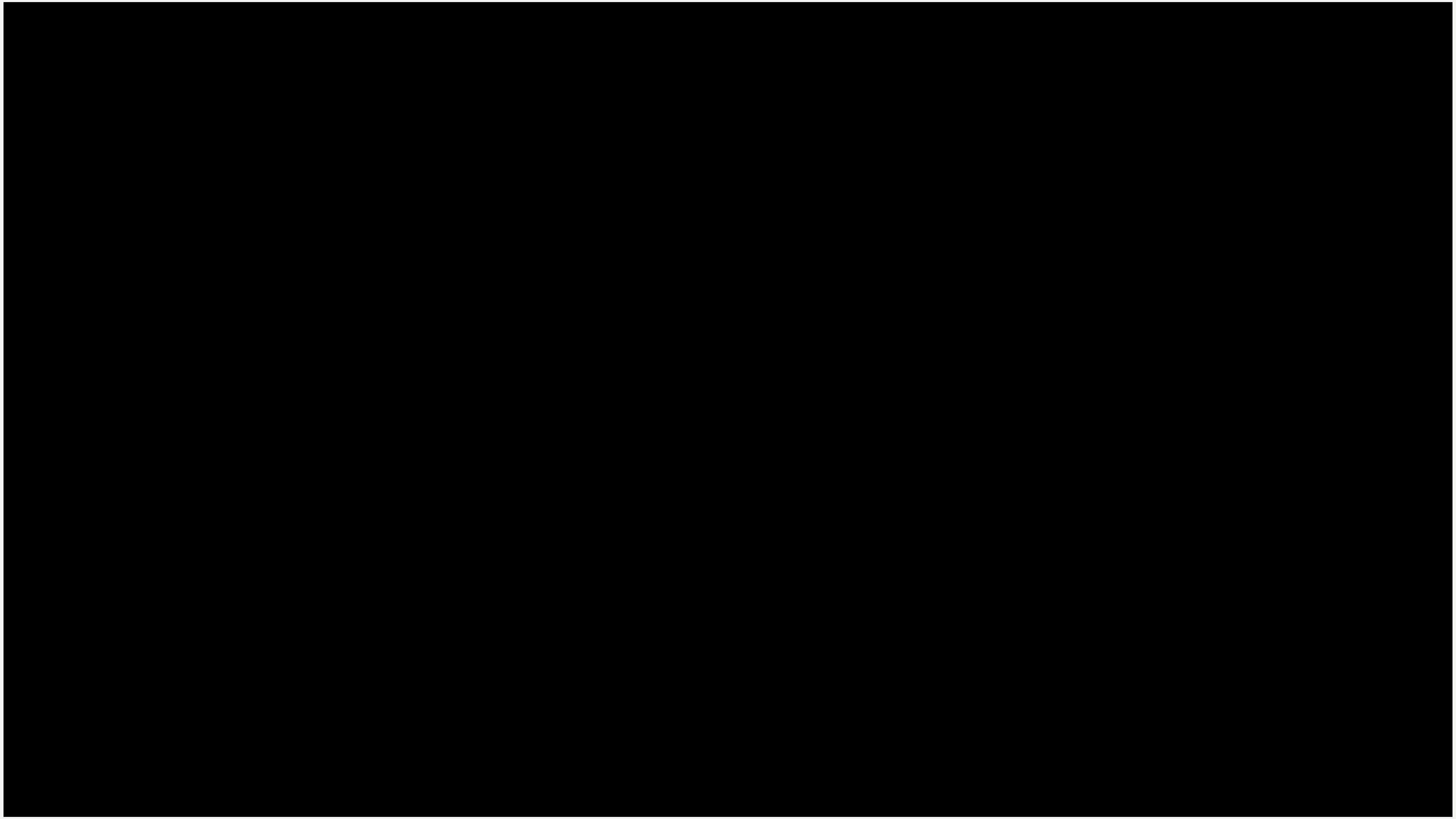
- **Platform** – Frequency input to motor controller -> Frequency detected correctly by sensors -> Frequency match with stop watch readings

- **Quadcopter - Unit Tests**
  1. Computer Vision first on laptop then on Odroid
  2. Quadcopter autonomous hover and land and manual control on simulation then real life.
  3. Quadcopter Velocity control on simulation first using laptop then Odroid
  4. Quadcopter Position control using velocity control on simulation first using laptop then Odroid
  5. **Computer Vision provides distances to the Quadcopter Motion control in simulation via Odroid**

PERFORMANCE EVALUATION - FVE

REQUIREMENTS	EXPECTATIONS	FVE	FVE-ENCORE
MFR1.1	Docking platform shall move according to the given input frequency	Successful	Successful
MFR2	Sensor gathers data from the motion of the platform and outputs the frequency	Successful	Successful
MFR2.1	Quadcopter shall be able to autonomously hover	Successful	Successful
MFR2.2	Quadcopter shall move autonomously from point A to point B	Failure due to lack of spare parts	Failure due to lack of spare parts
MFR2.1	AprilTag should detect the camera and ascertain the distance moved within 5% error	Successful	Successful
MFR2.1	AprilTag should detect the camera moving and therefore make the quadcopter move accordingly in the simulation	Not planned initially	Successful

TEASER



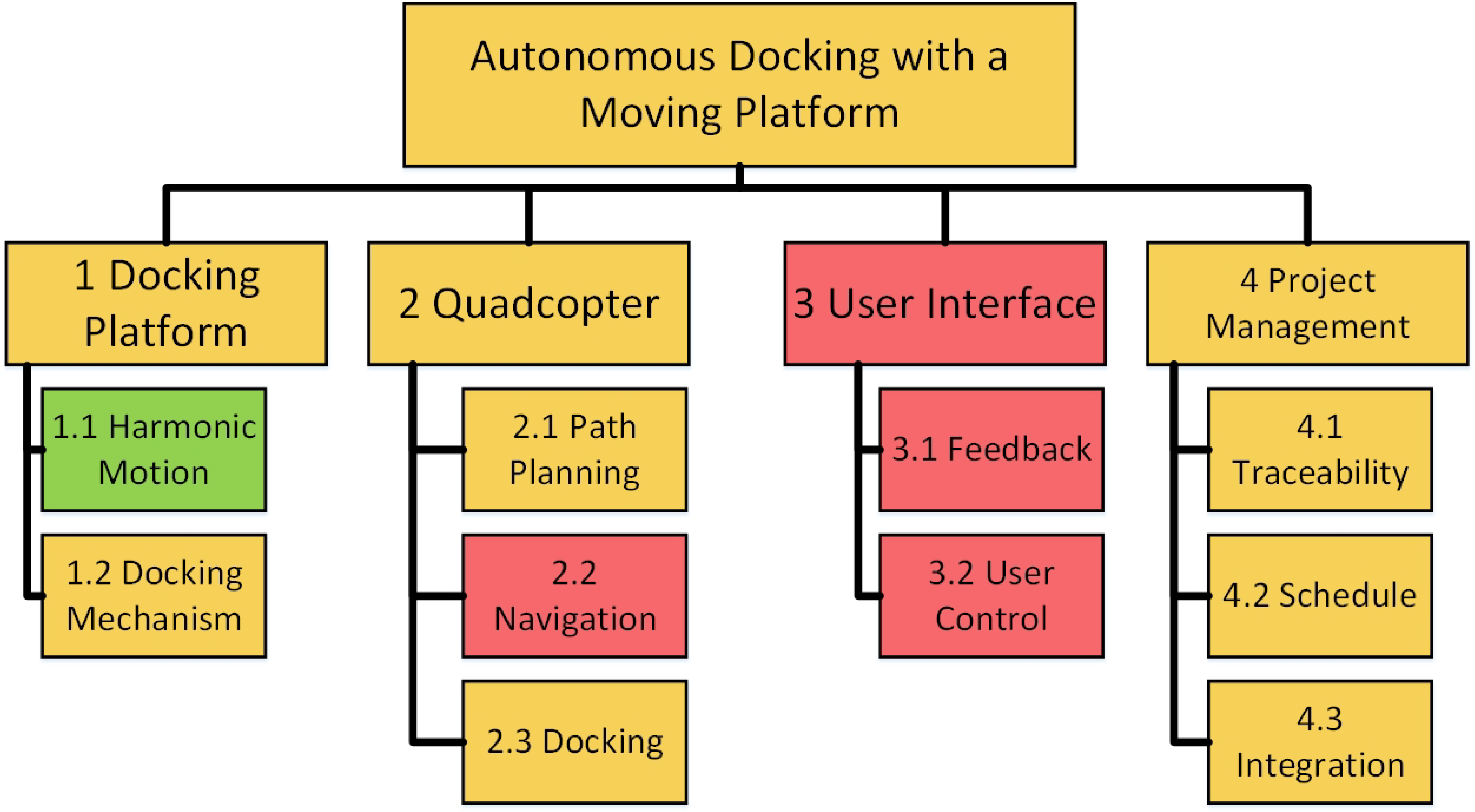
CONCLUSIONS

STRENGTHS	WEAKNESSES
Docking Platform is robust	Velocity control not stable in Matrice
Motor can withstand the desired load	Flight controller code is not accessible
April Tag works suitably even in low lighting	Cannot provide state estimation values to flight controller
IMU is giving accurate readings	Platform is heavy. Not able to find place to mount it permanently
Indoor hovering is stable using guidance	

PROJECT MANAGEMENT

PRESENTED BY: PAUL

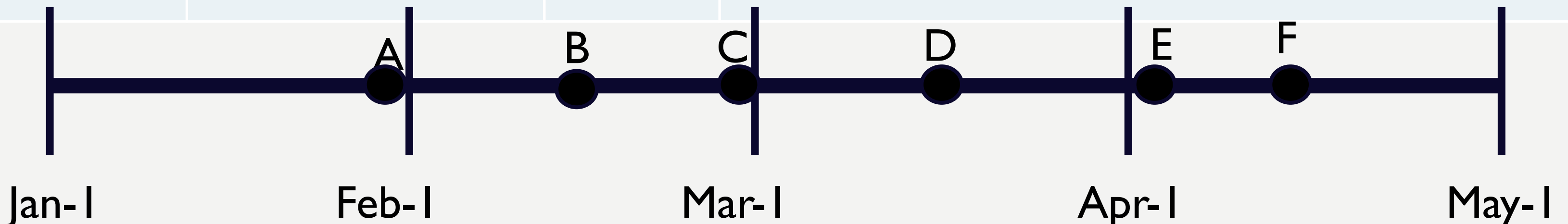
WORK BREAKDOWN STRUCTURE



SCHEDULE STATUS

Holding us behind schedule

Timeline	Progress Review	Label	Milestone
Late January	PR 7	A	Quadcopter motion from Point A to B
Mid-February	PR 8	B	• Determine position and velocity of platform using CV and sensors
Late February	PR 9	C	• Quadcopter localization with respect to the platform • Stabilization of Quadcopter under the Platform • Docking to the platform with the Nicadrone
Mid-March	PR 10	D	• Achieve docking on moving platform • UI integration
Early April	PR 11	E	Testing and refinement
Mid-April	PR 12	F	Testing and refinement

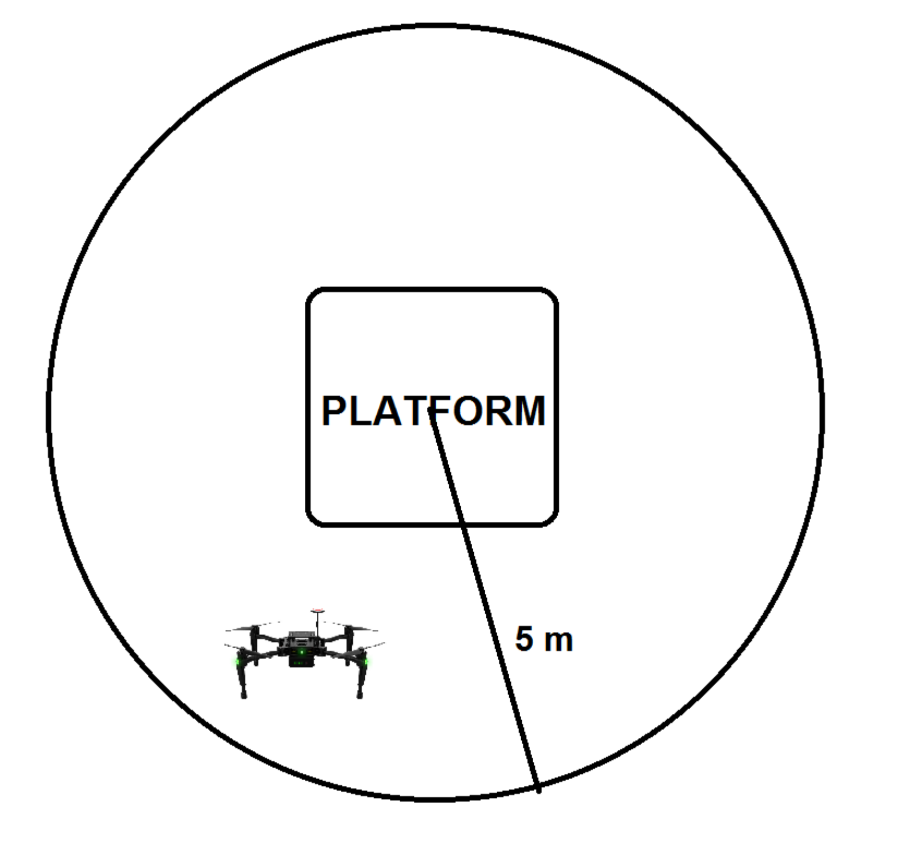


SPRING TEST PLANS

PR #	Timeline	Capability Milestones
PR 7	Late January	Autonomous navigation of quadcopter from point A to B within 5 m radius
PR 8	Mid-February	Robust estimation of position and velocity of platform using CV and sensors
PR 9	Late February	• System Integration - Quadcopter docks to stationary platform • User Interface designed and communicates with the quad & platform
PR 10	Mid-March	• Achieve docking on moving platform • UI receives status as requested by the user
PR 11	Early April	Testing and Refinement
PR 12	Mid-April	Testing and Refinement

SPRING VALIDATION EXPERIMENT

Where?	NSH B Level
Equipment used	DJI Matrice100, Guidance, Designed Platform, Power Supply, Smartphone
What will we show?	• The right instant to dock is decided based on the motion of the platform • Quadcopter docks to platform without collision

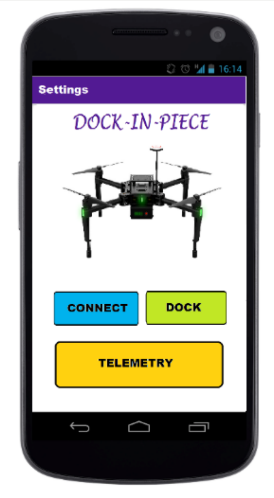


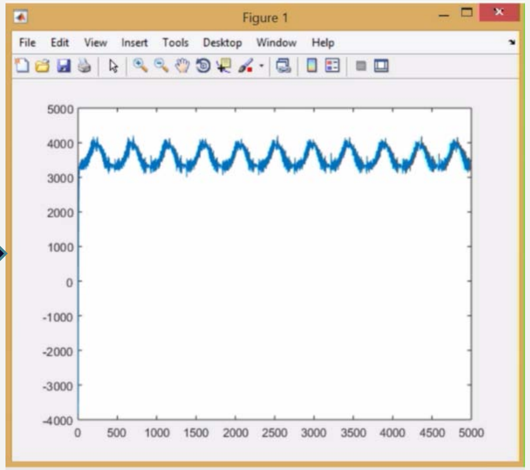
SPRING VALIDATION EXPERIMENT

PLATFORM :

Procedures and test metrics

- Turn on the power to platform
- Enter frequency for platform motion from user interface (Range 0.15 to 0.3 Hz)
- The frequency of platform motion changes as desired
- Motion is detected by sensors and graph plotted showing desired waveform
- The frequency detected by the sensors will be same as input value





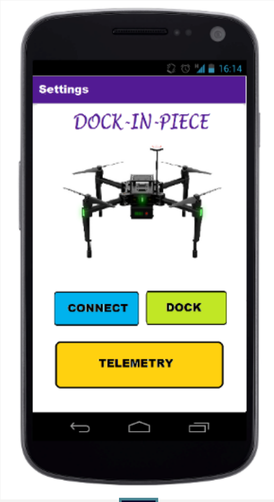
SPRING VALIDATION EXPERIMENT

QUADCOPTER :

Procedures and Test metrics
• Place the quadcopter on the ground within 5 m from the platform and turn on the power • Initiate docking operation from the UI & view flight status • The quadcopter will take off and search for the platform • The quadcopter will travel horizontally to below the platform • The quadcopter will hover 1m below the platform (within 0.5 m accuracy) to determine safe instant to dock • The quadcopter will dock to the platform without collision within 10 mins from initiation • The velocity of quadcopter w.r.t platform will be less than 50 cm/s when quadcopter is moving up towards the platform • Platform motion would stop and UI will display "DOCK SUCCESSFUL" • Repeat above steps 5 times with different starting positions & different frequencies of platform. Docking should be successful 70% of the times.

SPRING VALIDATION EXPERIMENT

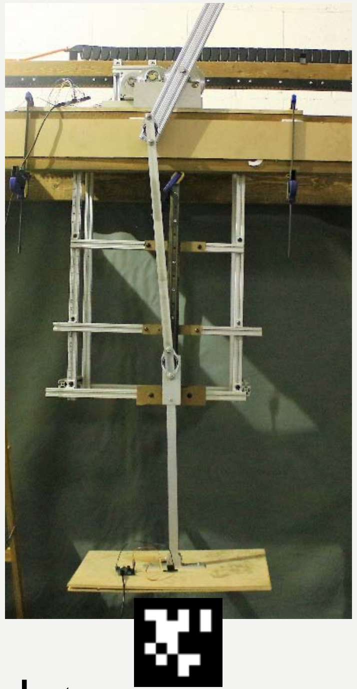
QUADCOPTER :



Initiate Docking

Nicadrone

Approach Dock



Attach to Platform

Stabilize Here

BUDGET STATUS

CMU total budget	$4,000.00	**Sponsor total budget**	$5,000.00
Total Executed from CMU	$1,954.53	**Total Executed from sponsor**	$4,726.00
CMU budget remaining	$2,045.47	**Sponsor budget remaining**	$274.00

Big-Ticket Items – Major Expenditure

Item	Cost	Type	Status	Funding Source	Comment
DJI Matrice 100	$3,299.00	Capital	Executed	Sponsor	Developer Quadcopter
DJI Guidance	$999.00	Capital	Executed	Sponsor	Sensor suite and collision avoidance for quadcopter

RISK MANAGEMENT

Severity

Probability		A Negligible	B Low	C Moderate	D Severe	E Catastrophic
5	Nearly Certain	0	0	0	1	0
4	Likely	0	1	0	5	0
3	Possible	0	1	2	1	0
2	Unlikely	0	1	1	0	0
1	Rare	0	1	0	1	3

Immediate Action
Urgent Action
Action
Monitor
No Action

RISK TABLE – TOP 10

ID	Risk	Likelihood	Severity	Requirement Impacted	Consequence	Mitigation	Risk Type	Action to Take
6	Quadcopter collision avoidance fails in flight	4	D	F2.1-4 , MP2.1-3	Damage to quadcopter	Keep Guidance On	Physical	Immediate Action
7	Quadcopter attempts to shut down engines after a false positive dock	1	D	F2.1-4 , MP2.1-3	Damage to quadcopter	Place net under platform	Physical	Monitor
8	Delays in shipping	4	D	ALL	Subsystems lack parts to be complete	Order in advance	Schedule	Immediate Action
13	Navigation algorithm more difficult than planned	3	C	F2.1-4 , MP2.1-3	Quadcopter navigation subsystem not completed on schedule	Keep in contact with other CMU developers	Schedule	Action
14	Indoor flight impossible	1	E	F2.1-4 , MP2.1-3	Cannot satisfy key requirements		Schedule	Monitor

RISK TABLE – TOP 10

ID	Risk	Likelihood	Severity	Requirement Impacted	Consequence	Mitigation	Risk Type	Action to Take
15	SDK Legal Issues Continue for significant time	0	0	F2.1-4 , MP2.1-3	Quadcopter subsystems not complete on schedule	Get a personal license	Schedule	No Action
17	Quadcopter Fails to Arrive in time for FVE	0	0	F2.1-4 , MP2.1-3	Demo cannot be completed	Lower expectations	Schedule Program matic	No Action
19	Quadcopter Spares Strategy Insufficient	3	D	F2.1-4 , MP2.1-3	Quadcopter testing delayed	Failure Mode Effects and Criticality Analysis	Schedule	Action
22	Arduino Not Fast Enough to Control Motor	4	B	MP1.2	Dock subsystem will require redesign	User Datagram Protocol Edits to the Driver Setting	Schedule	Action
24	Guidance Fails Midflight	4	D	F2.1-4 , MP2.1-3	Damage to quadcopter	Switch to Manual Control Faster	Physical	Immediate Action

CONCLUSIONS

PRESENTED BY: KEERTHANA

KEY FALL LESSONS

- Properly formulated requirements can minimize subsequent effort
- Stick to requirements – Don’t increase scope
- Trade studies should consider maintenance costs
- Take immediate steps for identified issues – bigger net
- One month margin for delay in delivery may not be enough!
- Carefully consider all the spares required
- Delays in schedule should be addressed
- Leave sufficient time for integration

KEY SPRING SEMESTER ACTIVITIES

- Finish mandatory requirements of both sub-systems before starting with desirable requirements of one sub-system
- Arrange a bigger net and proper testing area
- Order sufficient spares
- Improve team communication
- Improve individual contribution
- Keep 1½ month for integration

DOCKING…

Questions?