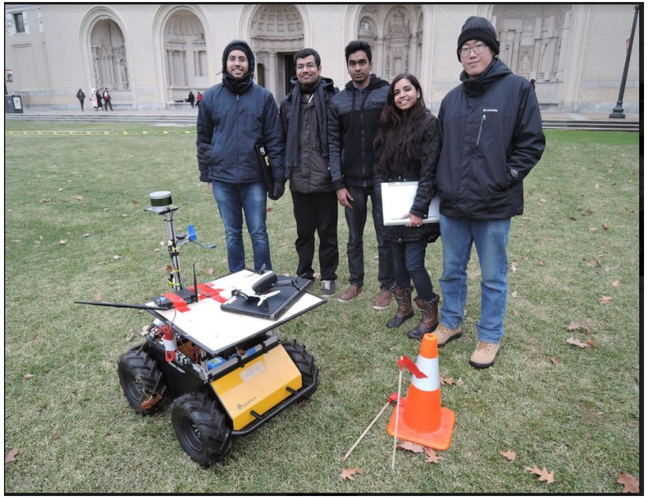
CARNEGIE MELLON UNIVERSITY
PITTSBURGH PENNSYLVANIA 1900

Multi-Agent Collaboration for navigation in disaster zone

THE ROBOTICS INSTITUTE

Team F- Falcon Eye

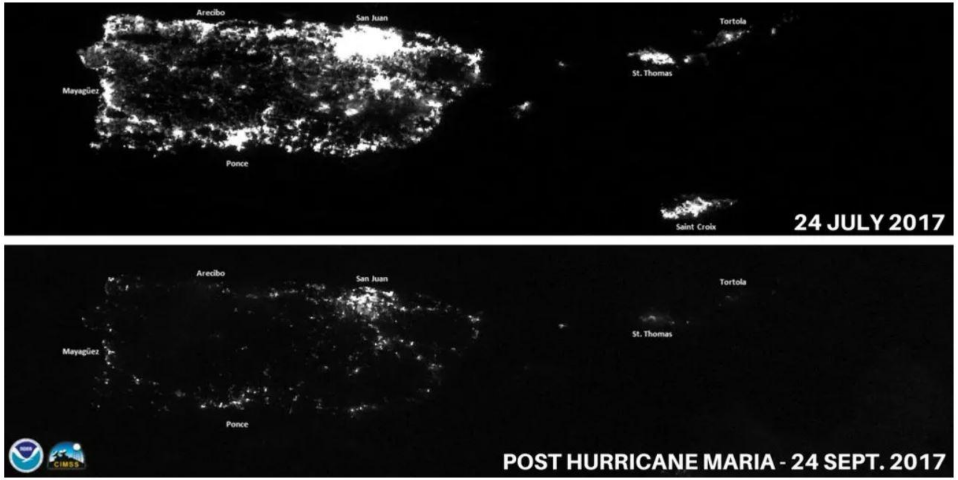
Yuchi Wang
Danendra Singh
Pulkit Goyal
Pratibha Tripathi
Rahul Ramakrishnan

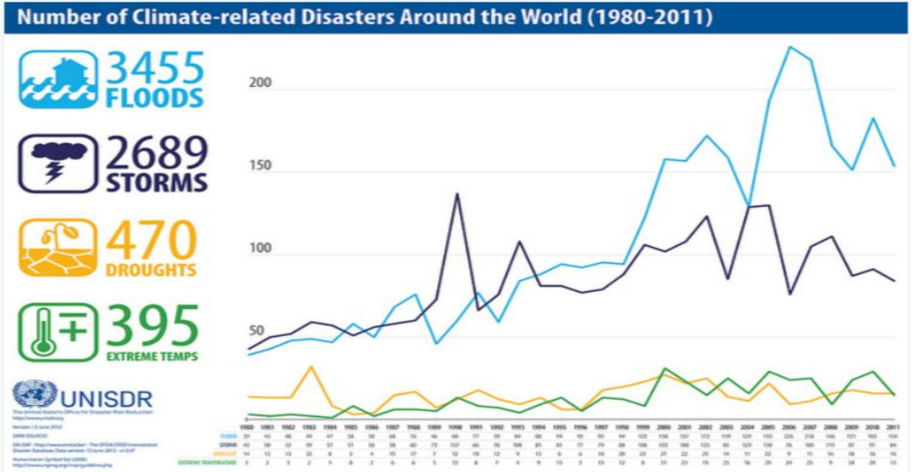


Project description

What Users Need:

- Quick access to ground data and information in disaster hit areas by various relief and response agencies without risking ground personnel.
- Send life-critical care package to the disaster victims.





Project description

What We Provide:

- Independent post-disaster data collection by a collaborative robotics system capable of operating on its own.
- Allows responders and relief agencies to plan their response effectively without sending personnel for damage assessment. Hence Reduce Risk!
- Send life-critical care package to the disaster victims via an autonomous ground vehicle.
- Robust Aerial and Ground based data gathering through mapping and vision sensors.

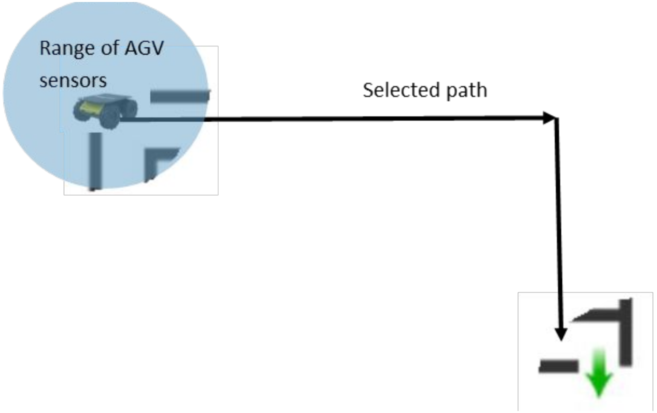
Project description

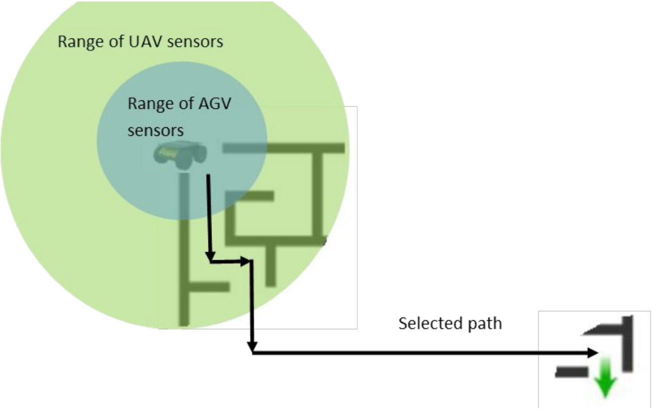
Motivation

- Path finding and localization using ground-level sensors is a difficult task when obstacles and dead ends are obstructed from the sensor's field of view.
- Leads to unacceptable performance in time-critical missions in unknown environments - such as disaster relief.

Objective

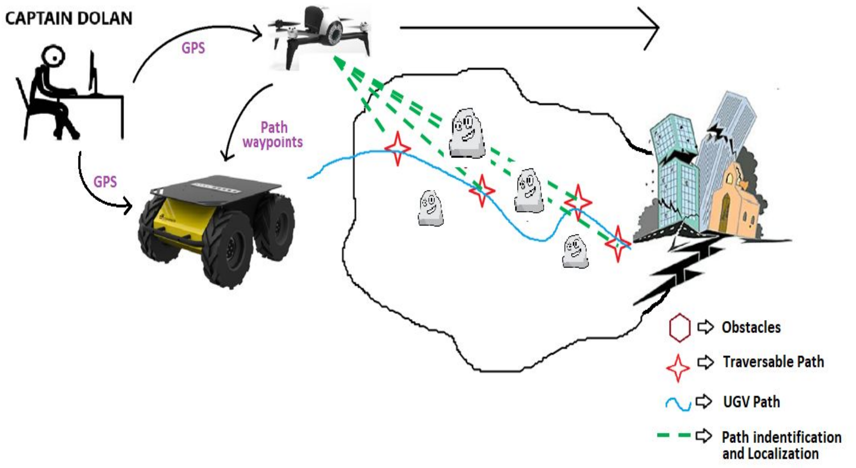
- Augment the localization and path planning capabilities of AGV's by integrating aerial sensor data from UAV's.





Use Case

1. An earthquake has struck in Louisiana. Capt. Dolan's Team is dispatched to provide assistance by gathering data and mapping disaster zone.
2. He has to send critical supplies to the victims along with surveying the intensity of damage in the area and to the major roads connecting the town.
3. He sets up the system, Falcon Eye team made for him and immediately starts collaborating with other responders, providing them with valuable aerial and ground imagery.



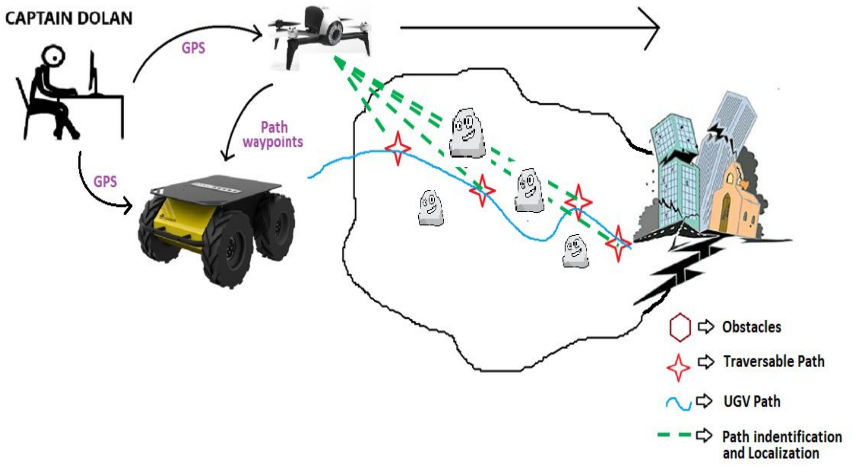
Use Case

4. Capt. inputs GPS location to the system of the area he wants to start the survey from.

5. The UAV takes off and moves to the given location with the AGV while identifying the best possible traversable path for the AGV to navigate.

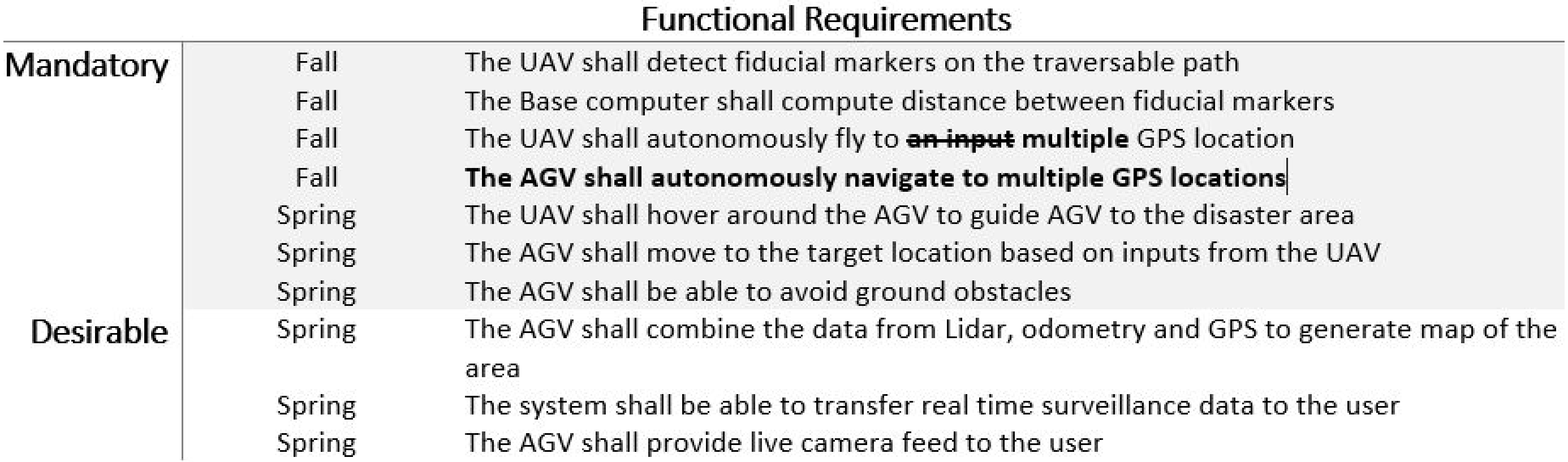
6. UAV provides the waypoints to the AGV for its navigation.The AGV navigates in accordance to the given waypoints avoiding obstacles on its way and reaches the disaster zone.

7. The AGV delivers care package to the victims.



System-level requirements

Functional



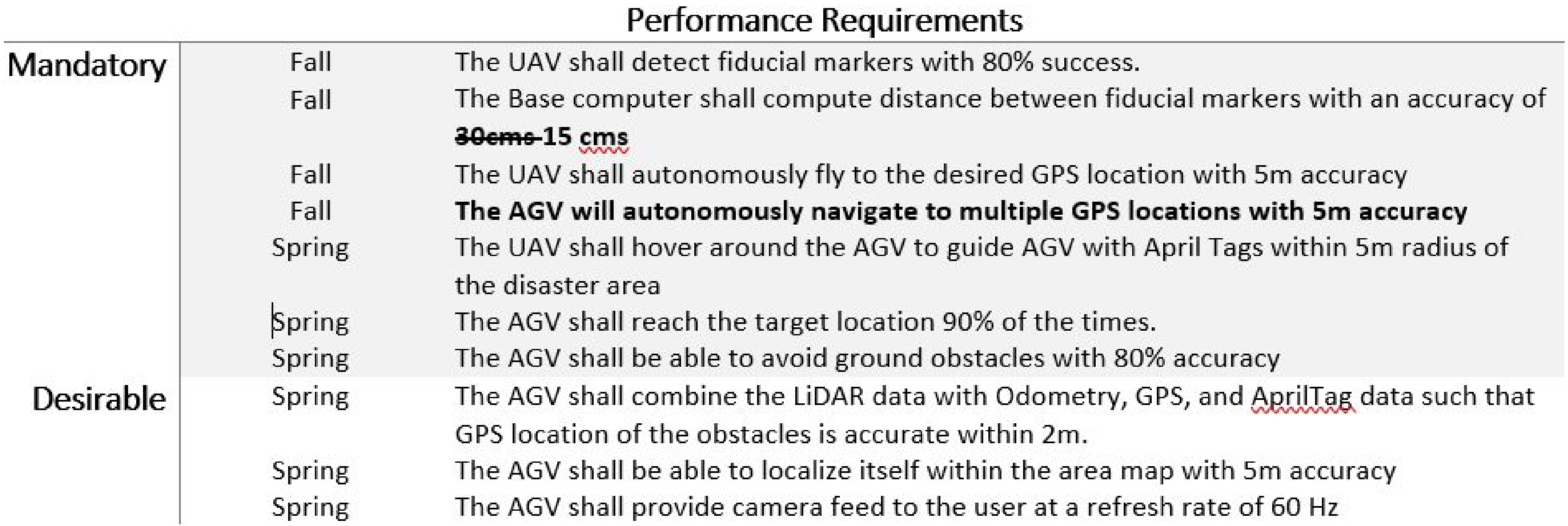
Functional Requirements

Mandatory	Fall	The UAV shall detect fiducial markers on the traversable path
	Fall	The Base computer shall compute distance between fiducial markers
	Fall	The UAV shall autonomously fly to ~~an input~~ **multiple** GPS location
	Fall	**The AGV shall autonomously navigate to multiple GPS locations**
	Spring	The UAV shall hover around the AGV to guide AGV to the disaster area
	Spring	The AGV shall move to the target location based on inputs from the UAV
	Spring	The AGV shall be able to avoid ground obstacles
Desirable	Spring	The AGV shall combine the data from Lidar, odometry and GPS to generate map of the area
	Spring	The system shall be able to transfer real time surveillance data to the user
	Spring	The AGV shall provide live camera feed to the user

- We have incorporated multi-waypoint navigation as the user might want to survey multiple locations.
- We have also added Autonomous navigation for Husky as a base for developing navigation stack for the husky in the spring

System-level requirements

Performance



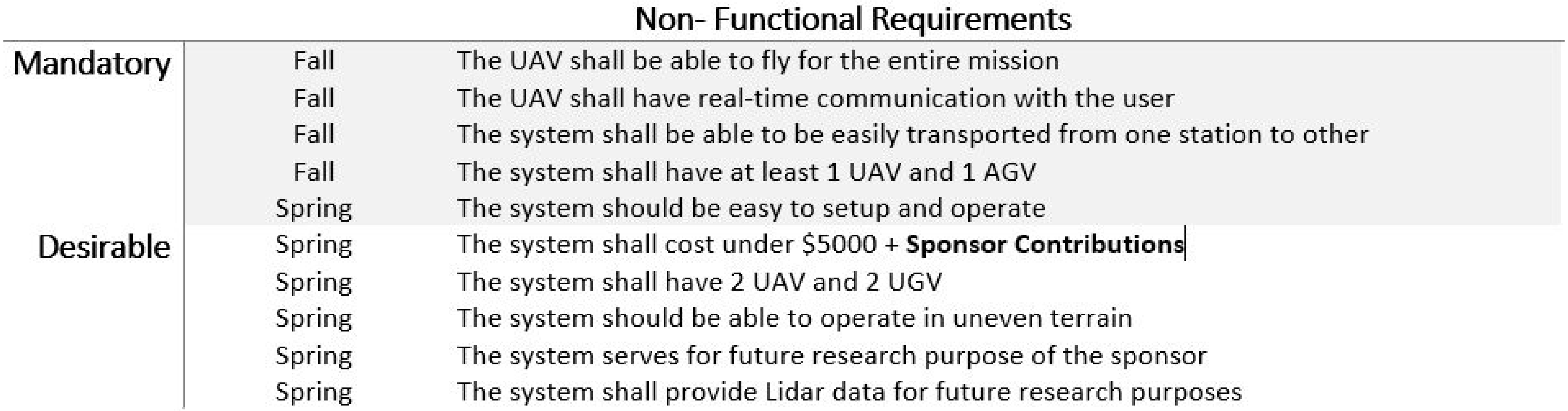
Performance Requirements

Mandatory	Fall	The UAV shall detect fiducial markers with 80% success.
	Fall	The Base computer shall compute distance between fiducial markers with an accuracy of **~~30cms~~ 15 cms**
	Fall	The UAV shall autonomously fly to the desired GPS location with 5m accuracy
	Fall	**The AGV will autonomously navigate to multiple GPS locations with 5m accuracy**
	Spring	The UAV shall hover around the AGV to guide AGV with April Tags within 5m radius of the disaster area
	Spring	The AGV shall reach the target location 90% of the times.
	Spring	The AGV shall be able to avoid ground obstacles with 80% accuracy
Desirable	Spring	The AGV shall combine the LiDAR data with Odometry, GPS, and AprilTag data such that GPS location of the obstacles is accurate within 2m.
	Spring	The AGV shall be able to localize itself within the area map with 5m accuracy
	Spring	The AGV shall provide camera feed to the user at a refresh rate of 60 Hz

- We have reduced the validation criteria for accuracy as test results have proved much better accuracy all the time.

System-level requirements

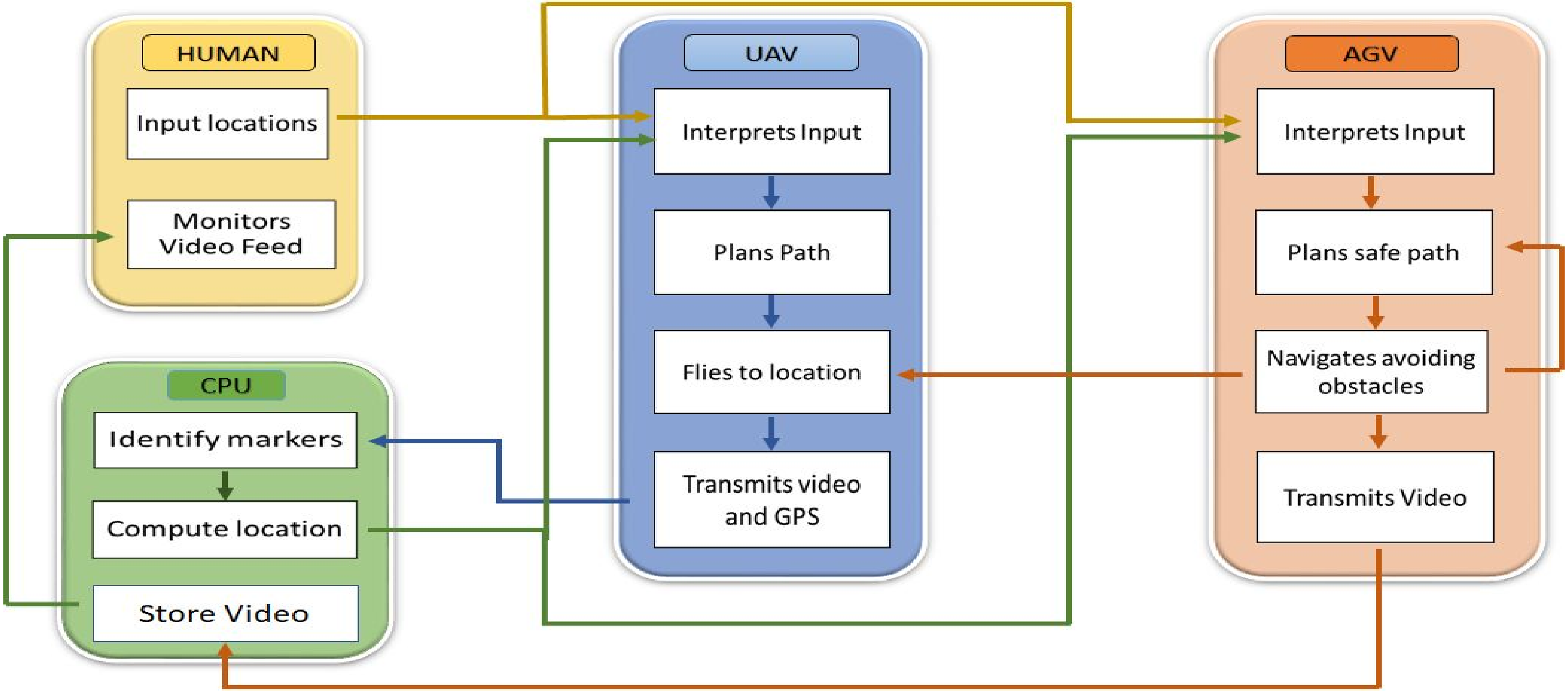
Non-Functional



Non- Functional Requirements

Mandatory	Fall	The UAV shall be able to fly for the entire mission
	Fall	The UAV shall have real-time communication with the user
	Fall	The system shall be able to be easily transported from one station to other
	Fall	The system shall have at least 1 UAV and 1 AGV
	Spring	The system should be easy to setup and operate
Desirable	Spring	The system shall cost under $5000 + **Sponsor Contributions**
	Spring	The system shall have 2 UAV and 2 UGV
	Spring	The system should be able to operate in uneven terrain
	Spring	The system serves for future research purpose of the sponsor
	Spring	The system shall provide Lidar data for future research purposes

Functional Architecture



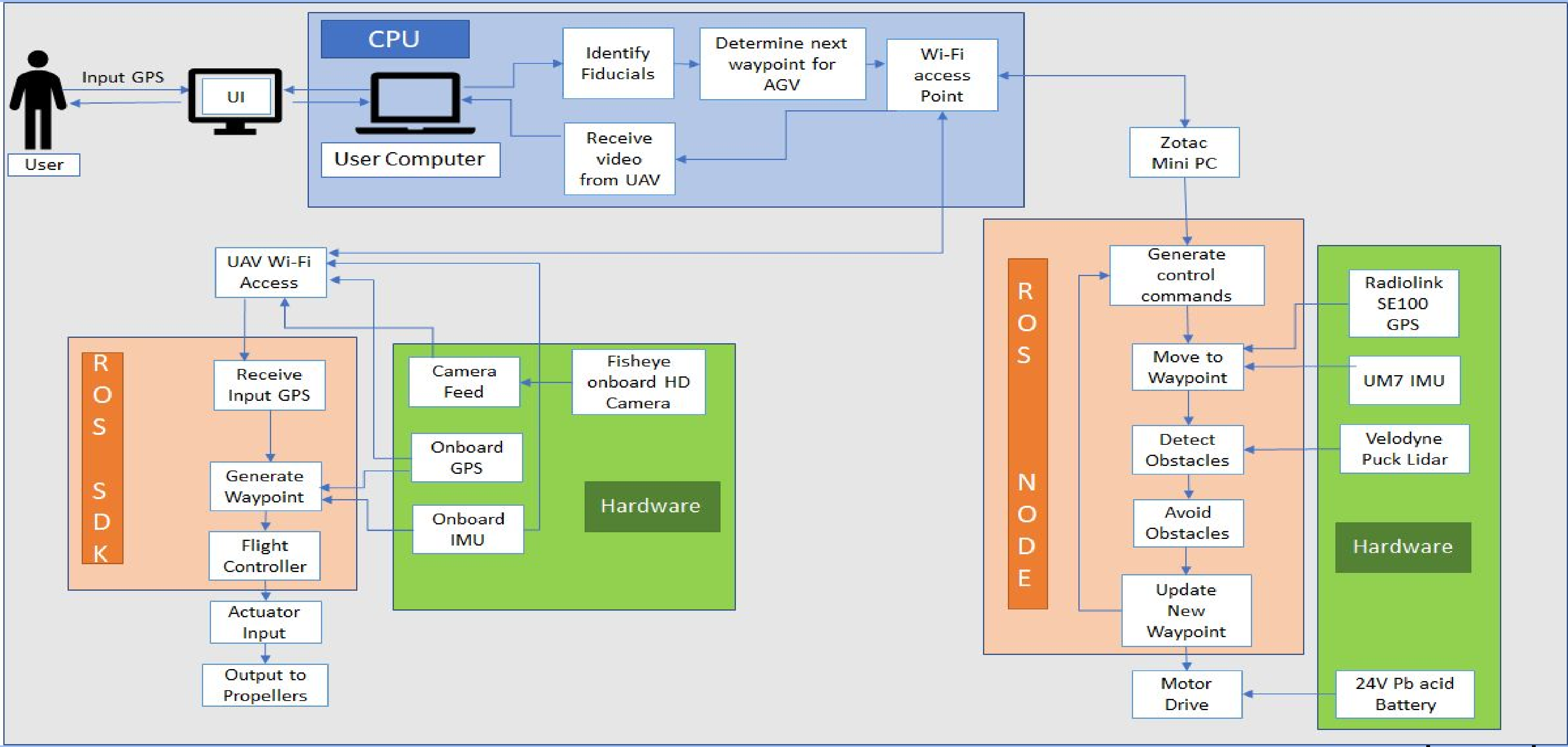
HUMAN
- Input locations
- Monitors Video Feed

UAV
- Interprets Input
- Plans Path
- Flies to location
- Transmits video and GPS

AGV
- Interprets Input
- Plans safe path
- Navigates avoiding obstacles
- Transmits Video

CPU
- Identify markers
- Compute location
- Store Video

Cyberphysical Architecture



User

Input GPS

UI

CPU

User Computer

Identify Fiducials

Determine next waypoint for AGV

Wi-Fi access Point

Receive video from UAV

Zotac Mini PC

UAV Wi-Fi Access

ROS SDK

Receive Input GPS

Generate Waypoint

Flight Controller

Actuator Input

Output to Propellers

Camera Feed

Fisheye onboard HD Camera

Onboard GPS

Onboard IMU

Hardware

ROS NODE

Generate control commands

Move to Waypoint

Detect Obstacles

Avoid Obstacles

Update New Waypoint

Motor Drive

Radiolink SE100 GPS

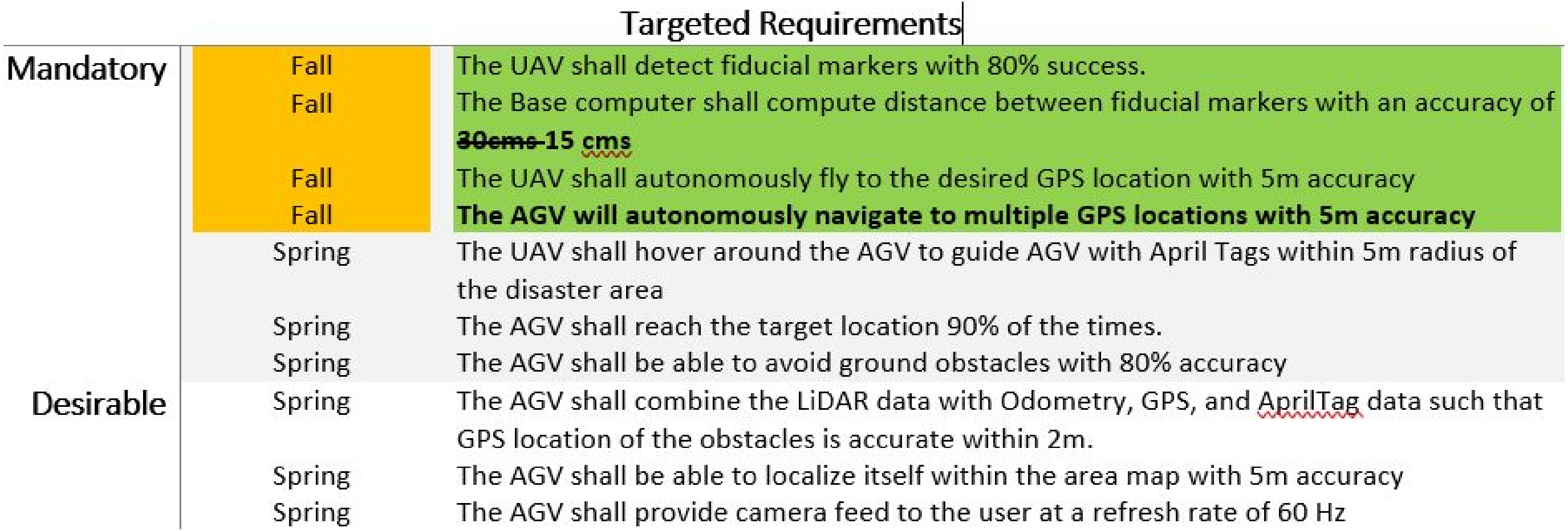
UM7 IMU

Velodyne Puck Lidar

Hardware

24V Pb acid Battery

Current System Status: Targeted Requirements

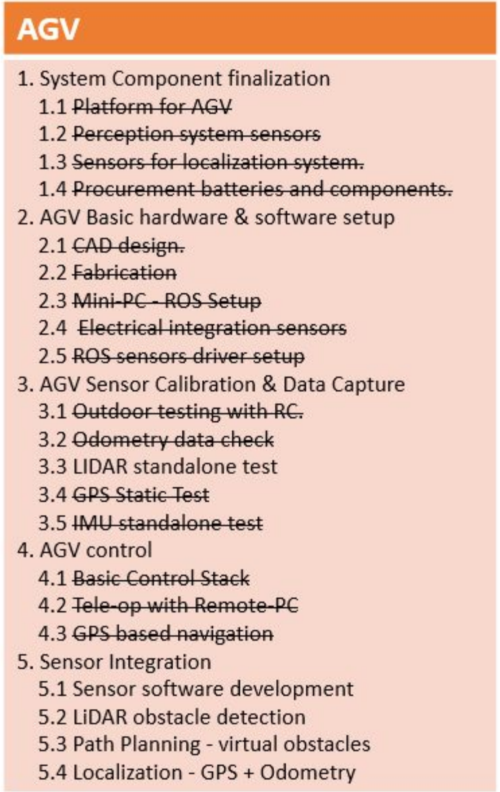


Targeted Requirements

Mandatory	Fall	The UAV shall detect fiducial markers with 80% success.
	Fall	The Base computer shall compute distance between fiducial markers with an accuracy of ~~30cms~~ **15 cms**
	Fall	The UAV shall autonomously fly to the desired GPS location with 5m accuracy
	Fall	**The AGV will autonomously navigate to multiple GPS locations with 5m accuracy**
	Spring	The UAV shall hover around the AGV to guide AGV with April Tags within 5m radius of the disaster area
	Spring	The AGV shall reach the target location 90% of the times.
	Spring	The AGV shall be able to avoid ground obstacles with 80% accuracy
Desirable	Spring	The AGV shall combine the LiDAR data with Odometry, GPS, and AprilTag data such that GPS location of the obstacles is accurate within 2m.
	Spring	The AGV shall be able to localize itself within the area map with 5m accuracy
	Spring	The AGV shall provide camera feed to the user at a refresh rate of 60 Hz

- The Green depicts we have achieved all the specified fall semester requirements

AGV Subsystem Status



AGV

1. System Component finalization
   - 1.1 ~~Platform for AGV~~
   - 1.2 ~~Perception system sensors~~
   - 1.3 ~~Sensors for localization system.~~
   - 1.4 ~~Procurement batteries and components.~~
2. AGV Basic hardware & software setup
   - 2.1 ~~CAD design.~~
   - 2.2 ~~Fabrication~~
   - 2.3 ~~Mini-PC - ROS Setup~~
   - 2.4 ~~Electrical integration sensors~~
   - 2.5 ~~ROS sensors driver setup~~
3. AGV Sensor Calibration & Data Capture
   - 3.1 ~~Outdoor testing with RC.~~
   - 3.2 ~~Odometry data check~~
   - 3.3 LIDAR standalone test
   - 3.4 ~~GPS Static Test~~
   - 3.5 ~~IMU standalone test~~
4. AGV control
   - 4.1 ~~Basic Control Stack~~
   - 4.2 ~~Tele-op with Remote-PC~~
   - 4.3 ~~GPS based navigation~~
5. Sensor Integration
   - 5.1 Sensor software development
   - 5.2 LiDAR obstacle detection
   - 5.3 Path Planning - virtual obstacles
   - 5.4 Localization - GPS + Odometry

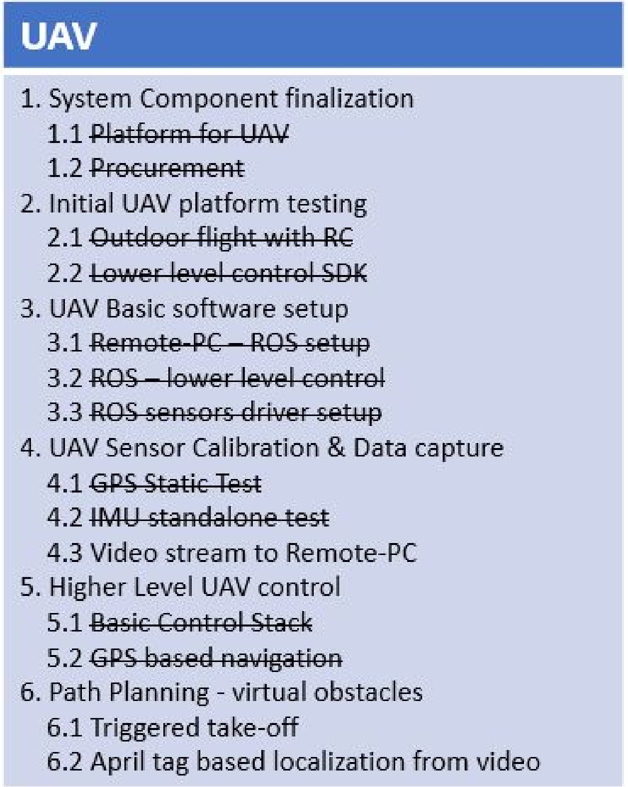
Completed:

1) GPS based waypoint navigation.

To be done:

1) LiDAR standalone testing and integrating with system.
2) Obstacle Detection
3) Path Planning
4) Localization.

UAV Subsystem Status



UAV

1. System Component finalization
    - 1.1 ~~Platform for UAV~~
    - 1.2 ~~Procurement~~
2. Initial UAV platform testing
    - 2.1 ~~Outdoor flight with RC~~
    - 2.2 ~~Lower level control SDK~~
3. UAV Basic software setup
    - 3.1 ~~Remote-PC – ROS setup~~
    - 3.2 ~~ROS – lower level control~~
    - 3.3 ~~ROS sensors driver setup~~
4. UAV Sensor Calibration & Data capture
    - 4.1 ~~GPS Static Test~~
    - 4.2 ~~IMU standalone test~~
    - 4.3 Video stream to Remote-PC
5. Higher Level UAV control
    - 5.1 ~~Basic Control Stack~~
    - 5.2 ~~GPS based navigation~~
6. Path Planning - virtual obstacles
    - 6.1 Triggered take-off
    - 6.2 April tag based localization from video

Completed:

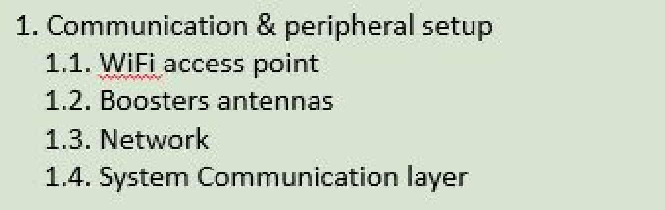
1) Accurate GPS based waypoint navigation.
2) April tag detection.
3) Localization of April Tags with respect to a home frame.

To be done:

1) Communication between UAV and UGV - to trigger UAV Takeoff.
2) Path identification by UAV on the basis of various april tags detected.
3) Communicating the path to UGV.
4) Able to get video from UAV on remote system, but need to figure that over the network.

Subsystem Integration status

Status



1. Communication & peripheral setup
    1.1. WiFi access point
    1.2. Boosters antennas
    1.3. Network
    1.4. System Communication layer

Internal Test

Test Sequence	Description	Performance Measures
1	Range test for the WiFi Network	Maintain connection till 50m distance
2	Single workstation for the entire system	Use one system for UAV and AGV

Analysis

1) We faced range issues with the WiFi. We had to mount the WiFi router over the Husky robot.
2) We faced some issues with multiple ROS master configuration, so we used separate workstations for UAV and AGV.

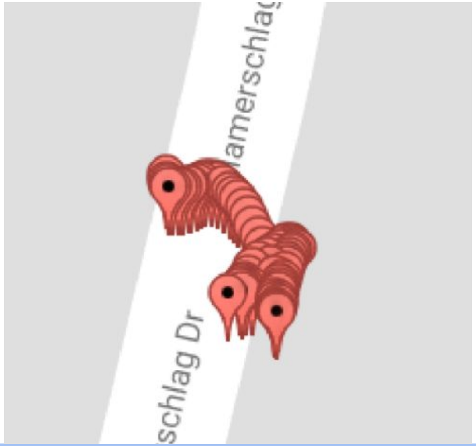
Power Distribution Board Status

- The board has been correctly populated and verified.
- Stable voltages to run two 5V devices (GPS and IMU) , one 12V Wifi Router and a 24V Mini-PC.
- We are getting nominal drift of 6% from the specified voltages that lies within the operating ranges of the devices.
- Since we already have stable DC output from the built-in DC-DC converters in Husky with less than 2% drift, we plan to use this board as a backup and for indoor testing.

Modeling, Analysis and Testing

- April Tag Localization
  - Performed multiple tests to accurately depict and localize set of april Tags.
  - Implemented a low-pass filter to improve detection

- GPS accuracy
  - Gathered data to find out the drift in UAV GPS and Radio Link GPS for AGV.
  - Found the accuracy to be about 3m.



Modeling, Analysis and Testing

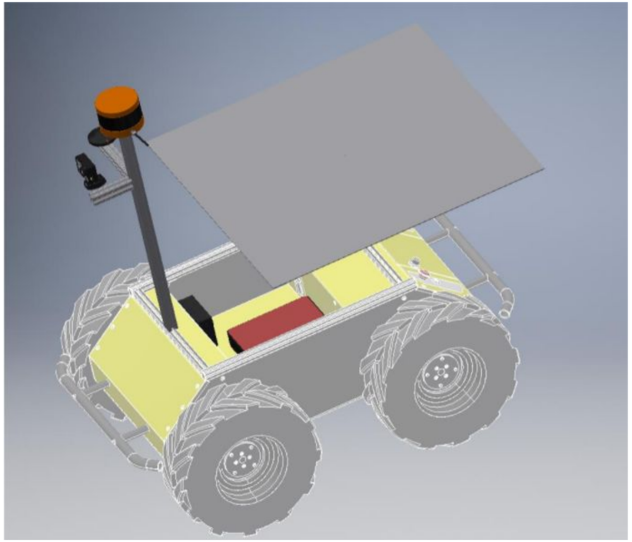
- UAV waypoint navigation
  - Tried using the built in GPS navigation capabilities
  - Implemented a custom Navigation Controller
  - Analyzed the results and improved the controller with smooth motion while approaching a waypoint.

- AGV waypoint navigation
  - Performed multiple test to check accuracy of IMU UM7
  - After analyzing drift even after calibration, utilized smartphone's IMU with ROS

Modeling, Analysis and Testing

- Network connections
  - Changed UAV configurations to function as a client (default is host).
  - Configured a router as a common network for UAV and AGV.
  - Tested the range of a common network that provides access point to both UAV and AGV.

- Husky remote operation and Mechanical Setup
  - Setup and configured a Mini PC for setting up Husky ROS node
  - Using remote access via a router, controlled the Husky from the maximum range of the router
  - Fabricated mounts for sensors and a platform for UAV on the AGV.



Fall Validation Experiment - Test A

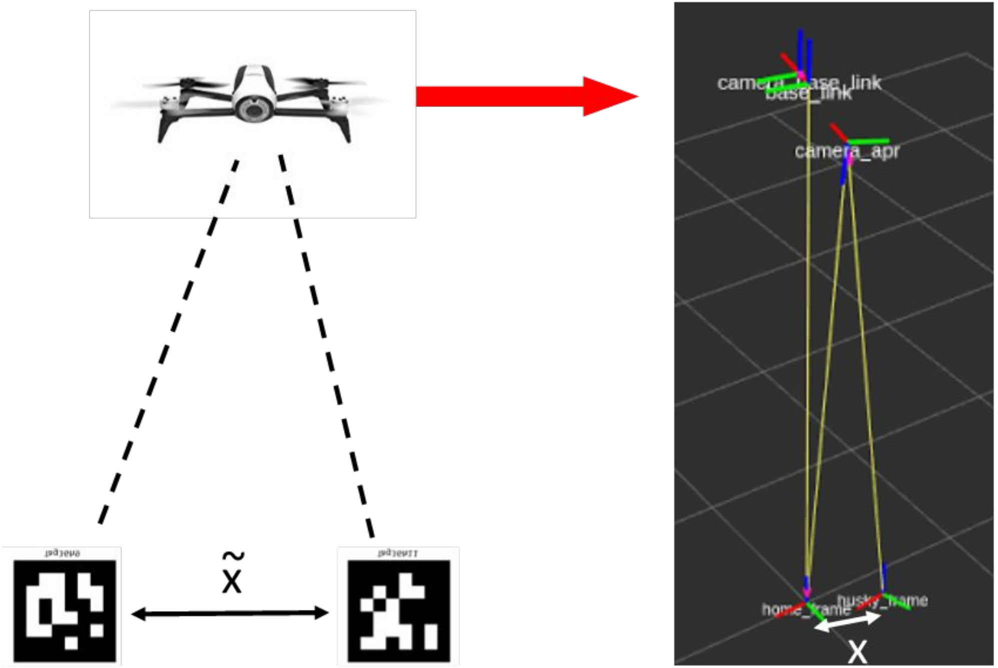
Test A: April Tags Localization test

Objective: To calculate the distance between two April tags from the video captured by the drone.

Test Sequence:

Test Sequence	Description	Performance Measures
A.1	Place UAV on ground. Give take-off command (teleop).	
A.2	UAV takes off and transmits video	
A.3	The CPU detects the two April tags placed on the ground through the camera feed of UAV	Accurately detect both the markers
A.4	Compute the distance between the two tags	Accuracy of distance computed with respect to manual measurement (+- 30cm)

Fall Validation Experiment - Test A



Fall Validation Experiment - Test A analysis

1) Able to localize another april tag w.r.t home frame successfully - **Actual distance (2m) - observed distance - 1.95m** .
2) **Safety** was a concern due to heavy wind during FVE, which we addressed during FVE Encore.
3) Need to identify **better material** to print April Tags.
4) If the material is better, with less reflection, we might be able to fly higher and even then detect all april tags accurately.
5) Need to plan better for **bad weather** (wind or snow) as it impacts the test.

Fall Validation Experiment - Test B

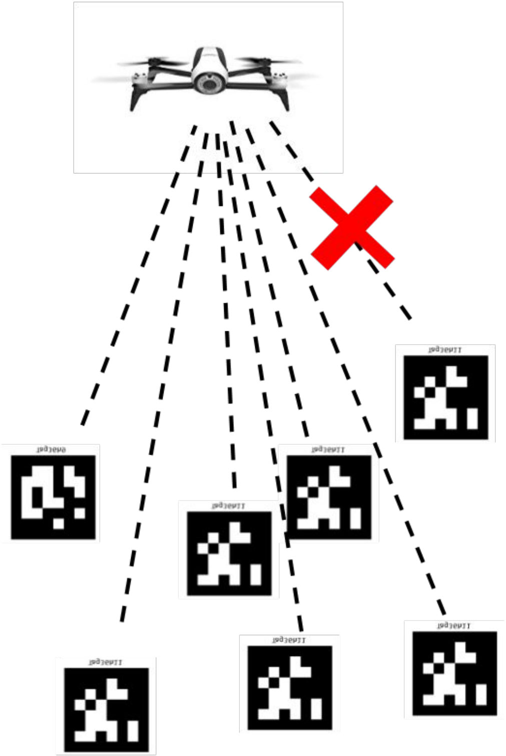
Test B: April Tags Detection test

Objective: To detect all the April tags placed in the test environment.

Test Sequence:

Test Sequence	Description	Performance Measures
B.1	Place the UAV on ground.	
B.2	Takeoff UAV (teleop)	
B.3	The CPU detects all markers placed in the test environment through camera feed from UAV	Accuracy of number of markers detected (80 %)

Fall Validation Experiment - Test B



Fall Validation Experiment - Test B analysis

1) Able to detect all **10 april tags** successfully.
2) **Safety** was a concern due to heavy wind during FVE, which we addressed during FVE Encore.
3) Need to identify **better material** to print April Tags.
4) If the material is better, with less reflection, we might be able to **fly higher** and even then detect all april tags accurately.
5) Need to plan better for **bad weather** (wind or snow) as it impacts the test.

Fall Validation Experiment - Test C

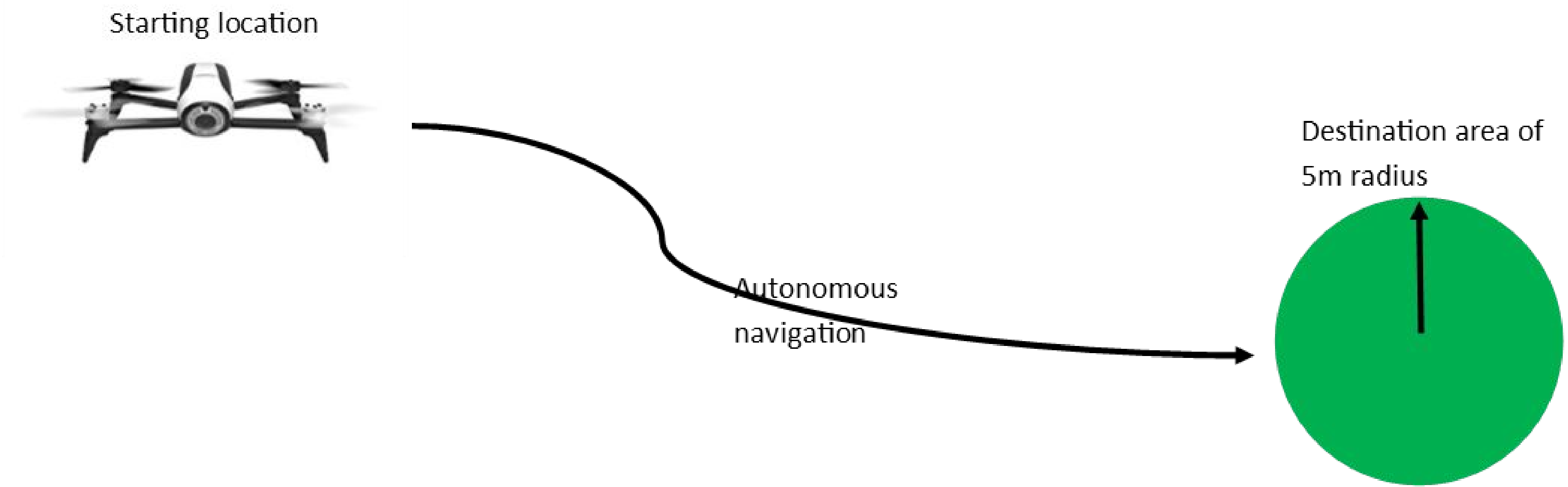
Test C: UAV waypoint navigation test

Objective: To validate the autonomous flight control and waypoint navigation capability of the UAV

Test Sequence:

Test Sequence	Description	Performance Measures
C.1	Feed three known GPS locations as destination.	
C.2	UAV flies to the given GPS locations and lands on the last location	Accuracy in reaching desired GPS location (+- 5m tolerance)

Fall Validation Experiment



Starting location

Autonomous navigation

Destination area of 5m radius

Fall Validation Experiment - Test C analysis

1) We had successfully **completed** the GPS based waypoint navigation for UAV.
2) We had taken the tolerance of **5m**, and were well within the tolerance(**<3m**).
3) We were **flying low**, due to bad weather conditions.

Fall Validation Experiment - Test D

Test C: AGV waypoint navigation test

Objective: To validate the autonomous waypoint navigation capability of the AGV

Test Sequence:

Test Sequence	Description	Performance Measures
C.1	Feed known three GPS locations as destination.	
C.2	AGV drives to the given GPS locations	Accuracy in reaching desired GPS location (+- 5m tolerance)

Fall Validation Experiment - Test D analysis

1) Able to navigate to all three GPS waypoints successfully.
2) For FVE we had taken the tolerance of **5m** , for FVE encore we reduced it to **3m**. We were well within range in both the experiments.
3) We used our phone for IMU, we were facing data accuracy issues with the IMU UM7.
4) Need to **replace phone** with a low cost IMU.

Fall Validation Experiment - Teaser

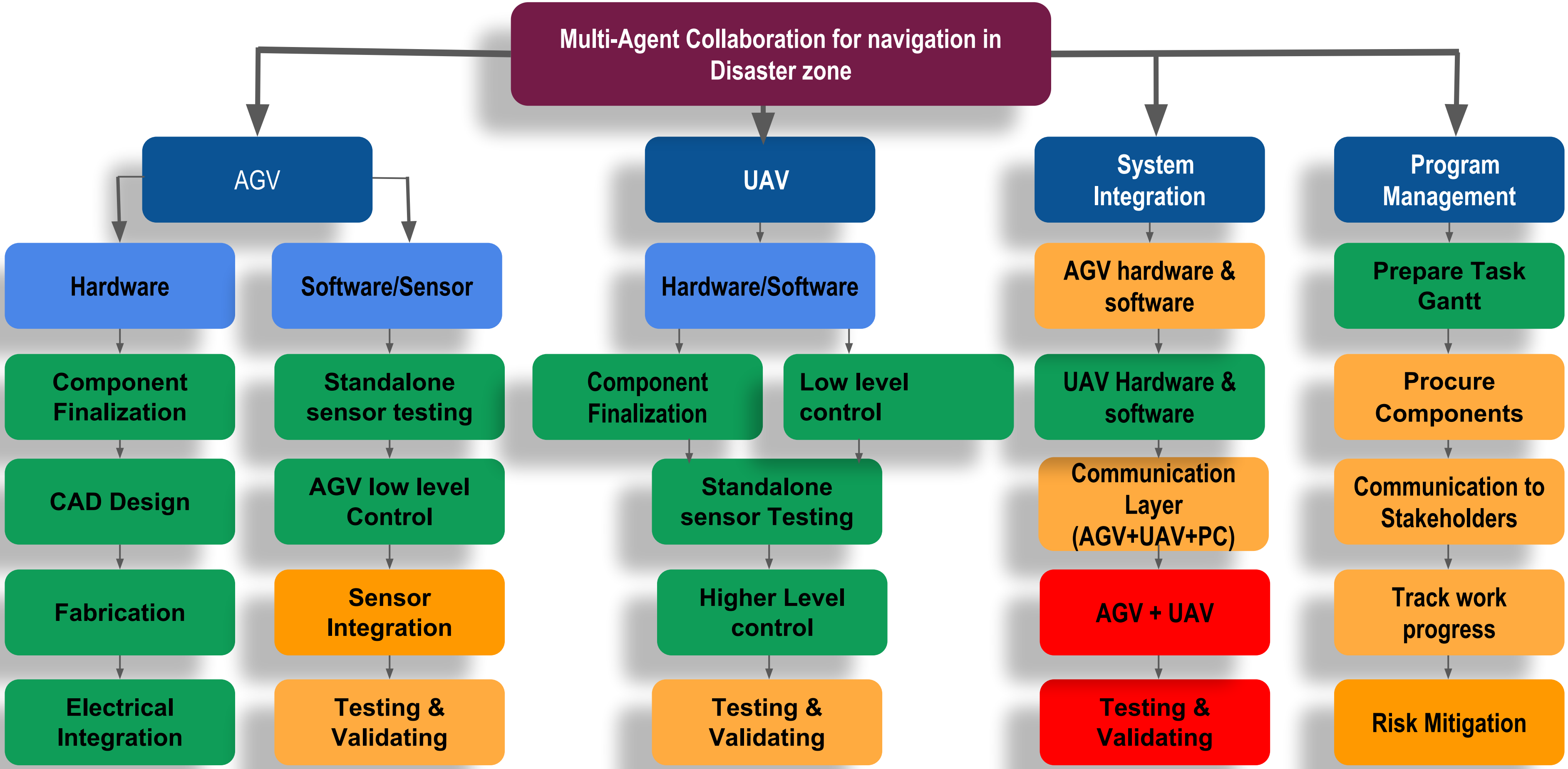
Fall Validation Experiment - Conclusions

- **Strong Points**
  - **Robust GPS** waypoint navigation capabilities of UAV and AGV.
  - **Stable** April Tag **Detection** from a height of <=5m.
  - **Accurate April Tag Localization** (much within the specified threshold).
  - **Easy** to **setup** system.
  - Great understanding and **collaboration** between **team** members.

- **Weak Points**
  - **Low flying** ceiling for the UAV.
  - Standalone **IMU** integration
  - Testing at night and in bad weather.

- **Refinement Areas**
  - Improvement in **safety measures** while demonstrating and testing subsystems.
  - Better introductions to various subsystems to be tested.
  - **Increase UAV flying height** with due considerations to safe ceiling

Updated Work Breakdown Structure



Multi-Agent Collaboration for navigation in Disaster zone

AGV

Hardware
Component Finalization
CAD Design
Fabrication
Electrical Integration

Software/Sensor
Standalone sensor testing
AGV low level Control
Sensor Integration
Testing & Validating

UAV

Hardware/Software
Component Finalization
Low level control
Standalone sensor Testing
Higher Level control
Testing & Validating

System Integration

AGV hardware & software
UAV Hardware & software
Communication Layer (AGV+UAV+PC)
AGV + UAV
Testing & Validating

Program Management

Prepare Task Gantt
Procure Components
Communication to Stakeholders
Track work progress
Risk Mitigation

Work Breakdown Structure (High-Level)

AGV

1. System Component finalization
2. AGV Basic hardware & software setup
3. AGV Sensor Calibration & Data Capture
4. AGV control
5. Sensor Integration

UAV

1. System Component finalization
2. Initial UAV platform testing
3. UAV Basic Software setup
4. UAV Sensor Calibration & Data Capture
5. Higher Level UAV control
6. UAV intelligence

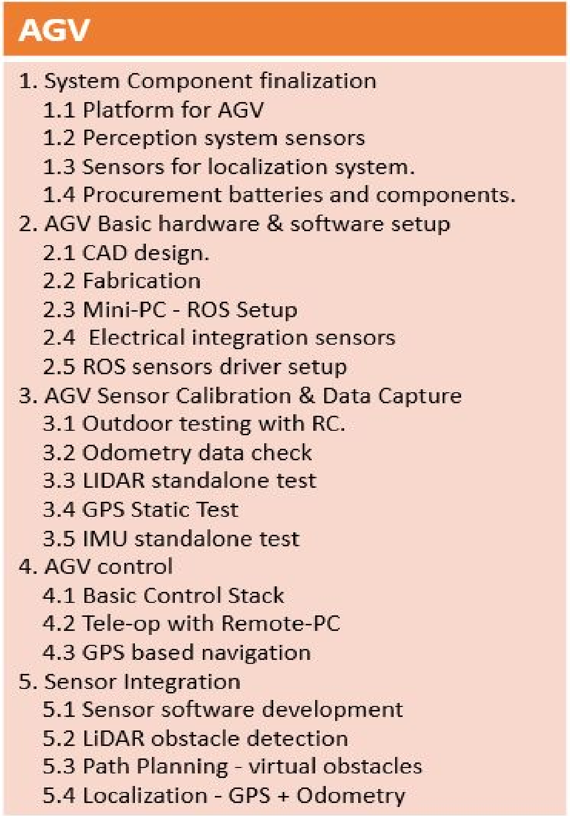
SYSTEM INTEGRATION

1. Communication & peripheral setup
2. Integration test for fall demo
3. Integration test for spring demo

PROJECT MANAGEMENT

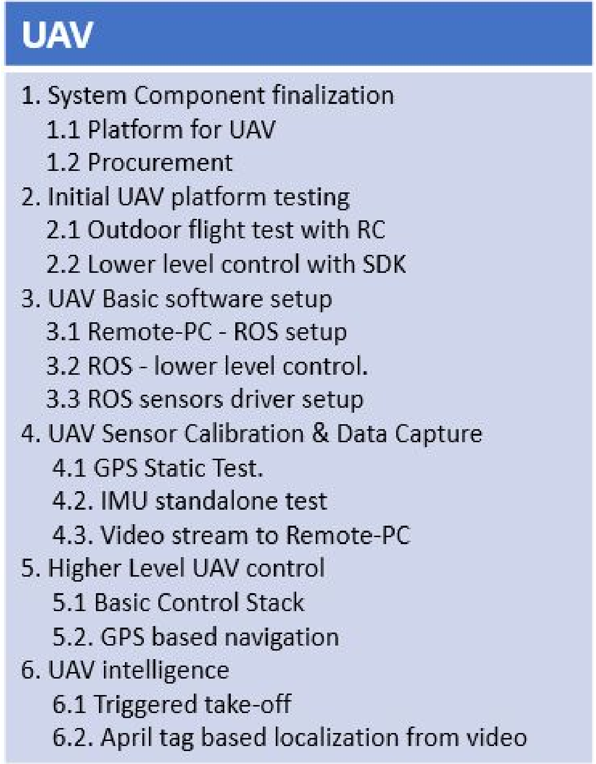
1. Prepare task Gantt
2. Procure Components
3. Communication to Stakeholders
4. Track Work Progress
5. Risk Mitigation

Work Breakdown Structure (Detailed)



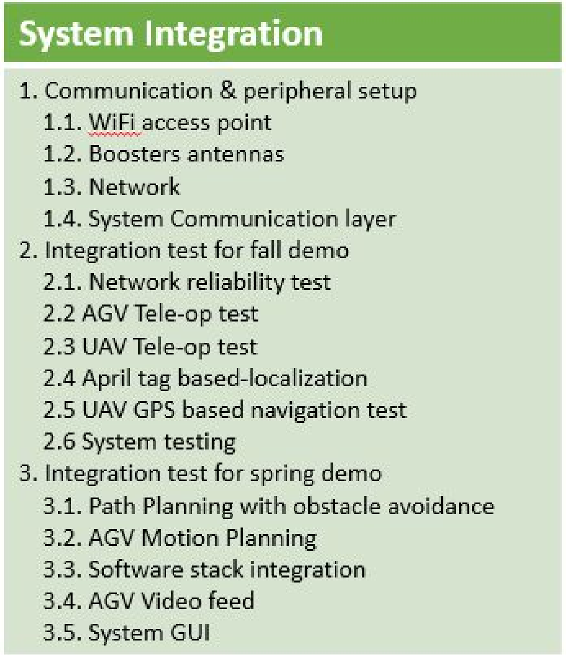
AGV

1. System Component finalization
   - 1.1 Platform for AGV
   - 1.2 Perception system sensors
   - 1.3 Sensors for localization system.
   - 1.4 Procurement batteries and components.
2. AGV Basic hardware & software setup
   - 2.1 CAD design.
   - 2.2 Fabrication
   - 2.3 Mini-PC - ROS Setup
   - 2.4 Electrical integration sensors
   - 2.5 ROS sensors driver setup
3. AGV Sensor Calibration & Data Capture
   - 3.1 Outdoor testing with RC.
   - 3.2 Odometry data check
   - 3.3 LIDAR standalone test
   - 3.4 GPS Static Test
   - 3.5 IMU standalone test
4. AGV control
   - 4.1 Basic Control Stack
   - 4.2 Tele-op with Remote-PC
   - 4.3 GPS based navigation
5. Sensor Integration
   - 5.1 Sensor software development
   - 5.2 LiDAR obstacle detection
   - 5.3 Path Planning - virtual obstacles
   - 5.4 Localization - GPS + Odometry



UAV

1. System Component finalization
   - 1.1 Platform for UAV
   - 1.2 Procurement
2. Initial UAV platform testing
   - 2.1 Outdoor flight test with RC
   - 2.2 Lower level control with SDK
3. UAV Basic software setup
   - 3.1 Remote-PC - ROS setup
   - 3.2 ROS - lower level control.
   - 3.3 ROS sensors driver setup
4. UAV Sensor Calibration & Data Capture
   - 4.1 GPS Static Test.
   - 4.2. IMU standalone test
   - 4.3. Video stream to Remote-PC
5. Higher Level UAV control
   - 5.1 Basic Control Stack
   - 5.2. GPS based navigation
6. UAV intelligence
   - 6.1 Triggered take-off
   - 6.2. April tag based localization from video



System Integration

1. Communication & peripheral setup
   - 1.1. WiFi access point
   - 1.2. Boosters antennas
   - 1.3. Network
   - 1.4. System Communication layer
2. Integration test for fall demo
   - 2.1. Network reliability test
   - 2.2 AGV Tele-op test
   - 2.3 UAV Tele-op test
   - 2.4 April tag based-localization
   - 2.5 UAV GPS based navigation test
   - 2.6 System testing
3. Integration test for spring demo
   - 3.1. Path Planning with obstacle avoidance
   - 3.2. AGV Motion Planning
   - 3.3. Software stack integration
   - 3.4. AGV Video feed
   - 3.5. System GUI

PROJECT MANAGEMENT

1. Prepare task Gantt
2. Procure Components
3. Communication to Stakeholders
4. Track Work Progress
5. Risk Mitigation

Schedule - FVE

	Sept		October								November								
Tasks	21-28	29-30	1-7	8-13	15-19	20	21-26	27	28-30	31	1-2	3	4-9	10	11-21	22	23-28	28-29	30
No. of days	8	2	7	6	5	1	6	1	3	1	2	1	6	1	11	1	6	2	1
AGV																			
System Component finalisation		1.1	1.4				1.2-1.3												
Basic hardware & software Dev				2.3			2.1		2.2		2.4								
sensor calibration & Data capture					3.1								3.2,3.5		3.3, 3.4				
Control															4.3				
sensor integration																			
UAV																			
System Component finalisation				1.1-1.2															
Initial UAV testing					2.1-2.2														
Basic ROS sensor software Dev					3.1-3.2		3.3		3.3										
sensor calibration & Data capture											4.1		4.2		4.2, 4.3				
Higher UAV Control											5.1		5.1		5.2				
UAV Intelligence																			
System Integration																			
Communication & peripheral setup													1.1		1.2,1.3				
Integrating system for FVE																	Testing for FVE		
Integrating system for SVE																			

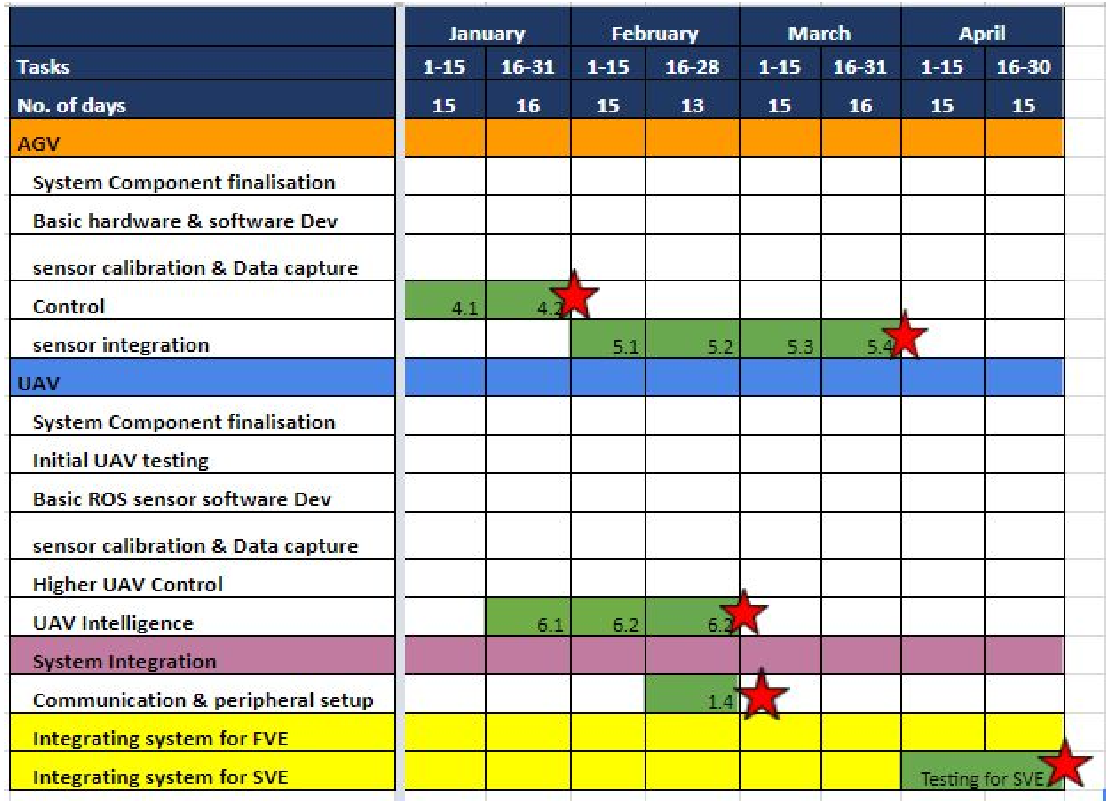
Program Management Status

PROJECT MANAGEMENT

1. Prepare task Gantt
2. Procure Components
3. Communication to Stakeholders
4. Track Work Progress
5. Risk Mitigation

1. We have prepared Gantt for tracking the work breakdown and timelines on weekly basis.
2. We keep doing the course correction based on hits and misses of the planned tasks.
3. Procurement for components is still on-going.
4. We are tracking the identified risks and mitigating them timely.

Schedule - SVE



	January		February		March		April	
Tasks	1-15	16-31	1-15	16-28	1-15	16-31	1-15	16-30
No. of days	15	16	15	13	15	16	15	15
AGV								
System Component finalisation								
Basic hardware & software Dev								
sensor calibration & Data capture								
Control	4.1	4.2 ★						
sensor integration			5.1	5.2	5.3	5.4 ★		
UAV								
System Component finalisation								
Initial UAV testing								
Basic ROS sensor software Dev								
sensor calibration & Data capture								
Higher UAV Control								
UAV Intelligence		6.1	6.2	6.2 ★				
System Integration								
Communication & peripheral setup				1.4 ★				
Integrating system for FVE								
Integrating system for SVE							Testing for SVE	★

★ Milestones

Test Plan

Date	PR	Milestone	Test method
Late-Jan	PR7	Location fusion & UAV Intelligence	Check difference in relative location between locations and test UAV path planning heuristics when AGV is teleoperated
Mid-Feb	PR8	Communication between UAV & AGV	UAV take off on trigger from AGV. AGV able to read locations given by UAV
Late-Feb	PR 9	AGV Path Planning and localization	Test AGVs ability to dynamically generate and follow AGV waypoints from combined map
Mid-Mar	PR10	Obstacle detection with LIDAR	Success rate of AGV to avoid static obstacles in its path
Early-Apr	PR11	System integration and testing	Refer to SVE test plan
Mid-Apr	PR12	Complete System integration	Refer to SVE test plan

Spring Validation Experiment

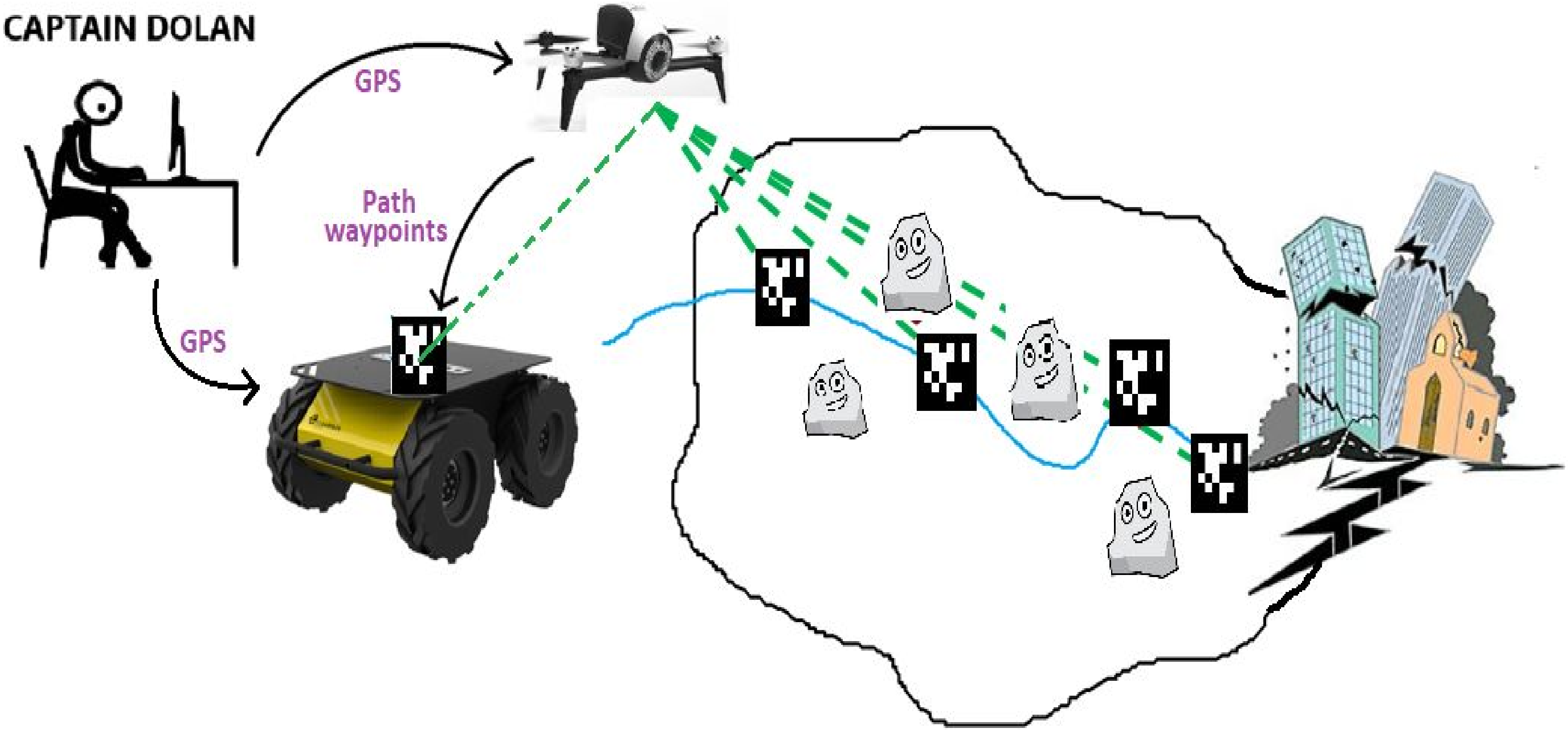
Location : CFA lawn

- Size: 50m X 50m
- Floor surface: Grass

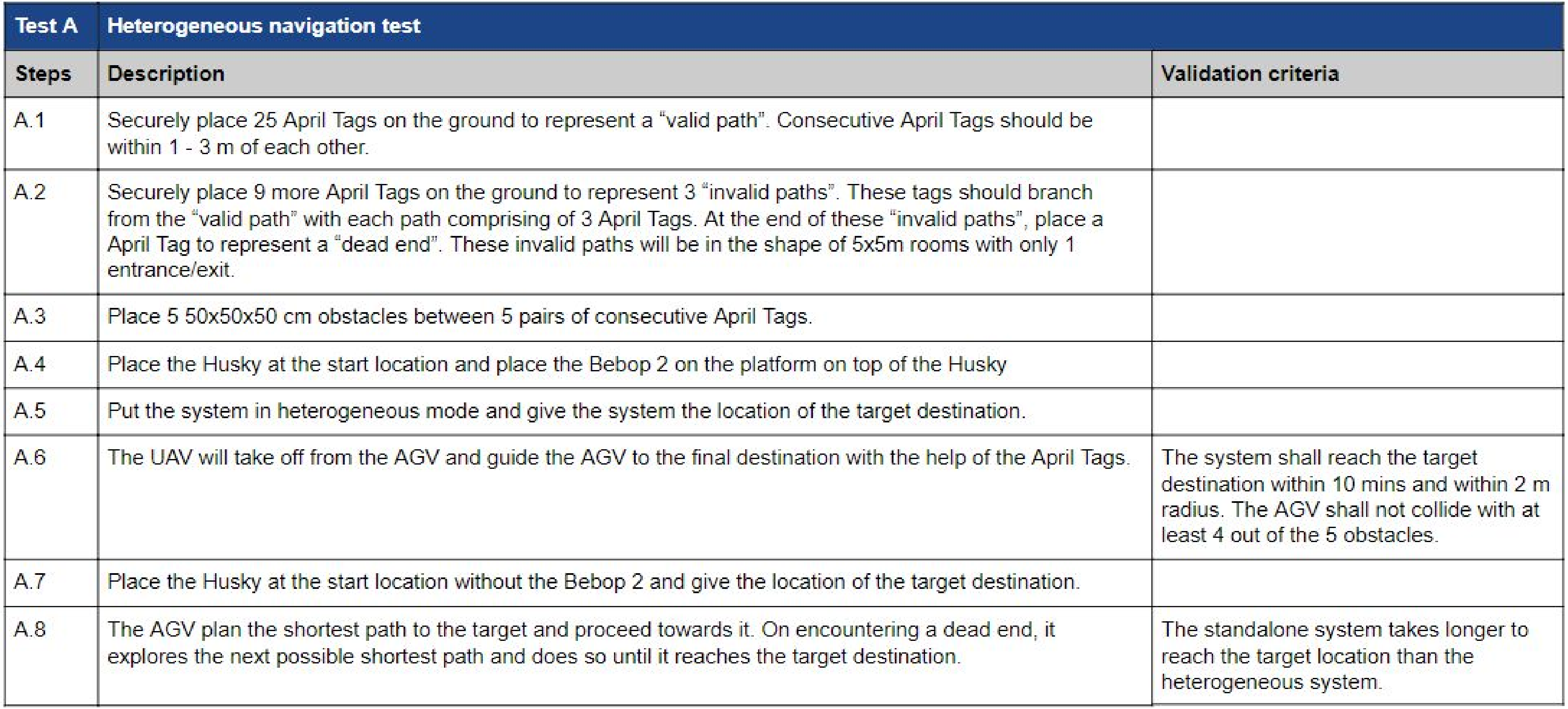
Setup:

- April tags : 11.7 x 16.5 in (Count : 37)
- Obstacles : 50 X 50 X 50 cm
- One UGV (Husky)
- One UAV (Bebop 2)
- One laptop connected wirelessly to UGV mini-PC
- One laptop connected wirelessly to UAV
- Five team Members

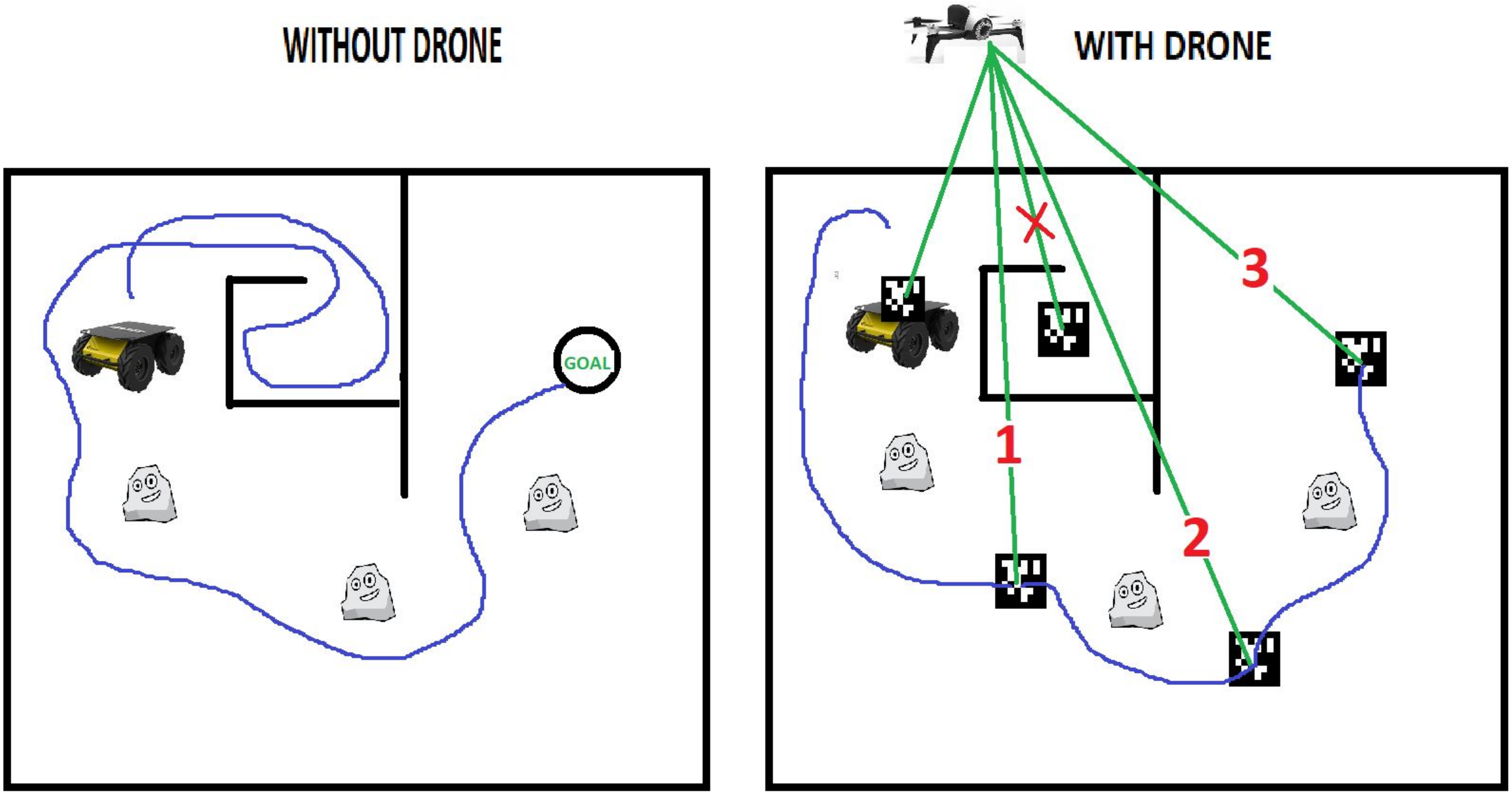
Spring Validation Experiment



Spring Validation Experiment

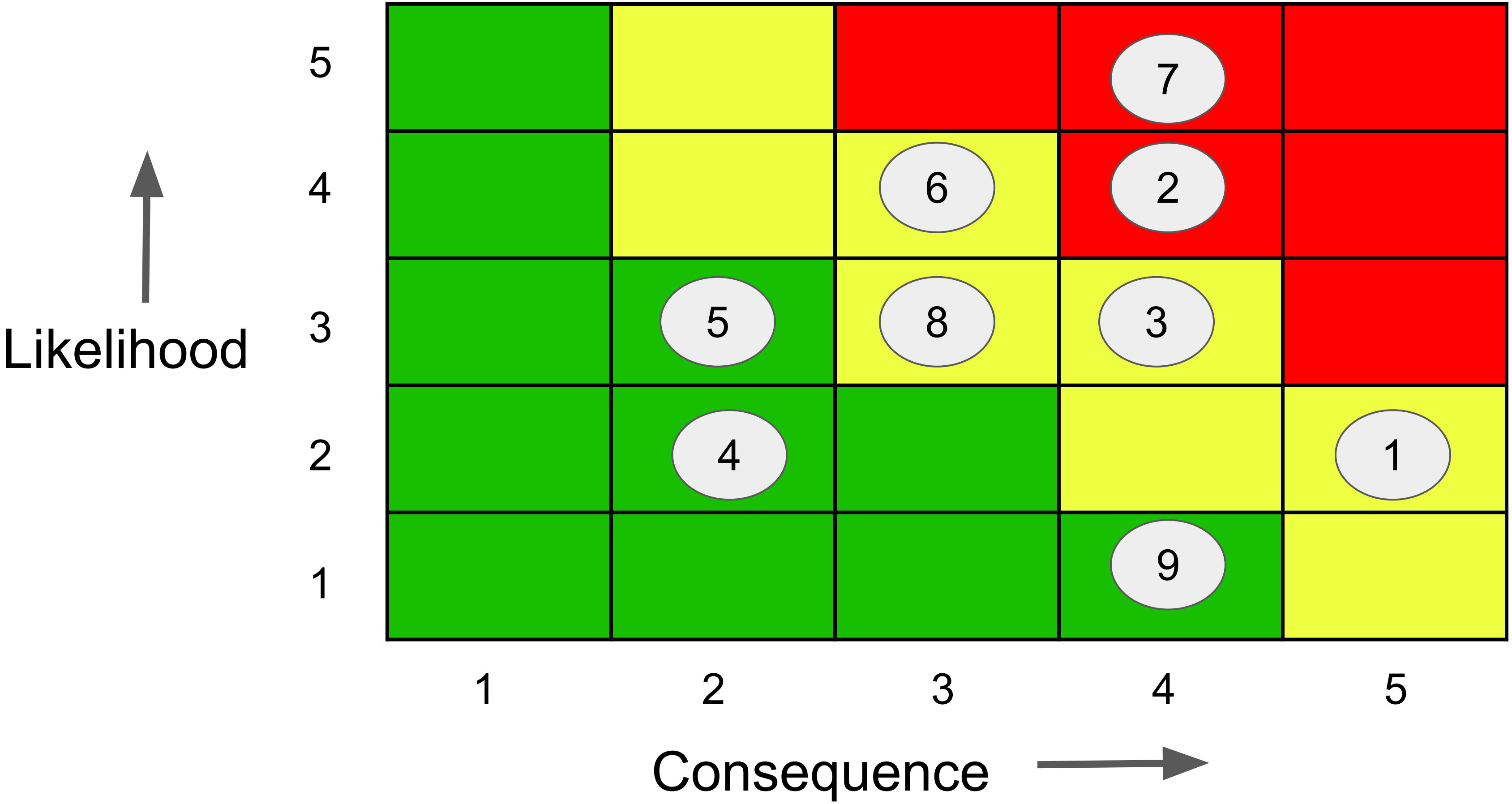


Test A	**Heterogeneous navigation test**	
Steps	**Description**	**Validation criteria**
A.1	Securely place 25 April Tags on the ground to represent a “valid path”. Consecutive April Tags should be within 1 - 3 m of each other.	
A.2	Securely place 9 more April Tags on the ground to represent 3 “invalid paths”. These tags should branch from the “valid path” with each path comprising of 3 April Tags. At the end of these “invalid paths”, place a April Tag to represent a “dead end”. These invalid paths will be in the shape of 5x5m rooms with only 1 entrance/exit.	
A.3	Place 5 50x50x50 cm obstacles between 5 pairs of consecutive April Tags.	
A.4	Place the Husky at the start location and place the Bebop 2 on the platform on top of the Husky	
A.5	Put the system in heterogeneous mode and give the system the location of the target destination.	
A.6	The UAV will take off from the AGV and guide the AGV to the final destination with the help of the April Tags.	The system shall reach the target destination within 10 mins and within 2 m radius. The AGV shall not collide with at least 4 out of the 5 obstacles.
A.7	Place the Husky at the start location without the Bebop 2 and give the location of the target destination.	
A.8	The AGV plan the shortest path to the target and proceed towards it. On encountering a dead end, it explores the next possible shortest path and does so until it reaches the target destination.	The standalone system takes longer to reach the target location than the heterogeneous system.


WITHOUT DRONE
GOAL
WITH DRONE
1
2
3

Major Risks and Mitigation

S. N	Risk	L	C	Type	Mitigation strategy	Owner
1	Drone testing not possible during night	2	5	Technical	Shift work schedule of members working on drone to maximize daytime testing	Danny
2	Outdoor testing problems due to weather	4	4	Schedule	Setup indoor testing platform and test often	Pulkit
3	Sensor data not sufficiently accurate	3	4	Technical	Test all required sensors early and finish testing by FVE. Order new sensors if required.	Rahul
4	Insufficient ROS support available from Bebop online community	2	2	Technical	Take guidance from the Robotics Institute's members working with drones	Yuchi
5	Limited system network bandwidth	3	2	Technical	Compress images received from the cameras	Pratibha
6	System integration takes up lot of time	4	3	Schedule	Start system integration early and integrate subsystems whenever possible	Pulkit
7	Husky localization not sufficiently accurate	5	4	Technical	Implement sensor fusion. Use visual odometry with LiDAR, increase April Tag size, use better GPS	Pratibha
8	Team Member unavailable due to sickness, personal matters etc	3	3	Schedule	Assign secondary holder to major tasks	Danny
9	Availability of Drone Testing Location and Licensing	1	4	Schedule	Acquire FAA licensing & acquire permissions to fly on campus	Yuchi



Likelihood	1	2	3	4	5
5	🟩	🟨	🟥	🟥 7	🟥
4	🟩	🟨	🟨 6	🟥 2	🟥
3	🟩	🟩 5	🟨 8	🟨 3	🟥
2	🟩	🟩 4	🟩	🟨	🟨 1
1	🟩	🟩	🟩	🟩 9	🟨

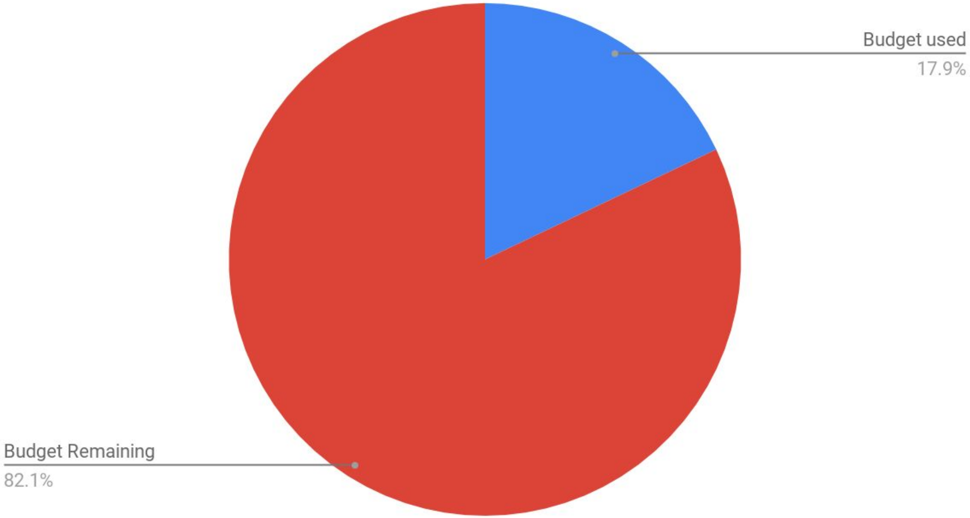
Consequence

Budget Status

BOUGHT

Upgrades for Mini-PC
IMU
Battery
Electronics
Safety equipments

Total Budget: $5000



Money Spent: $896.91
Balance Left: $4103.09

SPONSORED

Clearpath Husky
Parrot Bebop
Velodyne Puck
GPS
Camera
Zotac Mini-PC
Laptop
Monitor

Conclusions

- Lessons Learnt
    - **Collaboration** is important amongst teammates.
    - Everyone should be on the same page.
    - **Time management** is essential.
    - Systems sometimes don’t perform as we want them to.
    - **Mitigate risks** by planning for failsafes.
    - Record everything, People and **Robots have mood swings**.
    - Pittsburgh’s weather is unpredictable.

- Key Spring Activities
    - Implement **sensor fusion** for AGV localization.
    - Implement **path planning** for AGV navigation.
    - **Obstacle Avoidance** for AGV.
    - **System Integration** (AGV + UAV collaborative operation).
    - Testing the overall system for robustness.

Thank You!