Multi-Agent Collaboration for navigation in disaster zone

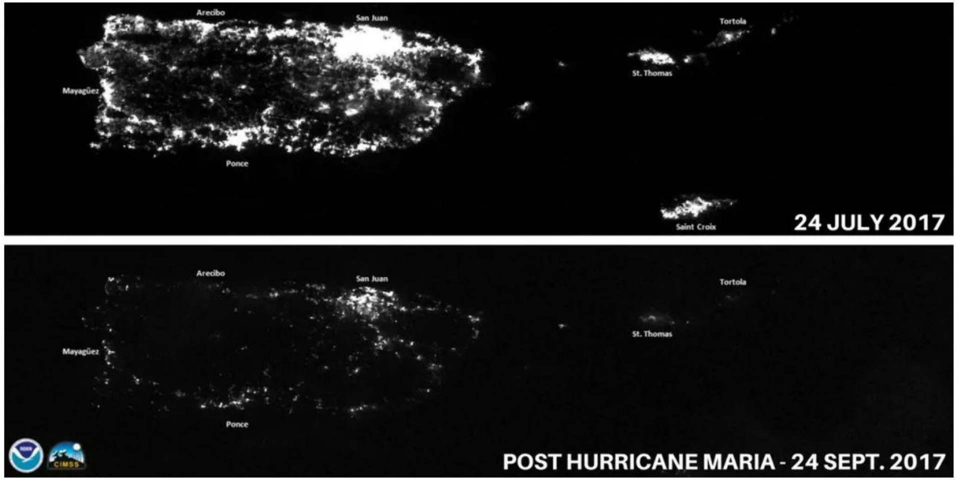
Team F- Falcon Eye

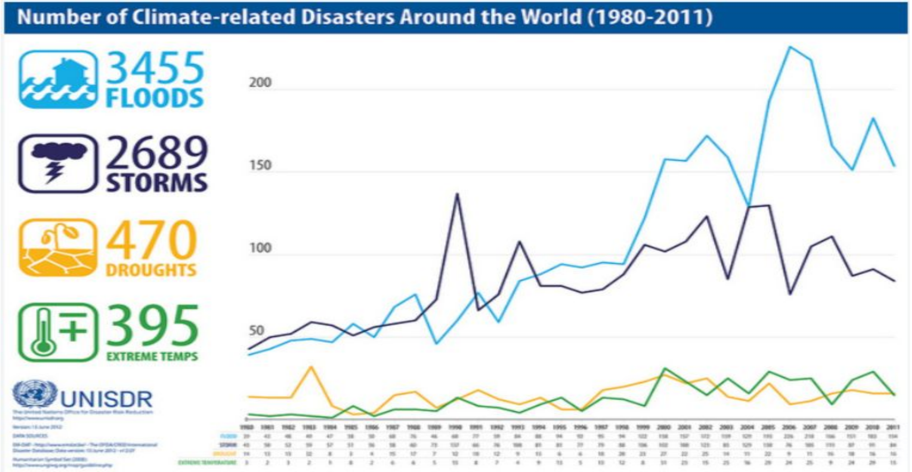
Yuchi Wang
Danendra Singh
Pulkit Goyal
Pratibha Tripathi
Rahul Ramakrishnan

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

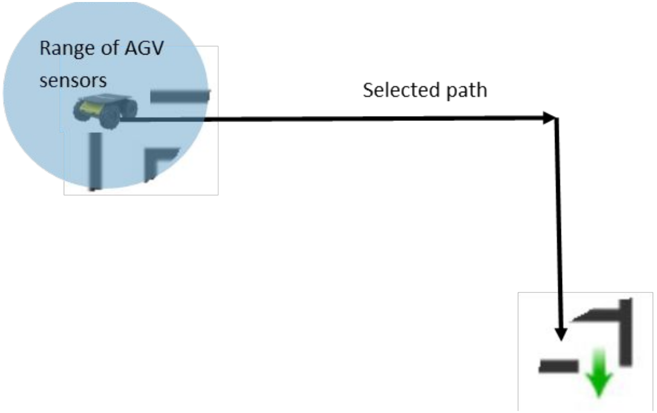
What We Provide:

- Independent post-disaster situational awareness (SA), damage assessment and 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

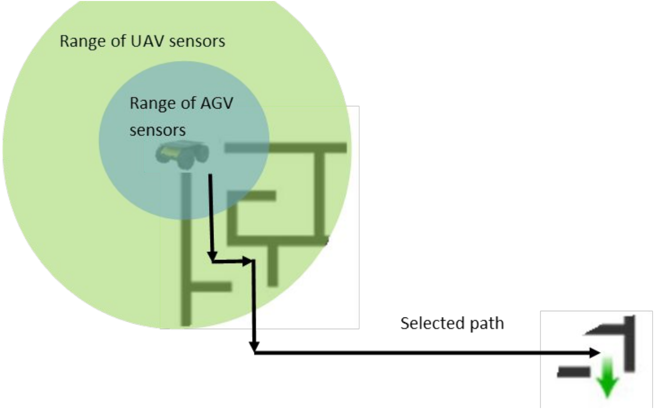
Motivation

- 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.

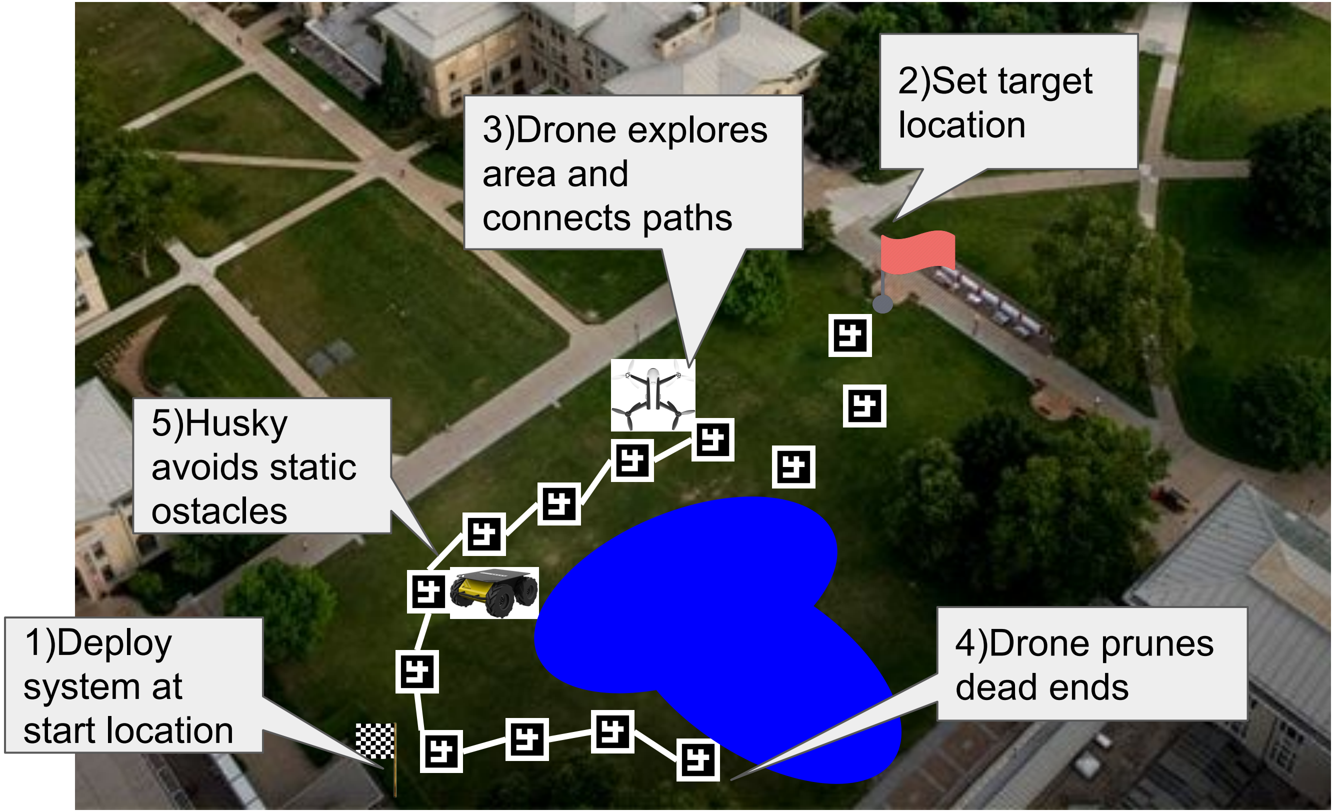


Objective

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

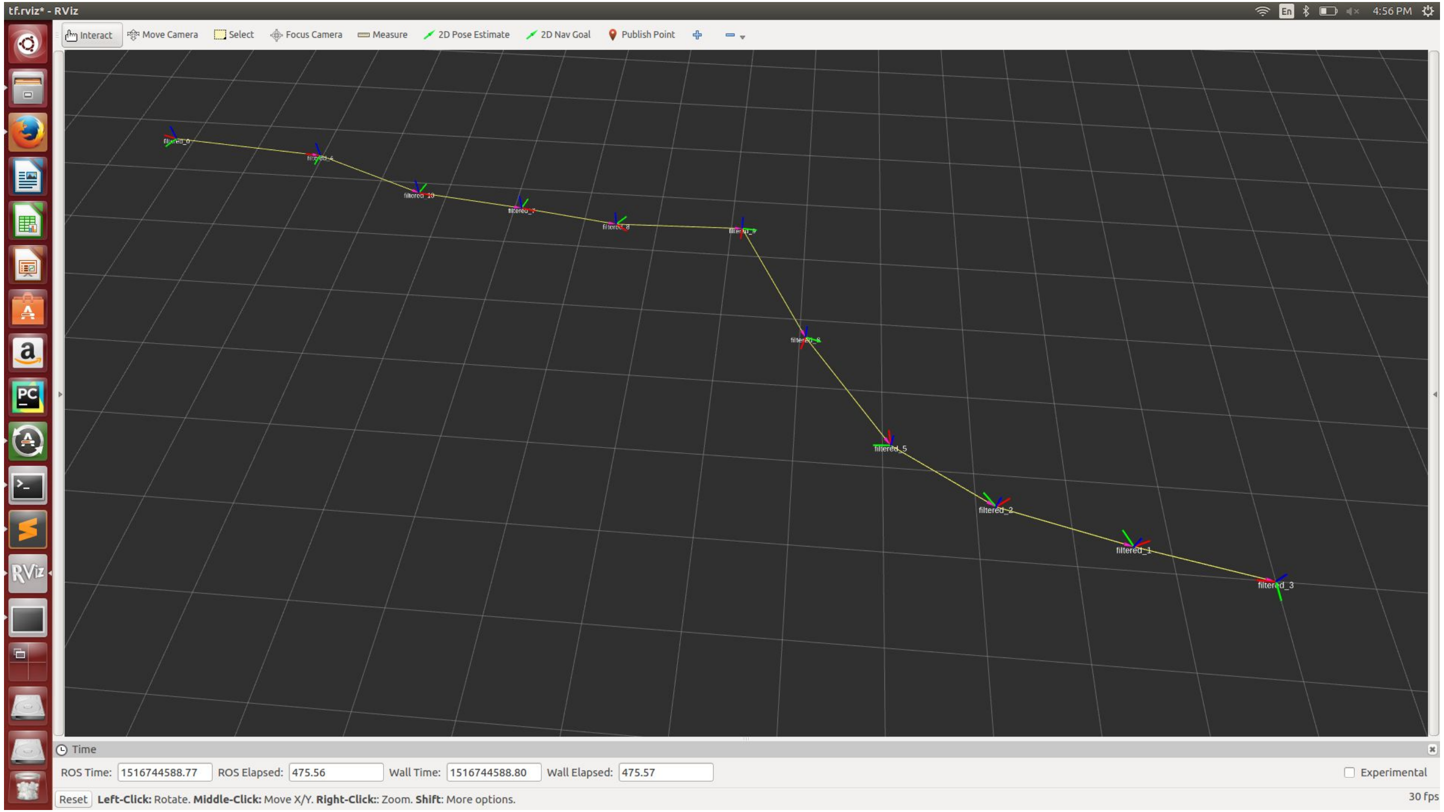


Use Case



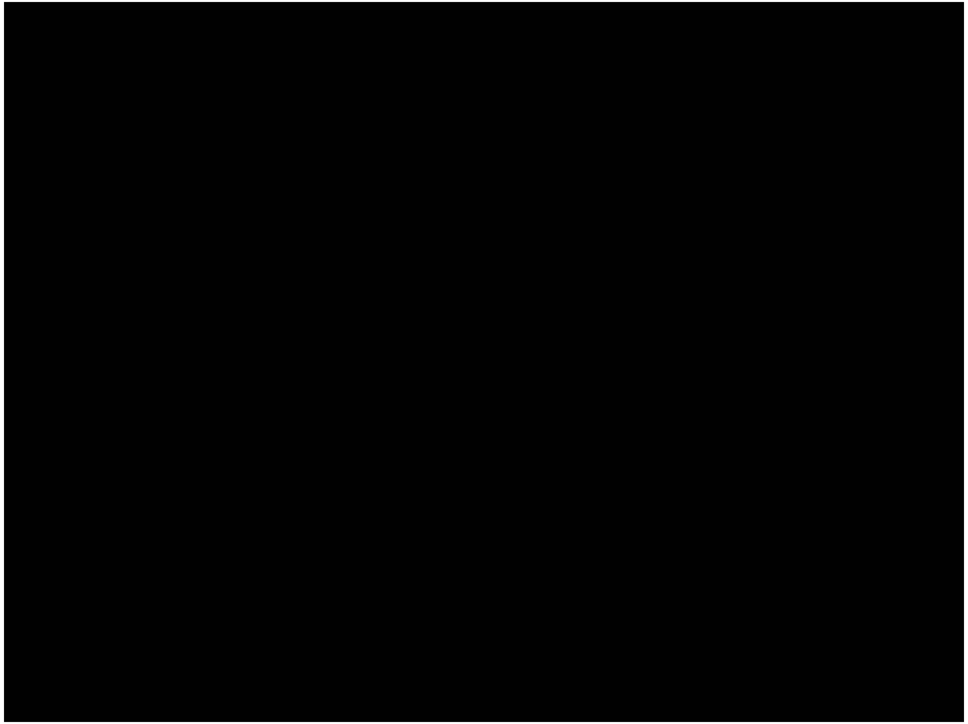
3)Drone explores area and connects paths
2)Set target location
5)Husky avoids static ostacles
1)Deploy system at start location
4)Drone prunes dead ends

Use Case - UAV


tf.rviz* - RViz
4:56 PM
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
filtered_5
filtered_2
filtered_1
filtered_3
Time
ROS Time: 1516744588.77
ROS Elapsed: 475.56
Wall Time: 1516744588.80
Wall Elapsed: 475.57
Experimental
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click:: Zoom. Shift: More options.
30 fps

Use Case - AGV

Use Case - UAV



Requirement Modifications

Added Three additional requirements:

1) The UAV shall explore the field and provide the waypoints to guide AGV to the disaster area.
   a) The UAV needs to explore the mission area and detect fiducials on the ground while recording the locations of each detected fiducials.

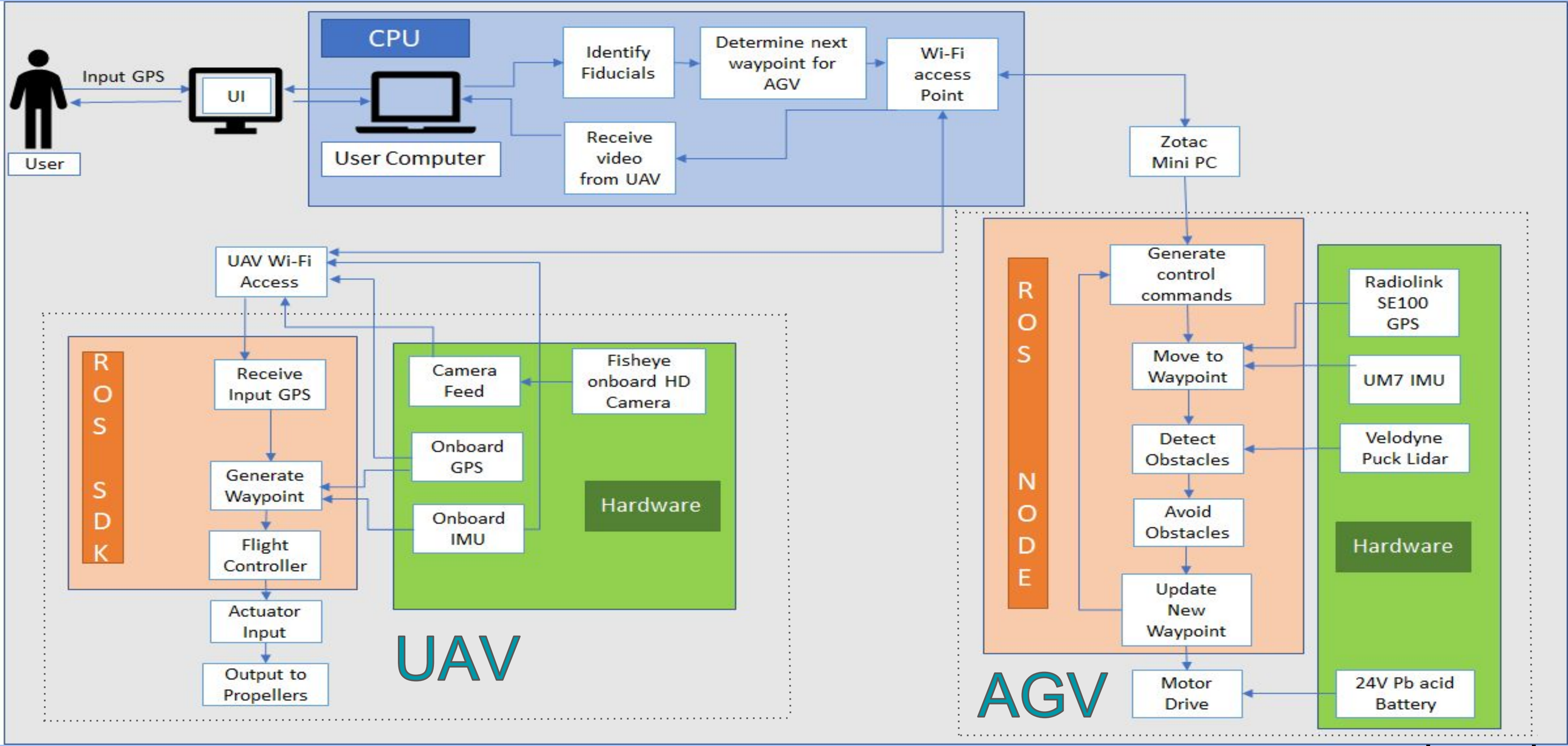
2) AGV shall move to target location based on the inputs from UAV.
   a) UAV provides real time path information to the Husky and it navigates to the goal location.

3) AGV shall move to target location autonomously by itself
   a) It avoids obstacles on the way to the target location.

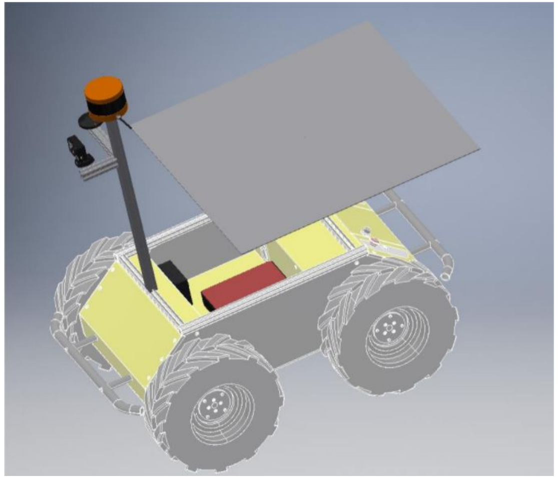
Current System Status

		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 explore the field and provide waypoints 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 move to the target location autonomously by itself
	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

System Overview:



AGV Subsystem - Description

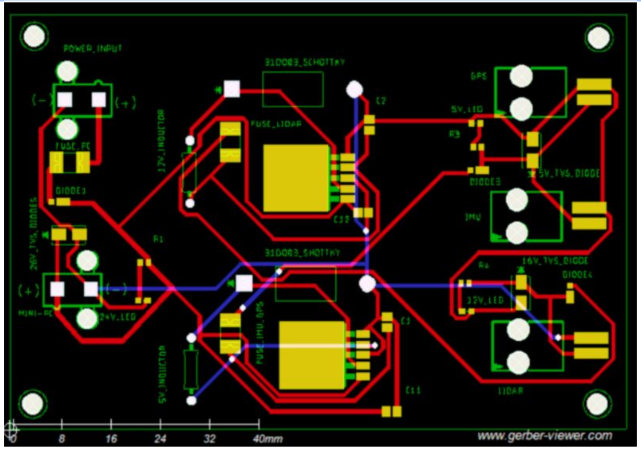


Mechanical:
1) Base - Husky (ClearPath)
2) Wooden Platform - to place UAV(Parrot Bebop2)
3) Mounting for sensors

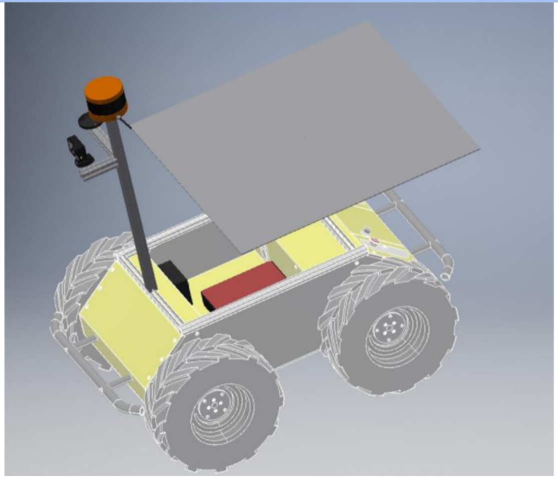
Sensors:
1) LIDAR - VLP-16
2) Razor- IMU - M0
3) RadioLink- GPS - SE-100

Control/Programming:
1) ROS development in C++

AGV Subsystem - Depictions



PDB Schematics and Testing



Lidar Mount and Bebop Base

AGV Subsystem - Current 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) Hardware Setup - Platform
2) Standalone sensor testing
3) GPS based autonomous navigation
4) LIDAR standalone testing

To be done:

1) ~~LIDAR standalone testing and Integrating with system.~~
2) ~~Obstacle Detection~~
3) Path Planning
4) ~~Localization.~~

Challenges:

1) Less Outdoor testing
2) Obstacle properties make detection difficult

UAV Subsystem Status- Description

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~~

Mechanical and sensing capabilities:

1. COTS platform and cannot support add ons.

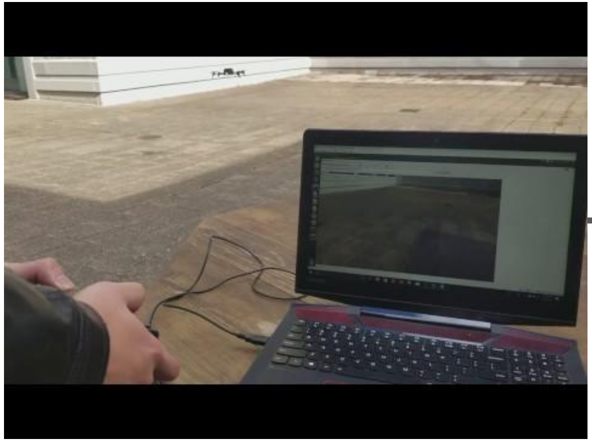
Software:

1. ROS to develop the software stack for autonomous navigation and area exploration by the drone.

Control:

1. To stabilize the drone movement during GPS navigation, we are using a PID control along with a low-pass filter.
2. We are using live video feed from the drone to recognize fiducials on the ground, capture their GPS locations and develop a map based on the fiducials detected.

UAV Subsystem Status- Depiction



Beginning

FVE

Current

UAV Subsystem Status- Current 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.
4) Path Identification by UAV (Based on April Tags Detected)
5) Received video from UAV connected via Access Point

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.~~
5) Bebop area exploration - testing and validation outdoors

Challenges:

1) Less onboard storage memory
2) No onboard Processing Capabilities

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

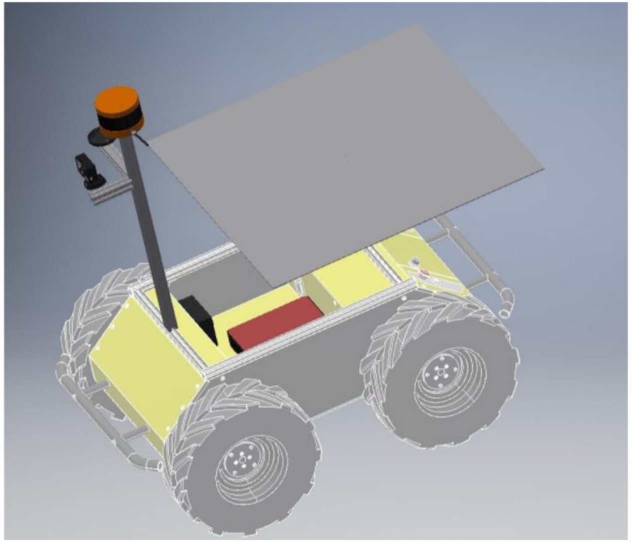
Analysis & Mitigation

1) We faced range issues with the WiFi. We had to mount the WiFi router over the Husky robot.
   - Set up a mesh network of Access Points.
2) We faced some issues with multiple ROS master configuration, so we used separate workstations for UAV and AGV. -Yet to be solved.

Modeling, Analysis and Test Results

- 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.



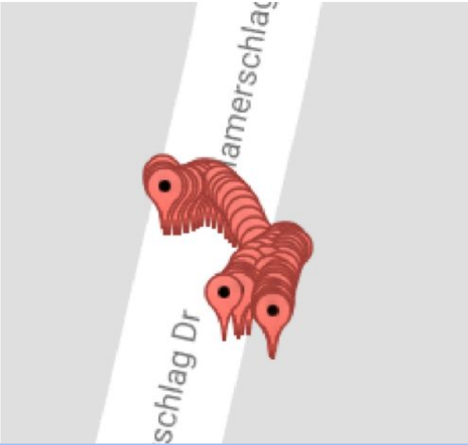
Results: Husky Sensor Mounting and Network Tested for AGV and UAV

Modeling, Analysis and Test Results

- 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.

<u>Results:</u> April Tags are accurately detected from a height of 5m and GPS deviation incorporated in the system.



Modeling, Analysis and Test Results

- 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

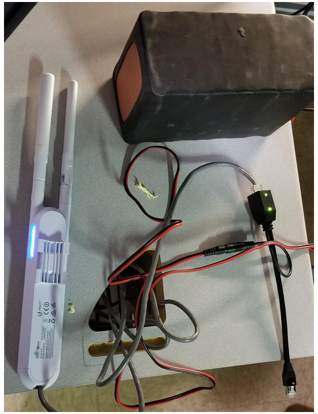
Results: GPS based waypoint navigation achieved well within the system requirements

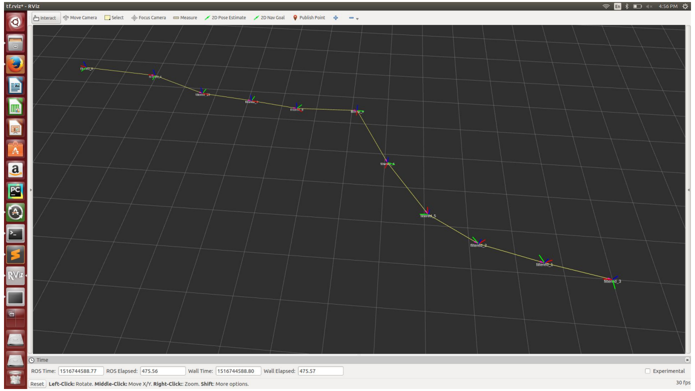
Modeling, Analysis and Test Results

- Access Point Network Setup
  - Switched from DHCP to a fixed IP Access Point Mesh network
  - Increased range of 130 m achieved.

- Husky Obstacle Detection & Avoidance
  - Used Velodyne Point cloud to detect and segment obstacles.
  - Applied reactive approach to identify and avoid obstacles

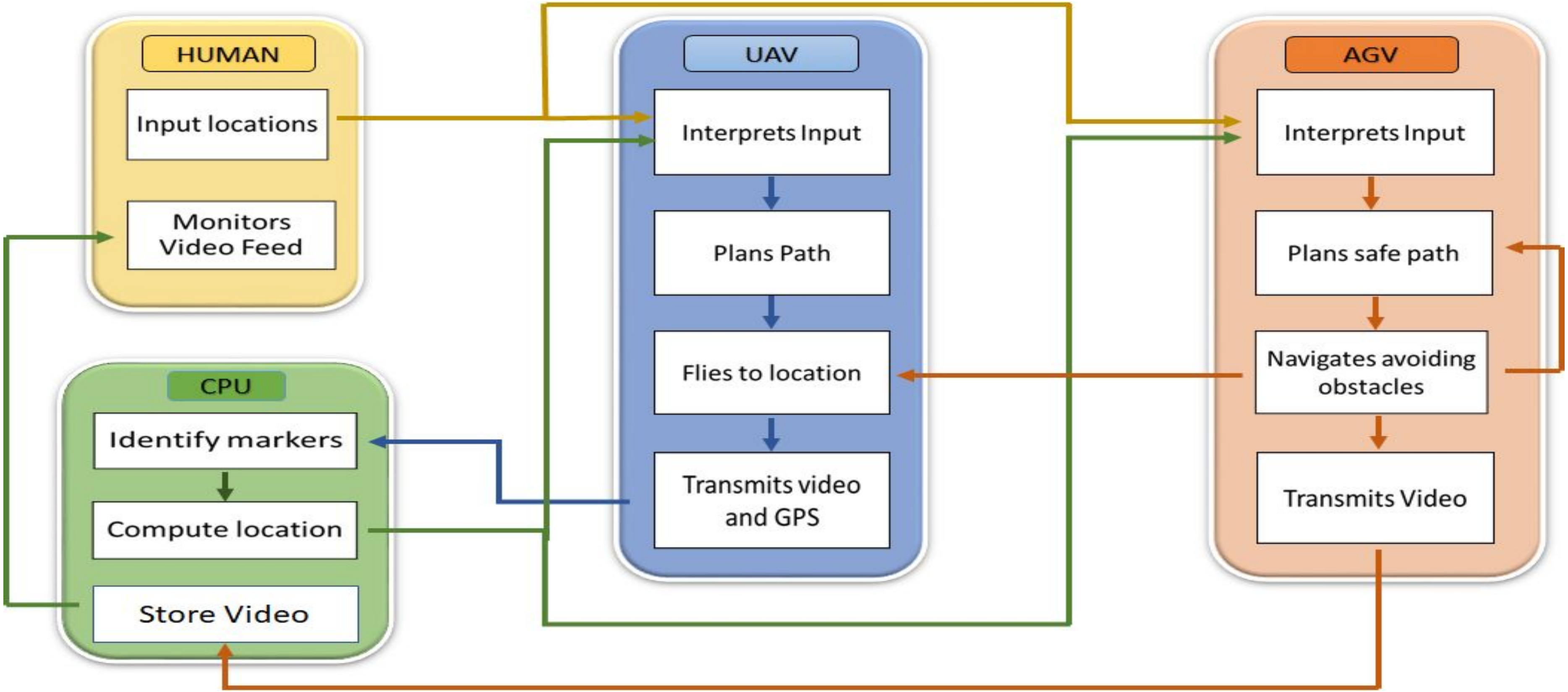
- Bebop Area Exploration
  - Generated a map of detected April Tags to be used for navigation by Husky.





<u>Results:</u> Configured a long range and secured Access Point Mesh Network and tested with both systems. Husky is able to avoid static obstacles on its path. Bebop is able to achieve area exploration to detect and map April Tags.

Functional Architecture and Status Update



Key design challenges and To Do

UAV

1) Improve outdoor detection of April Tags suppressing the effect of glare
2) Detection of walls from video frame using CV and pruning edges from the graph
3) Development of a path planning algorithm to expand the graph

AGV

1) Local path planning to avoid static obstacles and navigate to local objectives
2) Integration of UAV on a single network
3) Comparison of AGV v/s UAV-AGV system.

Schedule Status

	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	

5.4) Path Planning
6.3) UAV Exploration

1.4) Communication Layer

- mostly on schedule, setup of communication layer is bit delayed.
- First test AGV with UAV proxy
- After that communication layer will be setup

Project Management

Date	PR	Capability Milestone	Test Criteria
21 March	PR-10	AGV platform works with drone proxy	AGV should work with given GPS waypoints
4 April	PR-11	Integrate AGV and UAV subsystems	AGV and UAV communication and coordination
		Standalone AGV test	AGV should return from the dead-end path
16 April	PR-12	Fully integration test and record-run-through for SVE	Full system test for recording
25 April	SVE	Full System Capability	As explained in SVE

Spring Validation Experiment

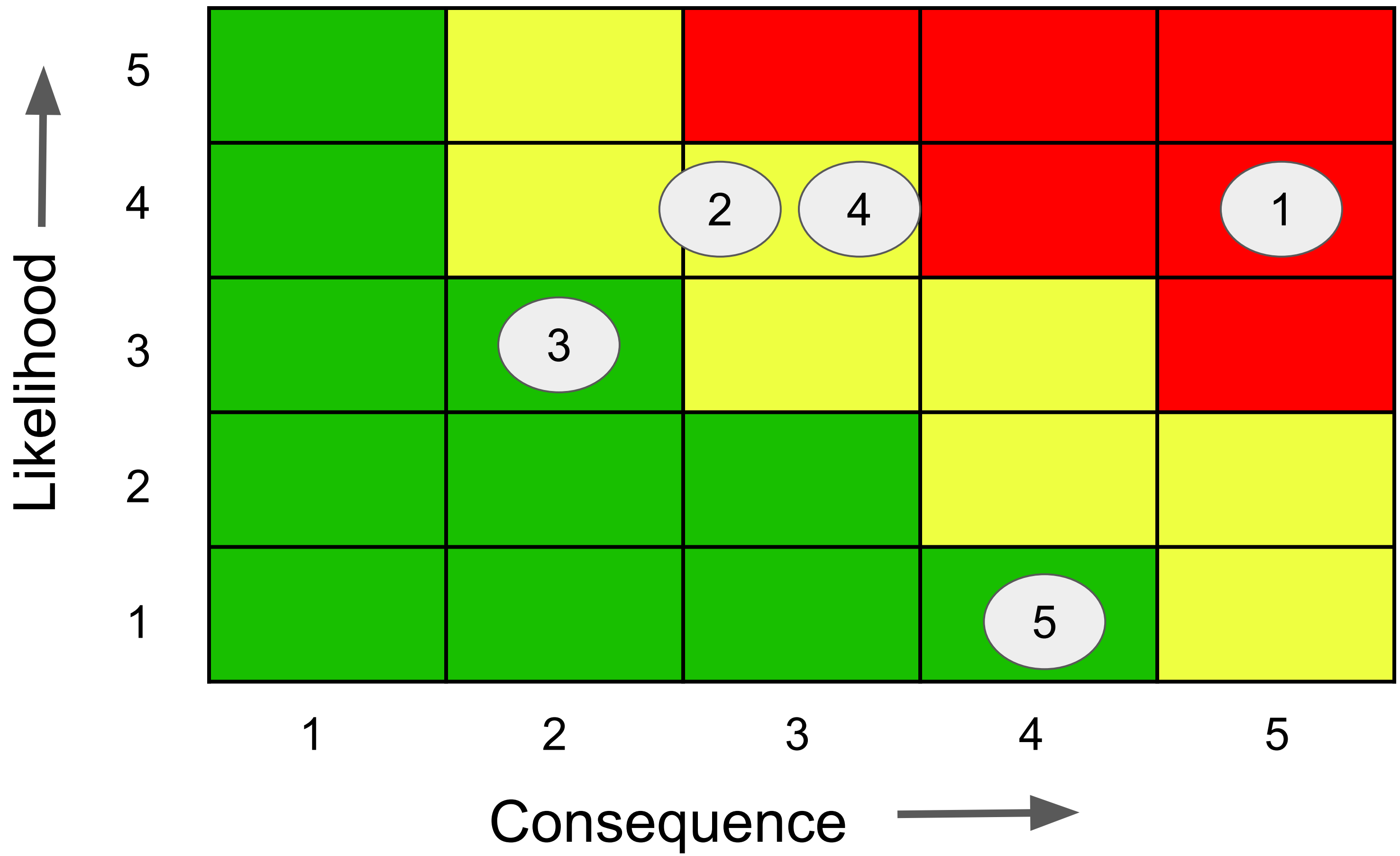
- Location: CFA Lawn
- Needed equipment: 50 April Tags, Husky, Bebop 2, pylons, safety tape, wooden walls/dividers
- Size of area: 50 x 50 m, flat grass
- Desired weather: Sunny with no wind

Spring Validation Experiment

Test Sequence	Description	Performance Measures
A.1	Setup the environment with April Tags, obstacles, and wooden walls	
A.2	Setup the system in heterogeneous mode and enter the target destination into the system	
A.3	The heterogeneous AGV + UAV system autonomously navigates to target destination	• UAV identifies 90% of fiducials • The AGV arrives within 3 meters of the target destination • The AGV navigates the path in less than 10 mins • The AGV avoids 80% of the stationary obstacles
A.4	Remove the UAV from the system and reset it up into AGV-only mode	
A.5	The AGV autonomously navigates to the target destination	• The AGV navigates slower (yet to quantify) than the heterogeneous system

Major Risks and Mitigation

S. N	Risk	L	C	Type	Mitigation strategy	Owner
1	Drone testing not possible during night and in bad weather	4	5	Schedule	Optimize weekends and good weather for testing	Danny
2	Drone getting stuck in Trees (not wanting to come down)	4	3	Technical	Buy more drones and allocate reserve budget	Yuchi
3	Limited system network bandwidth	3	2	Technical	Analyze offline planning	Pratibha
4	System integration takes up lot of time	4	3	Schedule	Start system integration early and integrate subsystems whenever possible	Pulkit
5	Availability of Drone Testing Location and Licensing	1	4	Schedule	Acquire FAA licensing & acquire permissions to fly on campus	Rahul



Likelihood
5
4
3
2
1
2
4
1
3
5
1
2
3
4
5
Consequence

Spending Plan

Sponsored Equipment

SN	Component	Cost	Source
1.	Parrot Bebop 2	$830	Sponsored
2.	Husky	$18600	Sponsored
3.	Velodyne Puck	$8000	Sponsored
4.	Ubiquiti Bullet	$89	Inventory

Bought Equipment

SN	Component	Total Cost
1.	ICs and PCB components	$205.51
2.	Battery	$120
3.	Wires and Electrical	$196.04
4.	Mini-PC upgrades	$468.94
5.	IMU	$49.95
6.	Backup Bebop	$275
7.	Ubiquiti Wifi Access point	$200
8.	Misc.	$596.56
	Total	**$2112**
	% Used	**42.24%**

Planning to buy

SN	Component	Total Cost
1.	Wall dividers for obstacles and misc items for SVE	$400
	Total	**$400**

Thank You!