Team **H**

Critical Design Review

PhoeniX

UAV-AGV Firefighting

Team Members:

- Akshit Gandhi
- Parv Parkhiya
- Shubham Garg
- Zhihao(Steve) Zhu

Sponsors:

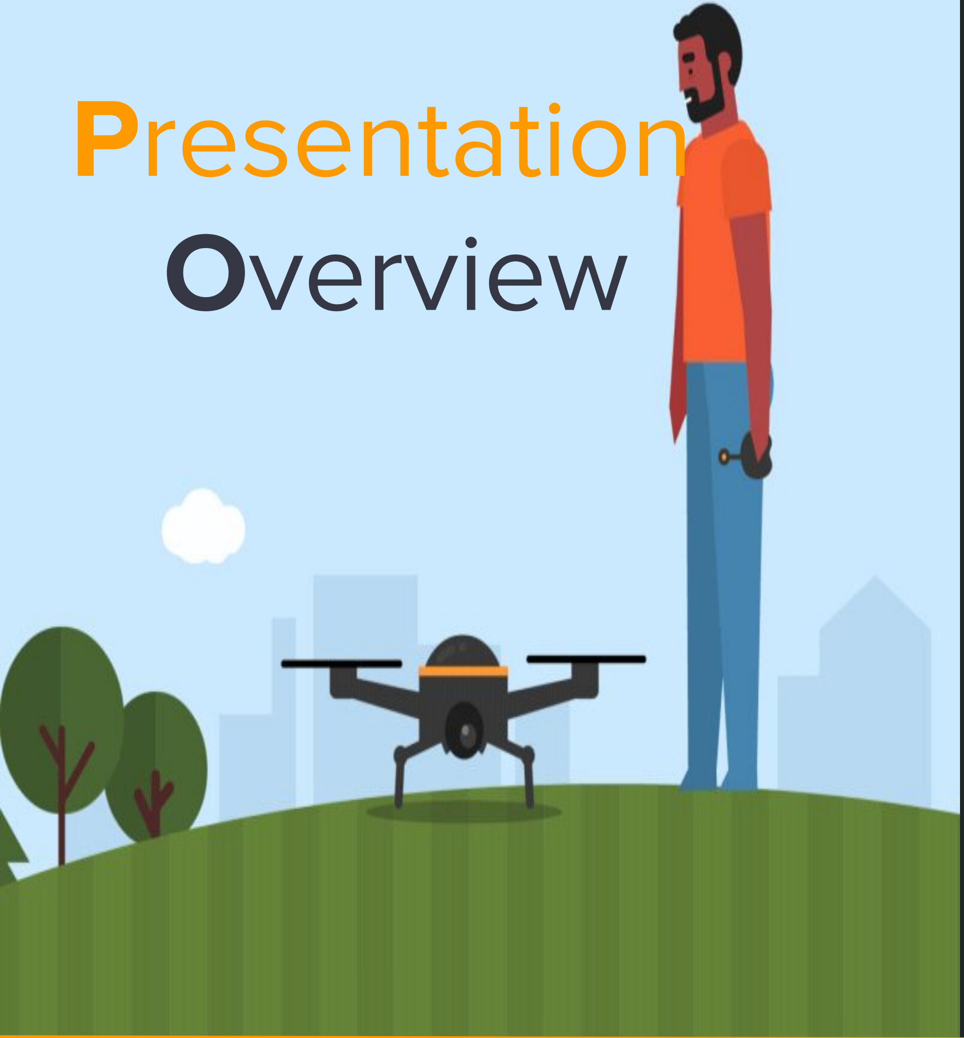
- Oliver Kroemer
- Scherer Sebastian

Mentors:

- Dimi Apostolopoulos
- John M. Dolan

THE ROBOTICS INSTITUTE

Presentation Overview



- Project Description
- Use Case
- System-Level Requirements
- Functional Architecture
- Cyber-Physical architecture
- Current system status
- Project management
- Conclusion

Project Description

PhoeniX team proposes an multiagent heterogeneous UAV (Unmanned Aerial Vehicle) & AGV (Autonomous Ground Vehicle) fire-fighting system with navigation, perception capabilities and mechanism to deploy fire extinguishing material at best can “extinguish” the “simulated” fire and at very least can act as first response collecting information about surroundings (map) and location of fire which human firefighters can use to make better judgements.

When given a fire alarm signal, the autonomous multi-agent system can autonomously search for fire inside the building as well as surrounding and deploy the extinguishing material to put out fire.

PhoeniX

Use Case

PhoeniX fire fighting system **receives** a notification (from any source) regarding a fire in a nearby building. The system becomes active and 3 UAVs **take-off** from the station (carrying 1 kg of extinguishing material each) and an AGV also **drives** towards the building.

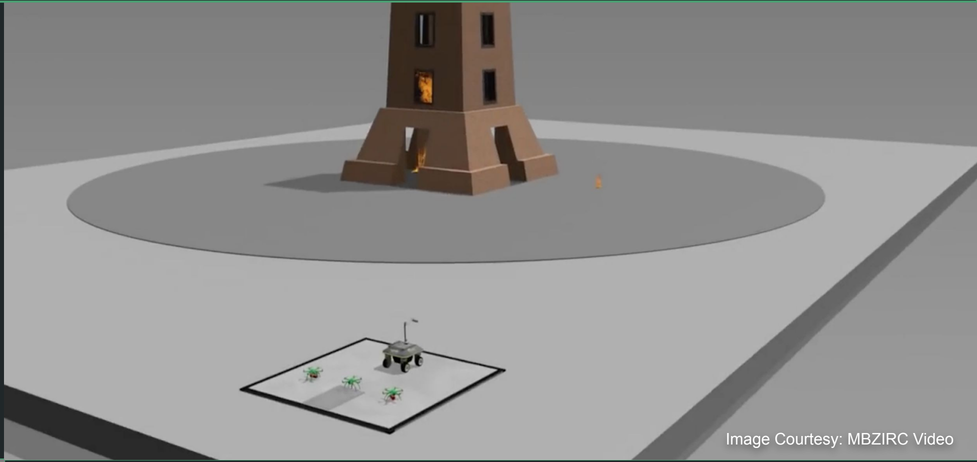


Image Courtesy: MBZIRC Video

Image Courtesy: MBZIRC Video

All the robots **coordinate** and **collaborate** to optimally **explore** the surrounding by **avoiding** obstacles while **creating a map** (accumulate less than 5 m drift for every 100 m of distance travelled) of the environment. UAV-2 **detects fire** (localize fire with less than 1 m error) in the building at two locations : ground floor and 1st floor, it shares that information with other systems.

Use Case Continued

The system **divides** the task of extinguishing fire at those two locations. The AGV is **assigned** the task of extinguishing fire at the ground floor and 2 UAVs are assigned the first floor. While the 3rd UAV is still **exploring** to find potential fire locations. The AGV uses **sweeping strategy** to extinguish fire whereas the UAVs use some different mechanism to **extinguish** fire depending on the fire location.

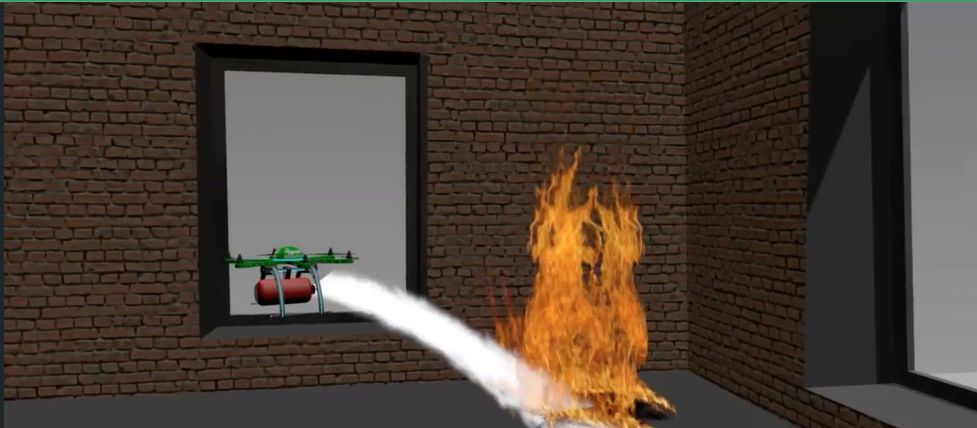


Image Courtesy: MBZIRC Video

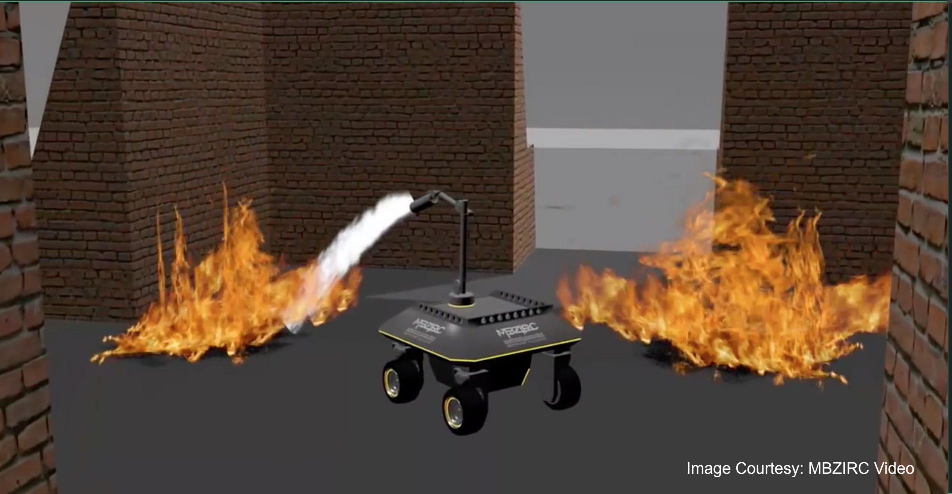


Image Courtesy: MBZIRC Video

Every robot **monitors** their fire extinguishing progress. AGV **reports** that it has successfully extinguished the fire. When the 1st and 2nd UAVs are out of the fire fighting material, they **request** help from the 3rd UAV. The third UAV comes and extinguishes the fire. After **ensuring** that there is no more fire in the building the UAVs **land** back within 5 minutes at the station along with the AGV driving back.

System-level Requirements

Mandatory functional and corresponding performance requirements:

Requirement ID	Requirement Description	
F.R.1	Take-Off and Land from base station	
	M.P.1	Land within 5 m radius from center of base station for UAV and 1 m for AGV
F.R.2	Plan Trajectory	
	M.P.2	Explore 50 m x 60 m x 20 m environment with greater than 60% coverage (robot has seen and identified potential fire) in 10 minutes or less
F.R.3	Create real-time map	
F.R.4	Localize itself in the environment	
	M.P.3	Accumulate less than 5 m drift for every 100 m of distance travelled
F.R.5	Traverse desired trajectory	
	M.P.4	Maximum error between desired and actual trajectory should be less than 1 m

System-level Requirements

F.R.6	Avoid collision with obstacles and other UAVs/AGV	
	M.P.5	Keep 0.75 m minimum distance between system and obstacles
F.R.7	Detect Fire	
	~~M.P.6~~	~~Achieve fire detection AUC (Area under curve) of ROC (Receiver Operating Characteristic) of 0.65~~
	M.P.6	Detect fire from a maximum 1.5 m away - in the line of sight of the UAV and UGV
F.R.8	Localize and Monitor Fire	
	M.P.7	Localize fire with less than 1 m error
F.R.9	Deploy material strategically	
	M.P.8	Carry 1 kg of extinguishing material each
	M.P.9	Deposit 40% deployed extinguishing material on the target area of minimum 0.5 m x 0.5 m
F.R.10	Coordinate between different UAVs & AGV	
	M.P.10	Reliable communication within 25 m

System-level Requirements

Mandatory non-functional requirements:

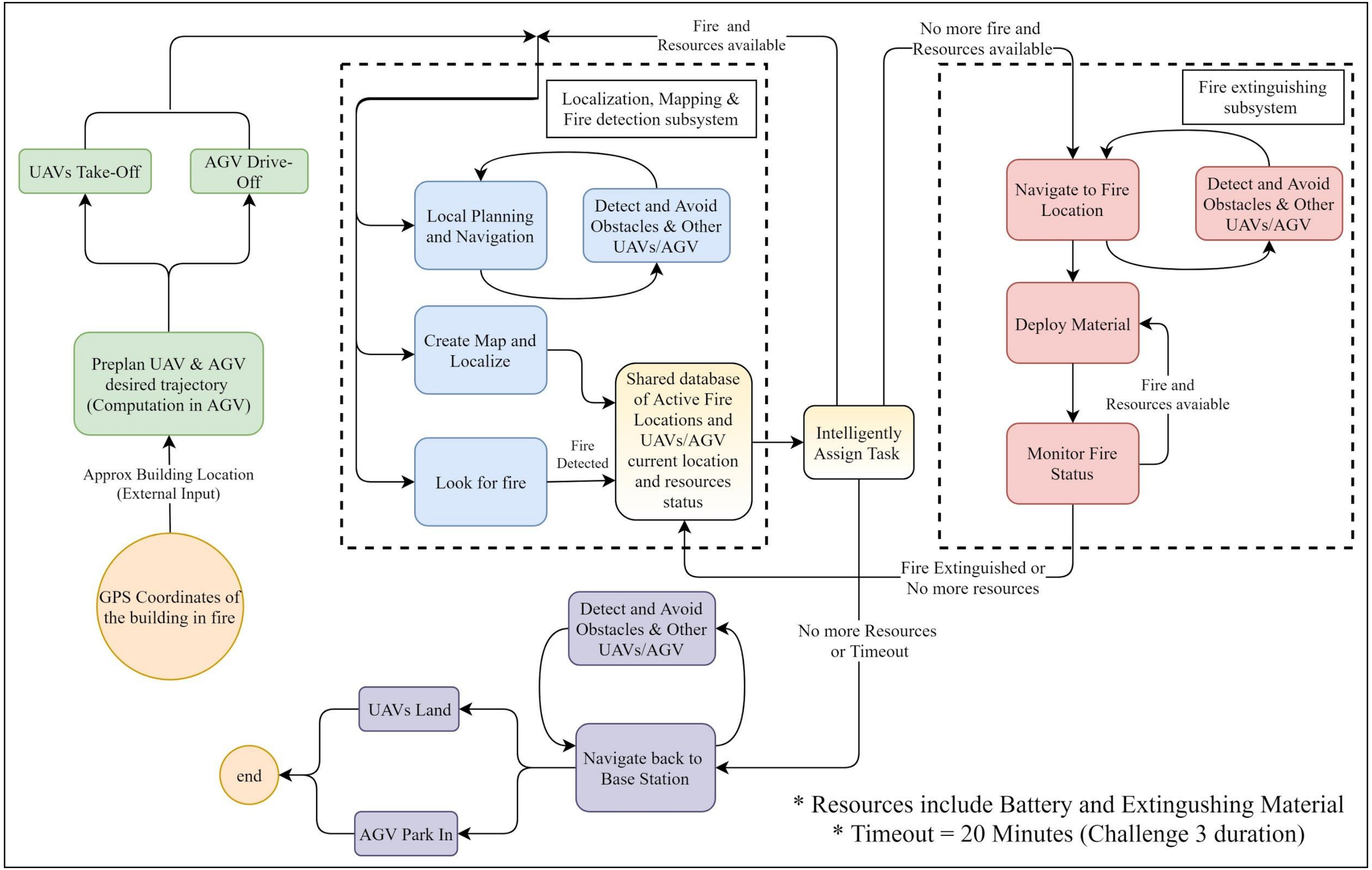
Requirement ID	Requirement Description
M.N.1	Fit in the size of 1.2m x 1.2m x 0.5m (UAV)
M.N.2	Fit in volume of 1.7m x 1.5 m x 2m (AGV)
M.N.3	Feature kill switch for safety
M.N.4	Feature user interface
M.N.5	Maintainable with easily replaceable components like motor, battery, ESCs etc
M.N.6	Resist wind speed upto 2-3 knots
M.N.7	Interoperate with other MBZIRC team’s systems by the means of functional modularity

System-level Requirements

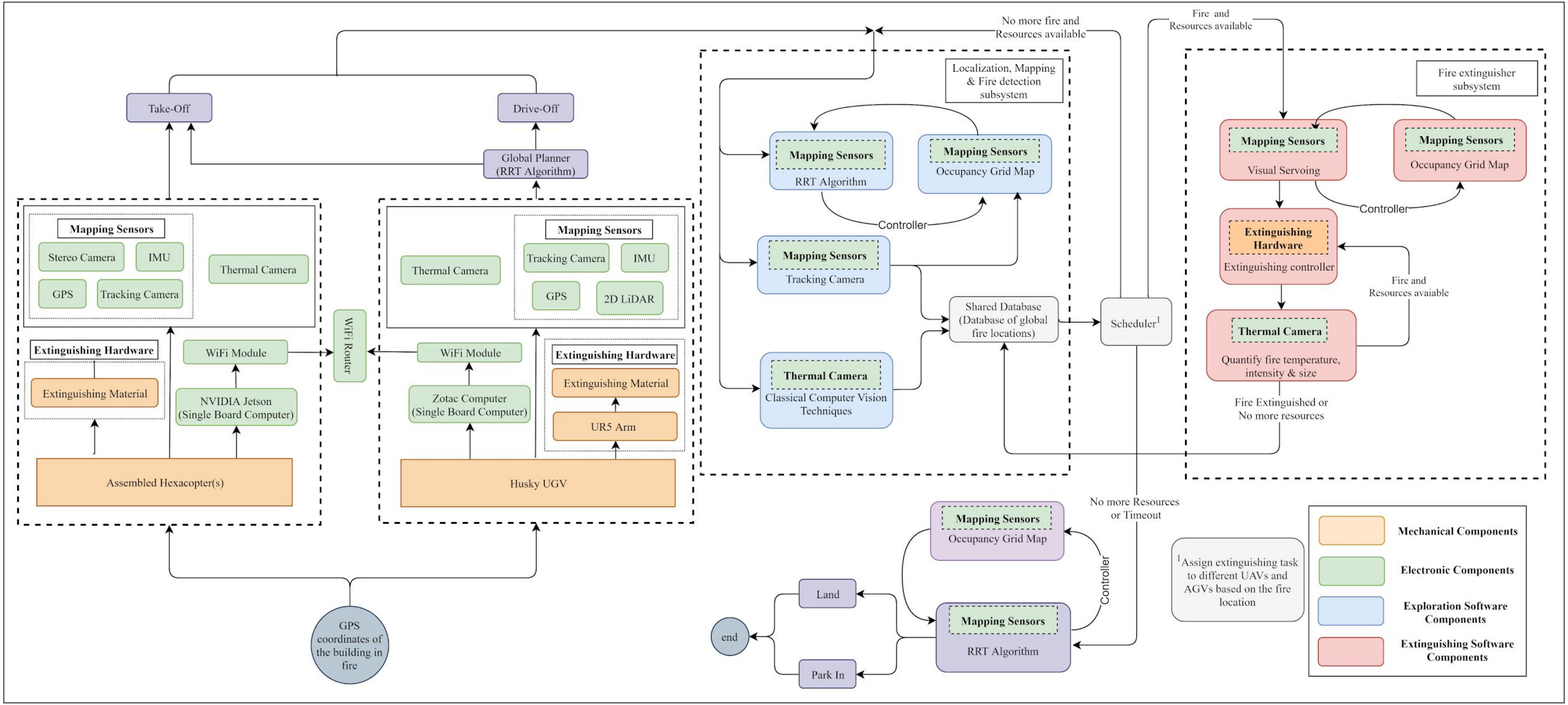
Desirable non-functional & performance requirements:

Requirement Type	Requirement ID	Requirement Description
Non-functional	D.N.1	Perform non-overlapping tasks (mapping, extinguishing, etc)
	D.N.2	Create a common global map by merging individual maps from different systems
	D.N.3	Portable (weight, compact size/form factor within 0.5m x 0.5m x 0.25m)
	D.N.4	Economical (system costs under $7000)
	D.N.5	Scalable (in terms of manufacturing)
	D.N.6	Feature Prop Guards
Performance	D.P.1	Dock to refill the extinguishing material when it is below 10% capacity within 5 minutes
	D.P.2	Dock for battery recharge/battery replacement when it is below 20% capacity within 5 minutes
	D.P.3	Detect humans trapped inside the building with 60% accuracy
	D.P.4	Notify authorities about the location of people trapped inside to plan rescue mission within 45 seconds of human detection

Functional Architecture



Cyber-Physical Architecture



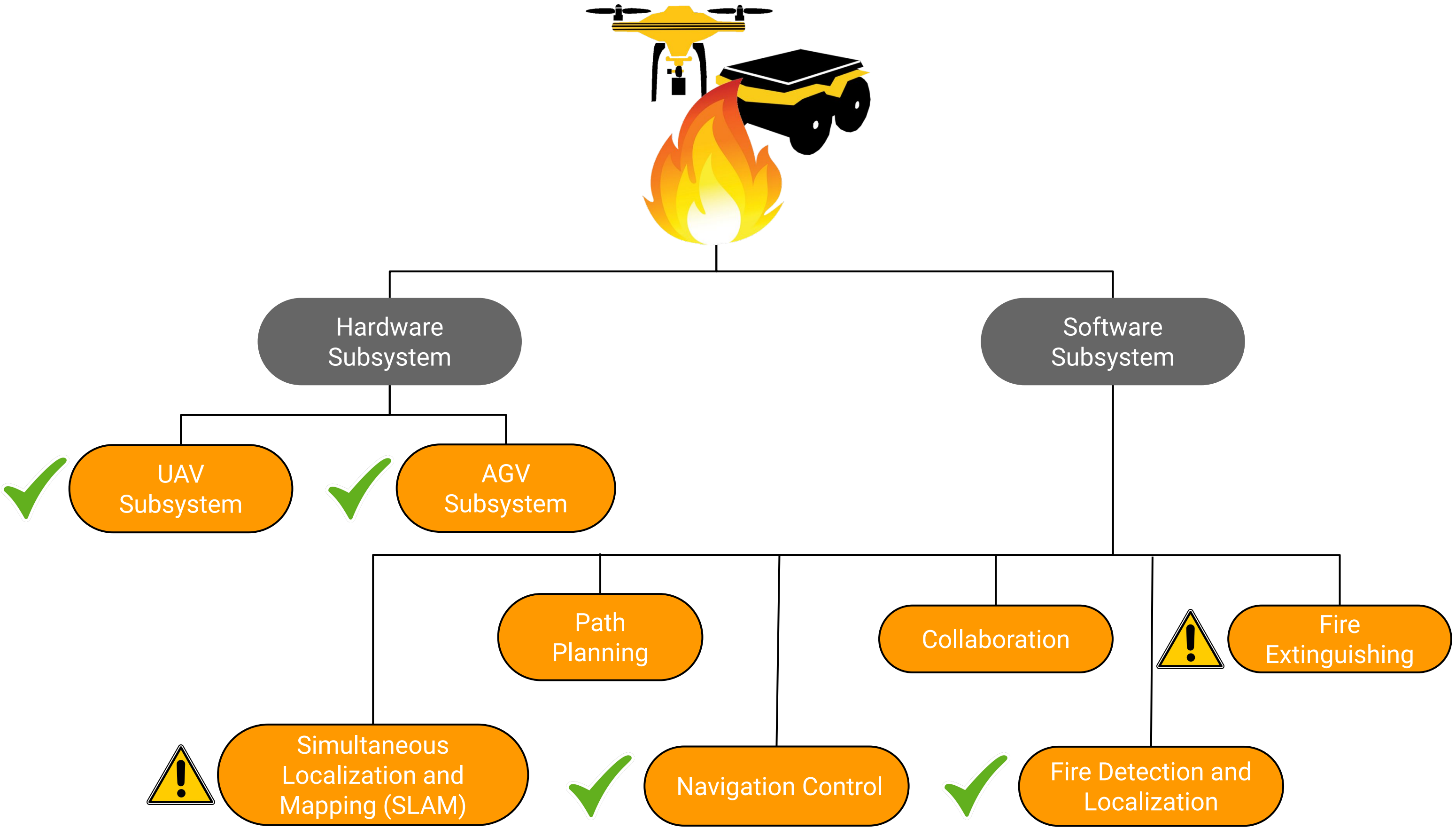
Current System Status: Targeted requirements

Requirement ID	Requirement Description	Corresponding Subsystem
M.P.1	AGV successfully parks in within **1 m** radius of the base station center.	Navigation Control
M.P.7	AGV successfully detects fire within **1.5 m** distance.	Fire detection and Localization
M.P.5	AGV successfully stops as soon as it detects obstacle **0.75 m** away and does not crash into it.	Navigation Control
M.P.8	AGV points a laser pointer within the water bag of size **7.5 inch x 9.5 inch**	Fire extinguisher

Current System Status: Targeted requirements

Requirement ID	Requirement Description	Corresponding Subsystem
M.P.4	UAV performs the desired movements within **1 m** error radius (following the trajectory)	Navigation Control
M.P.7	UAV successfully detects the hot water bag from a **maximum** of **1.5 m** distance.	Fire detection and localization
M.P.8	UAV points a laser pointer within the water bag of size **7.5 inch x 9.5 inch.**	Fire extinguisher
M.P.9	UAV successfully lifts **1.5 KG** payload	Navigation Control

PhoeniX Fire Fighting System



Subsystem Overview

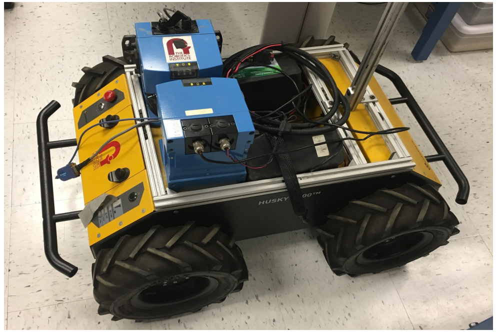
PhoeniX

Hardware Subsystem

UAV Subsystem

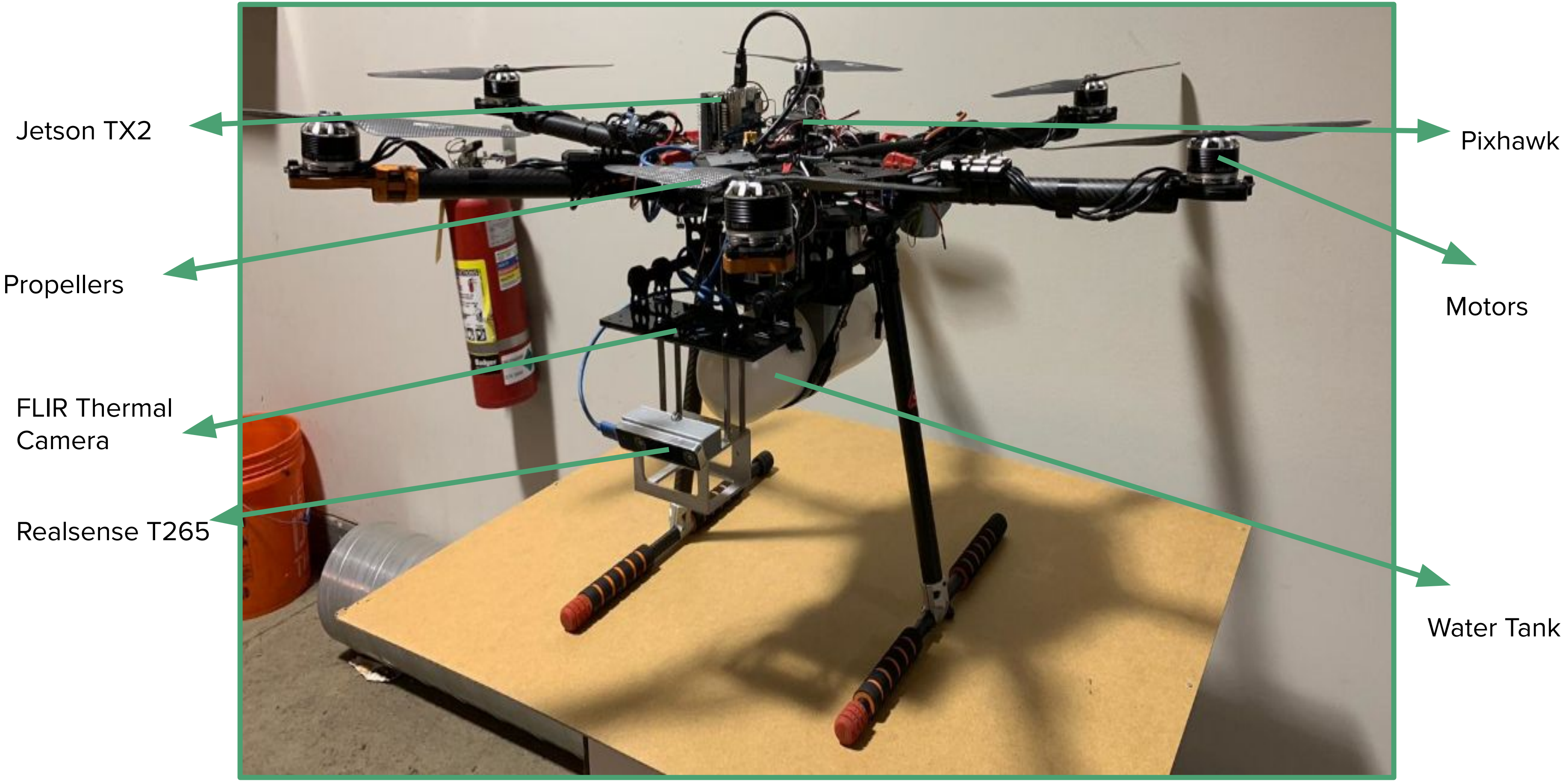
AGV Subsystem

Hexacopter



Husky AGV

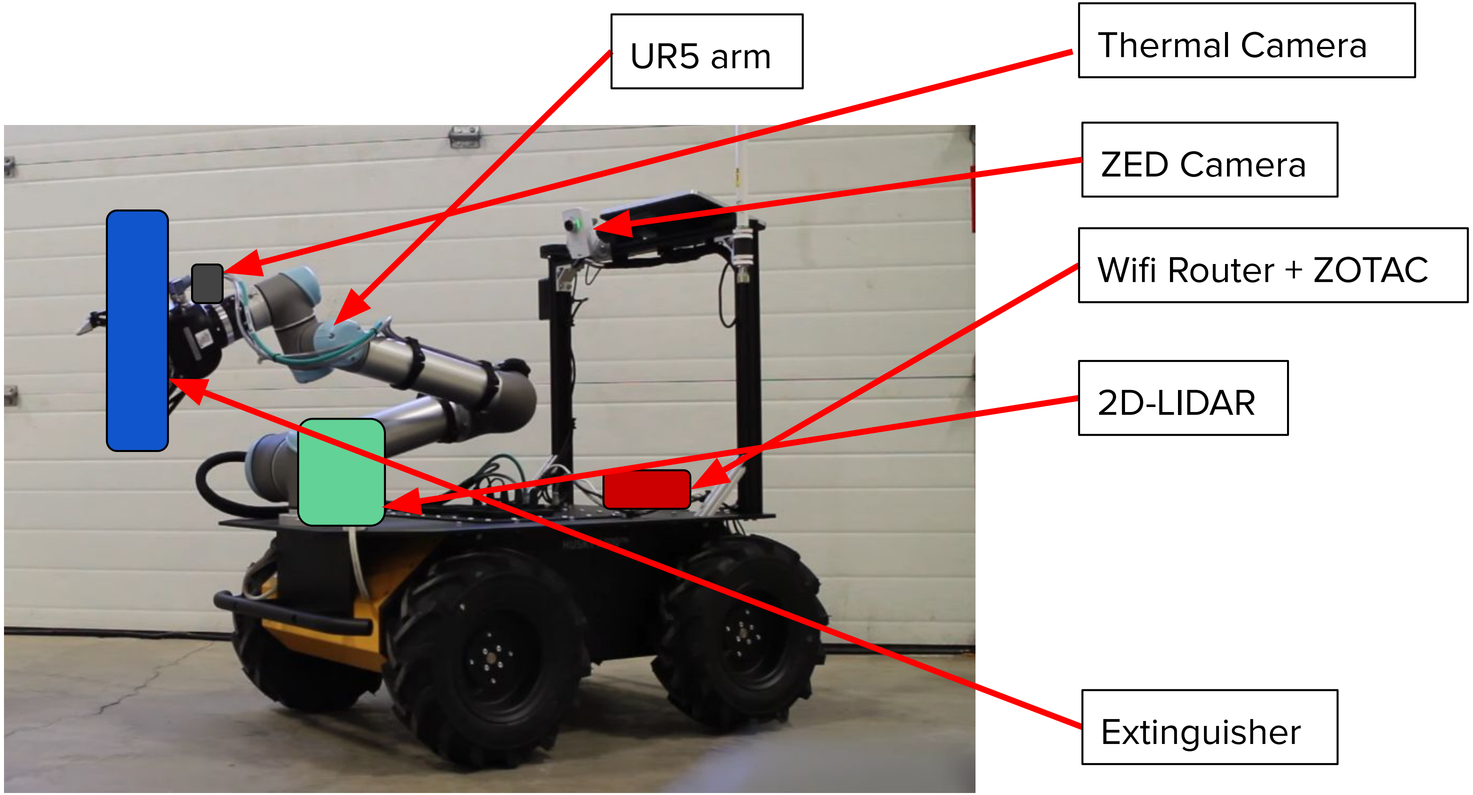
UAV Hardware Subsystem Current Status



Reserve UAV Current Status

Reserved UAV frame with shortened arm length as per MBZ size constraints

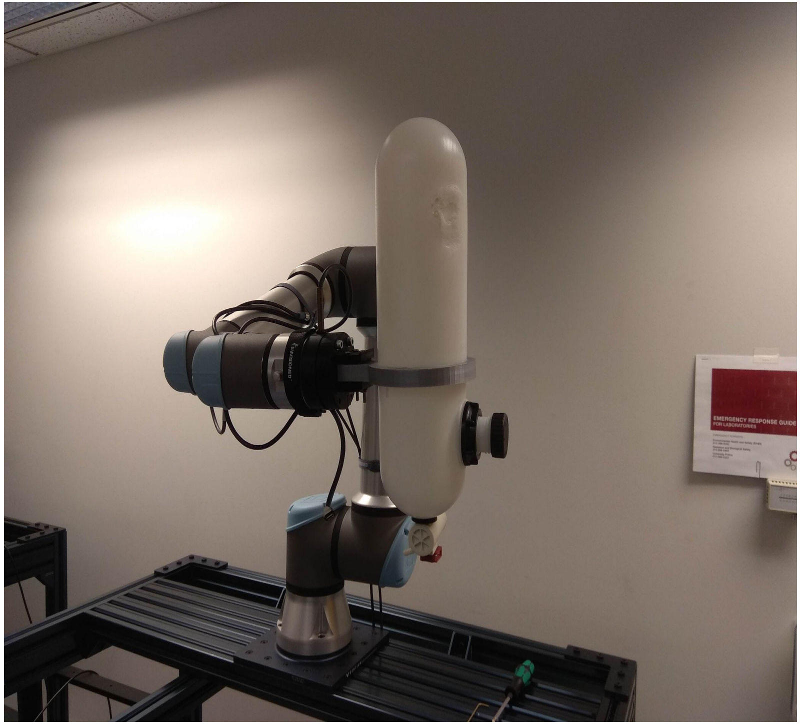
AGV Hardware Subsystem Overview



Husky with UR5 arm (Illustrative image)

AGV Current Status

Current Husky with LIDAR, ZOTAC and PCB



UR5 arm with the extinguisher attached using custom 3D printed attachment

UR5e Current Status

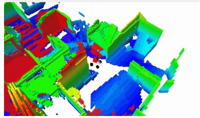
UR5e arm moving towards fire location using visual feedback

Subsystem Overview

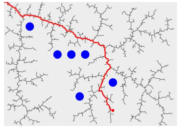
PhoeniX

Software Subsystem

Simultaneous Localization and Mapping (SLAM)



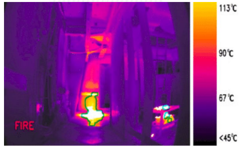
Path Planning



Navigation Control

Collaboration

Fire Detection and Localization



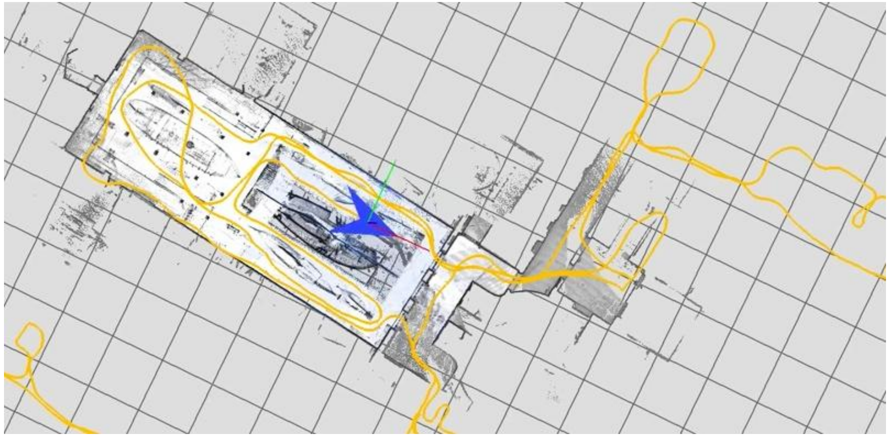
Fire Extinguishing

SLAM Subsystem Overview

- Our system needs to function outdoor and indoor both. GPS localization is not reliable indoors so we need a Simultaneous Localization and Mapping (SLAM) capabilities to keep track of current location of robot in global map.

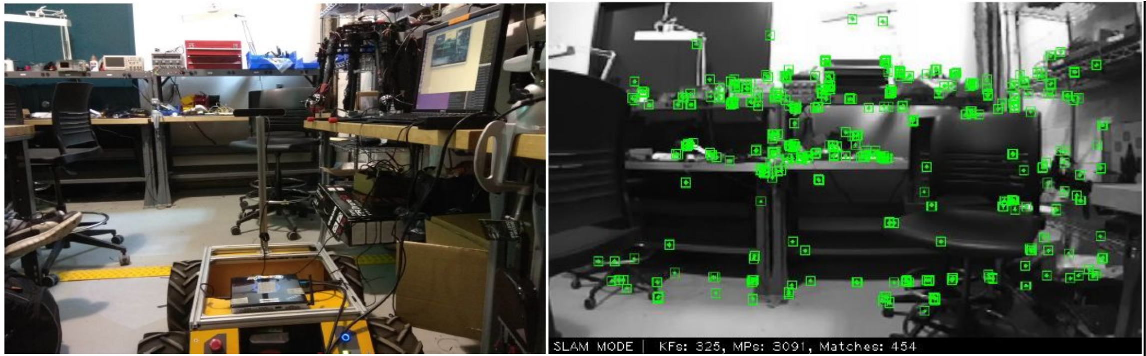
- System uses ORB-SLAM2 on stereo images from ZED camera. ORB-SLAM2 provides reliable loop closure ability which would allow it to relocalize in case feature tracking gets lost.

- Vision SLAM data will be fused with IMU, GPS sensors values to get even higher accuracy in localization.

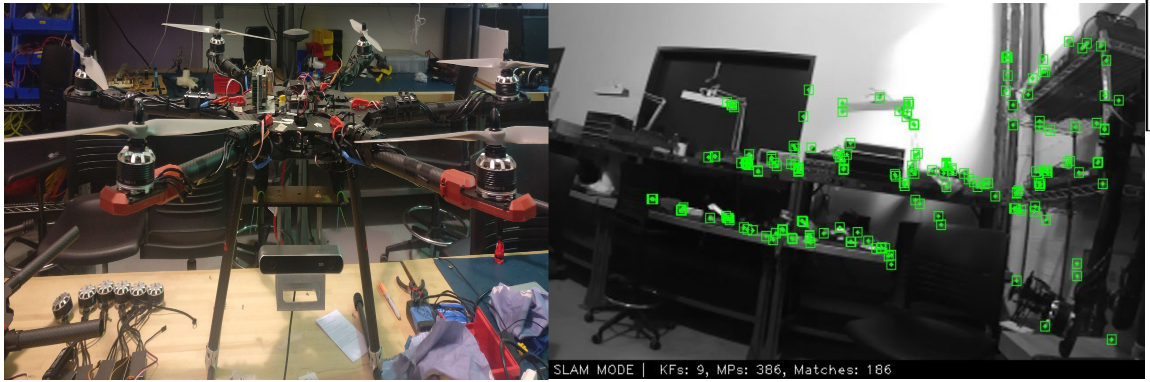


Example robot trajectory and current location in map

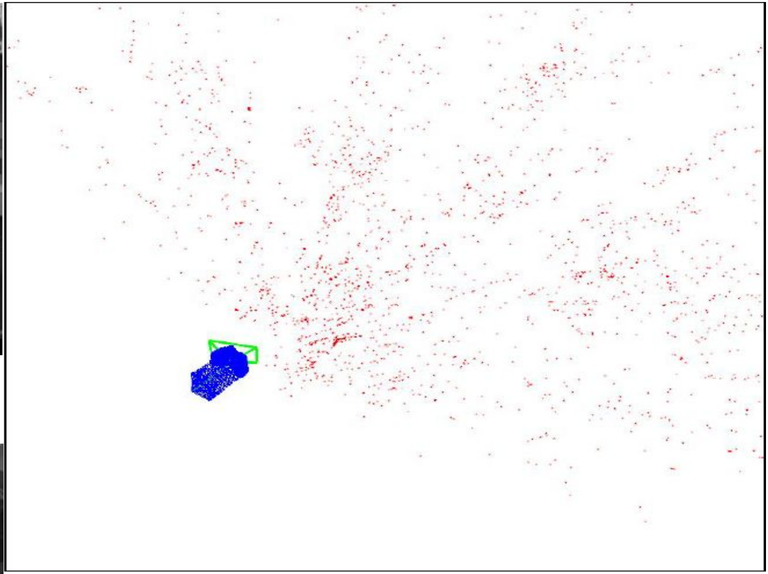
SLAM Subsystem Current Status



Right image shows features detected by AGV



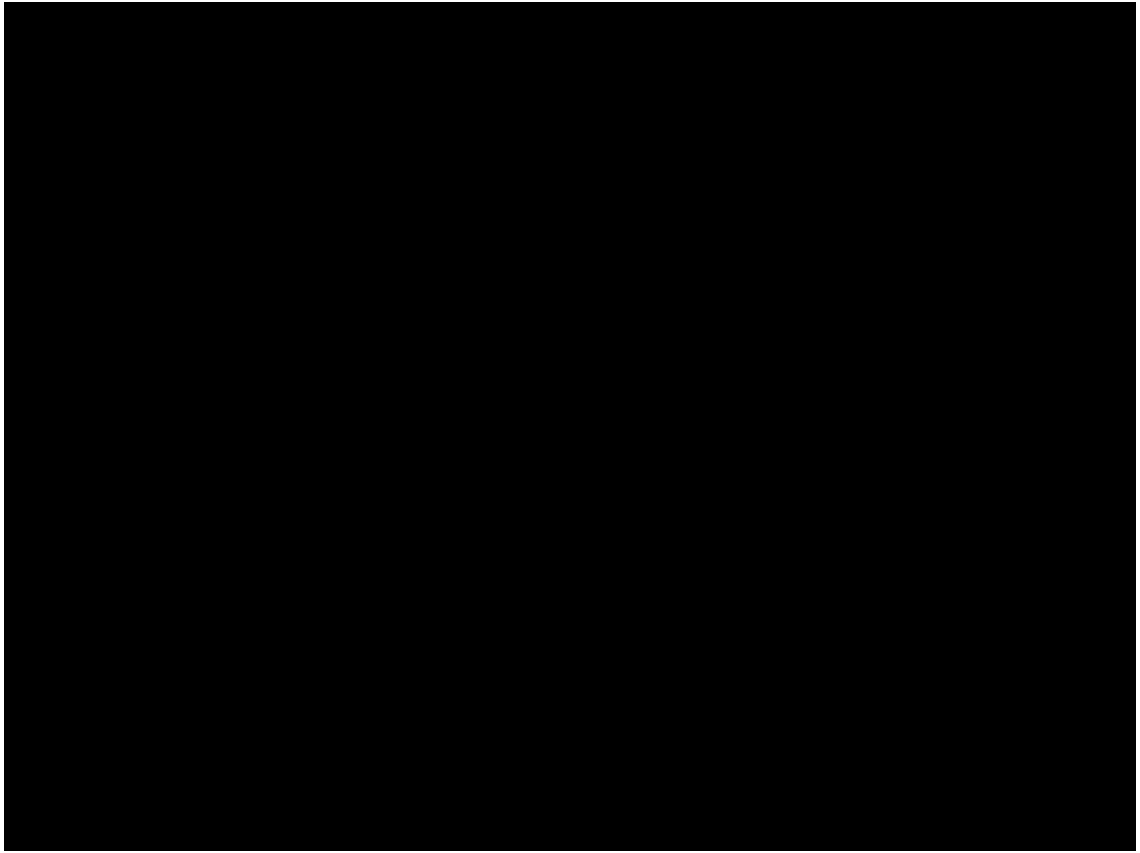
Right image shows features detected by UAV



Husky moved forward and rotated 45° (Red points shows the 3D point cloud of the detected features, blue rectangles represents past camera position and green rectangle represents current location.)

SLAM Subsystem Current Status

The tracking camera is used to get state estimation information for both of our UAV and UGV.



Navigation Control Subsystem Overview

- Navigation control system takes care of low level control of robot. It takes inputs from SLAM subsystem and path planning subsystem along with GPS, IMU, LIDAR sensor data to generate final control signals (eg MAVROS for UAV) to pixhawk controller

- It also creates fail safe to avoid collision with obstacles. While ORB_SLAM2 is great when it comes localization and loop closure but gives very sparse point cloud which is not usable for collision avoidance. System will use ZED stereo camera wrapper to generate dense point cloud.

- Navigation Control subsystem will use ROS Action Server framework to enable safety response in case of failure of higher level node.

MAVROS link between Jetson and Pixhawk Controller

Navigation Control Subsystem Current Status

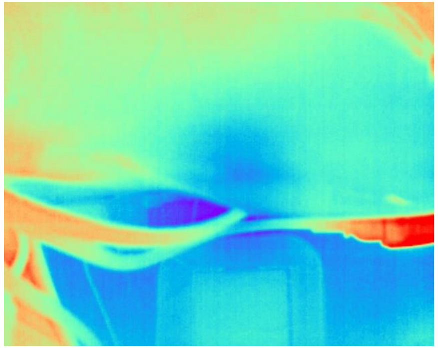
ROS motion server framework

- To improve the reliability of the system we are using ROS action server action which ensures that if any ROS node crashes there is a mechanism to help the node recover from that state.
- More specifically, motion server creates an object that has everything it needs to produce commands at fixed rate that actuate the robot in such a way to achieve a high level goal.
- Task Commands and Task States facilitate this process.
- Makes the high-level state machine very easy to write.
- Also, the user might want the ability to cancel the request during execution or get periodic feedback about how the request is progressing. So, we are integrating all the high level tasks in the ROS motion server framework.
  Ex: land, explore, fly to fire, extinguish fire etc.

Fire Detection & Localization Subsystem Overview

The fire detection subsystem has the task of detecting regions of interest which is the simulated fire and simultaneously capture the location of the fire so that it can be used to update the shared database which contains the list of all the fire locations.

- The subsystem uses a classical method of computer vision using various morphological operations coupled with a fusion from the RGB images obtained from the stereo pair which is onboard.
- The thermal camera captures the near infrared spectrum of the scene and which helps in segmenting the regions of high energy from the low energy resulting in some interesting interesting patterns.
- These patterns along with thresholding can give us the regions that we want but with shape approximation we can better rule out the various outliers.
- Once a region is identified we need the exact 3D world coordinates of the region which is done using the RGB stereo pair which is onboard the PhoeniX system.



Grayscale image converted to color based on intensities

Nature of Simulated Fire

We are planning to use high density rubber hot water bags for simulating fire for SVD & FVD.

Fire Detection & Localization Subsystem Status

Thermal Raw Input

Thermal Processed Input

Fire Extinguishing Subsystem Overview

- Orients UAV/AGV with respect to fire by providing waypoints to Navigation Control Subsystem using Visual Servoing method. e.g. if system is far away from the fire, waypoints that leads the system to fire are provided.

- Further, it provides control signals to microcontroller which in turn activates the actuators of extinguishing deploying mechanism to point it directly at fire. Finally, it gives signal to deploy the material and provides alert to Navigation Control Subsystem so that it can handle recoil and payload change.

- It keeps monitoring the fire status using thermal image data to stop deployment if fire is extinguished.

- It also keeps track of how much extinguishing material is available on UAV by computing current available thrust and using load sensor on AGV.

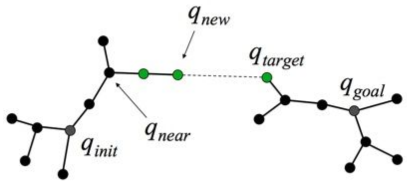
UAV deploying water on actual fire
(Illustrative Purpose Only)

Fire Extinguishing Current Status

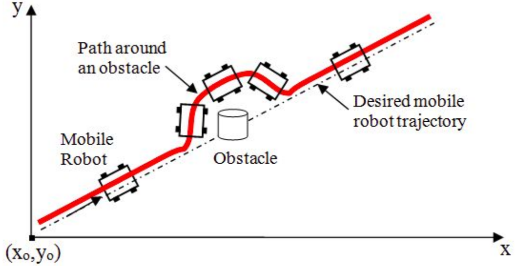
Path Planning Subsystem Overview

The path planning subsystem is divided into two smaller subsystems namely:

1. Global Planner
2. Local Planner



- The global planner is responsible for creating a trajectory from the base station to the fire location. Which is a building for the sake of the competition (using the GPS coordinates of the location).
- The subsystem uses a RRT algorithm to precompute trajectories for each of the PhoeniX UAV and UGV subsystem such that the UGV reaches the ground floor and the UAVs capture different levels of the building when they begin to look for fire and start to explore.



- The local planner which is a variant of RRT is responsible for obstacle avoidance while executing the trajectory generated by the global planner.
- The local planner also helps in navigating indoors by entering the building through a opening such as a window for UAV and a door for UGV.

Collaboration Subsystem Overview

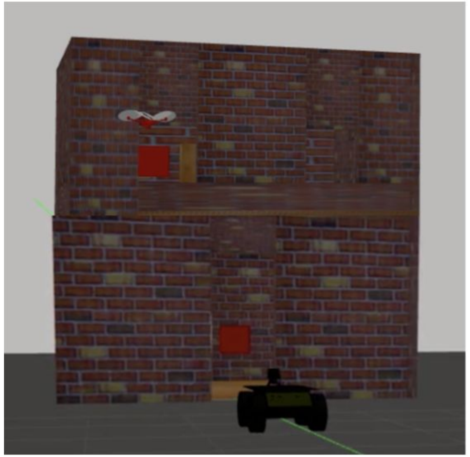
- In our system, different robots need to share information among them, and take actions collaboratively according to the shared information.
- System uses Wifi as the communication module. Different robot maps different regions of the entire area, and interest points of the map information is shared among each other.
- Each robot will report the locations of the detected fire,as well as their status (battery status and volume of remaining water) to the UGV, and UGV will coordinate the mission of each robot.

UGV and UAV communication

Current System Status: Modeling, analysis, testing

SLAM system test: from ORB SLAM2 to Intel Tracking Camera

Current System Status: Modeling, analysis, testing



Gazebo Simulation: robot control, navigation and fire detection

Fire source simulation/modeling:
Using massage bag filled with water as fire source.

Current System Status: Modeling, analysis, testing

PCB Power rating:

Number of Output Connectors : 3

- Output voltage 1: 12V, 4A
- Output voltage 2: 24V, 2A
- Output voltage 3: 5V, 2A

Voltage regulating circuit:

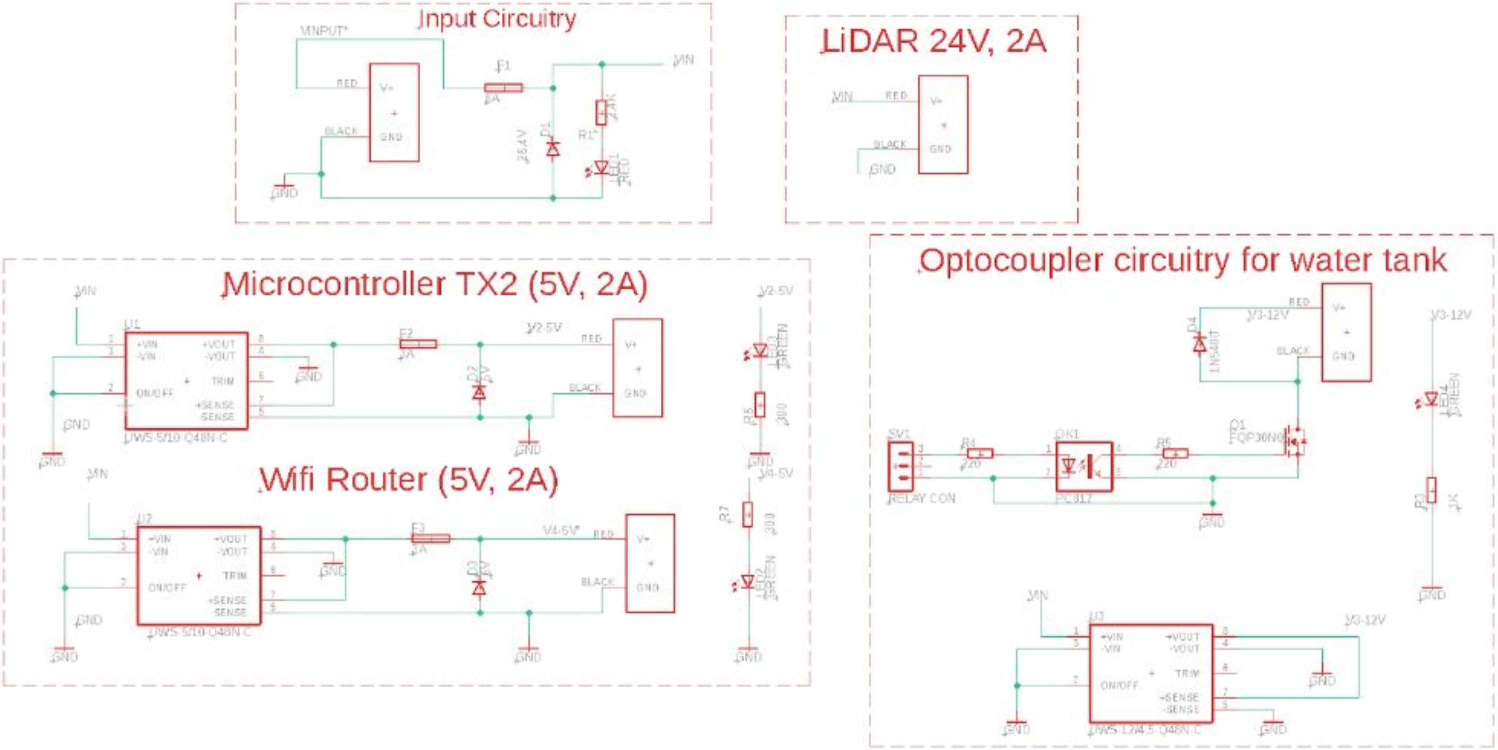
12V voltage regulator (DC-DC converter)

- Desired Efficiency: 85 %
- Peak Output current: 4A
- Output voltage: 12 +- 2% V

5V voltage regulator (DC-DC converter)

- Desired Efficiency: 85 %
- Peak Output current: 2A
- Output voltage: 5 +- 1% V

Current System Status: Modeling, analysis, testing



PCB Schematics

Current System Status: SVD Performance Evaluation

AGV:

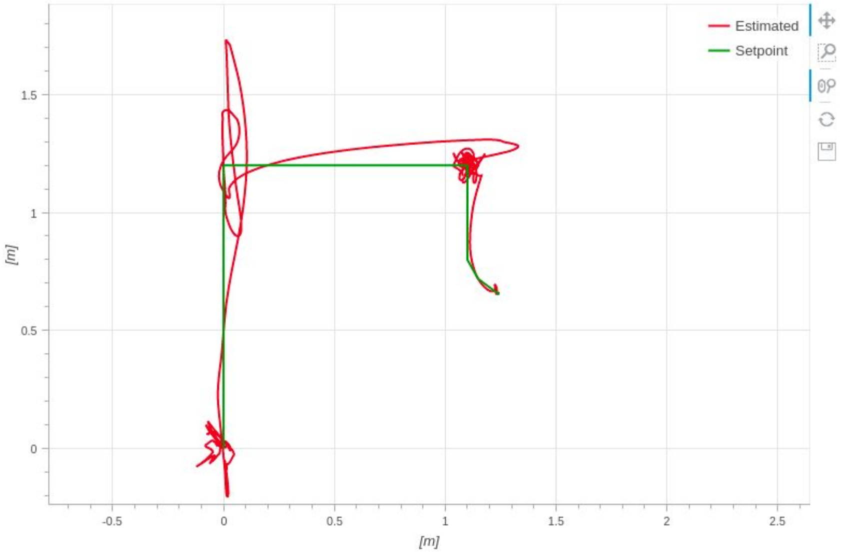
- AGV is capable of stopping while detecting obstacles within 75 cm range.
- AGV has maximum fire tracking error of 15 cm.
- AGV has maximum error of 30 cm between desired and the actual trajectory.
- AGV has maximum drift error of less than 50 cm in x and y direction for 24 m distance traveled.

UAV:

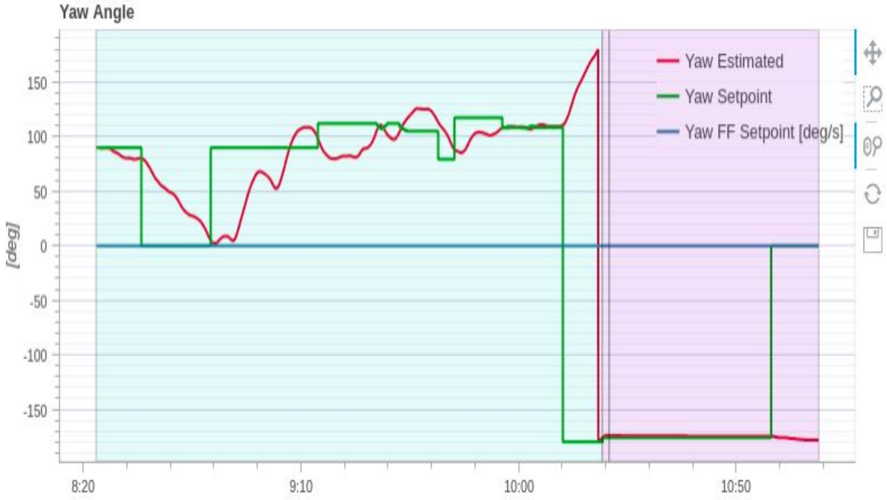
- UAV can carry 1.5 kg to 2 kg of payload.
- UAV has mean fire tracking error of 39 cm and maximum error of 45 cm.
- UAV has maximum tracking error of around 50 cm in x and y direction and 20 cm in z direction.

Both UAV and AGV can detect fire from a maximum of 1.5 m distance.

Current System Status: SVD Performance Evaluation

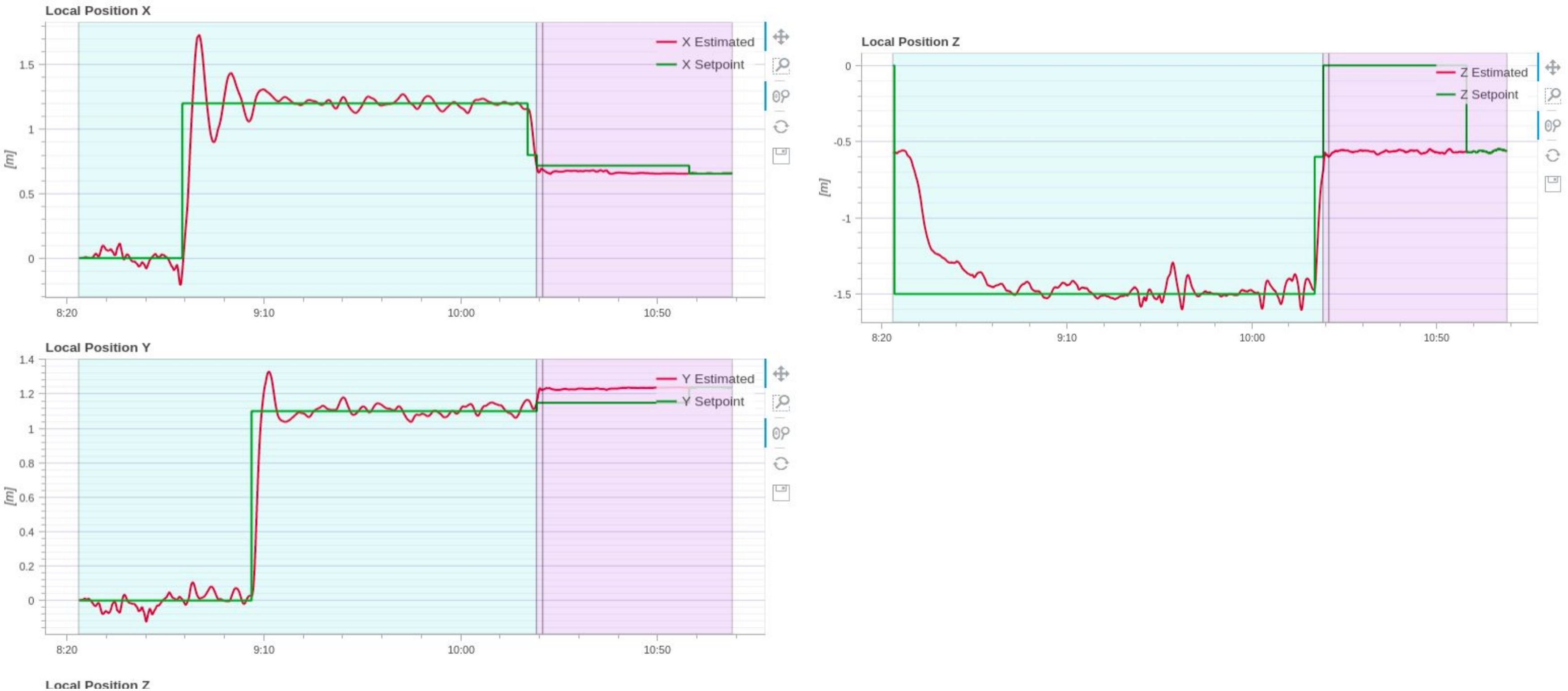


Estimated v/s Desired Trajectory for UAV



Estimated v/s Desired Yaw Angle for UAV

Current System Status: SVD Performance Evaluation



Estimated v/s Desired position in x, y & z direction for UAV

Current System Status: Spring Semester Video Excerpt

Current System Status: Strong/Weak Points

Strong Points:

- UAV system can perform stable autonomous flight
- UAV system can carry high payload (1.5 kg - 2 kg) and autonomously deploy the water
- AGV system can avoid collision using LIDAR
- UAV and AGV can localize in global map without GPS
- UAV and AGV can perform multistage autonomous mission with common ROS motion server framework
- UAV and AGV can detect and segment heated region using thermal camera and orient itself towards it
- UR5 arm can be controlled based on thermal feedback

Weak Points:

- AGV doesn’t have UR5 arm attached to it
- UAV can not detect obstacles
- UAV and AGV can not figure out how much water is left
- Water deploying mechanism is not able to spray water to greater distance (as it can’t resist the downward air turbulence from UAV)
- UAV yaw control is not perfect in the visual orienting mode

Project Work Breakdown Structure

1.0 Hardware:

1.1 Hardware Design	1.1.1 Finalize base part list for UAV and sensors for both UAV, AGV 1.1.2 Choose mechanism to deploy extinguishing material at target 1.1.3 Design mount for UAV that attaches extinguishing material deploying mechanism 1.1.4 Attach UR-5 arm along with the mount for AGV that attaches extinguishing material deploying mechanism 1.1.5 Choose UAV Power Distribution Board 1.1.6 Design AGV Power Distribution Board
1.2 Hardware Integration	1.2.1 Procurement of Parts 1.2.2 Fabricating/Procuring Power Distribution Board 1.2.3 Assembling base UAV system with help of AirLab, CMU 1.2.4 Integrate sensors with UAV 1.2.5 Integrate sensors with AGV 1.2.6 Building deploying mechanism 1.2.7 Building UAV arm with microcontroller integrated 1.2.8 Integrating deploying mechanism with arm and arm with UAV 1.2.9 Integrating deploying mechanism with arm and arm with AGV
1.3 Hardware Testing	1.3.1 Kill switch testing 1.3.2 UAV manual flight test 1.3.3 AGV manual drive test 1.3.4 Stereo Camera testing with Nvidia Jetson & ZOTAC 1.3.5 Thermal Camera testing with Nvidia Jetson & ZOTAC 1.3.6 2D-LiDAR testing with ZOTAC 1.3.7 Pixhawk testing with Nvidia Jetson 1.3.8 Microcontroller testing with Nvidia Jetson and ZOTAC 1.3.9 Testing UAV water deploying mechanism 1.3.10 Testing AGV arm and deploying mechanism 1.3.11 Testing wireless communication

Green: Completed; Orange: Partially Completed; Red: Incomplete

Project Work Breakdown Structure

2.0 Software:

2.1 SLAM Subsystem	2.1.1 Installing libraries dependencies in Nvidia Jetson & ZOTAC 2.1.2 Installing and integrating ORB-SLAM2 in ROS pipeline 2.1.3 Integrating odometry data coming from GPS and IMU sensors to improve SLAM result 2.1.4 Improving SLAM results
2.2 Path Planning Subsystem	2.2.1 Implementing global primitive paths that takes system to area of interest (building) from paths that achieve high coverage 2.2.2 Integrating implemented frontier based exploration planner in ROS pipeline 2.2.3 Integrating implemented RRT based local planner that takes 3D maps from SLAM and gives collision free paths
2.3 Navigation Control Substem	2.3.1 Establish MAVROS link from Nvidia Jetson to Pixhawk 2.3.2 Provide low level desired waypoints/trajectory to Pixhawk 2.3.3 Provide IMU, GPS data from pixhawk to SLAM subsystem and feed updated localization from SLAM subsystem to pixhawk 2.3.4 Create collision avoidance failsafe (eg using 2D-LiDAR in AGV)
2.4 Communication Subsystem	2.4.1 Establishing WiFi communication link between UAV and AGV 2.4.2 Creating shared database that can store active fire location, system location and available resources 2.4.3 Design and Implement optimum task assignment algorithm

Green: Completed; Orange: Partially Completed; Red: Incomplete

Project Work Breakdown Structure

2.0 Software:

2.5 Fire detection and Localization Subsystem	2.5.1 Classical segmentation approaches on thermal images to detect fire region 2.5.4 Implement module to Localize fire based on current system location and 3D map
2.6 Fire Extinguishing Subsystem	2.6.1 Implement/Integrate basic visual servoing method to orient system close to fire 2.6.2 Establish serial communication from Jetson to Microcontroller 2.6.3 Implement controller to orient AGV arm to point at fire 2.6.4 Implement controller to orient UAV to point at fire 2.6.5 Implement microcontroller signal to deploy material while monitoring fire 2.6.6 Improving deploying mechanism
2.7 Software Integration	2.7.1 Getting familiar with ROS (bonding and action server) 2.7.2 Implement code that initializes various ROS nodes 2.7.3 Implement main function that does high level decision making 2.7.4 Provide any missing link between various subsystem

Green: Completed; Orange: Partially Completed; Red: Incomplete

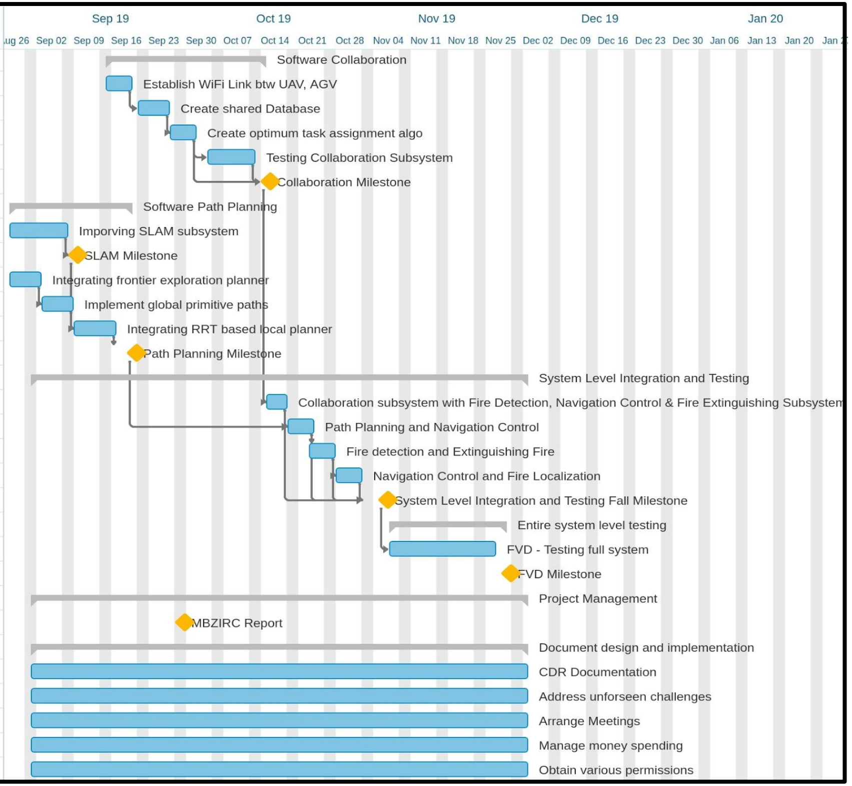
Project Work Breakdown Structure

3.0 System Level Integration and Testing	3.1 Integrating & testing Fire detection subsystem and Extinguishing Fire system 3.2 Integrating & testing Path Planning subsystem and Navigation Control subsystem 3.3 Integrating & testing Navigation Control subsystem and Fire Localization subsystem 3.4 Integrating & testing Collaboration subsystem with Fire Detection, Navigation Control & Fire Extinguishing Subsystem 3.5 Entire system testing
4.0 Project Management	4.1 Document design and implementation 4.2 Manage budget 4.3 Arrange meetings 4.4 Address unforeseen challenges 4.5 Obtain various permissions

Green: Completed; Orange: Partially Completed; Red: Incomplete

Project Milestone/Schedule Fall 2019

	PhoeniX Copy	Sub Tasks	Est. Hours	Start Date	Due Date	Task Prog.
	S Software Collaboration		-	15/Sep	14/Oct	0%
3	Establish WiFi Link btw UAV, AGV		-	15/Sep	19/Sep	0%
4	Create shared Database		-	21/Sep	26/Sep	0%
5	Create optimum task assignment algo		-	27/Sep	01/Oct	0%
6	Testing Collaboration Subsystem		-	04/Oct	12/Oct	0%
7	Collaboration Milestone		-	14/Oct	14/Oct	0%
	S Software Path Planning		-	28/Aug	19/Sep	0%
9	Imporving SLAM subsystem		-	28/Aug	07/Sep	0%
10	SLAM Milestone		-	08/Sep	08/Sep	0%
11	Integrating frontier exploration planner		-	28/Aug	02/Sep	0%
12	Implement global primitive paths		-	03/Sep	08/Sep	0%
13	Integrating RRT based local planner		-	09/Sep	16/Sep	0%
14	Path Planning Milestone		-	16/Sep	19/Sep	0%
	S System Level Integration and Testing		-	01/Sep	02/Dec	0%
16	Collaboration subsystem with Fire Detection…		-	15/Oct	18/Oct	0%
17	Path Planning and Navigation Control		-	19/Oct	23/Oct	0%
18	Fire detection and Extinguishing Fire		-	23/Oct	27/Oct	0%
19	Navigation Control and Fire Localization		-	28/Oct	01/Nov	0%
20	System Level Integration and Testing Fall Mi…		-	02/Nov	05/Nov	0%
	Entire system level testing		-	07/Nov	28/Nov	0%
22	FVD - Testing full system		-	07/Nov	26/Nov	0%
23	FVD Milestone		-	27/Nov	28/Nov	0%
	P Project Management		-	01/Sep	02/Dec	0%
25	MBZIRC Report		-	28/Sep	28/Sep	0%
	Document design and implementation		-	01/Sep	02/Dec	0%
27	CDR Documentation		-	01/Sep	02/Dec	0%
28	Address unforseen challenges		-	01/Sep	02/Dec	0%
29	Arrange Meetings		-	01/Sep	02/Dec	0%
30	Manage money spending		-	01/Sep	02/Dec	0%
31	Obtain various permissions		-	01/Sep	02/Dec	0%



Project Milestone/Schedule Fall 2019

Milestone	Date
SLAM Subsystem (Fall)	September 8
Path Planning Subsystem	September 20
MBZIRC Report 2	**September 28**
Collaboration Subsystem	October 15
Mid Sem Report (MRSD)	**October 22**
Hardware Integration (Fall)	October 25
Hardware Testing (Fall)	November 1
System Level Integration and Testing (Fall)	November 5
Fall Validation Experiment	**December 1**

Currently we are on schedule and all the milestones for the Spring semester are achieved along with some additional milestones which were due for the Fall semester.

High Level Test Plan

Progress Review	Milestones/Capability
PR 7	Fully functional localization system with mapping w/o tracking camera and obstacle detection for UAV using stereo vision
PR 8	1. Local Path Planning subsystem around obstacles 2. Global path planner for UAV and UGV tested in simulation
PR 9	WiFi connectivity on UAV and UGV with a central router and database server on UGV.
PR 10	Fire localization in the world coordinates and added to the database. Tested with a demo of fire locations added to the database.
PR 11	Testing the global planner in real world with a small mission to validate the trajectories generated coupled with the local planner
PR 12	Test the UAV and UGV doing a small autonomous mission to validate the collaboration subsystem by testing the systems moving towards a fire location

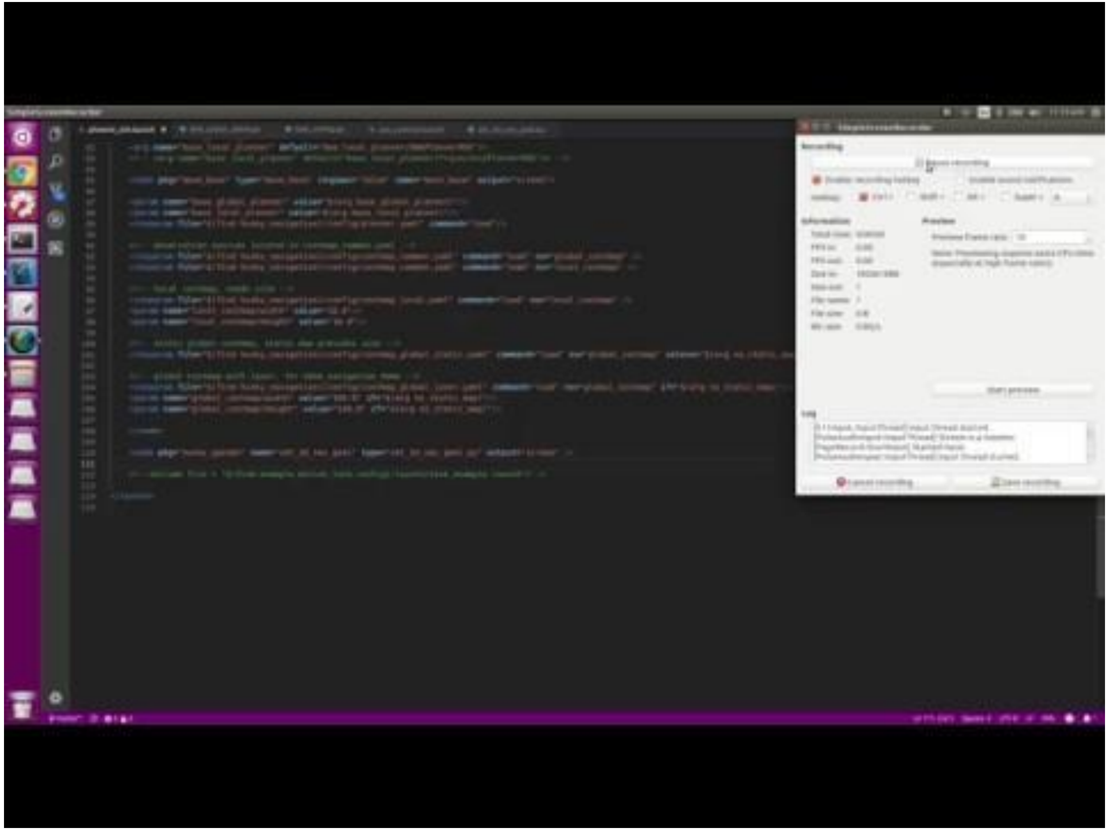
Fall Validation Demonstration

Objective	Demonstrate that the PhoeniX fire fighting system is capable of collaboratively extinguishing fire using UAV and AGV in a building or an equivalent simulated environment.
Sub-systems	• Global Trajectory Planner (Navigation) • Local Planner with Obstacle Avoidance • Visual Servoing • Mapping • Manipulator control • Fire extinguishing
Location	Dummy MBZIRC test site / “TBD”
Equipment	• PhoeniX UAV, UGV both with extinguishing material • Hot water bag • Kill Switch • Safety nets (if test location is indoors)

Fall Validation Demonstration

Testing Procedure	Validation Criteria
1. Operator will give the GPS location of the building in the form of a input to the PhoeniX firefighting system. 2. UAV and AGV will takeoff and drive off towards the known location of a structure/building containing potential fire spots. 3. During the movement of the systems they will create a real time map. 4. The systems shall continuously avoid obstacles in their way towards the structure by rerouting around the obstacles like other UAVs, AGV and the walls of the structure. 5. Systems will enter inside the building through the openings like windows and doors to detect fire. 6. When the systems detect fire locations, they will add its location in the shared database. 7. The same system or some other system shall then navigate to this fire location to extinguish fire. 8. The system will deploy extinguishing material on the simulated fire spot to simulate the extinguishing task. 9. Once all the fire locations have been extinguished the system shall come out of the building	1. 60% of the area of the test area (“TBD”) mapped within 10 minutes of operation. 2. Deposit 40% deployed extinguishing material. on the target area of minimum 0.5 m x 0.5 m 3. System accumulates less than 5m drift for every 100m distance travelled. 4. Localize fire with less than 1 m error. 5. System is able to communicate with each other within a radius of 25 m. 6. System carried 1 KG extinguishing material.

Fall Validation Demonstration



Project Management : Budget

Quantity	Part Name	Unit Price	Total Price
2	ZED Stereo Camera	$449.00	$898.00
1	Thermal Camera	$1,242.00	$1,242.00
2	Washer Bottle with pump	$72.85	$145.72
1	Intel Tracking Camera	$199.00	$199.00
8	Hex Standoff	$4.39	$35.12
4	Tarot rubber damper	$7.50	$30.00
4	Tarot extended rubber damper	$9.90	$39.60
8	Hex Standoff	$4.39	$35.12
1	Water heater	$39.99	$39.99

1. Total Budget (MRSD): $5,000; Spend: $ 3128
2. Big tickets items: Zed Stereo camera, Intel Realsense Tracking and Thermal Camera
3. Percentage we have spent to date: 62.56%
4. Funds used from MBZ (on building 2 hexacopters): $ 10,708.70

Risk Management

Risk Id	Risk	Category	L	C	Mitigation Strategy	Risk Owner
R.1	Data/Code Corruption	Technical	2	4	Always take a copy of the code/data, or distribute code on the cloud and share among team members	Akshit
R.2	No knowledge on the actual Fire simulation to be used in MBZIRC	Technical, Schedule	3	5	1.Get the sponsors (Oliver Kroemer) speak to the MBZ Committee in ICRA 2019. 2. Procure and use massage bags containing hot water to simulate fire till the 'TBD' status is resolved by 15 March 2019.	Shubham
R.3	Extra effort on repairing/maintaining the UAV and AGV	Schedule, Cost	5	2	1.Maintain a contingency reserve especially for the UAV like motors, propellers and ESC. 2.Also possibly assemble a new UAV for replacement if another one not working	Steve

*Green color indicates risks mitigated successfully
*Orange color indicates risk mitigation in progress
*White color indicates risk not encountered yet

Risk Management

R.4	UR 5 integration with the Husky	Schedule	2	3	1. Buy off the shelf arm (latest by 15 April 2019) 2.Ask sponsor for any manipulator in Oliver's lab which can meet the requirements	Steve
R.5	Tilted Hexacopter performance not up to the mark	Schedule, Technical	3	2	Tentatively move towards a non-tilted version	Parv
R.6	Test location for FVD	Logistics	5	4	1. Talk to the sponsors about some test facility like Near Earth Autonomy warehouse or NSH B Level 2.If no test site is available use the cage in FRC and notify Prof Howie Choset students about the date and time	Akshit

Risk Management

R.7	Install T265 bindings on Jetson	Schedule, Technical	5	5	1. Use Team RAMS kernel image	Akshit
R.8	Jetson USB buffer/processing not sufficient for multiple USB peripherals	Schedule, Technical	5	5	1. Port the whole system to Intel NUC if patches with TX2 don't work	Akshit

Risk Management

		R.2	R.6	R.7, R.8
	R.1			
	R.4			
		R.5		R3

Likelihood v/s Consequence Matrix

Red = Critical
Yellow = Moderately Critical
Green = Less Critical

Likelihood on the vertical axis and consequence on the horizontal axis

Conclusions

Key lessons learnt

- Before every UAV flight, test the code in simulator.
- UAV flight testing needs good amount of time due to gain tuning and longer repair/maintenance time.
- Intel Realsense tracking camera drifts in outdoor environment.
- Always keep spare mechanical mounts for all the subsystems.
- Take a OS snapshot after every milestone.

Key Activities

- Adhere the guideline of testing the code on simulator before every flight test.
- Allocate sufficient time for UAV flight testing.
- Add additional sensor modality (like GPS) for outdoor navigation.
- Develop robust state estimator for fusing IMU and Realsense tracking camera output.
- Always 3D print additional mechanical mounts for all the subsystems.

Questions?