PhoeniX
Collaborative Robotics

UAV - UGV Collaborative Robots for Fire-Fighting

- Team H:
  - Akshit Gandhi
  - Shubham Garg
  - Parv Parkhiya
  - Zhihao Zhu (Steve)
- Sponsors:
  - Oliver Kroemer
  - Sebastian Scherer
- Advisors:
  - John Dolan
  - Dimitrios Apostolopoulos

Date: May 9, 2019

Abstract

Structural-fire has caused 2,640 civilian death and material loss of 9.7 Billion USD in the US alone in the year 2011 as per the National Fire Protection Association. Time is of the essence when it comes to tackling most of the fire incidents. PhoeniX team proposes cutting edge, fully autonomous, heterogeneous, multi-agent robotics system to collaboratively locate and extinguish the fire without any human intervention in an unknown environment. Our system comprises of UAV (Unmanned Aerial Vehicle) and AGV (Automated Ground Vehicle) equipped with thermal/night vision camera which uses deep learning based methods for detecting and localizing the fire.

Our system uses a stereo camera to simultaneously create a real-time 3D map of the environment and localize itself in that map. The system uses state of the art algorithms to explore the environment while avoiding collisions. UAVs have roughly 20 minutes of flight time, high payload capacity (2 Kg), and improved stability that can be attributed to its tilted hexacopter design. Both vehicles carry extinguishing material which it can strategically deploy on the target fire. UAVs and the AGV share information of the fire location with each other, extinguishing material status to make smart decision resulting in a timely & efficient response. PhoeniX firefighting system attempts to push the technological boundaries to create a net positive impact on mankind.

This report outlines the progress of PhoeniX towards building a collaborative robotics system for fire-fighting.

Contents

1. Project Description

On 8th October 1871, a small barn in Chicago caught fire because of unknown reason and what followed was a conflagration lasting 3 days that killed up to 300 people and made 100,000 residents homeless. In the aftermath of this Great Chicago Fire, Chicago and many other cities updated and implemented better fire safety code.

Table 1 shows the number of fire incidents in the US in the year 2011[1].

Table 1: **Reported Fire Incidents**

Fire Location	Number of Incidents
Outside (Forest)	686,000
Structure (Building)	484,500
Vehicle	219,000
Total	1,389,500

Table 2 shows the damages caused by these fire incidents.

Table 2: **Damages due to these incidents**

Damage	Structure	Outside	Vehicle	Total
Property (Billion USD)	9.7	0.616	1.4	11.7
Civilian Injuries	15,635	675	1,190	17,500
Civilian Deaths	2,640	65	300	3,005

PhoeniX team proposes an autonomous multiagent system with navigation, perception capabilities and mechanism to deploy fire extinguishing material. Our system can also act as the first responder for collecting information about surrounding (map) and location of fire & trapped people which human firefighters can use to make better judgments.

2. Use case

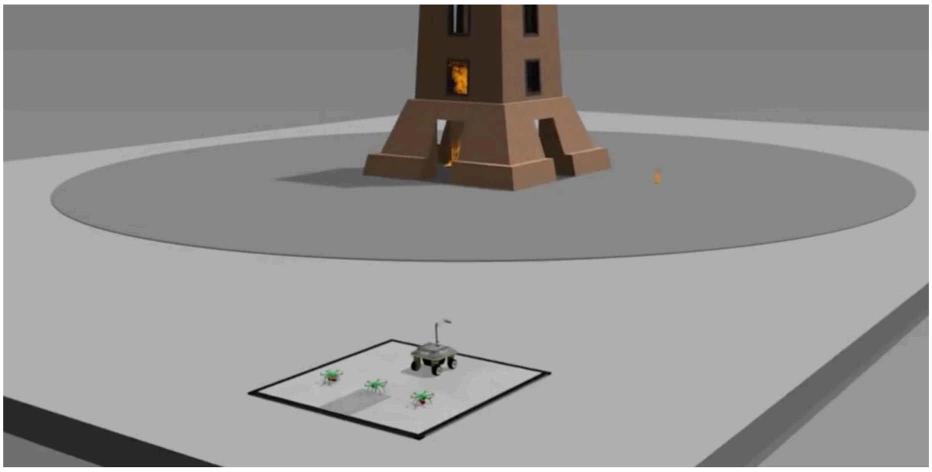


Figure 1: **UAVs and UGV at base station**

North Oakland is a Pittsburgh community located near the world-famous Carnegie Mellon University. At the night of 11th Nov 2018, a three-storey building caught fire. Few minutes after receiving the fire notification, firefighters reached an open area (base station is shown in Figure 1) near the target building, and they put PhoeniX firefighting system on the ground and set off the initiating signal. The system becomes active and 3 UAVs take-off from the station and an AGV also drives towards the building.

All the robots coordinate and collaborate to optimally explore the surrounding by avoiding obstacles while creating a map of the environment. UAV-2 detects fire in the building at two locations: ground floor and 1st floor, it shares that information with other systems.

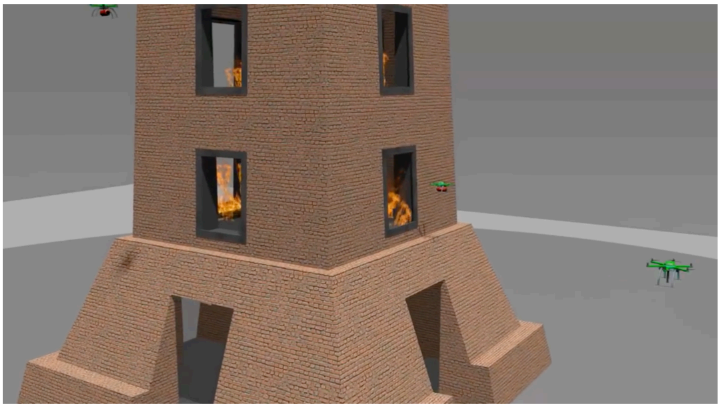


Figure 2: Robots moving towards fire locations

The system divides the task of extinguishing the fire at those two locations as shown in Figure 2. The AGV is assigned the task of extinguishing the 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. As shown in Figure 3 the AGV uses sweeping strategy to extinguish fire whereas the UAVs use some different mechanism to extinguish fire depending on the fire location.

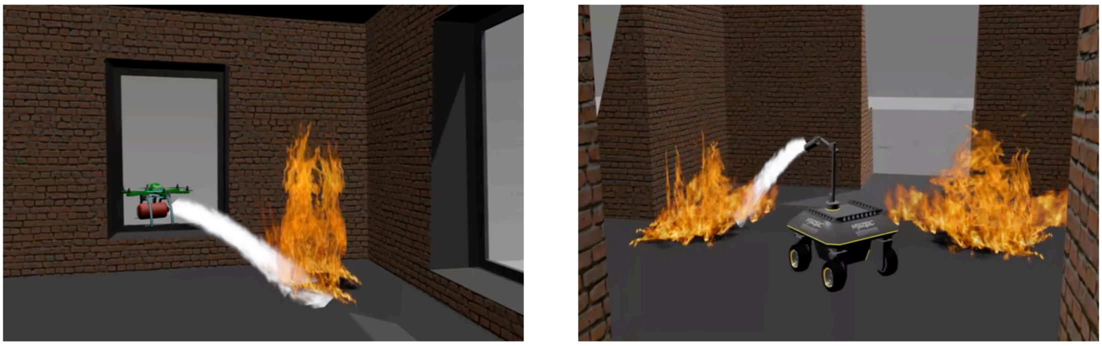


Figure 3: Robots extinguishing fire

Every robot monitors its fire extinguishing progress. AGV reports that it has successfully extinguished the fire. When the 1st and 2nd UAVs are out of the firefighting 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 at the station along with the AGV driving back.

3. System-level requirements

Since all of our performance and non-functional requirements have been derived from the functional requirements and the objective tree, the table 3 depicts a one-to-one mapping between the mandatory requirements. The naming convention is as follows:

- F.R (Functional requirement)
- M.P (Mandatory performance requirement)
- D.P (Desirable performance requirement)
- M.N (Mandatory non-functional requirement)
- D.N (Desirable non-functional requirement)

Table 3: **Mandatory functional and corresponding performance requirements**

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
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	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 and AGV	
	M.P.10	Reliable communication within 25 m

Table 4 shows mandatory non-functional requirements. Few non-functional requirements like size/form factor have been derived from the MBZ Challenge rule-book and thus they have remained consistent throughout the course of the project. They are subject to change only if any changes are made to the challenge. Rest of the requirements are added to make our system safe, modular, reliable and robust against environmental conditions (like wind).

Table 4: **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	Inter-operate with other MBZIRC team's systems by the means of functional modularity

Table 5 shows the desirable non-functional and performance requirements. Our desirable requirements aim to make the system portable, easy to manufacture & have some additional safety features. Using propeller guards will not increase the safety but will also reduce the wear and tear damage to the expensive carbon-fiber propellers. We also aim that the system has parallel and distributed processing architecture to maximize the collaborative effort.

Our desirable requirements also aim to have intelligence built into the system such as docking when battery or extinguishing material falls below a certain threshold. Creation of a common map of the building based on the individual maps generated by the UAV and AGV which can help the firefighters to make better judgment and decisions. Since each UAV and AGV works independently, our system is easily scalable and capable of covering a large amount of area.

We are using a thermal camera for detecting fire in the environment. Same thermal technology can be utilized to detect and localize trapped humans in the different parts of the building. This information is crucial for firefighters to plan the rescue missions. Based on this we added a desirable requirement of notifying authorities about the trapped people inside the building.

Table 5: **Desirable non-functional and performance requirements**

Type	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

4. Functional architecture

The functional architecture of the PhoeniX firefighting system has been depicted in fig 4. It captures all the functionalities we derived from the functional requirements and the objectives tree. The operation of the system begins when an operator (person/system) triggers a start signal to the system which will be in the form of the approximate GPS location of the building under fire. Given this start signal the UAVs take off and the AGV drives off following a trajectory precomputed by the AGV onboard and sent to the UAVs. While they travel towards the fire location, the systems start to map the environment while avoiding any obstacles. This map will be used by the system to plan the path from point A to point B.

While the systems are exploring the environment, they will also keep on checking for any potential fire locations by using the fire detection subsystem. Once a system identifies a location with fire, it will inform other systems by adding the location of the fire in a shared database. If there are fire extinguishing tasks pending in the database, an intelligent task assigner/scheduler will command an agent with enough extinguishing & battery resources to navigate to the fire location and extinguish the fire.

So, once the systems receive the coordinates of the fire location, they shall autonomously navigate in the environment while avoiding obstacles and now once they reach the proximity of the fire, they will orient themselves in an appropriate position to extinguish the fire. Now the system will deploy the extinguishing material using some strategy and update the database when they recognize that they have extinguished the fire. So once every potential fire location inside the building has been extinguished or a timeout of 20 minutes has passed, the system shall return back to the base station. The system may also return back to base if it runs out of the extinguisher material or battery.

Our functional architecture highlights the state-machine that we aim to implement. We have various modes such as navigation, exploration and mapping, fire extinguishing, collaboration, and scheduling. The timeout of 20 minutes has been decided based on the fact that the MBZ Challenge will last for 20 minutes and also getting a flight time over 20 minutes is extremely difficult with the UAV.

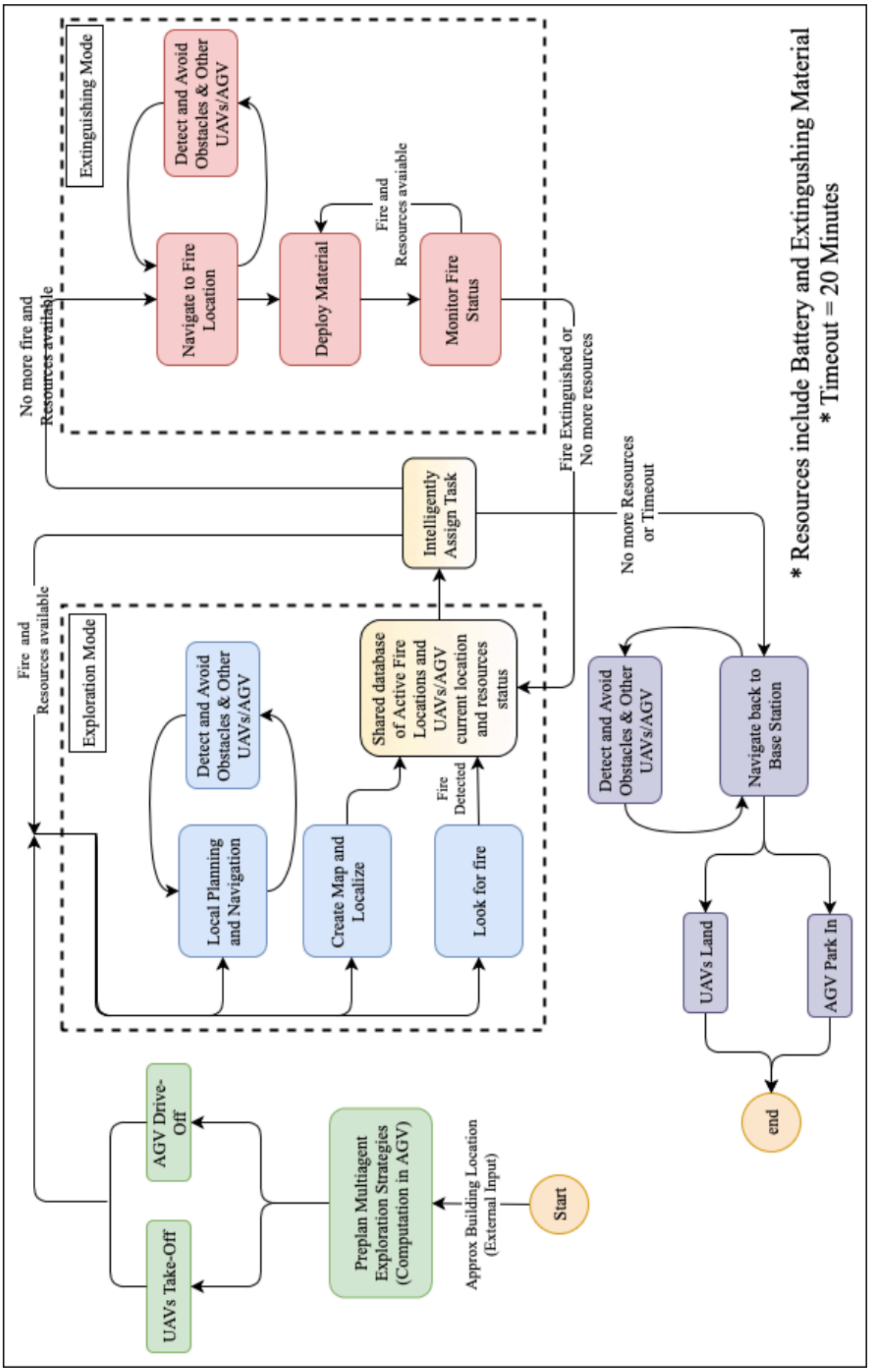


Figure 4: Functional Architecture

5. Cyberphysical architecture

Our Cyberphysical architecture is shown in figure 5 which maps the flow of data and energy between components and subsystems based on the results of our trade studies.

UAV/AGV System: Based on the trade studies, we have finalized combination of Stereo Camera (ZED), IMU, GPS as mapping sensors on UAV and an additional 2D LiDAR is added on AGV for better localization. Different mobile robots communicate via a WiFi link. Intel Tracking Camera T265 is added for better localization. On the UAV we will use the stereo camera for mapping.

Exploration Mode: Mapping sensors will generate an occupancy grid map for obstacle avoidance and planning local trajectory. Each mobile robot will use fire detection classifier for fire detection. The system will use RRT[3] path planning algorithm to create trajectories such that they cover a maximum possible area in minimum possible time. The exploration will also have a local planner that will use the obstacle detection hardware to generate strategies to avoid the obstacles. This mode uses the output of the localization module and the output of the thermal camera to detect the fire locations.

Scheduler: Each mobile robot will detect fire and update the shared database. Based on the fire location in the shared database, the scheduler will assign fire extinguishing task to different robots based on their locations from the fire. The scheduler will have heuristics such as proximity of the agent from fire location, battery status, extinguishing material status, etc.

Extinguishing Mode: Based on the global fire location, mobile robots will do visual servoing towards the assigned fire location, monitor and extinguish the fire. The system also relies on a local planner to navigate around obstacles. The system utilizes the extinguishing hardware which currently is a water tank and a water pump. Based on the images from the thermal camera the agent decides when to engage the water pump.

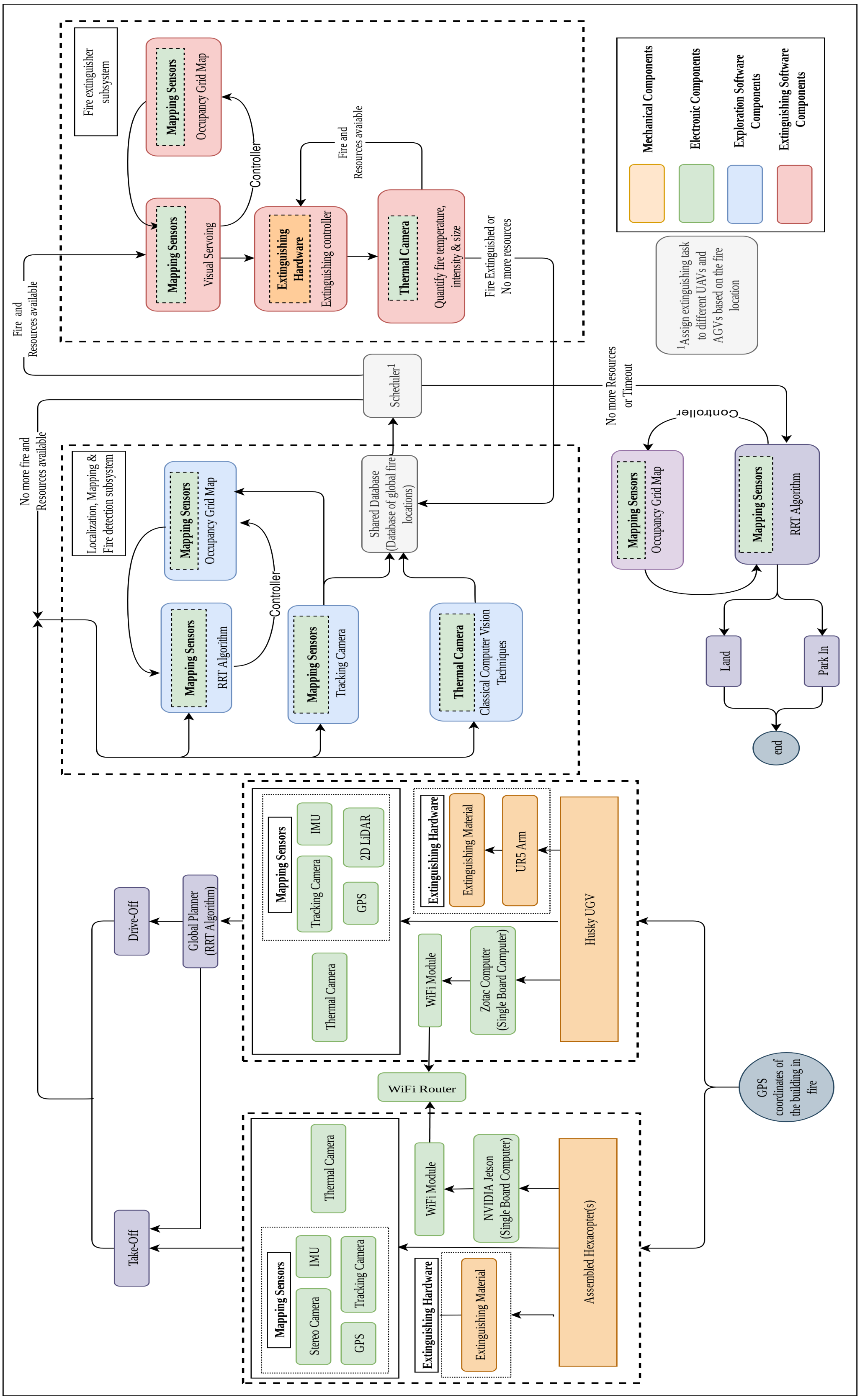


Figure 5: CyberPhysical Architecture

6. Current System Status

6.1. Targeted requirements

Table 6: **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 from a maximum 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
M.P.9	UAV successfully lifts 1.5 KG payload	Navigation Control
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.9	UAV points a laser pointer within the water bag of size 7.5 inch x 9.5 inch	Fire extinguisher

Table 6 shows the list of the system requirements and corresponding subsystems and system elements emphasized during the spring semester development. Most of the requirements in this semester were focused towards Navigation Control, Fire detection and Localization, Fire extinguishing subsystem. M.P.1 & M.P.4 verified our Navigation Control subsystem where a majority of the work was done this semester. This functionality enabled us to test and validate our M.P.7 (Fire detection and Localization subsystem). Further, we validated our Fire extinguisher subsystem through M.P.8 & M.P.9. Since our Fire detection and localization & Fire extinguisher subsystem code was made modular, the same code worked on both UAV and AGV without many changes. Surprisingly the SVD satisfies more than 50% of our mandatory performance requirements and thus we are optimistic in touching upon some desirable requirements if time permits in the next semester.

6.2. Subsystem descriptions/depictions

6.2.1. Hardware Subsystems

6.2.1.1 UAV Subsystem

While the design of the base UAV (tilted rotor hexacopter) is provided by the sponsors, the firefighting task necessitates modifying an existing design to integrate thermal and stereo camera as well as the mechanism for attaching and deploying payload (extinguishing material) at the target. For low-level control in deploying mechanism, micro-controller and actuators would be required. Integrating these modules would require us to redesign the power distribution board as well. UAV hardware subsystem will handle modifying an existing design to handle the above-mentioned requirements.

Currently the tilted hex platform is not stable and thus we are using a non-tilted version of the hexacopter for the platform. The platform currently looks as shown in the figure 6.

Figure 6: **PhoeniX UAV**

We have Intel Realsense T265 onboard for localization, Flir Boson 640 camera for thermal imaging and Pixhawk as an on-board flight controller highlighted in figure 7.

Figure 7: **T265 Tracking Camera (left), Flir 640 thermal camera (center), Pixhawk controller (right)**

To mount the necessary fire fighting hardware such as the water pump, we have developed a custom mount to be attached to the battery plate of the drone. The water tank and the mount can be seen in figure 8. The water tank has been modified to boost the projectile range. The water tank mount has been designed to reduce the amount of time needed to detach from the UAV and refill water.

Figure 8: Water Tank (left), 3D mount design (right)

6.2.1.2 UGV Subsystem

Pre-built AGV system (Husky) is provided by the sponsors but like UAV system certain modifications are required to do the firefighting task. The major difference compared to the UAV is that there are no power constraints or significant payload constraints which would allow us to integrate LiDAR for robust collision avoidance along with stereo, thermal camera and extra extinguishing material.

The UGV also has the same cameras as shown in fig 7, but additionally it also has a SICK LMS 200 LiDAR which helps us to perceive objects as far as 80m (fig 9). The UGV also uses a custom developed PCB to power the various peripherals such as the WiFi router which acts as a central hub for all communications, water tank, microcontroller, and the LiDAR.

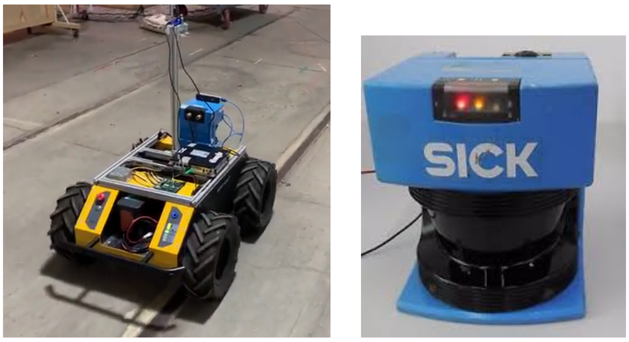


Figure 9: PhoeniX UGV (left), SICK LMS 200 (right)

6.2.2. Software Subsystems

6.2.2.1 Simultaneous Localization and Mapping (SLAM) Subsystem

We need to create a 3D map of the environment which would allow Path Planning Subsystem to avoid collision and the navigation control subsystem to generate better control signals. While distributed multi-agent SLAM[5] would lead to efficient mapping, our UAV/AGV system will not do distributed SLAM due to added complexity and reliance on high communication bandwidth. So, both the subsystems (AGV and UAV) will generate real time map of the environment independently using a stereo camera.

Currently we have tried ORB-SLAM2[2],[7] which has a sparse mapping capability as shown in figure 10. It also has a faster and reliable loop closure capability which is crucial for our application. The algorithm implementation, however, has some issues with the scaling factor which led us to use Intel Realsense T265 tracking camera for our state estimation. This Camera includes two fisheye lens, an IMU and an Intel processing unit which runs SLAM algorithm on board. But as of now, the tracking camera doesn't provide the support of fetching and saving the map on a host machine. So, we need to still to explore other mapping techniques to get a 3D map of the indoor/outdoor environment.

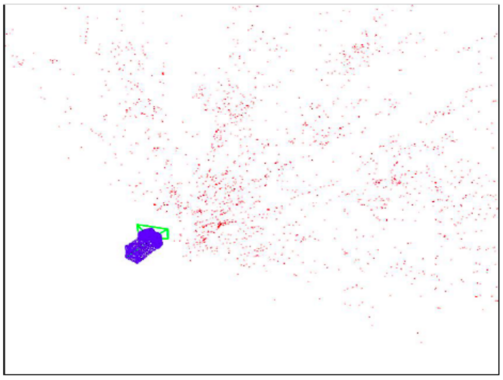


Figure 10: Husky moved forward and rotated 45° (Red points shows the 3D point cloud of the detected features, blue rectangles represents past camera position and green rectangle represents current location.)

6.2.2.2 Path Planning Subsystem

PhoeniX firefighting system is essentially a heterogeneous multiagent system. UAV and AGV both need to collaboratively search for fire in the environment efficiently i.e. we don't want both systems to search the overlapping area. Path planning subsystem consists of a global planner and local planner. Global planning includes high-level exploration path for both UAV, AGV in such a way that they jointly achieve high coverage while exploration (Like Frontier based exploration Algorithm). Local planner involves taking cues from the global planner and point cloud map from the mapping subsystem to produce a path that detours obstacles location to avoid any collision. Path planning subsystem will output the desired trajectory as waypoints to the Navigation Control subsystem.

This subsystem is to be implemented in the fall semester and hence no progress has been made into this.

6.2.2.3 Navigation Control Subsystem

Based on the mode of operation, the Navigation Control Subsystem takes input from the Path Planning Subsystem or Fire Localization Subsystem and generates control signals to go to the desired way-point or follow the desired trajectory. Navigation controller would take localization data from SLAM Subsystem and will fuse it with GPS and IMU data[9] to get accurate localization (feedback)[6].

Currently we have the full software and hardware stack ready to perform autonomous missions. We use a ROS action server architecture coupled with MAVROS which is a bridge helping us connect to the onboard flight controller and control the UAV and similarly for the UGV we have the husky-core service active.

6.2.2.4 Collaboration Subsystem

Collaboration Subsystem will provide a communication link to transfer vital information such as active fire location, other system's resources status. This information can be extremely useful for high-level decision making of what to do eg: explore, extinguish or return. Communication link would enable our system to fight the fire in a collaborative manner. Collaboration Subsystem will create a shared database containing active fire location and would assign tasks based on the priority queue.

This subsystem is to be implemented in the fall semester and hence no progress has been made into this.

6.2.2.5 Fire Detection and Localization Subsystem

As the name suggests the goal of this subsystem is to detect fire from the thermal image. Fire detection Subsystem continuously looks for fire. Based on the fire location in the thermal image, the approximate location of the fire is found in the global map . Finally, Fire Detection and Location Subsystem share this fire location to Extinguishing Fire Subsystem as well as Collaboration system so that other system can also become aware of the fire location.

Currently, we have the fully functional fire detection classifier which can accurately identify fire regions within 1.5m. We are using classical vision techniques to fetch the regions of high intensity in a thermal image and use prior knowledge of the shape of the fire region to make decisions. Extracting 3D global coordinates of the fire is not implemented in this semester and will be due Fall 2019 along with creating a global database of the fire locations.

6.2.2.6 Fire Extinguishing Subsystem

The extinguishing task is assigned after the agent has detected the fire. In the first step, the agent orients itself with respect to the fire by providing waypoints to Navigation Control Subsystem using Visual Servoing method. E.g. if the system is far away from the fire, waypoints that leads the system to fire are provided. In the second step, it provides control signals to the microcontroller which in turn activates the extinguishing mechanism to point it directly at fire. It continuously monitors the fire status using thermal image data and immediately stops deployment if the fire is extinguished. It also keeps track of how much extinguishing material is available using a load sensor.

Currently, we have a visual servoing algorithm which can control UAV and UGV when they are 1.5m away from the fire. We also have a master-slave communication architecture between the onboard computer and the Teensy micro-controller to engage and disengage the water pump.

6.3. Modeling, analysis, testing

1. Fire source simulation/modeling: Because we still have no idea on the nature of simulated fire source that will be used in MBZ challenge[4], we are using the massage bags filled with hot water as the simulated fire source. It is also shown in figure 11 (right). So, we adapt our fire detection algorithm according to the specification of such configuration of the simulated fire source (e.g. fire temperature and fire source volumes).

2. For the SLAM system, we previously tested the ORB SLAM2, but we noticed significant drift during an autonomous mission in an unseen feature-sparse environment. These constraints the capability of the robot's autonomous navigation. So we moved to Intel tracking camera[11] which has much robust performance.

3. Using Gazebo Simulation: We have built a Gazebo simulation environment, including AGV/UAV models, buildings, fire source, laser, obstacles (walls, closet), to test the functionality of the system. A snapshot of Gazebo simulation[10] is shown in Figure 11 (left). The simulation works as follows:

   (a) For UAV, it first navigates towards the building. Once it detects window, it flies into the room across the window. Then, it navigates inside the room and avoids obstacles along the way. When it detects the fire source, it flies towards it and positions itself right in front of the fire source by visual servoing. Once the UAV reaches a certain distance from the target fire source, it ignites the laser and points towards it. After all, these are done, the UAV navigates the room and finds its way out via windows, and finally return to the base station.

   (b) For AGV, it basically follows the same procedures as UAV, the only difference is that AGV will only search for the first floor while UAV searches for higher floors. Through such a simulation, we can design our system's specifications accordingly, and fix the functionality of subsystems by testing each unit.

4. As per MBZ challenge, our AGV should run at least for 20 minutes. So, the power consumption of our subsystem is critical. Based on the power analysis, our power distribution board has an efficiency of around 85%. We have used DC to DC buck converters for converting input 25.2V (from 6 cell Lithium Polymer battery) to 5V and 12V. PCB has an optocoupler circuitry which is responsible for turning water tank on/off based on the high-level command from the SBC. The detailed schematic can be found in figure 12.

5. Our PCB board is of 100 x 90 mm in dimensions and we have mounted our PCB on Husky (AGV) using four mounting holes of 6-32 screws. We are using Anderson Power connector for connecting input power supply & LiDAR. The CAD model of our PCB and the actual PCB can be seen in figure 13.

Figure 11: **Left: Gazebo Simulation for robot control, navigation and fire detection, Right: Fire source simulation/modeling using massage bag filled with water as fire source.**

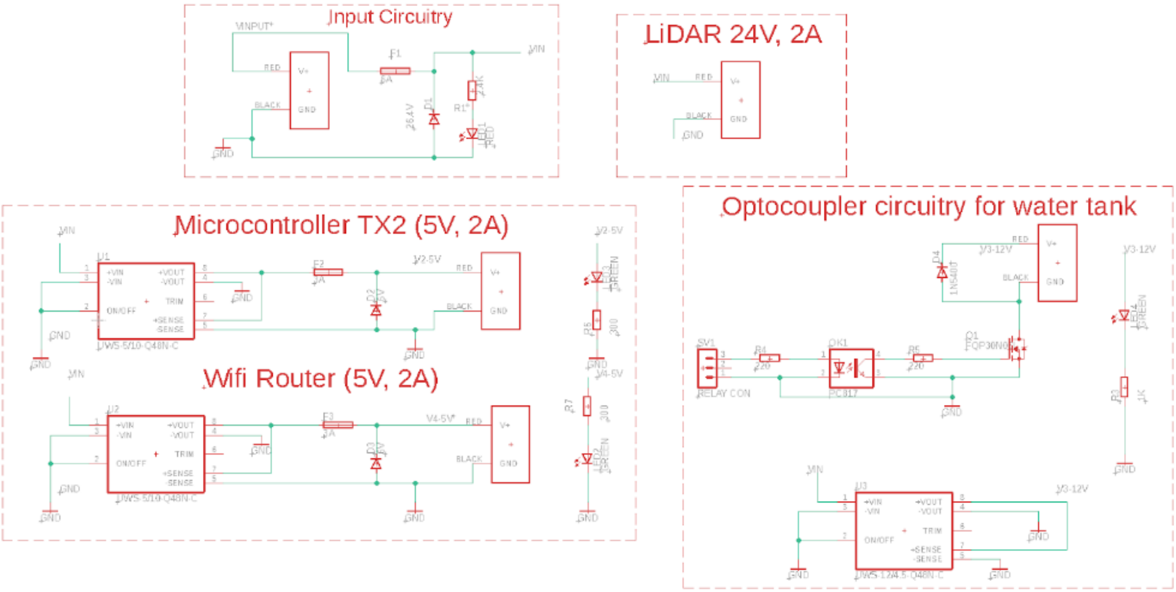


Figure 12: **PCB schematic**

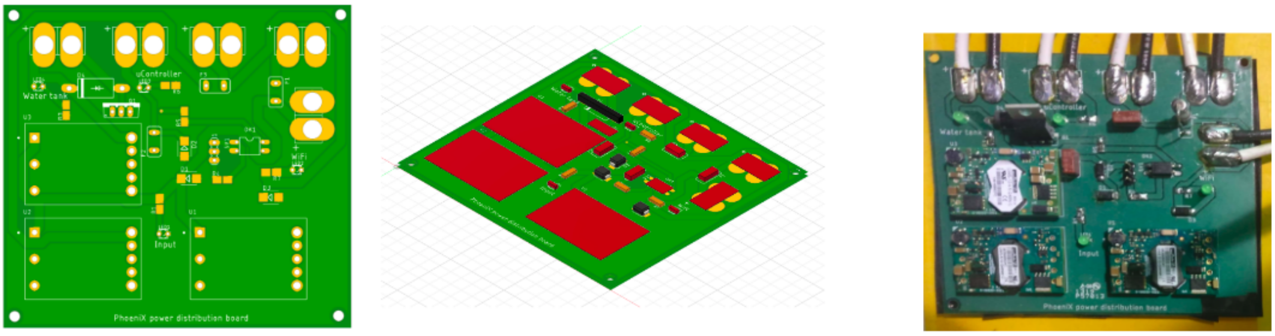


Figure 13: **Left: 3D layout of our PCB, Right: Populated PCB**

6.4. SVD performance evaluation

We performed three experiments for the spring validation demonstration. In the first experiment, UAV takes off from the base station with 1.5 kg payload and then autonomously follows the predefined trajectory of four waypoints in a GPS-denied environment. As shown in Figure 14, UAV follows all the required waypoint within the error limits of ± 50 cm. Similarly, as shown in Figure 16 & 17, there is a maximum error of ± 50 cm in the x and y-direction & ± 20 cm in the z-direction. UAV detects fire whenever its at max 150 cm away from the simulated fire. After reaching the third waypoint, UAV demonstrates its fire extinguishing capability by pointing a laser towards the simulated fire (Hot water bag). As shown in Figure 15, UAV was oscillating around the yaw axis while performing this task during SVD. We significantly reduced the yaw error in SVD-encore and maximum error was around 45 cm and the mean error was 39cm through the extinguisher task.

In the second experiment, AGV drives off from its base station and follows a predefined trajectory in GPS-denied environment with a maximum tracking error of 50 cm in x & y-direction and 20 cm in the z-direction. AGV is able to detect fire from maximum 150cm and it is able to track the fire with an error of around 15cm. AGV detects all the obstacles within 75cm of range and stops immediately.

In the third experiment, we have shown a video demonstration of UR5 arm video tracking the fire (hot water massage bag). It was able to track the fire with an error of ± 15cm. So, we were able to meet all our SVD performance metrics for both UAV & AGV.

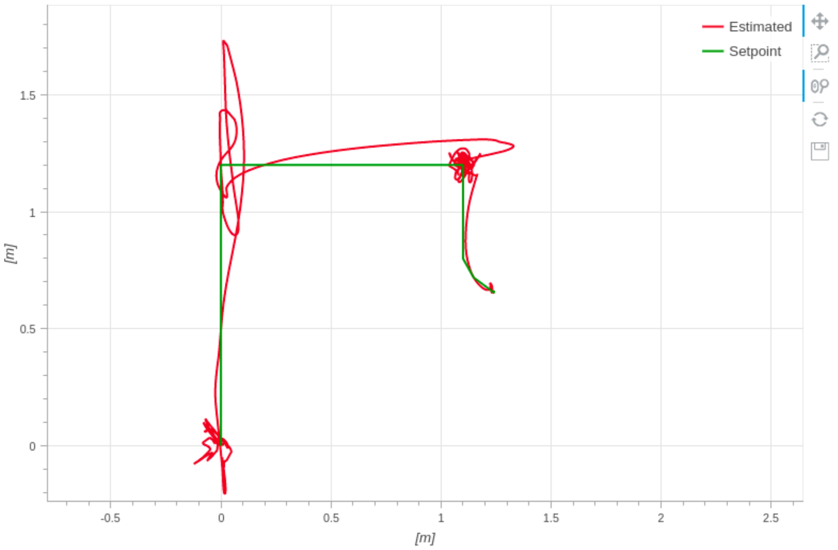


Figure 14: **Estimated v/s Desired Trajectory for UAV**

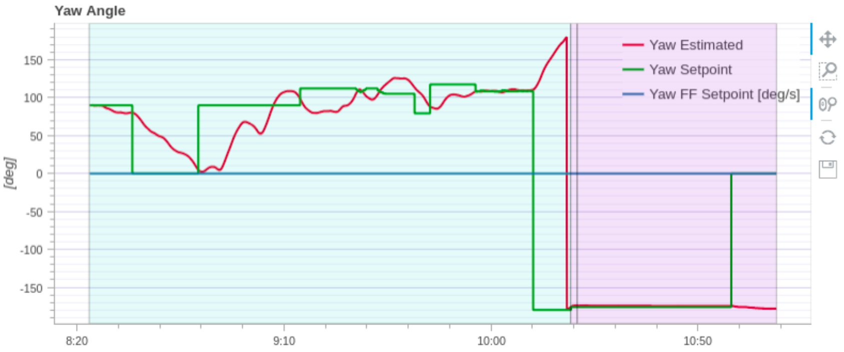


Figure 15: **Estimated v/s Desired Yaw Angle for UAV**

Figure 16: **Estimated v/s Desired X & Y position for UAV**

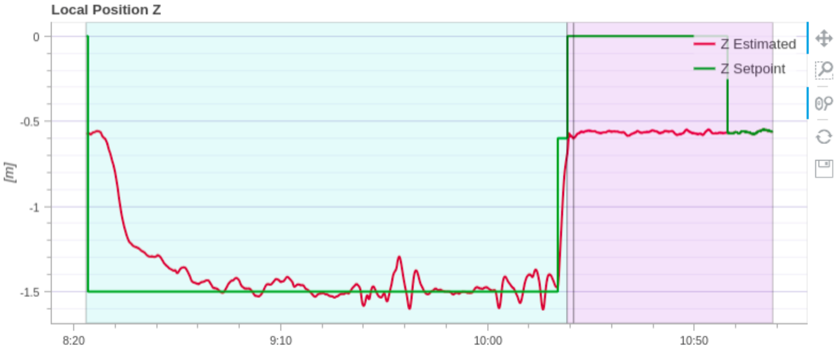


Figure 17: **Estimated v/s Desired Z position for UAV**

6.5. Strong and Weak points

Strong Points:

- After using carbon fiber propellers, UAV system is able to perform the stable autonomous flight.
- UAV system can carry a high payload of around 1.5 kg to 2 kg.
- UAV can autonomously detect fire and is capable of spraying water towards it.
- AGV system can avoid collision using LIDAR with very high accuracy.
- UAV and AGV can autonomously navigate in GPS denied environment.
- UAV and AGV can perform the multistage autonomous mission with common ROS action server framework.
- UAV and AGV can detect and segment the heated region using a thermal camera and orient itself towards it.
- UR5 arm can accurately point towards the fire based on thermal feedback.
- Reserve UAV is at standby and will heavily reduce the time to get airborne in case of a very bad crash with our primary drone.

Weak Points:

- We planned to mount our UR5 arm on the husky in the spring semester but the manufacturer couldn't ship the mounting kit on time. So, our AGV doesn’t have UR5 arm attached to it.
- UAV cannot detect obstacles.
- UAV and AGV cannot compute how much water is left in the water tank.
- Water deploying mechanism is not able to spray water to greater distances (as it can’t resist the downward air turbulence from UAV)
- Both UAV and AGV don't have mapping capability as ORB SLAM2 has scaling issues.
- Poor wireless communication link between UAV and host machine.
- UAV oscillates around the yaw axis when it tries to point towards the fire.

7. Project Management

7.1. Work Breakdown Structure

Table 7 shows the pending tasks for Fall 2019 from the work breakdown structure. Integration and project management are a continuous process within the team and will be carried on as described in the CoDR.

Table 7: **Work break-down**

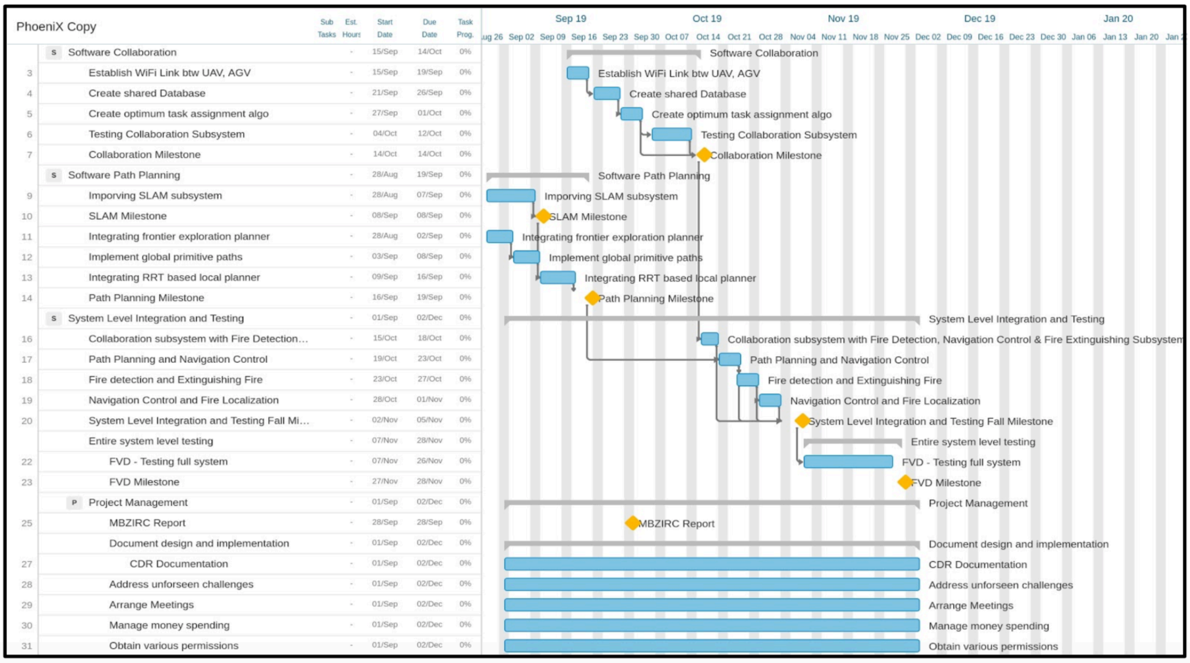
Task	Breakdown
Hardware Design	• Modify mechanism to deploy extinguishing material at the target
Hardware Integration	• Integrating deploying mechanism with arm and arm with AGV
Hardware Testing	• Testing UAV & AGV water deploying mechanism • Testing wireless communication
SLAM Subsystem	• Integrating odometry data coming from GPS and IMU sensors to improve SLAM result, coupled with a mapping component
Path Planning Subsystem	• Implementing global primitive paths that take the system to an area of interest (building) from paths that achieve high coverage and integrate into the ROS pipeline • Integrating implemented RRT based local planner that takes 3D maps from SLAM and gives collision-free paths
Communication Subsystem	• Establishing a WiFi communication link between UAV and AGV • Creating a shared database to store key information and implement a task allocation algorithm

7.2. Schedule Status

We are on schedule with our pre-specified plans. We have accomplished all of the pre-planned tasks including assembling of one UAV and one UGV with multiple sensors and onboard computing units, as well as software subsystem development including UAV/UGV control and the first version of fire detection classifier. But some of these subsystems need further refinements such as the localization and fire extinguishing subsystem (based on the MBZ challenge requirement). For the incoming fall semester, we have the following major system development milestones listed in Table 8:

Table 8: **Milestones for 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



	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%

Figure 18: **Fall 2019 Schedule**

7.3. Test Plan

7.3.1. Key Milestones - Fall 2019

Based on continued progress reviews throughout the fall semester, we have aligned the major milestones (as shown in table 9) which help us in putting up pieces together to construct the whole system that we intend to showcase at the Fall Validation Demonstration.

Table 9: **Progress reviews and milestones**

Progress Review	**Milestones/Capability**
PR 7	Fully functional localization system with mapping w/o tracking camera and obstacle detection for UAV using stereo vision. • The demo will show a UGV creating a live map of the area and demonstrate the SLAM subsystem. • For the UAV we will show a video demonstrating the same task
PR 8	1. Local Path Planning around obstacles: In this demo, UGV/UAV will be following a trajectory and an obstacle will be placed in front of it. (a) When it detects the obstacle it will use the local planner to move around the obstacle. (b) For the UAV we will show a video demonstration of the same task. 2. Global path planner for UAV and UGV tested in simulation. (a) In the simulation environment we will demo the systems moving towards the building following the trajectories generated by the planner, specifically the UAVs will occupy the higher levels of the building and the UGV will occupy the lowest level of the building.
PR 9	WiFi connectivity on UAV and UGV with a central router and database server on UGV. • The demo will have a UGV and UAV communicating with each other via WiFi and share vital information which will be stored in a database 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 and movement of an agent towards the location. 1. A UGV will see the fire location and it's world coordinates in the database, we will then release a UAV to reach the fire location validating our planner and communication system.
PR 11	Testing the global planner in the real world with a small mission to validate the trajectories generated coupled with the local planner. 1. At this point in the semester, we will demonstrate the global planner in real life on the MBZ dummy site and will validate the global planner with its mapping and obstacle avoidance system in real life.
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. 1. This will be the most important test before the Fall Validation Demonstration and will showcase a small scale Fall Validation demo wherein it will add more intelligent features to the PR 11, by incorporating the fire fighting task with the updated extinguishing subsystem.

The mapping component of the SLAM subsystem will be demonstrated as the first progress review in Fall 2019. Since the collaboration and path planning subsystem is due in the Fall semester we have lined them up as milestones in such a way that it gives us time to test out the planning subsystem in the early half of the semester such that if any discrepancies arrive, we could find a solution in the next half of the semester. The planning part will be tested along with the obstacle detection and avoidance capability using the local planner and sensors on-board which will be also thoroughly tested. Constructing an arena of 50mx60mx20m is not feasible and thus to test the system with the global planner we will demonstrate the subsystem in simulation. The communication system enables us to share the exploration task between multiple systems and thus it will also be a very crucial subsystem for the Fall semester and thus we have 2 progress reviews specifically dedicated for the task. In the last two reviews, we plan to demonstrate a collaborative effort in planning and mapping the area and movement towards fire locations using the intelligent scheduler.

7.3.2. Fall Validation Demonstration

The Fall Validation Demonstration will show all the subsystems developed in both the semesters working seamlessly and in harmony. The overall test plan for the next semester can be found the tables below.

Table 10: **Fall Validation Demonstration Objectives**

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 tested	• Global Trajectory Planner (Navigation) • Local Planner with Obstacle Avoidance • Visual Servoing • Mapping • Manipulator control • Fire Extinguishing • Collaboration/Communication
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)

In the FVD the operator will give a GPS coordinate for the location which is under fire. System will then compute trajectories and start to move towards that location by creating a map as they move. They will also avoid any obstacles in the way by taking a detour. Continuously they will be scanning the area for any signs of fire, if they find the location they will update the database and the scheduler will assign an agent to extinguish the fire at the locations identified. Once they extinguish all fires or gets out of any resources such as battery or extinguishing material, they will return back to the base station. The criteria's that will be validated during the experiment will be a certain % of the area mapped, drift accumulated, amount of material deployed on fire, localization error for fire, communication range and payload carried.

Table 11: **Fall Validation Demonstration Procedure**

Testing Procedure	Validation Criteria
1. Operator will give the GPS location of the building in the form of an 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 the 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 traveled. 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 of extinguishing material.

7.4. Budget Status

Table 12: **List of Major Expenses**

Quantity	**Part Name**	**Unit Price**	**Total Price**
2	ZED Stereo Camera	\$449.0	\$898
2	Washer Bottle with pump	\$72.85	\$145.72
1	Tarot X6 Landing gear connector (Aluminium)	\$14.95	\$14.95
1	Antenna for Nvidia Jetson TX2	\$8.59	\$8.59
1	Thermal Camera	\$1242	\$1242
1	Thermal Camera USB C	\$150	\$150
1	Intel Tracking Camera	\$199	\$199
3	Aluminum Motor Mounts (black)	\$25.90	\$77.70
1	Power converter	\$25.6	\$25.6
4	Tarot rubber damper	\$7.5	\$30.0
4	Tarot X6 Landing gear connector (Aluminium)	\$9.9	\$39.6
3	3M tape	\$9.24	\$27.72
8	Hex Standoff	\$4.39	\$35.12
4	Tarot rubber damper	\$7.5	\$30
4	Tarot extended rubber damper	\$9.9	\$39.6
8	Hex Standoff	\$4.39	\$35.12

Most of the expenses are done from our MBZIRC funds. We have spent around \$10708 from MBZ funds for building two hexacopter UAV platforms. From our \$5000 MRSD budget, we have spent around \$3128 i.e. 62.56% of our total budget. Some of the expenses are for purchasing parts for a repair task for the UAV and some of the expenses have been made for making mounts for the cameras to be attached onboard. The big-ticket items that have been purchased are Zed Stereo camera, Intel Realsense Tracking, and the FLIR Thermal Camera [8]. This portion of the budget and its breakdown is detailed in Table 12. There are no preplanned purchases for the next semester and we don't feel that we have any more big-ticket purchases. Since the current extinguishing mechanism is not working up to satisfaction we may buy new mechanism in Fall which should not be very expensive as compared to our current big-ticket purchases.

7.5. Risk Management

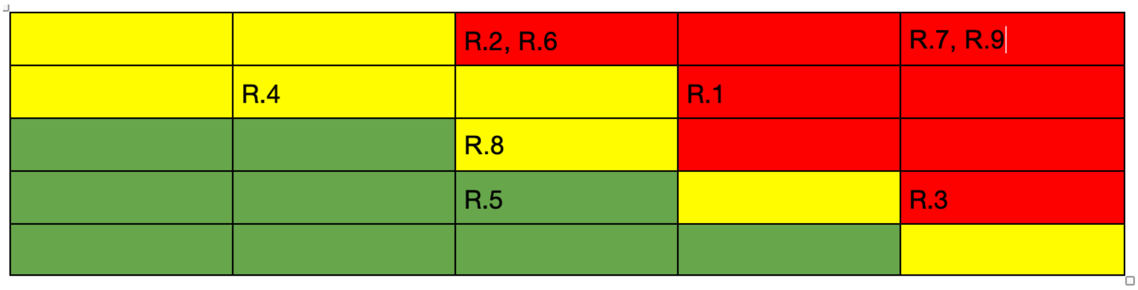
Highlighted below are the risks identified by the team. L and C stand for likelihood and consequences, on a scale from 1 (least) to 5 (most).

Table 13: **Risk Management**

Risk Id	**Risk**	**Category**	**L**	**C**	**Mitigation Strategy**	**Risk Owner**
R.1	Lack of availability of Test Site conforming MBZIRC Specifications	Schedule, Technical	4	4	Talk to Sebastian to at least create a dummy test site of smaller scale by 27 September 2019.	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 during ICRA 2019. 2. Procure and use Induction cooktops to simulate fire or some heating plate till the TBD status is resolved by 15 February 2019.	Shubham
R3	Extra effort on repairing/maintaining the UAV and AGV	Schedule, Cost	5	2	Maintain a contingency reserve especially for the UAV like motors, propellers and ESC	Akshit
R4	Data/Code Corruption	Technical, Schedule	2	4	Always take a copy of the code/data, or distribute code on the cloud and share among team members	Akshit
R5	Tilted Hexacopter performance not up to the mark	Schedule, Technical	3	2	Tentatively move towards a non-tilted version	Parv

Risk Id	Risk	Category	L	C	Mitigation Strategy	Risk Owner
R6	ORB SLAM scale issue	Schedule, Technical	3	5	1. Temporarily port to Intel Realsense Tracking Camera for SVD 2. Look for different SLAM algorithms which are dense and at par with ORB-SLAM2	Parv
R7	Install T265 bindings on Jetson	Schedule, Technical	5	5	Use Team RAMS patched kernel image	Akshit
R8	Communications link between systems not reliable for MBZIRC	Technical	3	3	Have a dual band high range comms link ready for MBZIRC which is compatible with TX2 and ZOTAC	Akshit
R9	Jetson USB buffer/processing not sufficient for multiple peripherals	Schedule, Technical	5	5	Port the whole system to Intel NUC if patches with jetson don't work	Akshit

Based on the above risks identified, some of the risks have been mitigated in the Spring semester and some of them span both the semesters of the project course. A likelihood v/s consequence matrix has been made which identifies the impact of having these risks on-board.



🟨	🟨	🟥 R.2, R.6	🟥	🟥 R.7, R.9
🟨	🟨 R.4	🟨	🟥 R.1	🟥
🟩	🟩	🟨 R.8	🟥	🟥
🟩	🟩	🟩 R.5	🟨	🟥 R.3
🟩	🟩	🟩	🟩	🟨

Likelihood v/s Consequence
Green = Low impact on risk occurrence, Yellow = higher and Red = Highest Impact

Figure 19: **Likelihood v/s consequence matrix**

8. Conclusion

8.1. Lessons Learned - Spring 2019

We faced multiple crashes when we tested our code directly on the UAV. So, we realized that before every UAV flight, we should first test the code in the simulator. It will not only avoid unexpected damage but will also save a lot of testing time. Moreover, we allocated/planned very less time for UAV testing but it took much longer to test the flights due to gain tuning and longer repair/maintenance time.

We spent a lot of time on stabilizing our UAV flight. Though we were using proven flight stack (PX4), we faced multiple stability issues due to tilted-rotor configuration. Most of our time was spent on testing/gain tuning tilted-rotor configuration which could have been avoided. Due to this, we couldn't have much progress in the first few weeks. When we moved back to the normal hex configuration it improved the stability of our system. And when we moved from plastic propellers to carbon fiber propellers it significantly improved the stability.

While testing Intel Realsense tracking camera on AGV, we found that it drifts in the outdoor environment. So, we need to fuse the tracking camera's output with other sensors (like GPS, IMU) to improve localization accuracy. Also, while installing Intel real sense camera drivers, we messed up our operating system and it stopped booting up. Despite having code backups and version control we had to install all the software tools like ROS, Nvidia drivers, etc. again on the system which consumed a lot of time. So, we realized that taking the OS snapshot after every milestone is very important.

On our UAV we are fusing the Intel tracking camera's output with PX4 IMU using onboard EKF2. 1 in 20 times, the UAV pose starts drifting even in a stationary position. We checked the tracking camera's output and onboard IMU's output and both were correct but the fused output was drifting. So, based on our analysis we realized that PX4 EKF2 is not reliable.

Just a day before Spring validation demonstration, our water tank mount broke and it took more than 3 hours to print. Unfortunately, MRSD lab's 3D printer was also not working. We spent a couple of hours to fix the printer and then waited for about 3 hours for the 3D part to get printed. We realized that we could have saved that time by keeping spare mechanical mounts for all the subsystems.

We relied on Nvidia Jetson TX2 on board's Wifi module for all the wireless communication. But its onboard Wifi module doesn't have good range and works only in the line of sight. We literally spent hours trying to connect (SSH). This could have been avoided by reviewing its Wifi module's performance online. So, we are planning to use Intel's dual-band antenna Wifi module for improving the range and connection reliability.

8.2. Key fall activities

We are planning to work mainly on path planning and collaboration subsystems during our fall semester. As mentioned in the project management section, a high-level Gantt chart is prepared to track all the activities for our fall semester. The first task will be to work on the mistakes done in the spring semester. So, we are planning to adhere to the guideline of testing the code on the simulator before every flight test to reduce the repair/maintenance time. Moreover, some reserve time will be allocated for UAV flight test to handle unexpected crashes/repair. In the spring semester, AGV localization was completely based on the Intel tracking camera's output. But to improve the reliability of the system, we are planning to add the sensor modality (like GPS, IMU) for the outdoor navigation. Developing robust state estimator for fusing IMU and Realsense tracking camera output on both UAV/AGV will significantly improve localization accuracy.

The second task will be to implement global primitive paths that take the system to an area of interest (building) from paths that achieve high coverage and integrate it into the existing ROS pipeline. Along with it, we will have to integrate implemented RRT based local planner that takes 3D maps from SLAM and gives collision-free paths.

The third major task will be to establish a reliable WiFi communication link between UAV and AGV. Further, we need to develop a shared database to store key information and implement a task allocation algorithm.

9. References

[1] National Fire Protection Association (https://nfpa.org)

[2] Raul Mur-Artal, Juan D. Tardos, J. M. M. Montiel and Dorian Galvez-Lopez Real-Time SLAM for Monocular, Stereo and RGB-D Cameras, with Loop Detection and Relocalization Capabilities

[3] Steven M LaValle: Rapidly-Exploring Random Tree: A new tool for path planning

[4] MBZ-International Robotics Competition: https://www.mbzirc.com/

[5] Titus Cieslewski, Siddharth Choudhary and Davide Scaramuzza: Data-Efficient Decentralized Visual SLAM

[6] Paul Furgale, Joern Rehder, Roland Siegwart (2013). Unified Temporal and Spatial Calibration for Multi-Sensor Systems. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan.

[7] Raul Mur-Artal and Juan D. Tard: ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras.

[8] https://www.flir.com/globalassets/imported-assets/document/boson-engineering-datasheet.pdf

[9] https://dev.px4.io/en/ros/external_position_estimation.html

[10] https://dev.px4.io/en/simulation/ros_interface.html

[11] https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/IntelRealSenseTrackingT265Datasheet.pdf
https://github.com/IntelRealSense/librealsense/blob/master/doc/t265.md