

# Heterogeneous Multi-Robot Sampling

## Preliminary Design Review



Team G: SAMP



# Outline:

1. Project description
2. Use case
3. System-level requirements
4. Functional architecture
5. Cyberphysical architecture
6. System/subsystem description & current status
7. Project management

# Project Description:

## Conventional Environmental Modelling

- Manual Sampling
- Discretized and limited coverage
- Tremendous manpower

## Robotics Automation Environmental Modelling

- Efficient and automated solution
- Better mobility and coverage
- Frequent modelling update

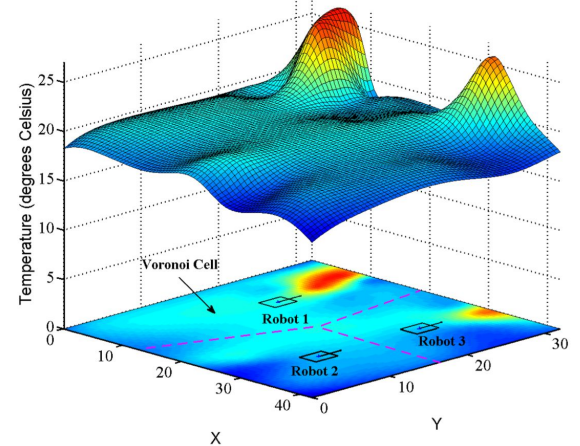
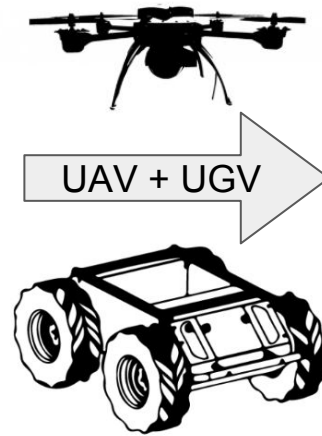


# Project Description:

We aim to deliver a UAV-UGV team that performs online environmental sampling and modeling collaboratively given an outdoor area with different terrains.



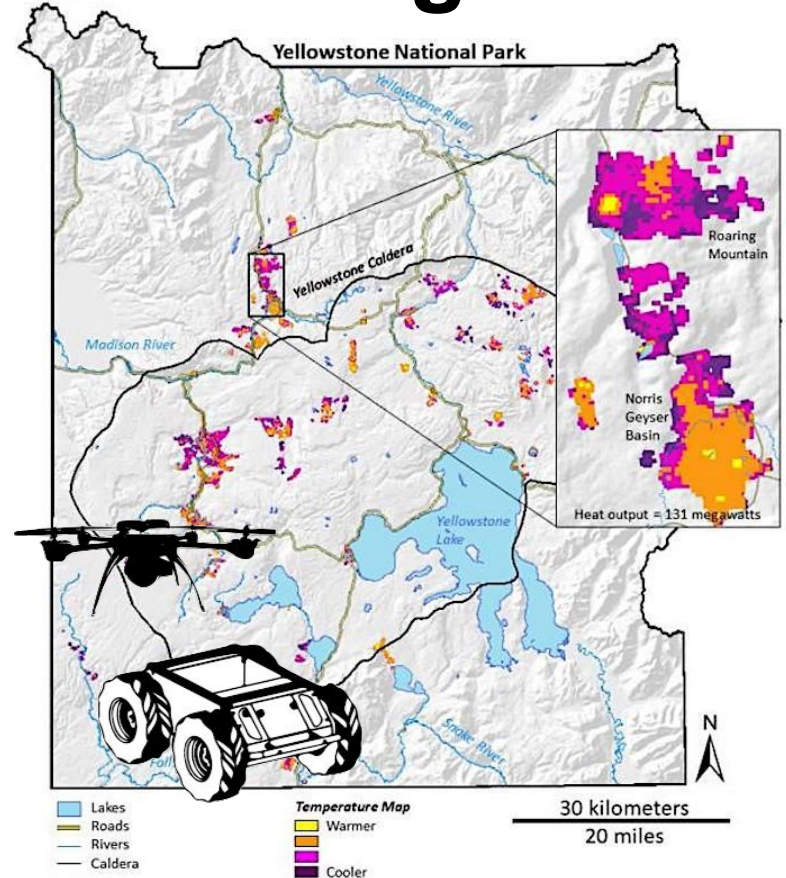
<https://secondnexus.com/environment/yellowstone-caldera-nasa-supervolcano/>



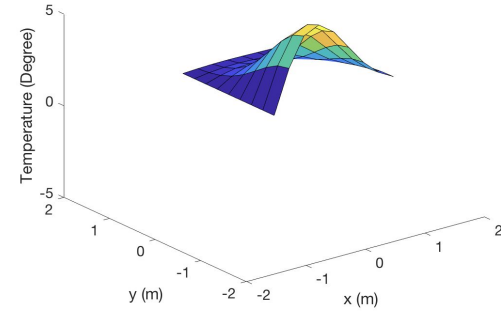
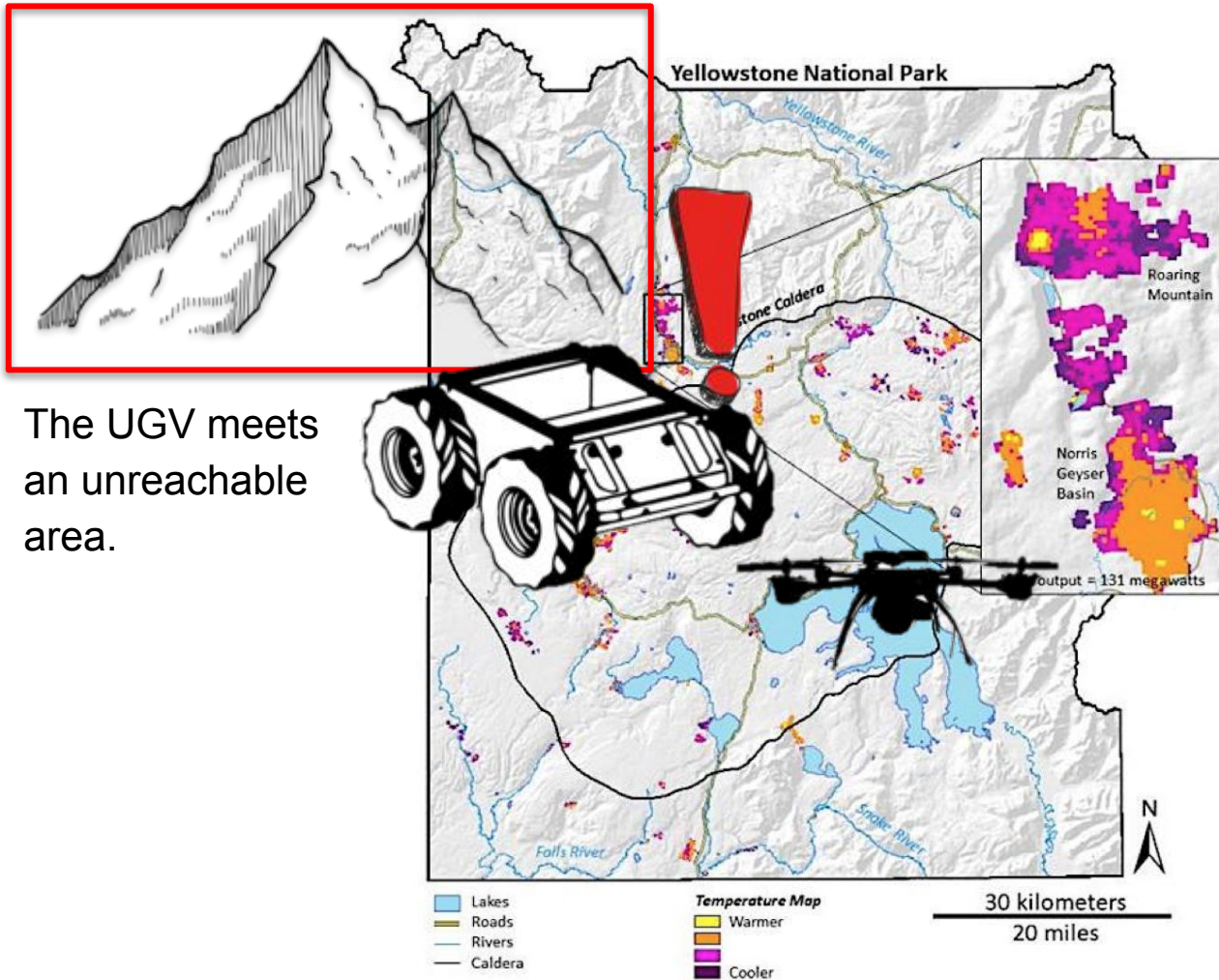
Wenhao and Katia, ICRA, 2018

# Use Case: Scientific Monitoring

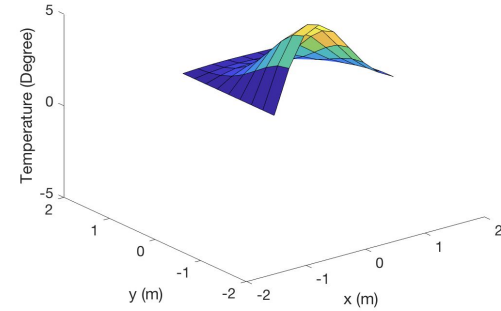
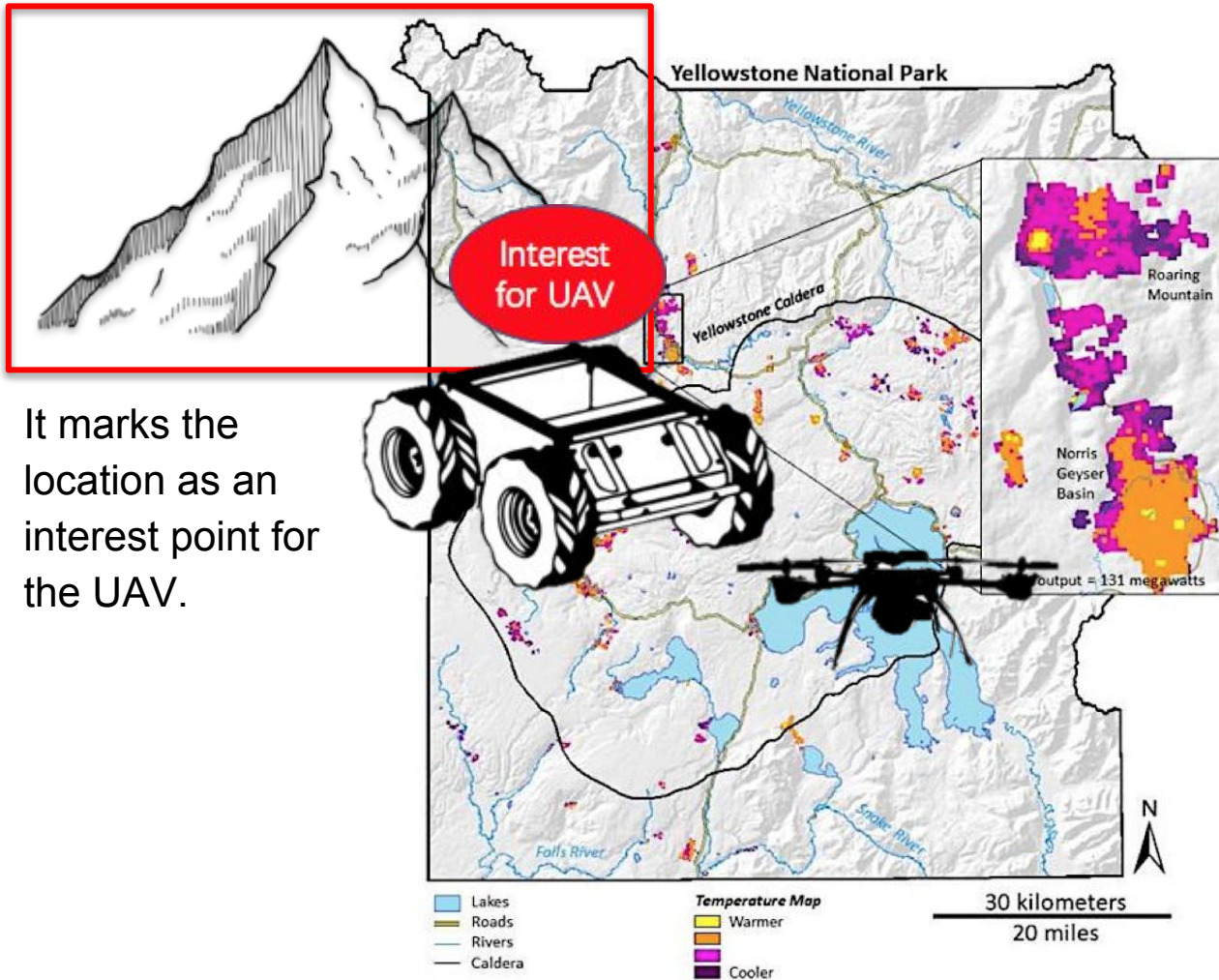
- Environmental Scientist Tom wants to study Yellowstone's thermal activity.
- Tom first tells the system the region where he wants to monitor the temperature.
- One UAV and one UGV enter the region and start collecting temperature samples collaboratively.



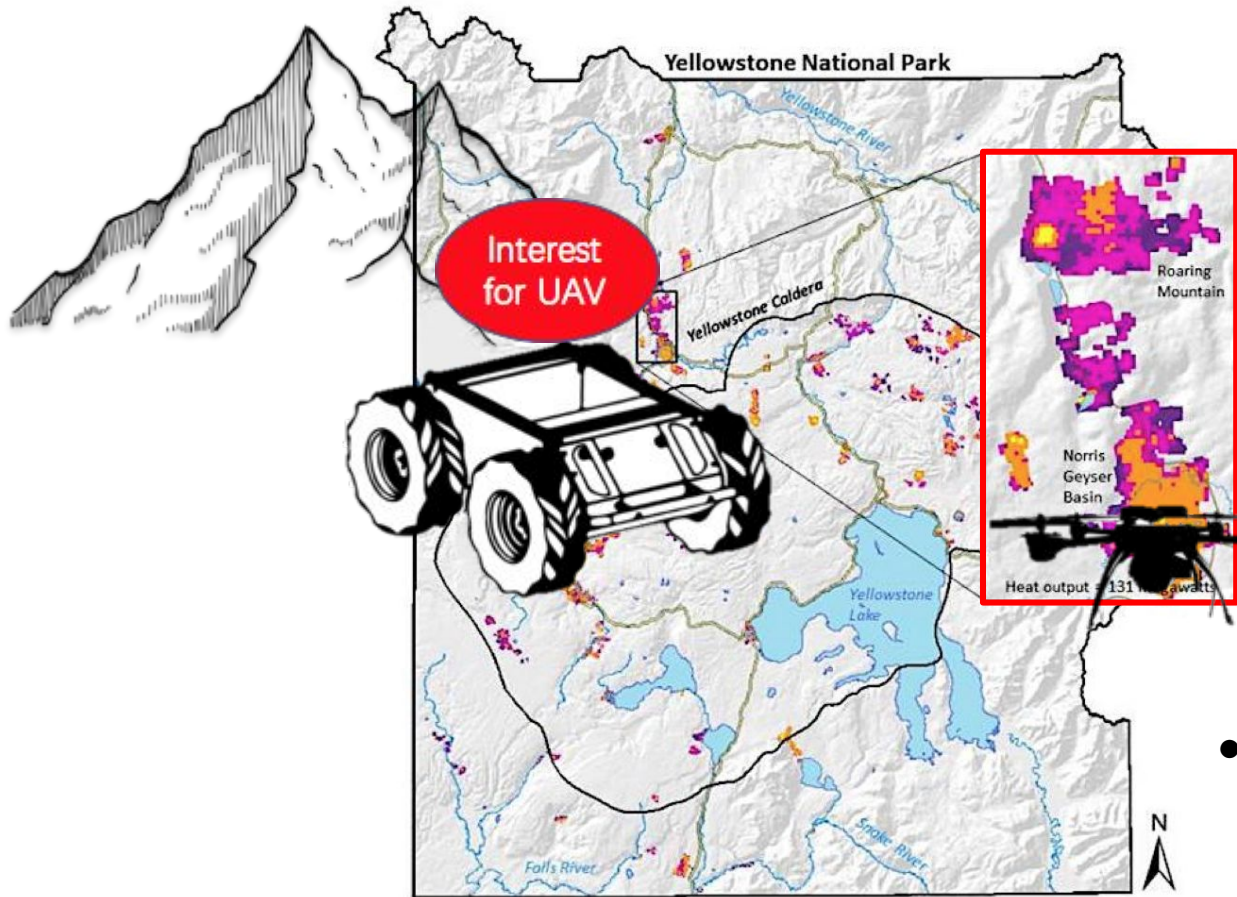




- The UGV meets an unreachable area.



- It marks the location as an interest point for the UAV.



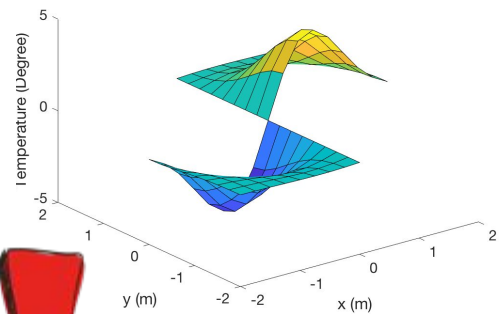
Interest for UAV

Heat output = 131 megawatts

- Lakes
- Roads
- Rivers
- Caldera

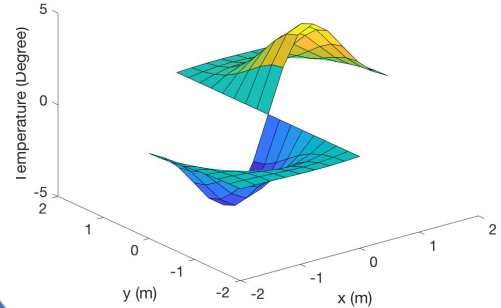
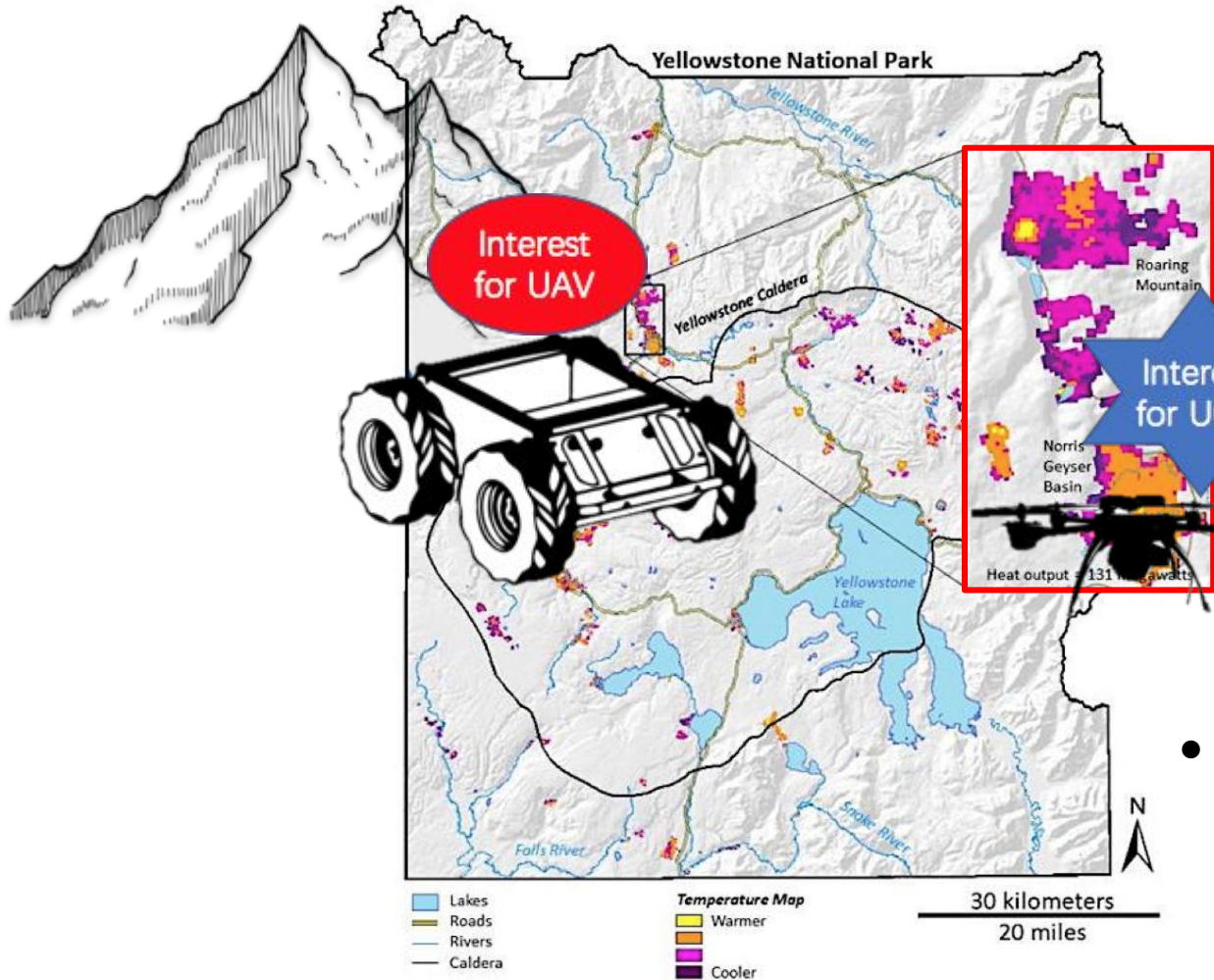
**Temperature Map**  
■ Warmer  
■  
■  
■ Cooler

30 kilometers  
 20 miles

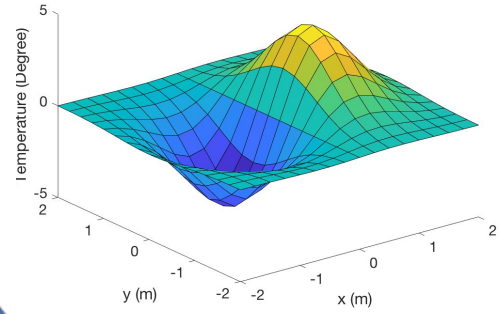
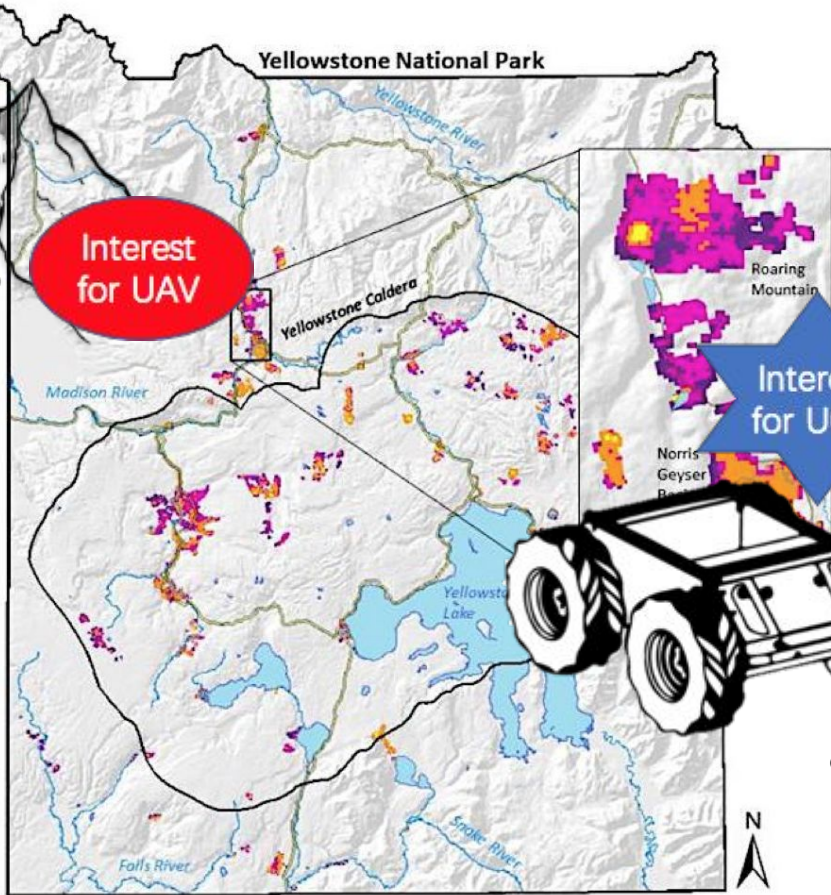


- The UAV meets an area with dense information





- It marks the location as an interest point for the UGV for accurate measuring.



- UAV and UGV go to next interest points for sampling.



- When finished, robot team returns to the starting point.

# Functional Requirements:

F.R.1 Generate an environmental phenomenon distribution model for the area of interest.

F.R.2 Self-Identify informative locations to take samples from.

F.R.3 Collect accurate samples at discrete locations across the area.

F.R.4 Correct and update the model during sampling.

F.R.5 Navigate autonomously in the given terrain.

F.R.6 Plan obstacle-free paths autonomously.

# Performance Requirements:

M.P.1 Generate an temperature distribution model for an area of interest within the dimension 20m x 20m x 5m.

M.P.2 The temperature distribution model accuracy greater than 80%.

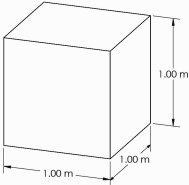
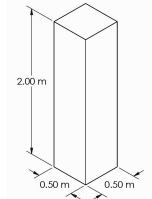
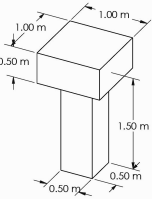
M.P.3 Self-select informative point which reduces local variance by at least 3% at each time.

M.P.4 Collect temperature sample with error within +/- 2 °C.

M.P.5 Update the model after receiving every 10 samples.



# Performance Requirements:

Obstacle	Quantity
	2
	2
	4

M.P.6 Navigate autonomously in the area with success rate greater than 80%.

M.P.7 Achieve sample localization accuracy better than +/- 2 m.

M.P.8 Plan obstacle-free path through randomly deployed obstacles.

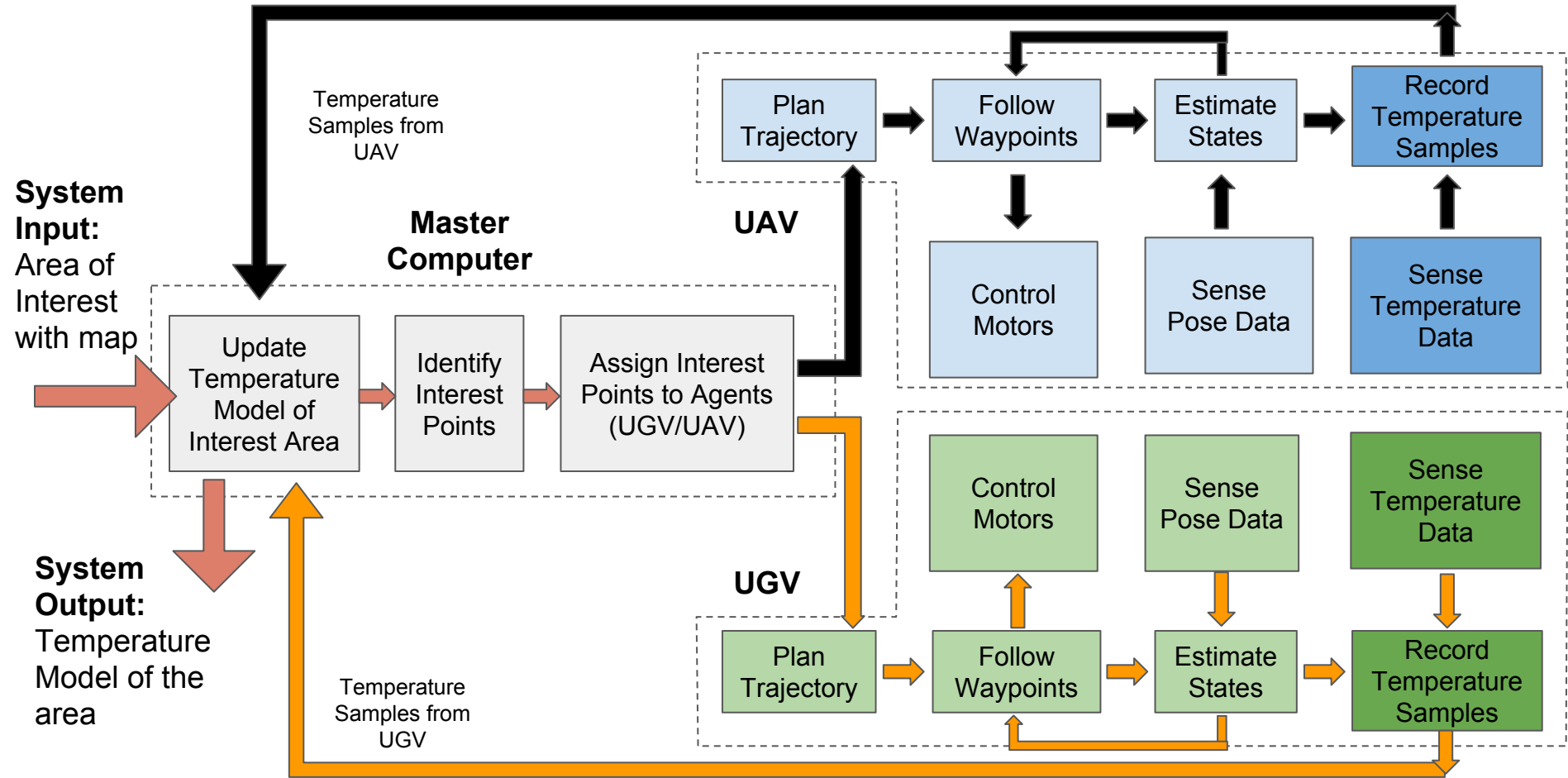
M.P.9 Last at least 15 minutes for one deployment.

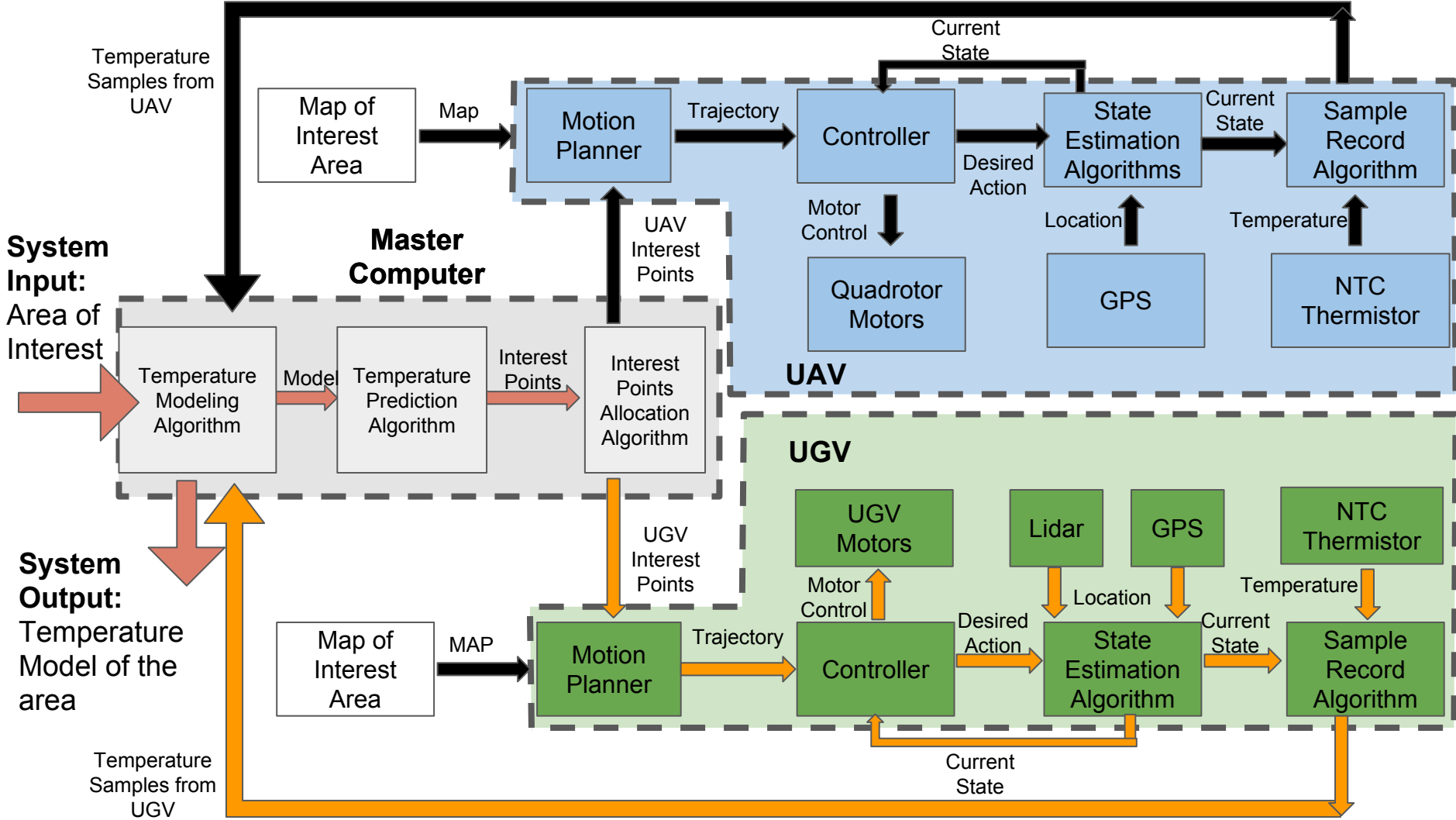
# Nonfunctional Requirements:

The system will:

1. Have safety features:
  - a. UAV and UGV have no sharp edges.
  - b. UAV has drone blade guards
  - c. Emergency Stop
2. Be environmental friendly:
  - a. Maintain a low noise level
  - b. Not damage operating environment
3. Have high extensibility:
  - a. Scale up to multiple heterogeneous robots





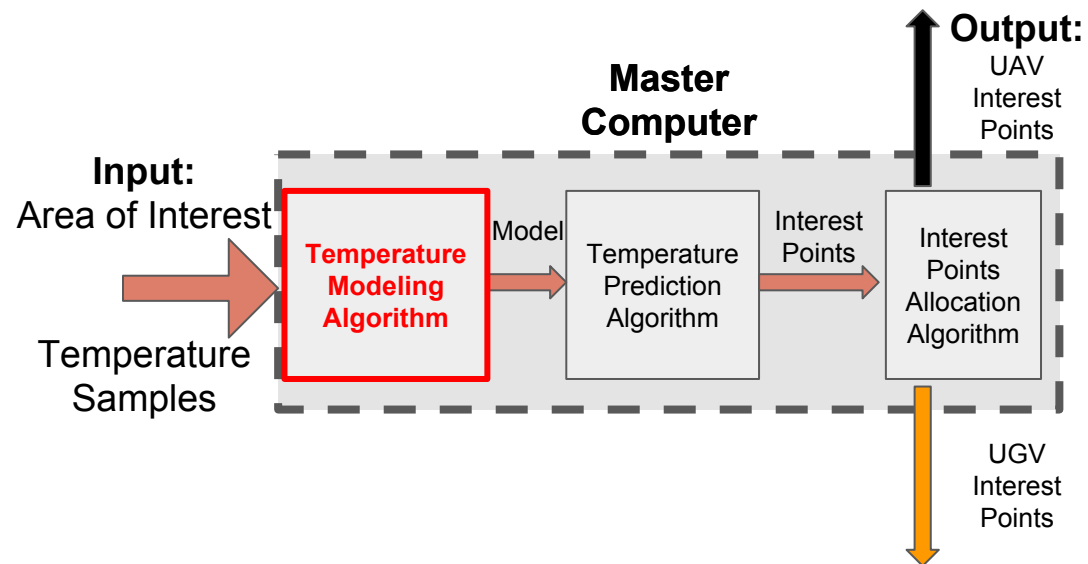




# Master Computer Description

Procedure:

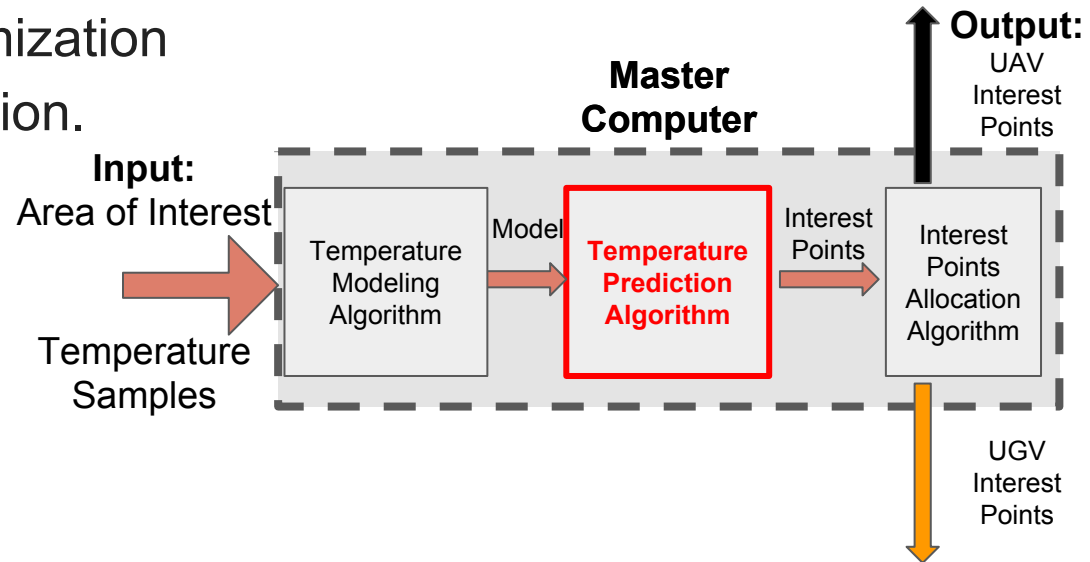
1. Mixture of Gaussian Process Models for temperature modeling.



# Master Computer Description

Procedure:

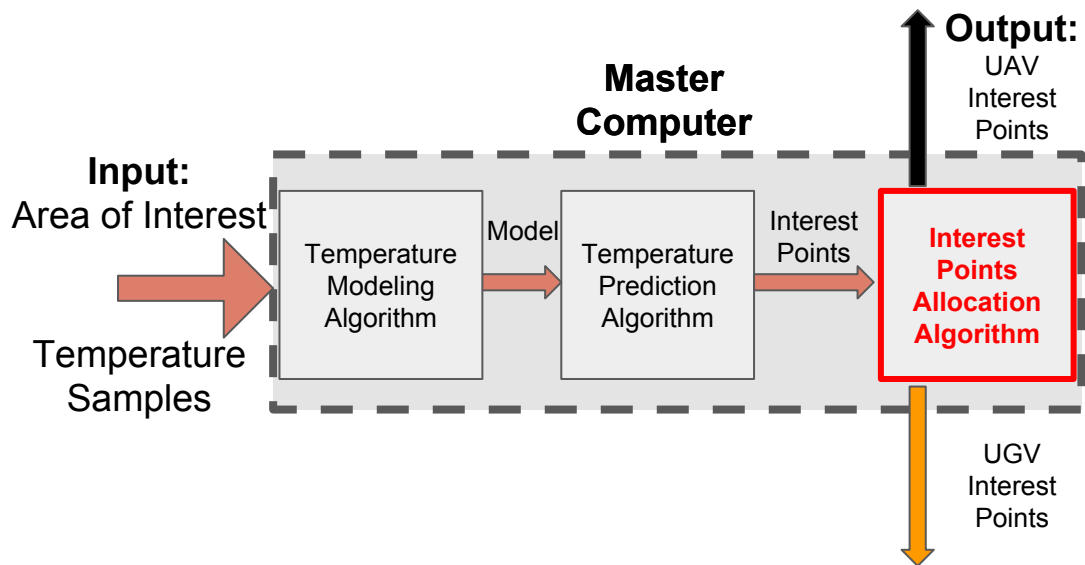
2. Expectation and Maximization for temperature prediction.



# Master Computer Description

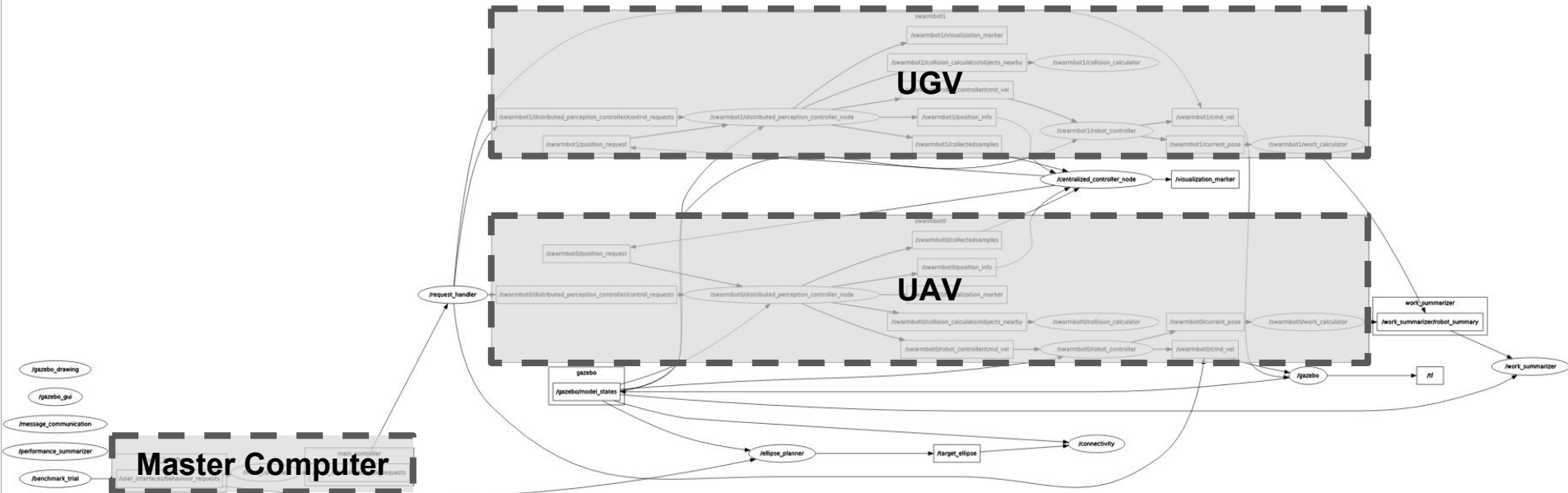
Procedure:

3. Interest point allocation
  - a. UAV/UGV Mobility
  - b. Utility
    - i. Prediction Accuracy
    - ii. Distance



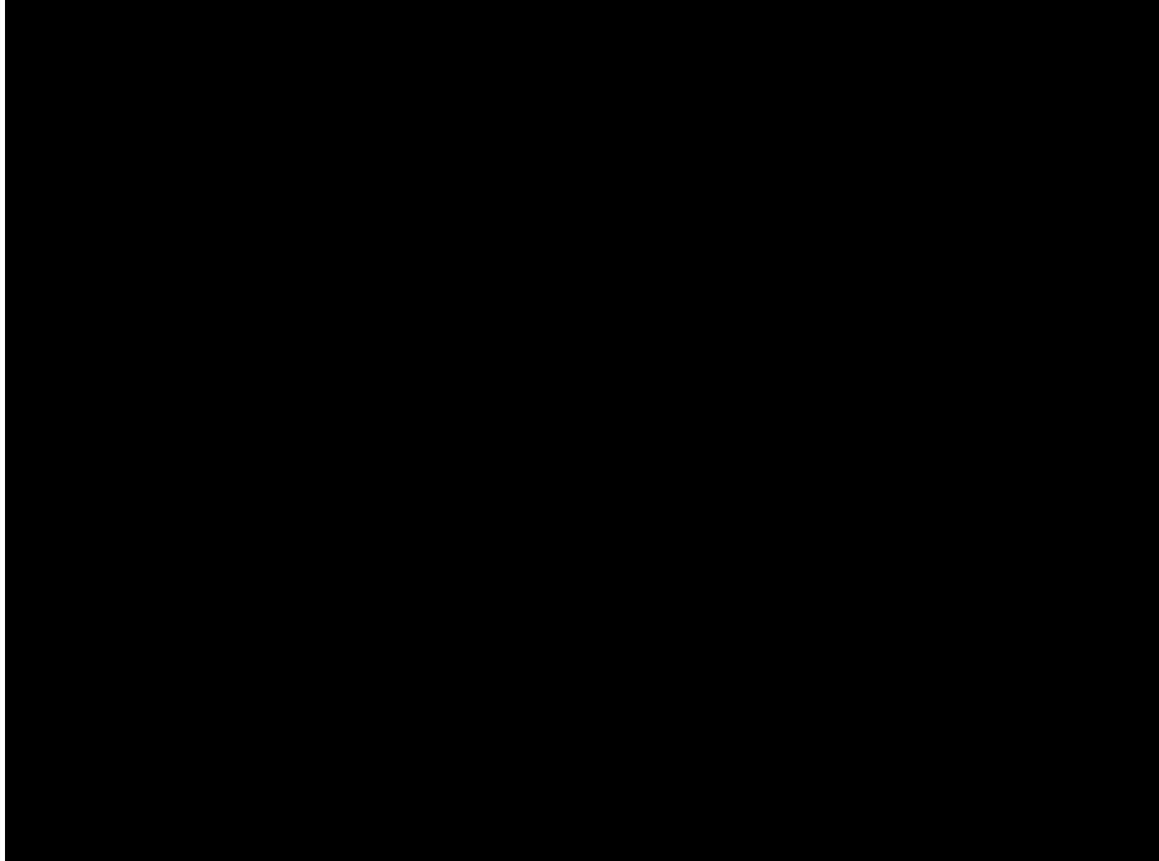
# Master Computer Status

Validated and simulated algorithms in ROS.





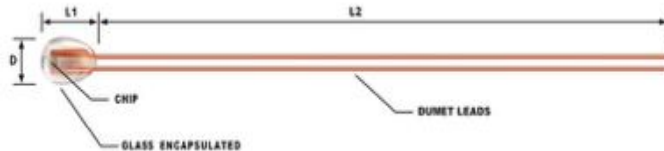
# Master Computer Status



# Master Computer Future Plan

1. Simulate heterogeneous robots
  - a. Create terrain features in the map, e.g. lake, cliff, hill
  - b. Assign interest points based on heterogeneous robots' different mobilities
2. Establish communication with Jackal and Pelican
  - a. Command the physical robots from master computer
  - b. Receive temperature and location information from the robots

# UGV Subsystem Description



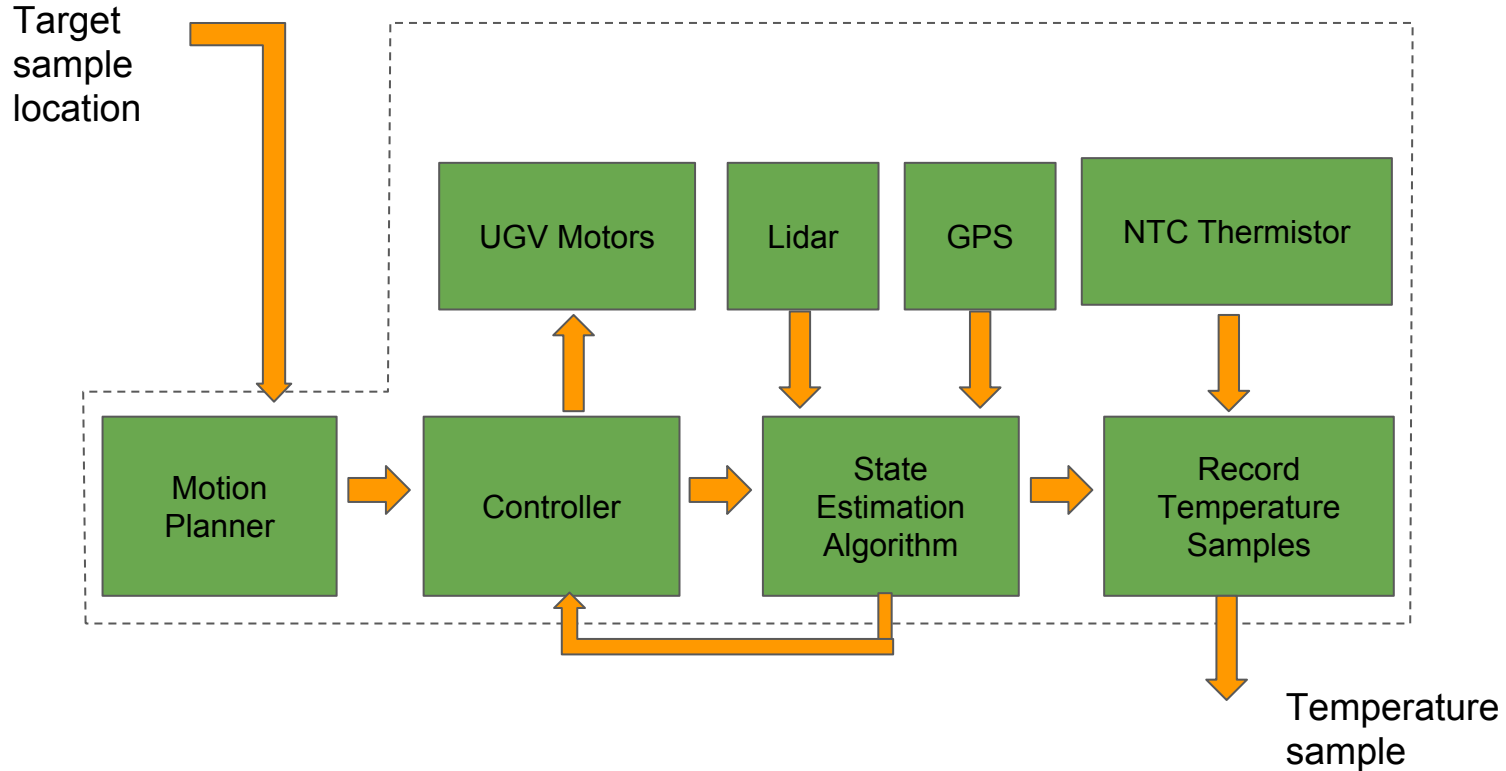
## Jackal UGV

- lightweight and waterproof
- flexible platform for integrating sensors and utilizing its ROS API
- Intel i5 onboard computer
- GPS
- wireless connectivity via both Bluetooth and wifi
- payload up to 20kg
- 4 hours duration with standard loads
- Velodyne VLP-16 LiDAR

## NTC Thermistor

- operating range:  $-55\text{ }^{\circ}\text{C}$  to  $+200\text{ }^{\circ}\text{C}$
- temperature sensitivities:  $-3\%$  to  $-6\%$  per  $^{\circ}\text{C}$
- experience large change in resistance per Celsius

# UGV Subsystem Description



# UGV Subsystem Status

1. Finished setting up onboard computer
2. Finished sensor data monitoring via ROS message
  - a. GPS
  - b. IMU
  - c. LIDAR

```
zy0zy-Blade:~$ rostopic echo /imu/data
header:
  seq: 61512
  stamp:
    secs: 1551836978
    nsecs: 962782103
  frame_id: "imu_link"
orientation:
  x: -0.000223615267207
  y: 0.00544155383426
  z: -0.145594733012
  w: 0.989329324951
orientation_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: -0.00256403154908
  y: 0.00111855398223
  z: 0.00337607615111
angular_velocity_covariance: [1.218467815533586e-07, 0.0, 0.0, 0.0, 1.218467815533586e-07, 0.0, 0.0, 0.0, 1.218467815533586e-07]
linear_acceleration:
  x: -0.0959428813466
  y: -0.0182150108661
  z: 9.7183762932
linear_acceleration_covariance: [8.661248102725949e-06, 0.0, 0.0, 0.0, 8.661248102725949e-06, 0.0, 0.0, 0.0, 8.661248102725949e-06]
---
header:
  seq: 61513
  stamp:
    secs: 1551836978
    nsecs: 975782103
  frame_id: "imu_link"
orientation:
  x: -0.000749605730448
  y: 0.00425931092381
  z: -0.14557960899
  w: 0.989337088059
orientation_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: -0.00196781757415
  y: 0.00153848876121
  z: 0.00293904106012
angular_velocity_covariance: [1.218467815533586e-07, 0.0, 0.0, 0.0, 1.218467815533586e-07, 0.0, 0.0, 0.0, 1.218467815533586e-07]
linear_acceleration:
  x: -0.0850895633837
  y: -0.0245602077888
  z: 9.7115206593
linear_acceleration_covariance: [8.661248102725949e-06, 0.0, 0.0, 0.0, 8.661248102725949e-06, 0.0, 0.0, 0.0, 8.661248102725949e-06]
---
```

# UGV Subsystem Status

3. Finished manual control test
4. Completed navigation simulation in Rviz

The screenshot displays the Rviz (Robot Visualization) interface. The central 3D view shows a robot (a small yellow and black vehicle) on a grid floor, with a colorful path (red, yellow, green, blue) leading around it. The 'Displays' panel on the left lists various visualization options, including 'Map' and 'Status: Warn'. The 'Views' panel on the right shows the current view settings, such as 'XYOrbit (rviz)' and 'Zero'. The status bar at the bottom indicates ROS Time: 1551760070.01, ROS Elapsed: 789.31, Wall Time: 1551760070.04, and Wall Elapsed: 789.24. The status bar also includes a 'Reset' button and a '31 fps' indicator.



# UGV Subsystem Future Plan

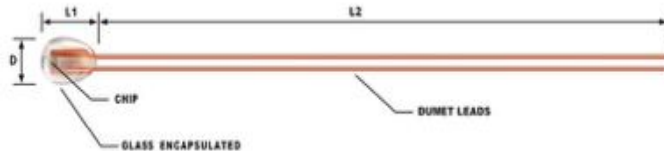
1. Integrate temperature sensor into Jackal
2. Realize GPS waypoint navigation
  - a. Integrate RTK GPS into Jackal for precise localization
  - b. Integrate waypoint navigation algorithm with RTK GPS reading
3. Receive commands from master computer

# UAV Subsystem Description



## AscTec Pelican UAV

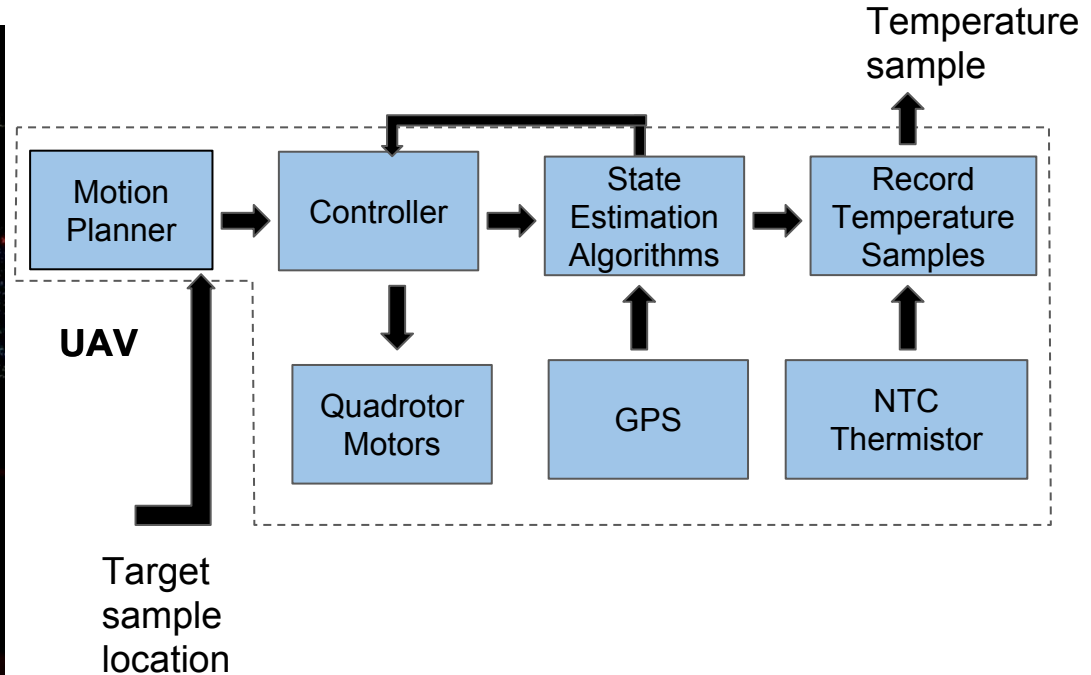
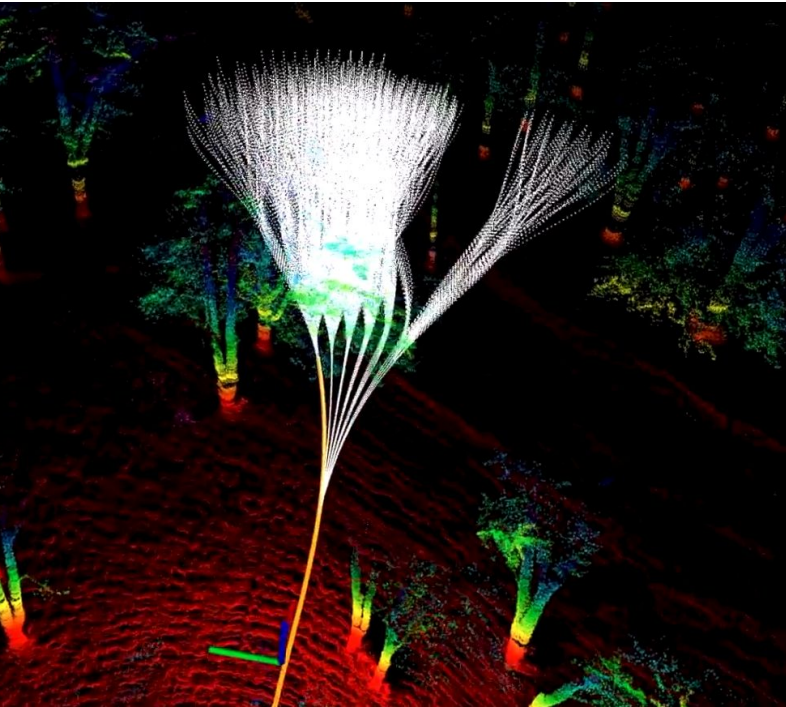
- Intel® Core™ i7 on-board computer
- Lightweight and robust
- Hokuyo Laser Scanner (up to 30m range)
- GPS
- Wifi and XBee (wireless serial)
- AscTec Autopilot sensor board
- 16 minutes flight time
- 16 m/s maximum speed



## NTC Thermistor

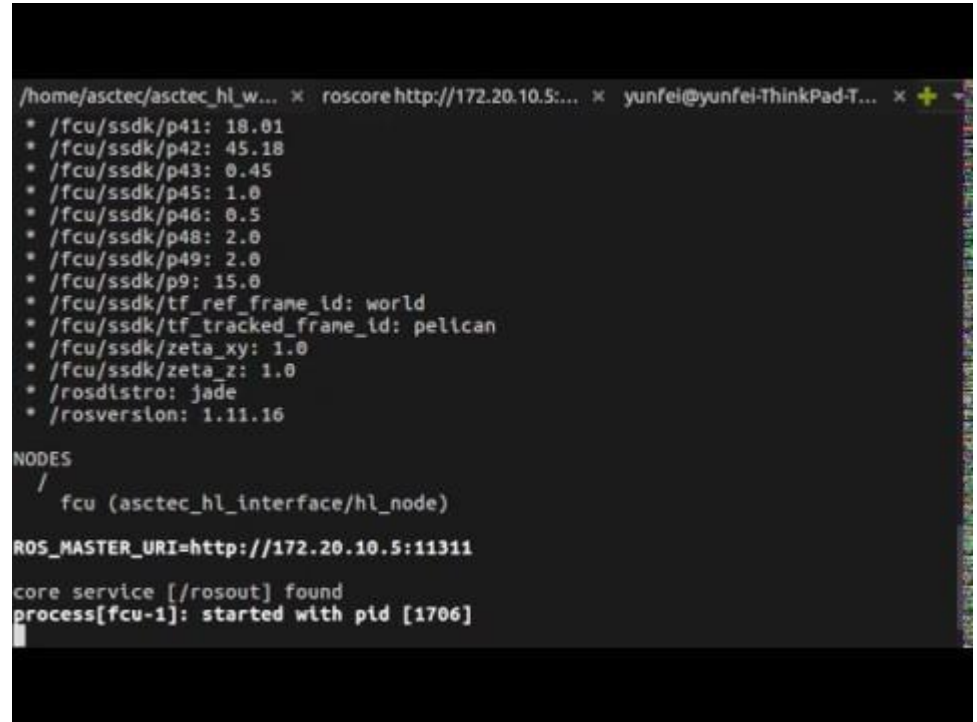
- operating range:  $-55\text{ }^{\circ}\text{C}$  to  $+200\text{ }^{\circ}\text{C}$
- temperature sensitivities:  $-3\%$  to  $-6\%$  per  $^{\circ}\text{C}$
- experience large change in resistance per Celsius

# UAV Subsystem Description



# UAV Subsystem Status

1. Finished setting up onboard computer OS
2. Completed communication specification between onboard computer and High Level Controller
3. Finished sensor data monitoring via ROS message

A terminal window screenshot showing ROS system status. The window title is "/home/asctec/asctec\_hl\_w...". The output lists various parameters for the /fcu/ssdk namespace, including p41 through p49, tf\_ref\_frame\_id, tf\_tracked\_frame\_id, zeta\_xy, zeta\_z, roslstro, and rosversion. Below this, it shows the NODES section with a single node 'fcu' of type 'asctec\_hl\_interface/hl\_node'. The ROS\_MASTER\_URI is set to 'http://172.20.10.5:11311'. Finally, it reports 'core service [/rosout] found' and 'process[fcu-1]: started with pid [1706]'.

```
/home/asctec/asctec_hl_w... x roscore http://172.20.10.5:... x yunfei@yunfei-ThinkPad-T... x +
* /fcu/ssdk/p41: 18.01
* /fcu/ssdk/p42: 45.18
* /fcu/ssdk/p43: 0.45
* /fcu/ssdk/p45: 1.0
* /fcu/ssdk/p46: 0.5
* /fcu/ssdk/p48: 2.0
* /fcu/ssdk/p49: 2.0
* /fcu/ssdk/p9: 15.0
* /fcu/ssdk/tf_ref_frame_id: world
* /fcu/ssdk/tf_tracked_frame_id: pelican
* /fcu/ssdk/zeta_xy: 1.0
* /fcu/ssdk/zeta_z: 1.0
* /roslstro: jade
* /rosversion: 1.11.16

NODES
/
  fcu (asctec_hl_interface/hl_node)

ROS_MASTER_URI=http://172.20.10.5:11311

core service [/rosout] found
process[fcu-1]: started with pid [1706]
```

# UAV Subsystem Status

4. Finished motor test using command via ROS message
5. Completed manual flight control test using Height Mode
6. Finished setting up temperature sensors on Ubuntu

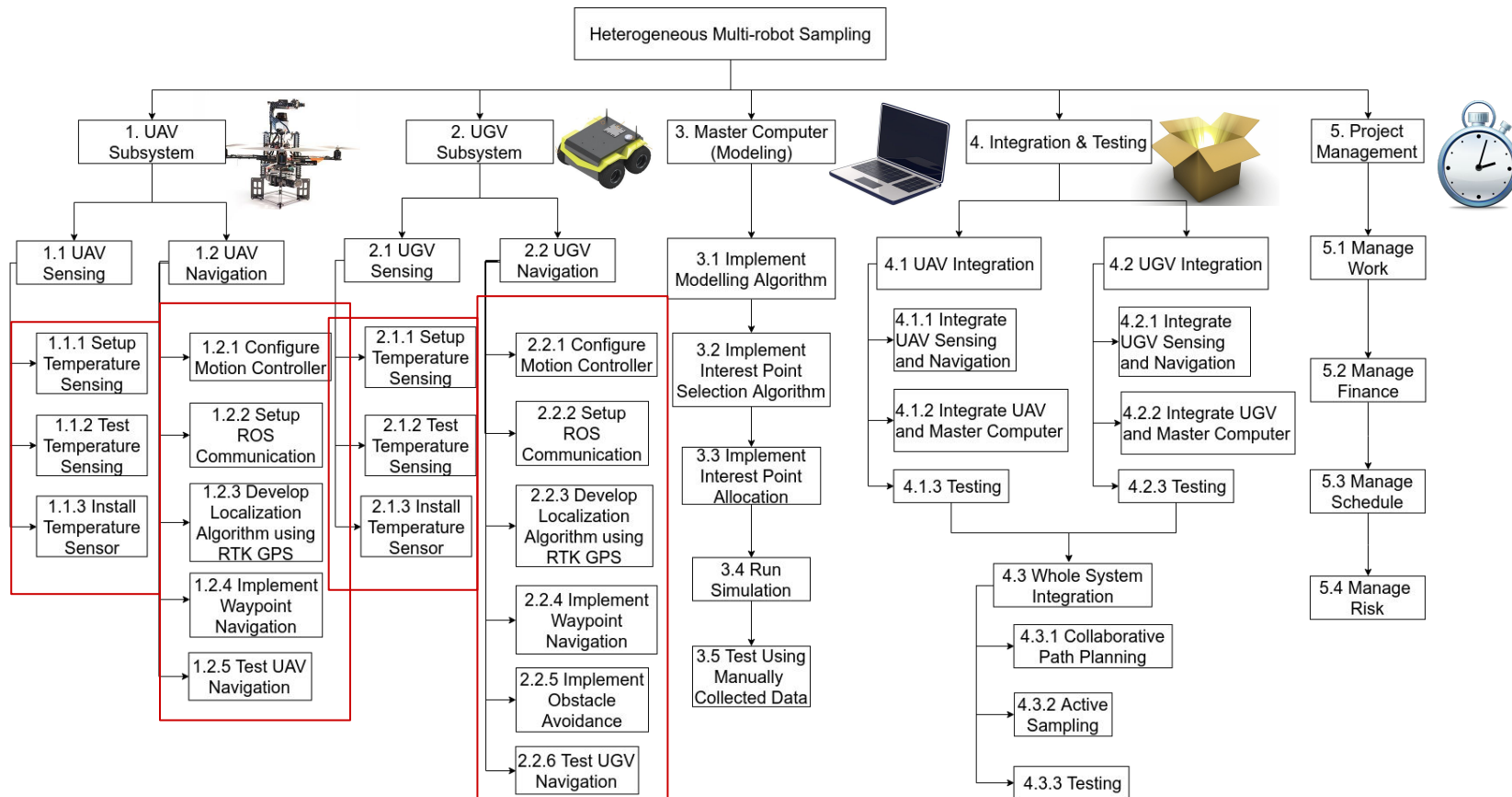


# UAV Subsystem Future Plan

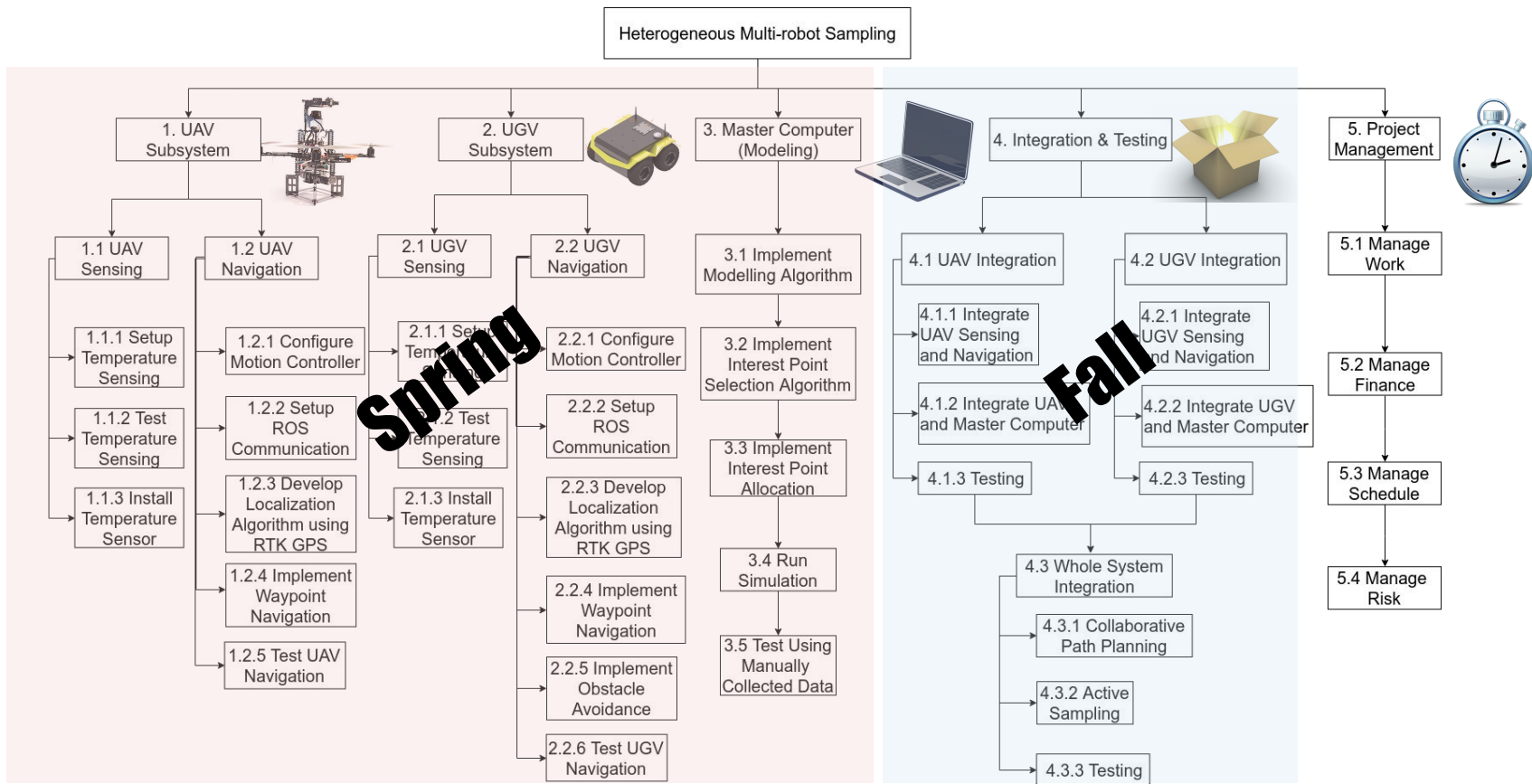
1. Integrate temperature sensor into AscTec Pelican
2. Archive autopilot algorithm
  - a. Manual flight test using GPS waypoint mode
  - b. Integrate RTK GPS reading with the control system
  - c. Flight test using waypoint command via ROS message



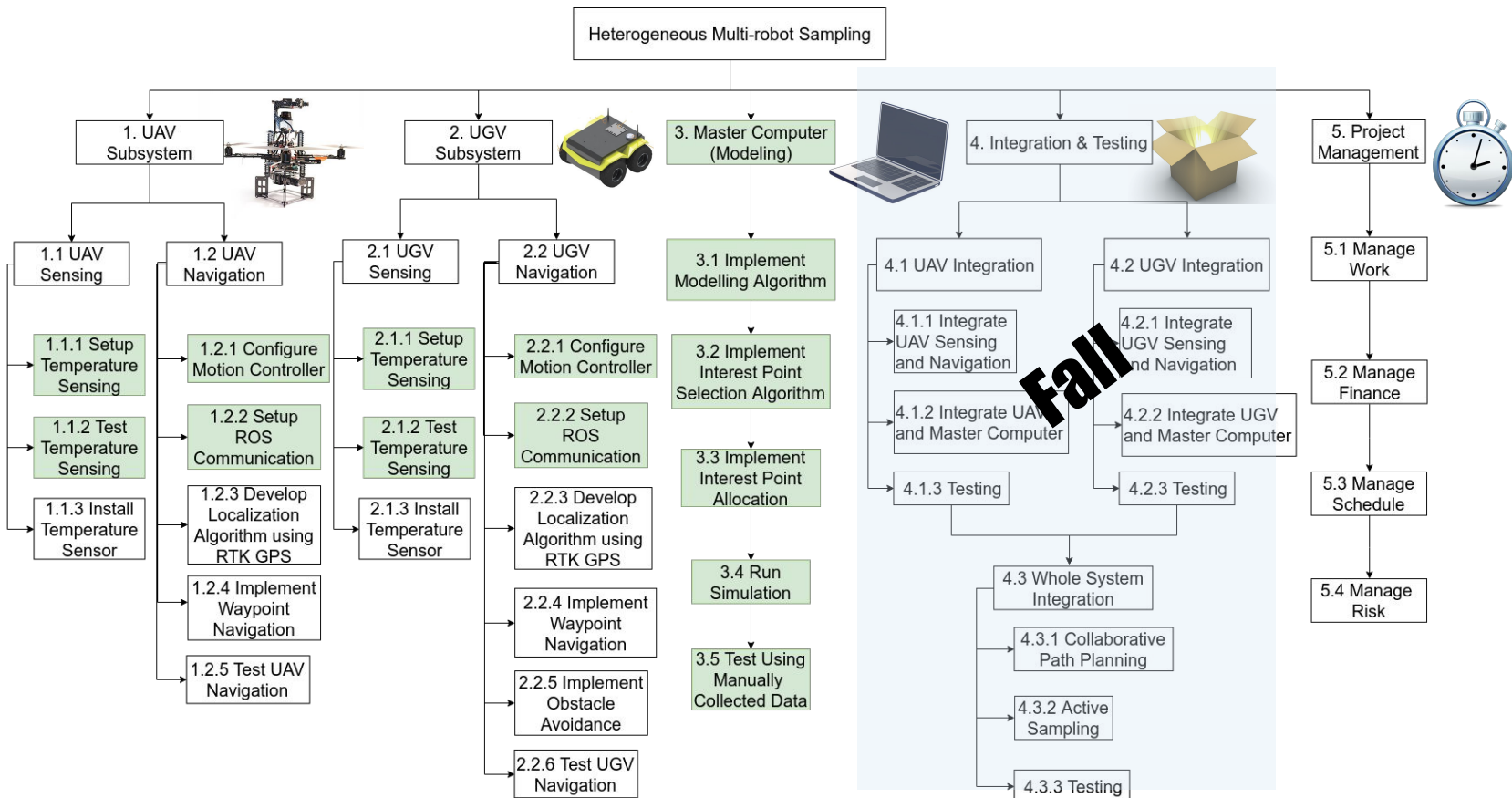
# Work Breakdown Structure



# Schedule



# Schedule



# Schedule

<b>Intended Finish Date</b>	<b>Milestone Description</b>	<b>Actual Finish Date</b>
9 Feb	Master Computer Simulation	23 Feb
20 Feb	Read data from temperature sensors	9 Mar
8 Mar	UGV localization using LiDAR	8 Mar
13 Mar	UGV Waypoint Navigation	-
31 Mar	UAV Controller	5 Mar

# Schedule

Date	Milestone Description
8 Apr	UAV Waypoint Navigation System
21 Apr	UAV Obstacle Avoidance
1 May	Spring Validation
6 May	Critical Design Review

# High-Level Test Plan

Milestone	Test Description
PR3	UGV Localization Test
PR4	UAV Localization Test
SVD	Master Computer Simulation, UAV & UGV Navigation
SVD Encore	Master Computer Simulation, UAV & UGV Navigation
PR5	UAV-Master Computer Integration Test
PR6	UGV-Master Computer Integration Test
FVD	Whole System Integration Test

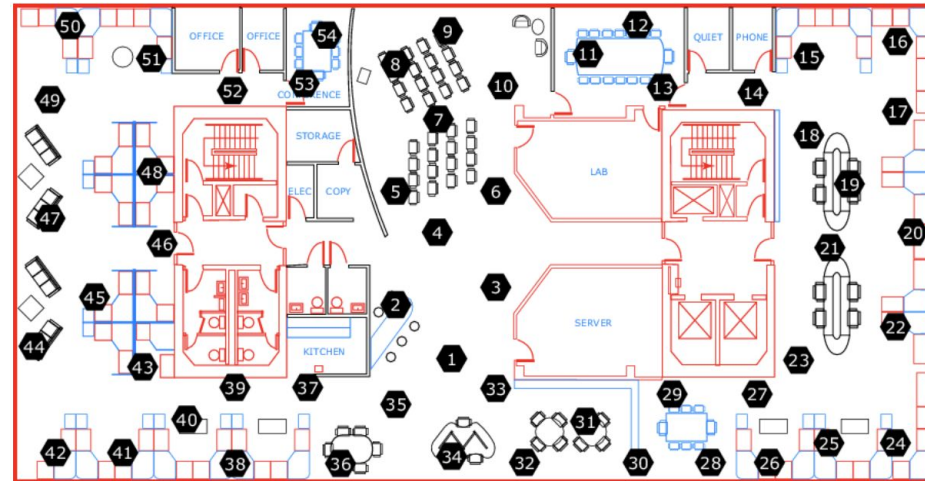
# Spring Validation Experiment

## Master Computer Validation Experiment

Equipment: Master Computer, Intel Lab Data

1. Master computer updates temperature model.
2. Master computer determines next interest points.
3. Master computer allocates points to UAV/UGV.
4. Return temperature data from dataset.
5. Loop through 1-4 until the model converges.
6. Compute root mean square error between the generated model and ground truth.

Pass Criteria: Error less than 1 °C.



<http://db.csail.mit.edu/labdata/labdata.html>



# Spring Validation Experiment

## UGV subsystem Validation Experiment

Location: Outdoor area within the dimension 20m x 20m x 5m

Criteria: Location error: less than 2m; Temperature error: less than 2 °C

Equipment: UGV, Tape measure, Stopwatch, Temperature Sensor

1. Power on UGV agent
2. Assign an interest point to UGV agent (and command it to stop once arrived)
3. After the UGV agent stops, measure the errors between the arrived location and desired location.
4. Export the recorded temperature sample from UGV and measure the error between the measured temperature and ground truth.

# Spring Validation Experiment

## UAV subsystem Validation Experiment



Location: Outdoor area within the dimension 20m x 20m x 5m

Criteria: Location error: less than 2m; Temperature error: less than 2 °C

Equipment: UAV, Tape measure, Stopwatch, Temperature Sensor

1. Power on UAV agent
2. Assign an interest point to UAV agent (and command it to collect temperature data and land once arrived)
3. After the UAV agent lands, measure the errors between the arrived location and desired location.
4. Export the recorded temperature sample from UAV and measure the error between the measured temperature and ground truth.

# Fall Validation Experiment

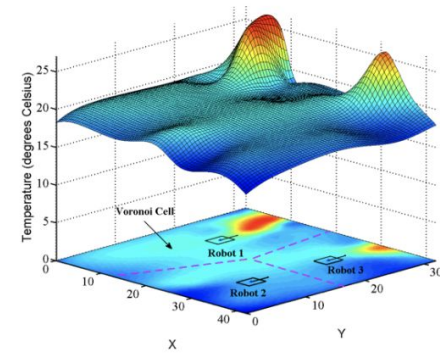
Integrated system Validation Experiment

Location: Outdoor area within the dimension 20m x 20m x 5m

Criteria: Temperature model accuracy: greater than 80%

Equipment: UGV, UAV, Master Computer, Stopwatch, Temperature Sensor

1. Measure ground truth temperature distribution manually.
2. Start master computer with an initial model.
3. Deploy UGV and UAV to collect temperature samples according to master computer's instructions.
4. After the temperature model converges, measure the difference between temperature ground truth and predictions.



# Budget

Part Name	Quantity	Unit Price	Total Price
Ground Truth Temperature Sensor	12	\$49.99	\$599.88
Ground Truth Temperature Sensor WiFi Gateway	1	\$99.95	\$99.95
Heat Source	2	\$16	\$32
Battery Monitor	1	\$9.89	\$9.89
On-board Temperature Sensor	4	\$13.99	\$55.96
Temperature Sensor Extension Cable	4	\$6.99	\$27.96
RTK-GPS Set (2 Rovers and 1 base)	1	\$2,000	\$2,000

**Total: \$2,865.64**

# Budget

Part Name	Quantity	Unit Price	Total Price
Ground Truth Temperature Sensor	2	\$49.99	\$599.88
Ground Truth Temperature Sensor WiFi Gateway	1	\$99.95	\$99.95
Heat Source	2	\$16	\$32
Battery Monitor	1	\$9.89	\$9.89
On-board Temperature Sensor	4	\$13.99	\$55.96
Temperature Sensor Extension Cable	4	\$6.99	\$27.96

**Current Spent:** \$325.32

**11.35%**

# Risk Management

ID	Risk	Type	Description	Likelihood	Consequence	Risk Reduction Plan
1	Electric System Failure	Technical	The battery or electric system fails due to incorrect operation.	3	4	<ul style="list-style-type: none"><li>• Add reverse voltage and overvoltage protection.</li><li>• Document and regulate operation.</li></ul>
2	Work Delay	Schedule	Heavy workload puts the team behind the schedule	5	5	<ul style="list-style-type: none"><li>• Optimize the WBS to break down the workload into manageable pieces.</li></ul>
3	Run Out of Budget	Financial	Run out of funds purchasing parts and repairing robots.	1	5	<ul style="list-style-type: none"><li>• Make purchasing decision carefully after trade study.</li></ul>
4	Latency for Real-time Operation	Technical	Communication latency between master computer and UGV fails real-time operation.	4	4	<ul style="list-style-type: none"><li>• Shut down unnecessary sensor publications.</li></ul>
5	Poor Weather for Validation Tests	Schedule	Poor weather prevents/delays the system from outdoor experiments.	4	3	<ul style="list-style-type: none"><li>• Monitor upcoming weather.</li><li>• Schedule tests beforehand.</li></ul>

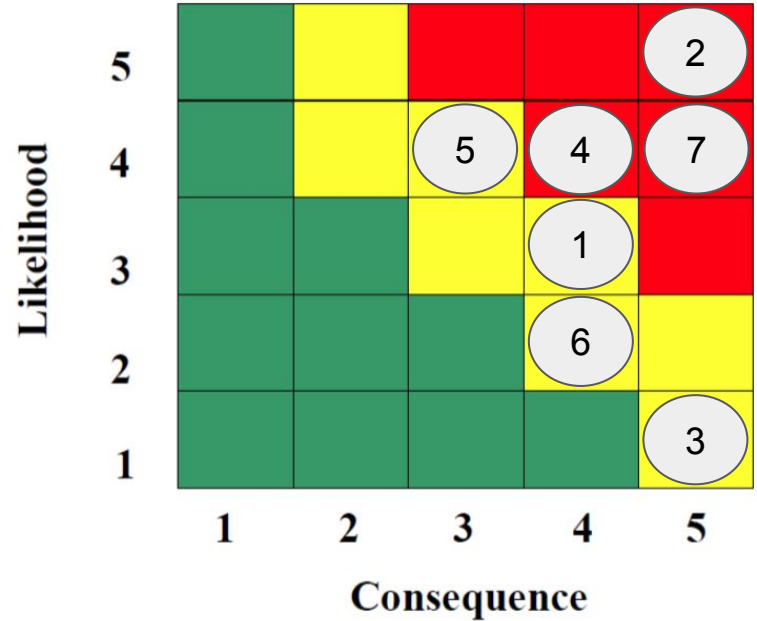
# Risk Management

ID	Risk	Type	Description	Likelihood	Consequence	Risk Reduction Plan
6	Even Temperature Distribution	Technical	Temperate difference in the test field is close to or smaller than sensor noise.	2	4	<ul style="list-style-type: none"><li>• Use sensors with higher sensitivities based on previous experiment results.</li><li>• Add heat sources to the test field to increase temperature variance.</li></ul>
7	Poor Localization Accuracy	Technical	Localization accuracy is not high enough considering the size of the test field.	4	5	<ul style="list-style-type: none"><li>• Use RTK GPS instead of built-in GPS.</li></ul>



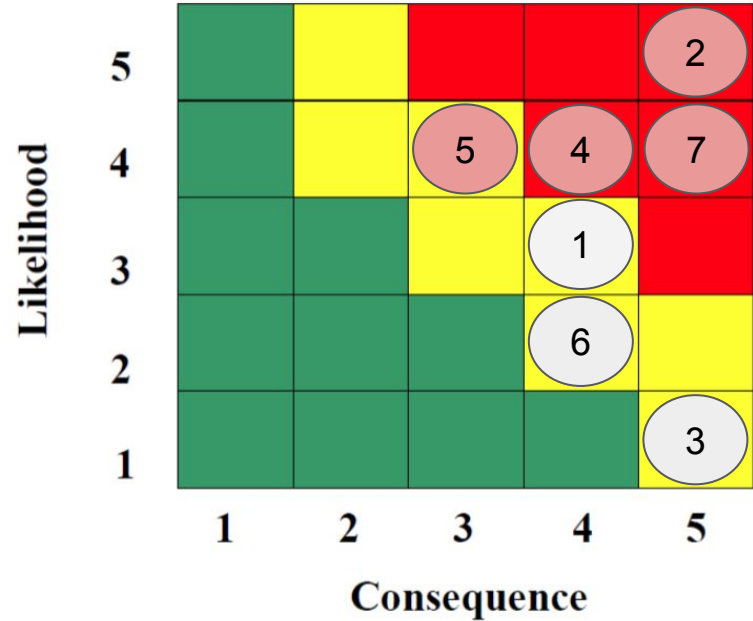
# Risk Management

1. Electric System Failure
2. Work Delay
3. Run Out of Budget
4. Latency for Real-time Operation
5. Poor Weather for Validation Tests
6. Even Temperature Distribution
7. Poor Localization Accuracy



# Risk Management

1. Electric System Failure
2. Work Delay
3. Run Out of Budget
4. Latency for Real-time Operation
5. Poor Weather for Validation Tests
6. Even Temperature Distribution
7. Poor Localization Accuracy



# Q&A

