MBZIRC 2023

White paper

Team Luna

1 Team information

1.1 Team members

Faculty Advisor: Dr Sujit P.B.
Student members:

- Swadhin Agrawal
- Gokul P
- Anushree Sabnis
- Raghav Thakar
- Manav Mishra
- Prithvi Poddar
- Kasi Viswanath
- Rajat Agrawal
- Prakrit Tyagi
- Yogesh Kumar
- Dhairya Shah
- Ninad Jangle
- Saad Hashmi
- Karthik Swaminathan
- Akshat Pandey
- Aditya Arun Nirmale

1.2 Teams contact

1.2.1 Main contact

Name: Dr Sujit P.B.
Email: sujit@iiserb.ac.in
Ph.no: +91 755 269 2649

1.2.2 Alternate contact points

Name: Swadhin Agrawal
Email: swadhin20@iiserb.ac.in
Ph.no: +91 7869 571 576

1.3 Team roaster

Name	**Affiliation**	**Email**	**Contact**
Dr Sujit P.B.	IISER Bhopal	sujit@iiserb.ac.in	+91 7552692649
Gokul. P	Manipal Institute of Technology	gokulgns@gmail.com	+91 9840797134
Raghav Thakar	Manipal Institute of Technology	raghavthakar12@gmail.com	+91 9910067476
Anushree Sabnis	VJTI Mumbai	absabnis_b19@me.vjti.ac.in	+91 9967585649
Swadhin Agrawal	IISER Bhopal	swadhin20@iiserb.ac.in	+91 7869571576
Manav Mishra	IISER Bhopal	manav20@iiserb.ac.in	+91 8369941750
Prithvi Poddar	IISER Bhopal	prithvi17@iiserb.ac.in	+91 9960283977
Kasi Viswanath	IISER Bhopal	kasi18@iiserb.ac.in	+91 9685016527
Rajat Agrawal	IISER Bhopal	rajatagrawal1307@gmail.com	+91 8923399636
Prakrit Tyagi	IISER Bhopal	ptyagi@iiserb.ac.in	+91 9004013464
Yogesh Kumar	IIIT Delhi	yogeshk@iiitd.ac.in	+91 8755919307
Akshat Pandey	Manipal Institute of Technology	akshatpandeyplus41@gmail.com	+91 8287388679
Aditya Arun Nirmale	Manipal Institute of Technology	adityanirmaleofficial@gmail.com	+91 9769298001
Ninad Jangle	VJTI Mumbai	nsjangle_b19@el.vjti.ac.in	+91 8879017402
Dhairya Shah	VJTI Mumbai	djshah_b19@et.vjti.ac.in	+91 9769016020
Karthik Swaminathan	VJTI Mumbai	kswaminathan_b19@me.vjti.ac.in	+91 9987605392
Saad Hashmi	VJTI Mumbai	shmohammed_b19@me.vjti.ac.in	+91 9925936955

1.4 Past experience

Our team is a group of diverse members constituting from 3 different universities. All of them working on different professional areas in robotics that includes computer vision, multi-autonomous vehicle systems, swarm robotics, robotic manipulator arms, robotic control, localization and navigation, and artificial intelligence like ML, deep learning and reinforcement learning. All of them have experience with at least one project or competition in their respective fields.

Dr Sujit P.B. himself is an expert in the field of Robotics (Multi-robotic system planning, controls and guidance). For the cooperative target search and collective payload transport, this team consists of four students Swadhin, Raghav and Manav. Swadhin and Manav are PhD students whose research area mainly focuses on swarm intelligence and multi-robot systems. Raghav brings in his experience on multi-robot task planning. This challenge requires expertise on computer vision since it will take place in a GNSS denied environment. Therefore, we have Gokul and Kasi bringing in their experience on SLAM, visual navigation and object detection. The final component is the robotic manipulator, for which Anushree brings in her expertise in mechatronics. She has also worked on multiple mobile manipulator systems. Adding up to her expertise, Rajat and Prakrit are experts in the same field. All the systems are required to perform the tasks in noisy sea environment for which we have a TCS research scholar Yogesh. He brings in his expertise in controls and guidance that includes visual servoing for moving targets, quad rotor dynamics and control, target tracking using MPC.

Our team members have experiences with different projects and competitions. Raghav and Gokul are members of the Project-MANAS, a team developing India's first autonomous car. Gokul was an RA at Robotics Research Center, IIIT-Hyderabad wherein he developed a novel navigation and planning algorithm specifically for multi-UAV system aimed at target search. He was also awarded the MITACS Fellowship and was a researcher at Queen's University, Kingston where he developed the navigation and multi-agent interaction controllers for self-driving vehicles in an urban environment. The study and the software is to be used for the SAE-Autodrive Challenge. He has represented the team and the country at Intelligent Ground Vehicle Challenge 2019, where the team won the Grandprize, the LESCOE Cup. Raghav is also an expert in creating animations and videos helping the team to realize the ideas in visual format. Ninad, Dhairya, Karthik and Saad have bagged the second prize in the 7th Delta International Smart and Green Manufacturing Contest; which is a globally conducted industrial manufacturing automation contest organized by The Ministry for Chinese Association of Automation and Delta Electronics. Anushree has made contributions to the Navigation 2 stack as an intern in the Open Robotics. Swadhin and Prithvi have previously contributed on IISc's team in the previous edition of MBZIRC in 2020 wherein the team reached the final demonstration phase. They have experience on depth estimation from monocular images. Apart from that, Swadhin, Prithvi, and Kasi have also participated in the VORC challenge in 2020 where they worked on path planning for an autonomous vessel for it to travel through the obstacles while detecting dangerous objects and the boundary line. Swadhin and Prithvi bring in their expertise and experience in the field of slam as they have also participated in tartan air slam competition. Swadhin's research work mainly focuses on collective decision making. Prithvi is experienced in working with reinforcement learning and computer vision. Anushree and Manav have prior experience working with quadruped locomotion of robots (Stoch2 and Stoch3 project) in Robert Bosch Centre for Cyber Physical Systems, IISc Bangalore. Apart from this, Manav has ample experience on working with multi-agent reinforcement learning systems. His expertise earned him the prestigious Prime Ministers Research Fellowship (PMRF) in Indian academics. His thesis work involves working on the problem of persistent monitoring of a bounded region using multiple learning based approaches. Kasi has been actively involved in projects related to computer vision - he had earlier worked on a social distancing API project during the COVID 2020 which made up to the national news. We also have other members in the group listed above who are well versed with handling of hardware's and piloting UAVs and possessing the desired skills set required during for the challenge.

1.4.1 Explanation and link to the video

The video contains the brief doodle animation of our idea presented in this white-paper along with some simulation results obtained as the proof of concept to achieve the goals of this challenge.

Link to the explanation video.

2 Introduction

Day-by-day illegal actions are growing off the coast. Fighting issues like smuggling occurring through the coast and other water bodies is important for the internal harmony, security and well-being of a region. To tackle such problems, responsible and farsighted leaders have come up with ways to trigger the brains of the young minds. Some of them are conducting this grand challenge called MBZIRC 2023 to utilize latest swarm technologies and the autonomous robots to

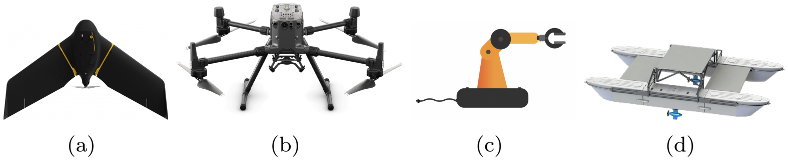


Figure 1: Schematics of the robots, (a) Delta wings, (b) Drones, (c) Robotic arm, (d) USV

automate coastal inspection and intervention in a secure and safe way. The task in this competition is broken into two phases:

- Inspection, and
- Intervention

In the inspection phase, a swarm of UAVs needs to monitor a large area of a water body like a coastal region near sea ports to identify moving suspected vessels from the other vessels. Once a vessel is detected, it has to be inspected to match it with the specifications of a target suspicious vessel. Information related to the vessel will be transmitted to the swarm through the USV along with a boolean question for the collective decision making in order to confirm if the identified vessel is the target vessel or not.

Once the target is confirmed through an operator, the USV+UAV swarm has to intervene for further scanning of the vessel with a live video stream showing recognized payloads on the vessel and boolean questions being transmitted asking the operator to confirm the recognized suspected payloads that needs to be picked from the target vessel. If the recognition of the target vessel goes wrong with no target payload on it, the entire swarm acquires a time penalty with increasing the mission completion time. Otherwise, the swarm of drones either lift the payloads individually if the payload is small or they lift the payload collectively and bring it closer to the manipulator arm on the USV. Then the arm on picks up the payload and stack it up on the USV. Once all the payloads are grabbed, the swarm proceeds towards the next target vessel and the process continues until either the mission timeouts, or if the system fails, or if the system falls back to the fail-safe mode or if all the target vessels are determined in time.

Based on the information provided during the series of live webinars we make the following assumptions for the white paper phase of the competition.

- There will be at maximum two target vessels among all the vessels.
- On each target vessels, there will be two payloads at maximum, with one being a smaller (≤ 1 Kg) payload and another larger (≤ 10 Kg).
- The target vessels will be randomly moving around in the search area with a velocity (≤ 4 Knots/sec).
- Among the information that will be given to us for recognition and localization, we expect that the following information will be available to the robots at the start of the mission:
  1. The boundary of the search area will be provided before the start of the mission. Since it will be GNSS-denied environment, active Geo-fencing won't be useful rather a passive Geo-fencing using high frequency RFID or coloured totems should be provided along with their location coordinates.
  2. The coordinates of the starting Gate and end Gate will be available.
  3. The images and other form of information related to the target vessels and the corresponding payloads will be given.
- The maximum size of the area that needs to be monitored is 10 sqKm.
- In general scenario, we assume that the communication link between the GCS and the USV remains intact at all times with no kind of hindrance on line-of-sight with the GCS.

3 Technical approach

In order to achieve the goals of this challenge, we plan to deploy the set of robotic systems shown in Fig.1. The detailed configuration of each kind of UAVs are listed in 3.1.1 and the components of USV and the robotic arm is listed in 3.1.4.

3.1 Technical approach to solving the inspection and intervention tasks in GNSS-denied environment

3.1.1 Description of UAV capabilities and sensors

We propose to use two different class of UAVs as listed below, one for the search task and one for the payload unloading task from the target vessels. We do this because both the task have different physical requirements which cannot be achieved in a single physical form of the device due to the complementing characteristics of the two task.

1. Delta wing UAVs (FW-UAV)
2. Drones

We plan to use the fixed-wings for the search task and drones for lifting the payloads and bring it closer to the manipulator arm. We list here the basic physical requirements for each task.

Requirements for Delta fixed wings:

1. 4 FW-UAVs, 2 for area coverage and 2 for examination since we assumed less than or equal to two vessels at maximum.
2. Flight time >1 hour
3. Cruise speed: 20m/s
4. Pix cube (75g, 15W)
5. Jetson Nano (100g, 10W)
6. Provided Access point on an extra FW for no-LOS communication with GCS (1Kg, 25W)
7. Cameras and Lidar ($\leq$ 1Kg, 60W, DJI Zenmuse L1)
8. TP-link Wifi adapters
9. Payload capacity 1.2 Kg

Requirements for Drones:

1. 16 Drones
2. Robotic arms ($\leq$ 500g)
3. Flight time >25 minutes
4. Pix cube (75g, 15W)
5. Jetson Nano (100g, 10W)
6. Cameras (e-CAM130A CUXVR) and Lidar (Velodyne VLP-16) ($\leq$ 1Kg, 60W)
7. TP-link Wifi adapters.
8. NRF 24L01 communication modules.
9. Payload capacity 2.7 Kg

3.1.2 Description of autonomous multi-UAV interaction and coordination methods for search, identification and localization

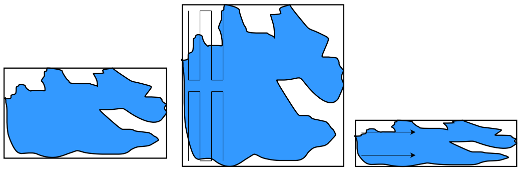
We propose to implement the lawn mower strategy in order to cover the entire search area using the delta-wings. The set of four delta-wings is divided into two sub classes called as the leader wings and the follower wings. The two leader delta-wings mainly trace the pre-generated way-points using the boundary coordinates at an altitude of about 400m to 500m. The other two follower delta-wings mainly trace the mid points of the lawn mower iterations at lower altitude of around 10m to 15m.

With the following arguments, we can guarantee that the target vessel cannot sneak into the searched area without passing through the searching FW-UAVs with target vessels moving at a velocity of $\leq$ 4 Knots/sec.

First of all, given the boundary of the competition area, we compute the convex hull and approximate the area to the smallest rectangle containing the entire region as shown in Fig.2(a). On this rectangle, we generate the way-points such that the entire region is covered using maximum of two FW-UAVs. The two worst case area that needs to be searched through such a method are:

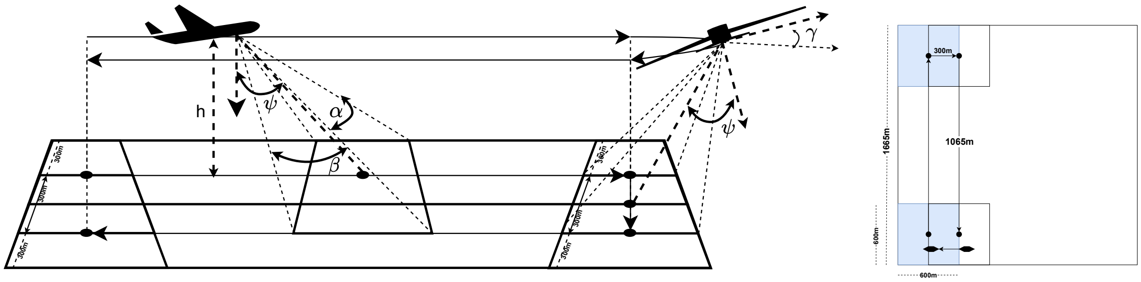
1. A square area of side 3.33Km.
2. A Rectangle area of size $1Km \times 10Km$.

In these two scenarios, the lawn-mower is reconfigured by modifying the lengths of the iterations as shown in Fig.2(b) & (c). In order to guarantee the no sneak condition for the target vessels to the previously searched area, we compute the time that each FW-UAV takes to finish one complete iteration of the lawn-mower at any time t as shown in Fig.3(b). Without loss of generality, we can consider the case that is shown in Fig.3(b). The squares depict the area under the vision at an instant. In worst case I, the FW-UAVs will travel 1065m to and 1065m fro to complete one iteration and while changing the row, the FW-UAV covers 300m extra. Therefore, a total of 2430m. As the cruise speed of the FW-UAVs are 20m/s, this distance can be covered in approx 125sec. Given the maximum speed of target vessels, the target vessel can cover not more than 250m in this time period. Since, the iterations of lawn-mower has an overlap area of 300m, the target vessel will be caught before it crosses the vision range of the searching FW-UAVs.



(a) (b) (c)

Figure 2: Search area, (a) Convex hull approximation, (b) Worst case I, (c) Worst case II



(a) (b)

Figure 3: Area coverage and no intrusion condition, (a) FOV and camera angle control, (b) Guaranteeing no intrusion

In this method, in worst case I, the entire region will be searched once completely in approximately 12 minutes. Whereas in worst case II, it will be covered in approximately 8.5 minutes. So, we can guarantee that in 12 minutes, we should be able to search all the vessels and confirm the target vessels with the help of inspecting follower FW-UAVs.

The camera angle while moving along the straight line is calculated using the Eq.1[1] along with countering the roll angle of the delta-wing on the camera.

$$\psi = 90^o - \arcsin\left((2h\tan\alpha/2)/600\right) + \beta/2 \tag{1}$$

However, while turning during changing the row of the lawn mower, we rotate the camera using Eq.2 along the z-axis of the delta-wing body frame.

$$180^o - vt/r \tag{2}$$

During turning the angle ψ is controlled using the Eq.3 along with countering the pitch angle of the delta-wing on the camera.

$$\psi = 90^o - \arcsin\left((2h\tan\alpha/2)/600\right) + \beta/2 + \arctan\left(r\sin(vt/r)/h\right) \tag{3}$$

After the searching FW-UAVs spot any vessel using the recognition method described in 3.1.6, it is added to a queue maintaining the relative pose, spotting FW-UAVs Id, time stamp of the spotted vessels. This queue is then communicated to the follower wing. The followers then start inspecting each vessel by searching in a circular area whose radius is calculated as described in Eq.4.

$$(t_{\text{spotted}} - t_{\text{queue element selected}} + \frac{|Pose_{\text{follower}} - Pose_{\text{vessel}}|}{20m/s}) * 2m/s \tag{4}$$

After the vessel is inspected, based on whether the vessel matches with target vessel or not, it is streamed back to the operator to confirm if the target vessel is recognized correctly. If it has recognized the target vessel, then the inspecting follower switches to the object tracking mode described in 3.1.6. The other follower keeps inspecting the vessels in queue until the other target vessel is found.

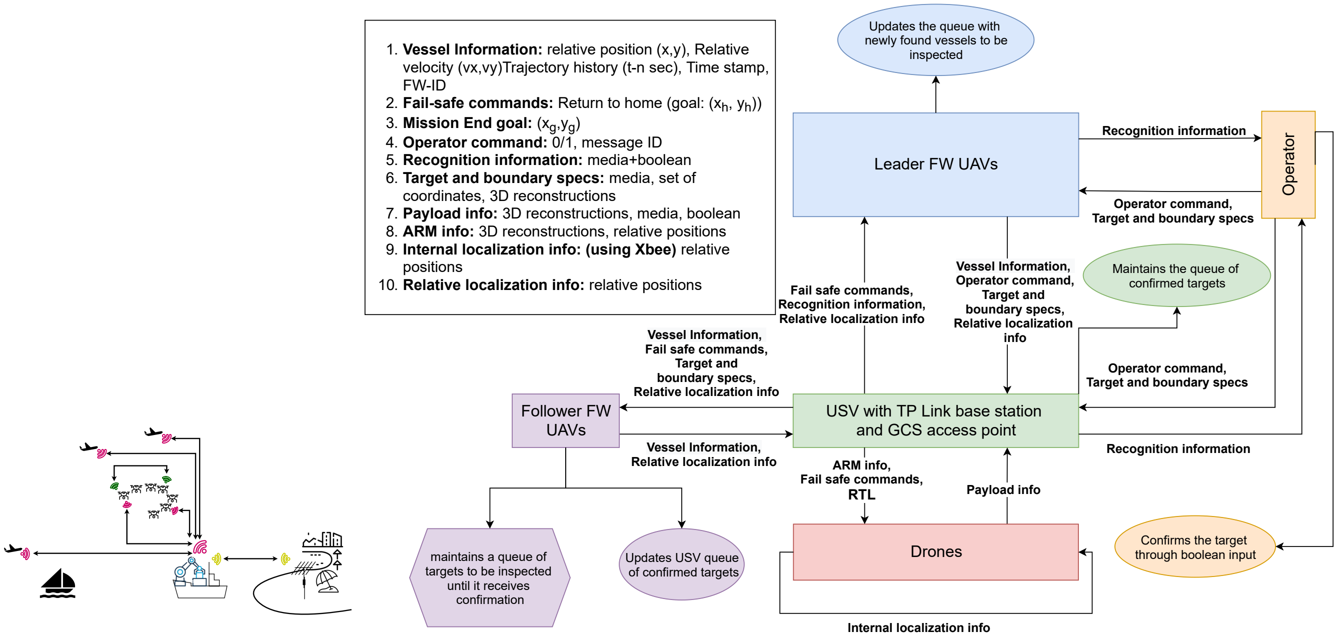

1. Vessel Information: relative position (x,y), Relative velocity (vx,vy)Trajectory history (t-n sec), Time stamp, FW-ID
2. Fail-safe commands: Return to home (goal: (x_h, y_h))
3. Mission End goal: (x_g,y_g)
4. Operator command: 0/1, message ID
5. Recognition information: media+boolean
6. Target and boundary specs: media, set of coordinates, 3D reconstructions
7. Payload info: 3D reconstructions, media, boolean
8. ARM info: 3D reconstructions, relative positions
9. Internal localization info: (using Xbee) relative positions
10. Relative localization info: relative positions
Updates the queue with newly found vessels to be inspected
Leader FW UAVs
Recognition information
Operator
Operator command, Target and boundary specs
Vessel Information, Operator command, Target and boundary specs, Relative localization info
Fail safe commands, Recognition information, Relative localization info
Maintains the queue of confirmed targets
Vessel Information, Fail safe commands, Target and boundary specs, Relative localization info
Operator command, Target and boundary specs
Follower FW UAVs
USV with TP Link base station and GCS access point
Vessel Information, Relative localization info
Recognition information
ARM info, Fail safe commands, RTL
Payload info
maintains a queue of targets to be inspected until it receives confirmation
Updates USV queue of confirmed targets
Drones
Confirms the target through boolean input
Internal localization info

Figure 4: Communication network [2]

3.1.3 Description of communication and video streaming methodology between UAV+USV system

For communication, we assume that we will be provided with an access point for the USV-UAV swarm to communicate with the GCS. We propose to mount the access point on USV with the condition that the line of sight never breaks off. For the USV-UAV swarm communication, we propose to use the TP link WBS210 on the USV with tp-link wifi adapters on all the UAVs and along with that, we will be using NRFL01 modules on all the drones for the inter drones communication for highly accurate localization due to low-latency communication.

3.1.4 Approach to mounting and controlling the Robotic arm on USV for the marine environment

The USV will host the following set of devices for computation, communication, sensing and controlling the ARM:

- Pix cube
- AGX Xavier
- Lidar
- e-CAM130A_CUXVR Cameras
- TP-Link WBS210 Base station

Given a payload of 10kg, the manipulator selected for this task is the UR10e manipulator with a payload lifting capacity of 12.5 kg and a reach of 1.3 m (51.2 in). Owing to the high-payload to be grasped, there is a need of grasping large objects and achieving high strength grasps. Furthermore, there is uncertainty regarding the shape and material of the object to be grasped.

We propose the use of a combination of fluidic elastomer actuators and gecko-inspired adhesives to both enhance existing soft gripper properties and generate new capabilities. The actuator is made proprioceptive with the use of a bend sensor for proprioception. Elastomer actuators provide an active mechanism for controlling the surface area in contact. Positive pressure bends the actuators towards an object. Applying a vacuum bends the actuator away and disengages the gripper. Increasing actuator pressure also stiffens the actuator, useful for applying torques.

The gripper can operate at higher pressure to maximize the contribution of both adhesion-controlled friction and bending stiffness from the elastomer actuators. The elastomer actuators create force closure and the gecko-inspired adhesives prevent shear sliding, significantly enhancing pull out strength.

Properties of Actuator Material				
Silicone name	**Demold Time**	**Elongation at Break**	**Tear Strength**	**Mass density** (gcc^-1)
DragonSkin 20	4 hours	620%	120 pli	1.31

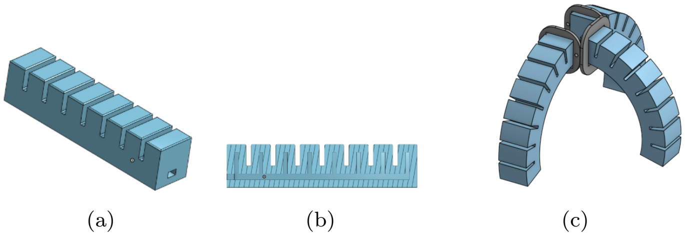


Figure 5: Gripper Design, (a) A single actuator, (b) Channels for fluid, (c) Gripper

The proposed gripper design will be a multi-actuator soft robotic system consisting of three adhesive actuators, and will be able to lift 11.3kg payload at around 40 kPa of pressure [3]

Mounting of Manipulator on U.S.V.

The arm is mounted on the usv using a mounting plate fitted with an IMU, at a height greater than 500mm. An IMU sensor is fitted on the mounting plate to detect turbulence. A custom-designed variable height stand is mounted rigidly onto the USV.

Weight of Manipulator System for Intervention Task	
Component	Weight (kg)
UR10e arm	33
Soft robotic Gripper	1.2
Mounting System	5.6

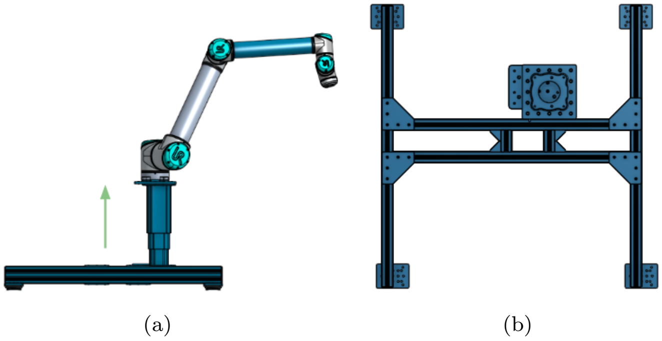


Figure 6: Mount, (a) Right view of the mounting system, (b) Top view of the mounting system

3.1.5 Description of physical interaction and motion coordination between UAVs, USV and the Robot arm

Impedance Control

The pick-and-place task is a contact task, as it produces an interaction force with the external environment. In the contact-type tasks of the robot, it is necessary to introduce force control in order to adjust the output force of the end-effector in real time and ensure the safety of the target itself. Impedance control allows the manipulator to obtain better compliance with the outside world by adjusting the impedance parameters of the target environment.

We propose a hybrid d position/force control with a virtual impedance model of robot manipulators. It distinguishes the position control and force control of the manipulator according to the axis based on the existing impedance control, and converts the change of the desired force at the top of the manipulator into a position compensation in the impedance control system based on the kinematics and dynamics of the robot, and superimpose it on the target position.

In order to achieve the impedance control of the manipulator, we use a virtual dynamic model composed of mass, spring and damper between the manipulator and the target. This impedance model based on virtual dynamics embodies the balance between the change in the position of the end effector and the change in the external environmental force it

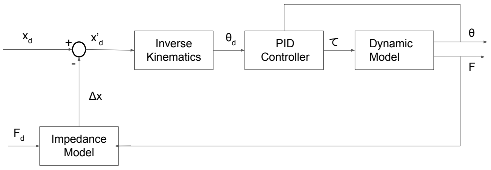
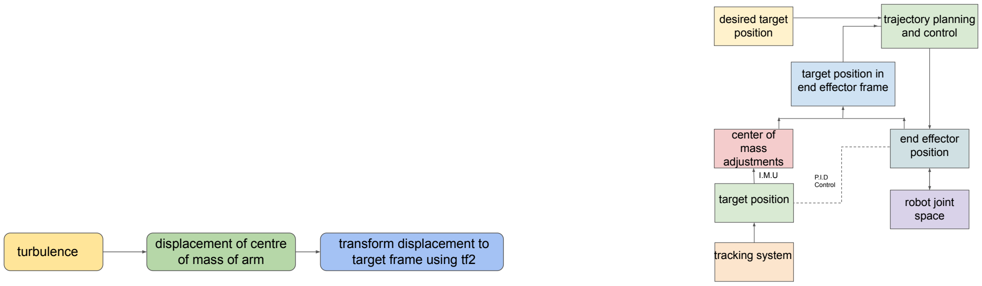


Figure 7: Block diagram of force-position mixed impedance control


turbulence
displacement of centre of mass of arm
transform displacement to target frame using tf2
desired target position
trajectory planning and control
target position in end effector frame
center of mass adjustments
end effector position
I.M.U
P.I.D Control
target position
robot joint space
tracking system

(a)

(b)

Figure 8: (a) Centre-of-Mass Tracking (b) Overview of Manipulator Control System

receives. Taking the change in position of the end-effector in the X-axis, the mathematical expression of the impedance model based on virtual dynamics is as follows:

$$\Delta F = M_i\, \Delta\ddot{x}\; + D_i\, \dot{x}\; + K_i\, x \tag{5}$$

$$\Delta F = F - F_d \tag{6}$$

$$\Delta x = x - x_d \tag{7}$$

ΔF is the change between the external force and the target force. X, $\dot{x}$, $\ddot{x}$ respectively are the difference between the position of the top of manipulator and the target position, and its velocity and acceleration.

External force is obtained using a force sensor. The target values for position are obtained from a camera mounted on the base of the gripper, and the target values of force are obtained from manipulator dynamics, given the payload weight.

Using Eq. 5 ΔF can be converted to Δx by equations 8 and 9

$$\Delta\ddot{x} = (\Delta F\; - D_i\, \dot{x}\; - K_i\, x)/M_i \tag{8}$$

$$\Delta x = \int (\int \Delta\ddot{x} dt)dt \tag{9}$$

The Δx obtained according to equation 9 is superimposed on the target position x_d of the end-effector as a compensation to the target position, thereby achieving the target force F_d. In a similar manner, the impedance model-based dynamic control can be obtained in the Y-axis and Z-axis.

Compensation for Turbulence

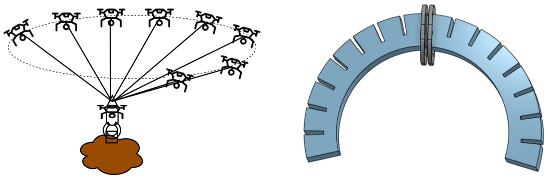
$$^{B}_{G}\boldsymbol{\epsilon}\; = \frac{^{B}_{G}\mathrm{T}\, \sum_{i=1}^{6} m_i\, ^{B}_{Ri}\mathrm{T}\, ^{R_i}\boldsymbol{\epsilon}_i}{\sum_{i=1}^{6} m_i} \tag{10}$$

where $^{B}_{G}\boldsymbol{\epsilon}$ is the arm's centre of mass in the global frame, $^{B}_{G}\mathrm{T}$ is the homogeneous transformation from the body frame (B) to the global frame (G), $^{B}_{Ri}\mathrm{T}$ is the homogeneous transformation from the i-th manipulator link frame (R_i) to the body frame, $^{R_i}\boldsymbol{\epsilon}_i$ is the centre of mass of the i-th manipulator link expressed in the i-th manipulator link frame (R_i), and mi is the mass of the i-th manipulator link.

The movement of this center of mass of the arm is calibrated upon mounting on the U.S.V. using the I.M.U. to obtain a stable base position. In high turbulence, the I.M.U. data gives us the moment of the base of the arm. That movement is transformed to the center of mass frame of the arm. The displacement of the center of mass frame due to turbulence is compensated by a displacing the target position in a similar manner. An overview of the centre-of-mass tracking algorithm is shown in 8(a)

For collective formation, we propose to use the multiple dog single sheep model from the non-cooperative shepherding swarm model [4].

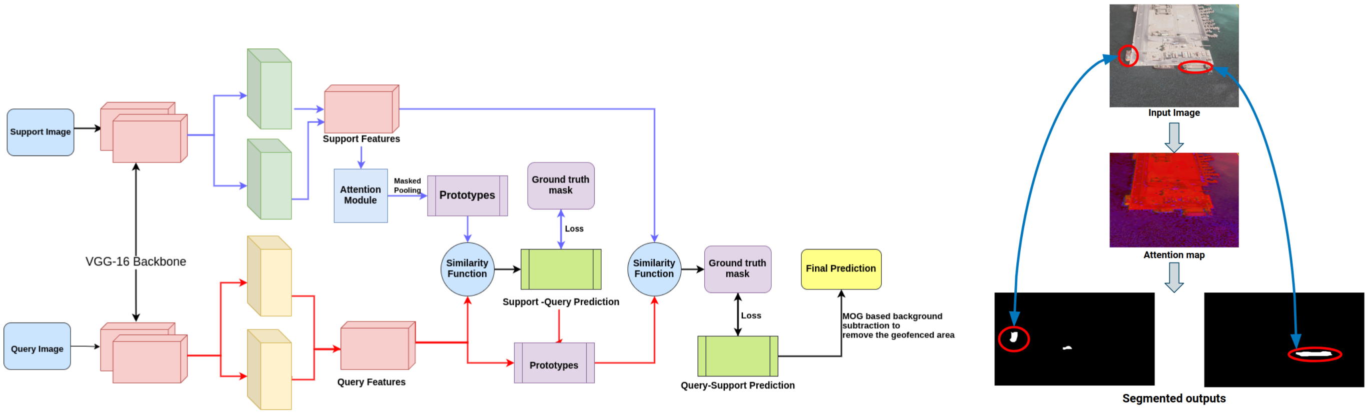
The path planning and the transport while avoiding obstacles will be done utilizing the method explained in [5].



(a) Overview (b) Gripper

Figure 9: Collective Lifting

3.1.6 Description of Vision based algorithm for target search and tracking



(a) Architecture (b) Output

Figure 10: Few Shot Semantic Segmentation Architecture

In a GNSS denied environment, active perception is of utmost importance to better understand the surroundings. The vision pipeline allows the UAVs to detect the target, track and monitor them and also detect the payloads that needs to be picked from the target. As mentioned in the information session, the intelligence report that will be provided is assumed to contain badly augmented images of the vessel (blurred out images or images taken from a good distance or images containing part of the vessel).

Different approaches can be developed for the inspection task (i.e. the target and payload inspection). Deep Learning based approaches[6], inspection based on the shape of the vessel [7] or inspection based on the colour of the vessel are some of the easier but more effective ways to solve the statement. But, the biggest problem with the above mentioned approaches so far is that, since we have only a handful of images, the use of any deep learning techniques to train on these images would be very hard. Similarly, since the images are expected to be badly augmented, it becomes difficult to generalize based on the shape or the colour. Because the data being dealt with is intrinsically rare, it becomes very hard to use a classical method to inspect and track the vessels.

Thus, to solve this, our proposed approach develops on a few-shot semantic segmentation. It involves splitting the dataset into two, support set and test set. The model makes predictions on the query set based on the training done on the support set. Unlike supervised semantic segmentation approaches [8], few-shot semantic segmentation can generalize better for unseen images (water bodies) with the help of few manually annotated images. Leveraging an extended version of the prototypical network[9], in the few-shot learning architecture, we have exploited the use of prototype alignment regularization between the support and query.

Dataset preperation and pre-processing

In order to test our network, images of similar looking vessels were scraped off the internet. The dataset contained images from different different regions which made sure that we were looking at different kinds of ocean environment. The train dataset contained over 565 images. These images were then augmented to match the description mentioned in the intelligence report. This was achieved by randomly augmenting images by blurring part of the image, removing a part of it and even using images where the vessels are quite far(but visible) from the camera location. This was then converted to 2,448 x 2,448 resolution and then tiled to create sub images of 612 x 612 pixels. This tiling results in sub-images that form the training set, D_{train}.

On the other hand, the test dataset was also scraped and augmented to match the real life simulation situation. The datasets contains four classes: Vessels, Land Cover, Water Bodies and Random Objects in the ocean. The vessels are further classified into Target Vessels and Non-Target Vessels. The input images in train and test set were resized to 256 x 256 pixels and augmented randomly. The model was trained with SGD optimizer with a learning rate of $1e^{-1}$ and momentum of 0.856 for 5,000 iterations. According to the standard practice for few-shot semantic segmentation, the number of query images was fixed as one.

Overview of the few-shot segmentation architecture In a semantic segmentation task, the support set contains K images from C classes thereby forming C-way K-shot segmentation architecture. Added to this is a collection of support pictures(which are annotated) and query pictures in each iteration. The annotated images form the C_{seen} class and the frames from the live video forms the C_{unseen} class. The performance of the model is evaluated on the query image from the same class C. For instance, a support set of five images containing pixels belonging to two classes will be referred to as 2-way 5-shot semantic segmentation.

Both the training and testing consists of multiple episodes. Each episode is composed of a set of support images S and a set of query images Q. Each episode forms a C-way K-shot segmentation task. The semantic segmentation of the query image is performed using the feature retrieved from the limited given images during the competition. The model is trained on the scraped images and tested on the live video output stream as mentioned in the data pre-processing subsection Initially, every image is processed to weed out the unknown classes (e.g. obstructing objects). The unknown classes are labelled under the Random Objects class. After that, the pictures are selected at random and averaged across three runs of 200 epochs each. The input to the network, like in a few-shot semantic segmentation technique, is a series of support pictures with the labels to conduct segmentation on the same class's query image. For each episode, the support and query images are embedded into deep features with the help of the dense backbone. And then using pooling, the prototypes are obtained and segmentation over the live video frames is performed by labelling every pixel in the frame to a certain class by connecting it to the nearest prototype. To improve the consistency of embedding prototypes, a PAR (Prototype Alignment Regularization) which was described in [10] is used.

A backbone architecture (VGG-16) extracts features from the support set as well as the query pictures. The first six layers extracts the features and to learn the prototypes of foreground and background classes, a masked average pooling is performed to the features of support set images: To generate and label a class, the distance between the query image prototypes and support image prototypes is calculated using the cosine distance

The label of the closest prototype is assigned via non-parametric metric learning. To compute the segmentation loss, we utilise the probability map.

This is then fed into the texture extraction model where the attention maps are created. The attention module utilizes the inter-spatial relationship of the features and generates a spatial attention map. Subsequently, prototype alignment regularization is applied:

In order to use attention based labelling, there needs to be something that is recognizable in both tactile and optical ways which represents variations at a much lower level and it assesses factors such as the bristliness, smoothness and evenness of the image. One way of doing it would be using shape based attention models, which is attention based on the shape of the vessel, but since we do not have any prior information regarding this, it would be better to opt for a much more robust way to solve this. Consistency and character based attention module is being applied in many places including biomedical, automation, remote sensing and also for image analysis . Consistency extraction, on a higher level, is based on the concept of establishing borders and areas where consistency properties differ. For e.g. the color consistency of an ocean is completely different from that of the land similarly it will be different from that of the obstacles in the ocean and so on. The most efficient analysis may be classified into three categories on a wide scale: statistical analysis, graph based method, and model-based approach. The geographical distribution is used as the descriptors in a model based method . Despite the fact that certain approaches are less computationally difficult, statistical methods do not provide a reliable measure of coarseness and homogeneity. The Graph Based techniques depends on graphs derived from the input picture to extract the textural features. Model based approach aims at extracting textures based on mathematical models. Our work focuses on using autoregressive model for extraction. At a lower level, it works on expanding in all areas in a picture at the same time, starting with internal regions that are already defined and progressing until smooth borders are established between all neighbouring parts.

Foreground-Background subtraction After the images are being segmented, there is a good chance of it being filled with unnecessary noise. This can be computationally redundant. Therefore, in order to filter this out, we subtract the foreground of the image from the current background. The foreground chosen is the segmented target vessel and the bakground is the live frame. Automatic extraction was proposed by [11] in 2010 extracts foreground captured from multiple viewpoints.

3.1.7 Object Tracking

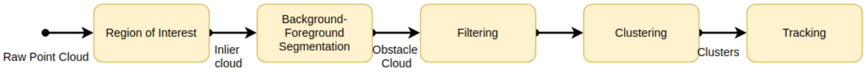


Figure 11: A high level overview of the tracking algorithm

The pointcloud output from the lidar has to be pre-processed to extract the objects of interest. The vision node which makes use of the DETR algorithm detects all the objects and passes it as a message via the ROS Bridge to the lidar node. Every frame has to be segmented into ground plane and object essentially drawing out the foreground and the background. The ego vehicle is segmented out, since the dimensions of the vehicle is well known and the point clouds that lie within those regions are nullified. The point clouds are filtered [**b13**] before clustering[**b14**], this involves removal of a set points or noise. Segmentation [**b12**] based approach was employed to separate the background from the foreground and remove the noise. We employ instance segmentation to categorize every point in the point cloud data. The point clouds belonging to the obstacles are clustered to identify sub goups in the point clouds using the k-means clustering algorithm. The main idea being iteratively partition the dataset into k defined clusters. It assigns datapoints to the clusters such that the squared distance between the centroid and the datapoints is minimum. It follows the expectation-minimization problem where the objective function is:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} ||x^i - \mu_k||^2 \tag{11}$$

where μ_k is the centroid of x_i's cluster.

$w_{ik} = 1$ if x_i belongs to cluster k; else $w_{ik} = 0$

To minimize J, differentiating wrt w:

$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} ||x^i - \mu_k||^2 \tag{12}$$

where $w_{ik} = 1$ if $k = argmin_j ||x^i - \mu_j||^2$ and 0 otherwise.

For M-step:

$$\frac{\partial J}{\partial w_{ik}} = 2 \sum_{i=1}^{m} w_{ik}(x^i - \mu_k) = 0 \tag{13}$$

$$\mu_k = \frac{\sum_{i=1}^{m} w_{ik} x^i}{\sum_{i=1}^{m} w_{ik}} \tag{14}$$

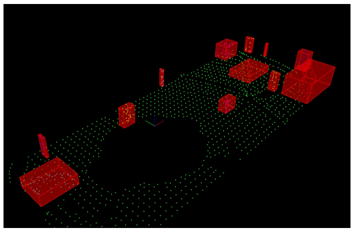


Figure 12: This figure depicts the object tracking output from a scene in a highly congested environment

3.1.8 Description and overview of localisation and sensor fusion

In order to localise in a GNSS denied environment, our proposed architecture makes use of both visual odometry and sensor fusion to better estimate the relative position of the UAV wrt the starting point and the boundary of the search area.

In order to localise the XAVs, we developed a novel object aware SLAM with the main aim of localising instead of mapping. In an environment such as an ocean, the number of position descriptors are very limited. So in such situations, it becomes increasingly difficult to localise using pre-existing visual odometry techniques. To mitigate this, we developed a transformers based object aware SLAM with the main intention of working in a loosely cluttered environment.

The backbone of the object aware SLAM is a DETR [12] based transformer network. In an image which has multiple objects, object detection gets harder especially in scenarios where there are multiple classes and instances of the same class in the same frame.

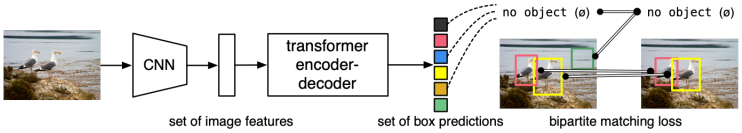


Figure 13: High Level Architecture of the DETR[12] model

The unlabelled image is first run through a Convolutional Neural Network encoder which gives a set of image features.

The set of image features is then put through a transformer encoder decoder (DETR) which gives a set of predictions which is a tuple consisting of the class and the position of the bounding box. This also includes classes that are a null set.

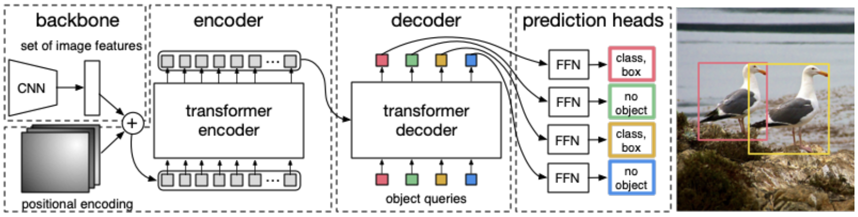


Figure 14: DETR[12] implementation architecture

The backbone is a CNN with positional encoding after which they are unrolled and flattened before inputting it into the transformer encoder. This sequence is transformed into an equally long sequence which is side inputted into the transformer decoder which conditions information. The input sequence is the object queries which are N random vectors and the output is fed into a classifier which gives the prediction.

A loss function L (bipartite matching loss [13]), is used which reads the predicted output of the model and the ground truth and computes how well both of them agree with each other. This also covers cases where the classes agree to each other while the position of the bounding boxes do not and vice versa. The bipartite matching finds an one to one assignment between the predictions and the ground truth such that the total loss is minimized using the Hungarian matching. The loss function for the class labels is a cross-entropy loss and the loss function for the bounding boxes is a L1 loss which compares two bounding boxes and the iou loss which computes what fraction of bounding box overlaps with the ground truth. This L is considered as the training loss.

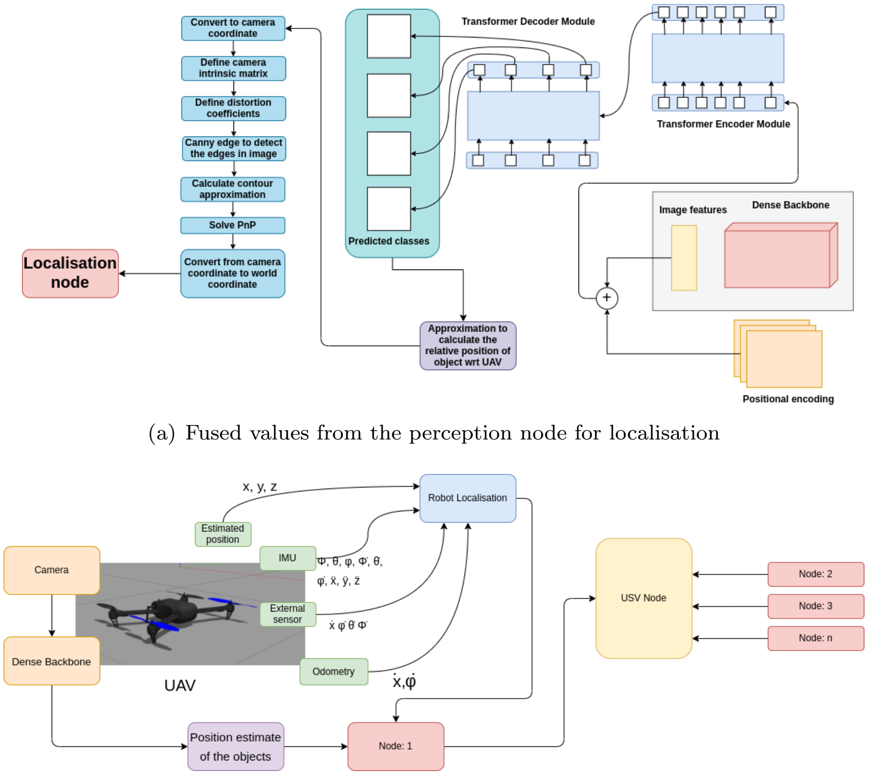
Once the position descriptors are detected, we use either lidar based data or camera based coordinate conversion to calculate the distance. This value is then matched with the sensor fusion based localisation to reduce the errors.

Sensor Fusion

Sensors used

- Dead Reckoning- to estimate the current position of the UAV.
- Camera
- IMU(Inertial Measurement Unit)
- Lidar

Using dead reckoning and the fused values from various sensors, we can have a better estimate of the relative position of the UAV. We used the above mentioned suite of sensors as the errors are uncorrelated. If one sensor fails, it will not


Convert to camera coordinate
Define camera intrinsic matrix
Define distortion coefficients
Canny edge to detect the edges in image
Calculate contour approximation
Solve PnP
Convert from camera coordinate to world coordinate
Localisation node
Predicted classes
Transformer Decoder Module
Transformer Encoder Module
Image features
Dense Backbone
+
Approximation to calculate the relative position of object wrt UAV
Positional encoding

(a) Fused values from the perception node for localisation

x, y, z
Estimated position
Robot Localisation
IMU
Φ, θ, φ, Φ̇, θ̇,
φ̇, ẍ, ÿ, z̈
Camera
External sensor
ẋ φ̇ θ̇ Φ̇
Dense Backbone
Odometry
ẋ,φ̇
UAV
USV Node
Node: 2
Node: 3
Node: n
Position estimate of the objects
Node: 1

(b) Localisation architecture

Figure 15: High level localisation architecture

affect the others. Since all of the sensors use different measurements they are unlikely to fail for the same reason. IMU acts as a high rate smoother.The dead reckoning setup acts as a supplementary smoother for the IMU. Lidar can be used for accurate local state estimation. In a pre-calculated map (from the SLAM pipeline), we can use current position to compare the values we are getting from a lidar and correct the estimate accordingly. Thereby complimenting each other.

Implementation pipeline The prediction model generally comprises of the dead reckoning and/or the inertial navigation to predict the new state whereas the observation model takes the values from the vision pipeline and other sensors like lidar to correct the position value. We use the IMU and dead reckoned measurements as our input to our motion model. This will give us our predicted state, which will update every time we have an IMU reading. This happens at a very high frequency since the update values are very high. The IMU output will be combined into a single input vector. The accelerometer and the gyroscope biases will be put into the state vector, estimated, and subtracted off the IMU readings to make it as unbiased as possible.

The update frequency of the vision pipeline is much lower in comparison to the IMU or the encoder. Hence it is important to calibrate the sensors accordingly. At a much lower frequency, we use the vision measurements whenever they become available. This estimate is used to correct our predicted state.

The values from the dead reckoning will be converted into forward velocity. But the IMU and the vision pipeline will have 2 separate coordinate systems, to have the two coordinate systems working as one, several transformations must be made: First, we will apply a projection on the geographic coordinates to obtain plane coordinates, like a paper map. Then we determine the complete transformation, composed of a translation and a rotation between the reference plane given by the IMU. Essentially, to get the data into the robot's world frame, we convert the vision data into UTM coordinates and use the initial UTM coordinate, EKF output and the IMU to generate a static transform (T) from the UTM grid to the robot's world frame. And then transform all the future pose estimates measurements using T. The output is fed back to the EKF. The output from the EKF is fed to the actuators.

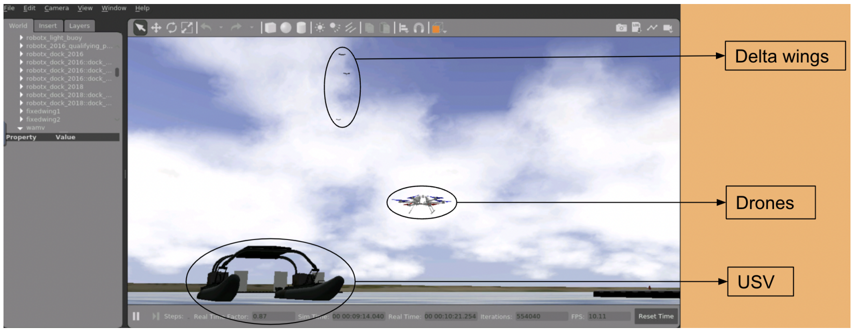
Equations 8, 9, 10, and 11 are the final set of equations that will be used to compare and fuse the values of the IMU and the dead reckoning. This will be fed into the motion model and will be used for predicting the state. Equations 25, 26, 27, 28, 29, 30 and 31 are the final set of equations that will be used for sensor fusion of the Lidar point cloud and the position estimates to calculate the corrected states by feeding it into the prediction model.

In order to improve the accuracy, we triangulate this with respect to 3 positions on the ocean environment: the starting point(where the UAVs take off), the position descriptors in the environment and finally the closest UAV to the current UAV/position of the USV wrt the UAV.

3.1.9 Description of simulation methodologies used for proof of concept

We used the Gazebo simulator with Ros melodic in an effort to try and create the simulation world for the competition. We combined few different existing packages to build the required environment. The existing packages includes:

1. RotorS [14] (For drone)
2. RosPlane [15] (For Delta-wing UAV)
3. osrf/VRX Gazebo [16] (For USV)
4. osrf/VORC Gazebo [16] (For target vessels)
5. ros-industrial/universal_robot Gazebo [17] (For manipulator)



Steps: Real Time Factor: 0.87 Sim Time: 00 00:09:14.040 Real Time: 00 00:10:21.254 Iterations: 554040 FPS: 10.11

Figure 16: Simulation environment showing UAV, FW-UAV and USV in an ocean world

We have also done some simulations in MATLAB for the collective transport. The results of the simulations are shown in the video.

3.2 Uniqueness of approach and its contribution to the advancement in the state-of-the-art in autonomous robotics

Our solution to the challenge is unique in terms of handling the overall mission. Existing maritime robotic automation research focuses more towards network coverage [18], object segmentation in marine environments [19, 20, 21], tracking moving vessels [22], search and rescue missions [23, 24]. Also, we have utilized some of the past works and showed their potential usage in real world applications. However, those are very specific research that targets to solve a very specific problem. For instance, there are few papers that solves only the area coverage problem, there are few that solves only the object tracking problem, some solve the object recognition problem, etc. Instead, in this work, we integrate the existing techniques by modifying them to automate the coastal inspection and intervention task that concerns the security of a geographic region. Therefore, the contribution of this work is unique in terms of it's contribution to the society.

Looking at the modifications done in each of the past works, brings our attention along the direction towards which those studies need to move to solve real world problems.

3.3 Safety considerations to ensure UAV returns to the base on command

In this white paper, we provide our ideas part of which are tested and argued to work without any fail. However, during the simulation phase, after receiving the simulation environment, we will verify whether the proposed algorithms work without any GNSS data. We will verify the algorithms through SITL based evaluations with various fail-safe situations and the risks described in 3.5.

3.4 Expected commercial and societal impact and application of the technical solution

The planned system has a direct impact on the society in various ways. This system is scalable, therefore, it can be used to inspect very large areas. Use of delta wings makes it even more scalable in terms of the size of area that can be monitored. Hence, this system can be useful for defence and security at water bodies and by replacing the USV with a Rover, it can also be used on large open ground like border areas. There are variety of objects that are being smuggled across the border, ranging from drugs to human trafficking. Many other illegal activities take place at large water bodies near the coast that remains unmonitored and unsafe to be fought against. This system, helps stopping such activities and also makes this process of intervention safe for the service persons.

This system can also be used for automated search and rescue missions on sea. However, this requires some pre built hubs where humans cannot be available all the time with such systems in place to cover huge areas and make the rescue process quicker. Looking at the level of maintenance required for this system, and the kind of applications it can be used for, it is mainly be bought and utilized by government agencies across the world. Therefore it has huge potential market.

3.5 Risks and mitigation approach

Risks	**Risk level**	**Mitigation**
Power loss	high	We have power monitoring program that changes systems state from operational mode to fail-safe mode
Robots leaving the competition area	very high	We will do a thorough verification of our ideas in simulation environment
Loosing line of sight while communication with GCS	very low	We can deploy an extra delta-wing always staying above the USV with an extra access point to the GCS
Weather turbulence	high	Switch to fail-safe mode above a threshold wind disturbance
Fixed wing catapult not provided	high	We can use VTOL Delta wing UAVs

In fail-safe mode, the robot searches for the nearest hard surface and lands there and waits until it gets rescued.

References

[1] Yan Li et al. "Coverage path planning for UAVs based on enhanced exact cellular decomposition method". In: *Mechatronics* 21.5 (2011). Special Issue on Development of Autonomous Unmanned Aerial Vehicles, pp. 876–885. ISSN: 0957-4158. DOI: https://doi.org/10.1016/j.mechatronics.2010.10.009. URL: https://www.sciencedirect.com/science/article/pii/S0957415810001893.

[2] Jiri Pokorny et al. "Prototype Design and Experimental Evaluation of Autonomous Collaborative Communication System for Emerging Maritime Use Cases". In: *Sensors* 21.11 (2021). ISSN: 1424-8220. DOI: 10.3390/s21113871. URL: https://www.mdpi.com/1424-8220/21/11/3871.

[3] Paul Glick et al. "A Soft Robotic Gripper With Gecko-Inspired Adhesive". In: *IEEE Robotics and Automation Letters* 3.2 (2018), pp. 903–910. DOI: 10.1109/LRA.2018.2792688.

[4] Alyssa Pierson and Mac Schwager. "Bio-inspired non-cooperative multi-robot herding". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 1843–1849. DOI: 10.1109/ICRA.2015.7139438.

[5] Alyssa Pierson and Mac Schwager. "Controlling Noncooperative Herds with Robotic Herders". In: *IEEE Transactions on Robotics* 34.2 (2018), pp. 517–525. DOI: 10.1109/TRO.2017.2776308.

[6] *Deep learning for autonomous ship-oriented small ship detection - ScienceDirect*. https://www.sciencedirect.com/science/article/abs/pii/S0925753520302095. (Accessed on 01/29/2022).

[7] *Shape based Object Recognition in Images: A Review*. https://research.ijcaonline.org/volume58/number21/pxc3883684.pdf. (Accessed on 01/29/2022).

[8] *[1606.00915] DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. https://arxiv.org/abs/1606.00915. (Accessed on 01/29/2022).

[9] *[1703.05175] Prototypical Networks for Few-shot Learning.* https://arxiv.org/abs/1703.05175. (Accessed on 01/29/2022).

[10] *Few-shot prototype alignment regularization network for document image layout segementation - ScienceDirect.* (Accessed on 01/29/2022).

[11] *(PDF) Automatic Foreground Extraction Based on Difference of Gaussian.* https://www.researchgate.net/publication/264902150_Automatic_Foreground_Extraction_Based_on_Difference_of_Gaussian. (Accessed on 01/29/2022).

[12] *[2005.12872] End-to-End Object Detection with Transformers.* https://arxiv.org/abs/2005.12872. (Accessed on 01/29/2022).

[13] *bipartite.pdf.* http://www.cs.toronto.edu/~sven/Papers/bipartite.pdf. (Accessed on 01/29/2022).

[14] Fadri Furrer et al. "RotorS—A Modular Gazebo MAV Simulator Framework". In: *Robot Operating System (ROS): The Complete Reference (Volume 1).* Ed. by Anis Koubaa. Cham: Springer International Publishing, 2016, pp. 595–625. ISBN: 978-3-319-26054-9. DOI: 10.1007/978-3-319-26054-9_23. URL: https://doi.org/10.1007/978-3-319-26054-9_23.

[15] Gary Ellingson and Tim McLain. "ROSplane: Fixed-wing Autopilot for Education and Research". In: *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on.* IEEE. 2017.

[16] Brian Bingham et al. "Toward Maritime Robotic Simulation in Gazebo". In: *Proceedings of MTS/IEEE OCEANS Conference.* Seattle, WA, Oct. 2019.

[17] Universal Robots. *universal$_r$obots.* https://github.com/ros-industrial/universal_robot. 2019.

[18] Sheikh Salman Hassan et al. *Seamless and Energy Efficient Maritime Coverage in Coordinated 6G Space-Air-Sea Non-Terrestrial Networks.* 2022. arXiv: 2201.08605 [cs.NI].

[19] Wentao Lu and Claude Sammut. *D-Flow: A Real Time Spatial Temporal Model for Target Area Segmentation.* 2021. arXiv: 2111.04525 [cs.RO].

[20] Dejan Štepec, Tomaž Martinčič, and Danijel Skočaj. "Automated System for Ship Detection from Medium Resolution Satellite Optical Imagery". In: *OCEANS 2019 MTS/IEEE SEATTLE.* 2019, pp. 1–10. DOI: 10.23919/OCEANS40490.2019.8962707.

[21] Borja Bovcon et al. "MODS–A USV-Oriented Object Detection and Obstacle Segmentation Benchmark". In: *IEEE Transactions on Intelligent Transportation Systems* (2021), pp. 1–16. ISSN: 1558-0016. DOI: 10.1109/tits.2021.3124192. URL: http://dx.doi.org/10.1109/tits.2021.3124192.

[22] Jianli Wei, Guanyu Xu, and Alper Yilmaz. "DeepTracks: Geopositioning Maritime Vehicles in Video Acquired from a Moving Platform". In: *2021 IEEE Sensors.* 2021, pp. 1–4. DOI: 10.1109/SENSORS47087.2021.9639595.

[23] Jorge Peña Queralta et al. "Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision". In: *IEEE Access* 8 (2020), pp. 191617–191643. DOI: 10.1109/ACCESS.2020.3030190.

[24] Leon Amadeus Varga et al. *SeaDronesSee: A Maritime Benchmark for Detecting Humans in Open Water.* 2021. arXiv: 2105.01922 [cs.CV].