2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

A Robotic System of Systems for Human-Robot Collaboration in Search and Rescue Operations

Teng Hooi Chan[1], James Kusuma Dewa Halim[1], Kian Wee Tan[1], Emmanuel Tang[1], Wei Jun Ang[1], Jin Yuan Tan[1], Samuel Cheong[2], Hoan-Nghia Ho[2], Benson Kuan[4], Muhammad Shalihan[3], Ran Liu[3], Gim Song Soh[2], Chau Yuen[3], U-Xuan Tan[3], Lionel Heng[4], Shaohui Foong[1]

***Abstract*— The progress in robot autonomy has opened up opportunities for various applications, notably in autonomous navigation and mapping missions with mobile platforms. This motivates us to exploit such technologies to develop a human-robot collaboration system. Such a system improves task efficiencies and ensures the safety of human counterparts in search and rescue operations and site surveillance missions. In this paper, we present a robotic system of systems as a strategy for human-robot teaming missions in unexplored and unstructured environments. The system comprises a single human operator and multiple custom-built aerial robots equipped with various sensors for localization, mapping, and object detection. It enables the human operator to set operation modes and assign tasks to the robots individually or as a group via a human-robot interaction device, allowing the human operator to focus on critical mission objectives and decision-making. In each operation mode, the robot(s) navigates the environment autonomously while avoiding obstacles for a given set of waypoints. Additionally, a formation planning policy has been developed for group navigation and relative poses between the human operator and robots are estimated using UWB ranging and odometry measurements to improve the human operator's IMU positioning accuracy. The robots are fitted with RGB-D cameras for object detection and real-time image streaming to the operator. Results from the deployment of the system in indoor settings are presented to demonstrate the feasibility of a human-robot collaboration system in an unknown environment.**

I. Introduction

Recent advancements in robot autonomy technology have led to a proliferation of opportunities for its application, rapidly expanding the potential for these systems in various fields. One such application is the human-robot collaboration system for urban search and rescue (USAR) and site surveillance operations. These systems can perform tasks such as exploring hazardous environments, mapping and surveying disaster sites, detecting survivors, and transmitting real-time information back to human operators. By leveraging the strengths of both humans and robots, they can significantly improve the efficiency and effectiveness of search and rescue (SAR) operations as well as reduce the risk to human operators.

Fig. 1: A robotic system of systems that features a rescuer (the human operator) and three aerial robots that would enable better situational awareness and prompt access to casualty (mannequin) in a USAR scenario.

The goal of this project is to develop a human-robot teaming system capable of navigating through unexplored and unstructured indoor environments in close proximity under global navigation satellite system (GNSS)-denied conditions and detect objects of interest. The system consists of a single human operator and three aerial robots or unmanned aerial vehicles (UAVs) as shown in Fig. 1. The human operator focuses on high-level mission goals and critical decision-making while the UAVs must each be able to demonstrate a high level of autonomy to navigate challenging environments. Such autonomy includes the ability to self-localize, environment mapping, and obstacle avoidance. The UAVs extend the reach of the human operator into different areas of a site to provide real-time visual information and better situational awareness. Further adding to this challenge, the UAVs are also required to navigate through narrow passages in indoor environments in close proximity. Through collaborative interaction, the subsystems complement each other and work toward the achievement of a common goal.

The scope of this project includes the development and use of both hardware and software solutions. The human operator is equipped with a custom-built Human-Robot Interaction (HRI) system that includes a pointing device enabling multiple operation modes selection and task allocation to single

This project was supported by the DSO National Laboratories, Singapore and Temasek Labs SUTD, Singapore.

[1]Aerial Innovation Research Lab, Singapore University of Technology and Design, Singapore.
[2]Robotics Innovation Lab, Singapore University of Technology and Design, Singapore.
[3]EPD Pillar, Singapore University of Technology and Design, Singapore.
[4]Robotics Autonomy Lab, Robotics Division, DSO National Laboratories, Singapore.

978-1-6654-7633-1/23/$31.00 ©2023 IEEE

or multiple robots. The aerial robots are custom-built coaxial quadcopters equipped with multi-modal sensors for self-localization, mapping, and object detection. Apart from using our developed solutions, we utilized existing ones for some of the UAVs' autonomous functions such as localization and object detection.

The experiments were conducted in an indoor search and rescue scenario with common indoor architectures that include a narrow passage, obstacles, rooms, and a casualty (mannequin). This scenario tests the feasibility of the various systems that enable indoor group navigation in close proximity, task allocation for the UAVs, and casualty detection.

II. Related Work

Driven by advancements in robot autonomy technology, various private and government organizations have been seeking solutions for human-robot teaming applications. These initiatives have sparked significant interest among the robotics community, academia, and industry alike, to develop innovative solutions to support such operations.

The European Commission-funded SHERPA [1] and ICARUS [2] projects aim to develop heterogeneous robotic systems to complement their human counterparts for SAR missions in complex and harsh open environments, such as alpine areas and shipwreck zones. The projects featured multiple UAVs and unmanned ground robots (UGVs) for land missions and additionally unmanned surface vehicles for coastal operations. These systems ensure optimal human-robot collaboration in SAR missions for disaster victims in wide, open environments.

In 2017, the US Defense Advanced Research Projects Agency (DARPA) organized the Subterranean (SubT) Challenge to spur innovations in human-robot exploration in complex underground environments. These environments include tunnel systems, cave networks, and urban underground spaces, which pose tremendous challenges for first responders. The top two contestants of the challenge, CERBERUS [3] and CSIRO Data61 [4], each provided a system of systems that comprises of UGVs, including quadrupeds, small and large size UAVs, and a remote human supervisor. Their systems displayed exceptional autonomous exploration capabilities while being monitored by the supervisor at the base station.

Our system differs from the aforementioned work in that it is developed for human-robot collaboration in close proximity in indoor settings with active, high-level task assignments from the human operator to the UAVs.

III. System Description

This section describes the subsystems including their hardware and respective functions.

A. Human-Robot Interaction System

The HRI system aims to be lightweight, simple, and intuitive for guiding and commanding the UAVs [5]. The operator can use it to command the UAVs to perform three modes of operations: "Follow Me", "Go There", and "Casualty Detection" (see Section IV). The system comprises a pointing device held by the operator and a touchscreen user interface (UI) worn on the wrist.

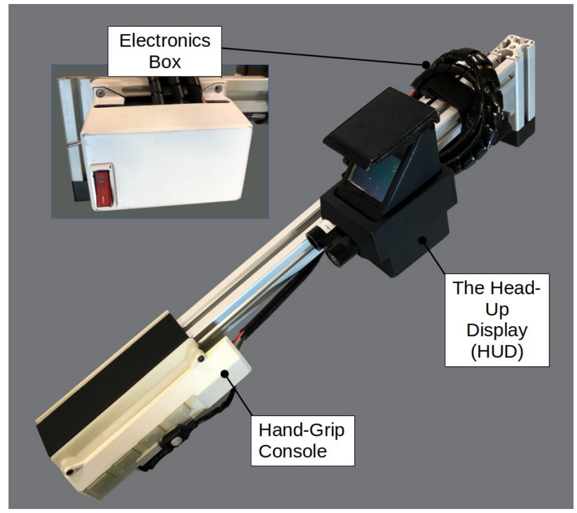
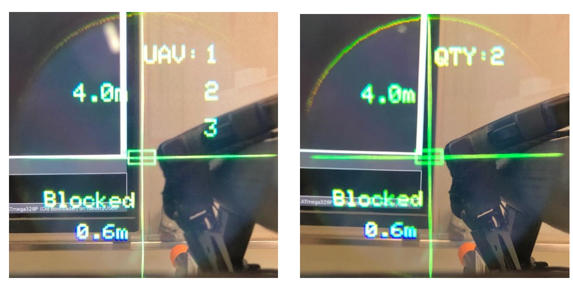


Fig. 2: The pointing device prototype.

1) The Pointing Device

The pointing device shown in Fig. 2 comprises a Head-up Display (HUD) for targeting and providing the operator with visual feedback, a hand-grip console, and an electronics box to house the device circuitry. The hand-grip console has a push-button for selecting individual UAVs to command, a rocker switch for adjusting the endpoint distance, and another push-button for sending command confirmations to the UAVs. The device has an integrated Bluetooth transmitter that relays data to a portable single-board computer.



(a) UAV ID selection mode (b) Quantity ID selection mode

Fig. 3: The reticles of the HUD display range-finding information (bottom left quadrant), user-selected endpoint distance (top left quadrant), and the feedback from the respective UAV selection mode (top right quadrant).

a) Selecting UAVs through the pointing device

The operator may select UAVs through the button console of the pointing device in two ways as shown in Fig. 3. Firstly, the operator may select the number corresponding to the UAV's numerical identification (i.e. UAV1, UAV2, and UAV3, see Fig. 3a). Secondly, the user can select the number of UAVs to command. In this second way, a specific function selects the UAV(s) closest to the operator based on proximity (see Fig. 3b).

b) Assigning waypoints for "Go There" mode

To assign waypoints for the UAVs to navigate to, the operator first selects the UAVs to command, either by ID or quantity. Then, the operator aims to use the HUD sight and adjusts the desired distance to the endpoint using the rocker switch on the hand-grip console. Finally, the operator presses the button to confirm the endpoint. The endpoint data is then relayed to the operator's portable computer, which then transmits it to the selected UAVs.

c) Commanding UAVs to "Follow Me"

The "Follow Me" mode starts with the operator selecting the UAVs. The operator then releases the button on the hand-grip console for one second. This action is interpreted by the pointing device as a switch to "Follow Me" mode and triggers the device to relay the command for the selected UAVs to follow the operator.

The "Casualty Detection" mode is handled by the UI.

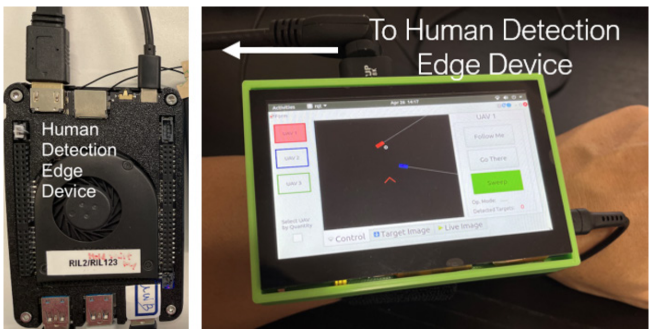


Fig. 4: The wrist-worn device setup.

2) The Wrist-worn Device

The wrist-worn touchscreen device provides the operator with a first- and third-person view of the UAVs for mission control and situation awareness. It works in conjunction with the pointing device to enable the operator to seamlessly carry out the missions and monitor their progress. The device includes a touch screen UI and a LattePanda Alpha edge computer (human detection edge device) as shown in Fig. 4.

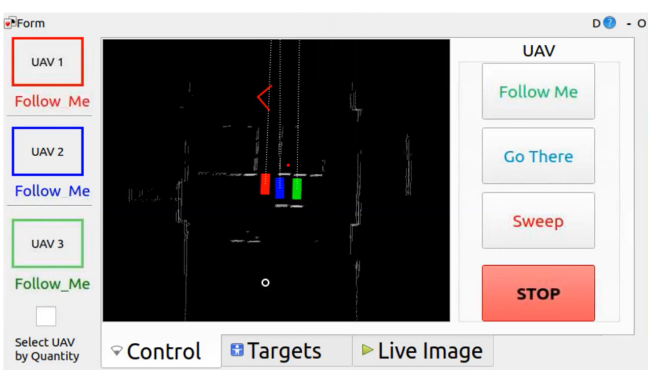


Fig. 5: GUI of the wrist-worn device.

a) The graphical user interface (GUI)

The GUI for the wrist-worn device was developed using the Robot Operating System's (ROS) rqt package [6] as shown in Fig. 5. The left button panel provides an alternative to the pointing device and allows the operator to select one or more UAVs, which are color-coded to match the selection buttons. UAV1, UAV2, and UAV3 are color-coded in red, blue, and green respectively.

The right panel of the GUI comprises button switches for different operation modes, including "Follow Me", "Go There", "Sweep", and "Stop". "Sweep" mode is equivalent to "Casualty Detection" mode. The operator selects an operation mode and the mission command along with the selected UAV IDs sent to the UAV Mode Manager to either execute the command or halt the UAVs on position.

The bottom panel comprises three tabs: "Control", "Targets", and "Live Image" which can be selected to access different functionalities. Under the "Control" tab, the display area shows the poses of the UAVs, operators, and detected targets. The combined point clouds acquired by the three UAVs' lidars in the global frame are also displayed in the top view to provide a layout of the surrounding environment to the operator. The position and orientation of the operator are shown in real-time with a red arrowhead. Detected targets are represented as small dots, with the color of the dot representing the respective UAV that detected the casualty.

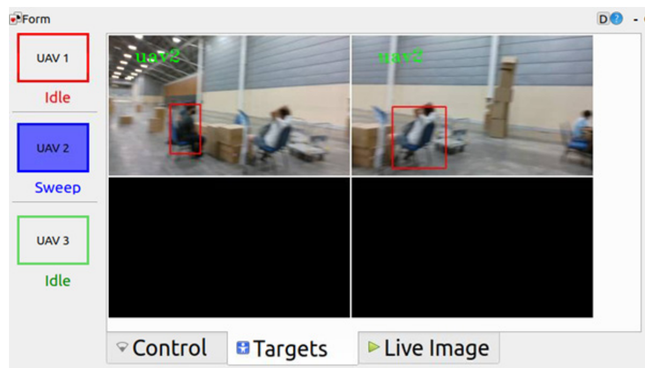


Fig. 6: "Targets" tab displaying images of targets.

Under the "Targets" tab, all detected targets in the images are enclosed with red bounding boxes, as shown in Fig. 6. The operator can select a casualty to issue a related command for a selected UAV by simply tapping on the respective image.

Under the "Live Image" tab, the live video feed from a selected UAV is displayed, allowing the operator to view the scene from the UAV's perspective. This functionality effectively improves the operator's situational awareness by providing access to the view of obstructed areas.

b) Operation mode via the GUI

The general workflow to execute an operation mode through the GUI consists of two steps: (1) selecting a specific or multiple UAVs, and (2) activating the operation mode. Both "Follow Me" and "Go There" modes are straightforward and require no additional steps after being triggered. For "Casualty Detection," the operator must first select the UAVs and then press the "Sweep" button. This sends a command to the selected UAVs to scan their surroundings, detect, and display the location of the targets in the GUI. If successful, the operator can view images of the detected targets under the "Targets" tab.

3) System Architecture

Fig. 7 shows the HRI system. It begins with input from the user via the pointing device and the touchscreen UI. Both generate a standardized custom ROS message that comprises:

- A "UAV Selection Mode" integer (i.e. '0' for selection by ID and '1' for selection by quantity).
- A "UAV ID" list of integers containing the IDs of the selected UAVs, if selection by ID is chosen.
- A "UAV Quantity" integer to indicate the number of UAVs selected, if selection by quantity is chosen.
- A "HRI Mode" integer to indicate the mode of control over the UAVs.

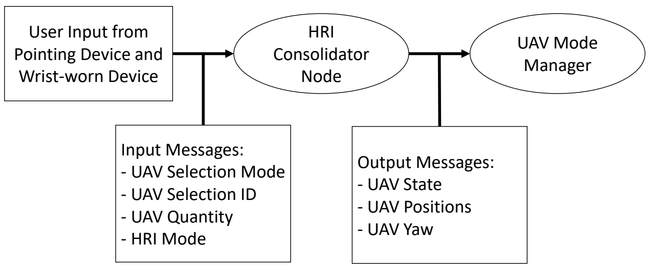


Fig. 7: The HRI system architecture.

This information is received by the HRI Consolidator Node, which consolidates the input from the pointing and wrist-worn devices respectively. The node then sends an output message to the UAV Mode Manager with the following information:

- A "UAV State" list of integers indicating the command states that UAVs are in.
- A "UAV Positions" array containing the positions of the UAVs, corresponding to its command state (e.g. If in "Go There" mode", the corresponding position in the array will be the endpoint position).

B. UAV System Hardware

The UAV platforms must have motors that can produce a substantial amount of thrust to support the weight of the various sensors and the onboard computer. They must also have adequate endurance for the mission and sufficient power for propulsion, computing, and sensing systems.

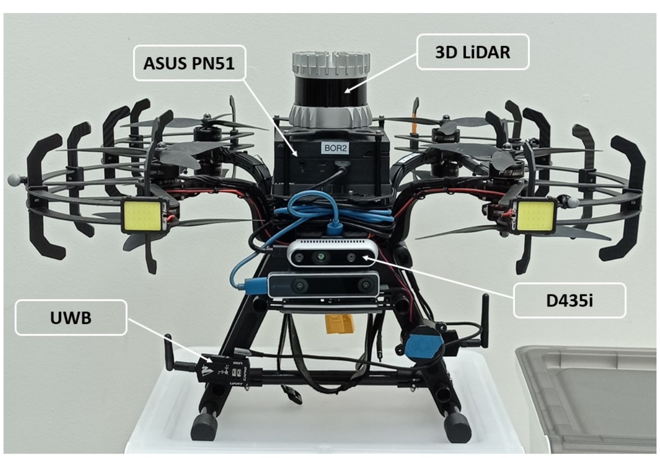


Fig. 8: UAV system platform.

We selected a 600mm by 600mm X8 coaxial configuration as shown in Fig. 8 along with smaller tri-blade 8-inch propellers. The additionally thrust from the additional motors in coaxial form enables fast dynamic changes in body velocities. The motors used were 8 T-Motor F100s, with Pixhawk 2.1 Orange Flight Control Unit (FCU) and a ASUS PN51 onboard computer. Equipped with a 279Wh battery, the platform has an endurance of 15 minutes or 10 minutes for aggressive off-board flight.

The UAV is also equipped with an Intel D435i RGB-D camera, an Ouster OS-1 32 lidar, and a Nooploop LinkTrack Ultra-Wide Band (UWB) transceiver for its sensor suite.

C. Communication Network

As part of the communication network system, two access points were connected to a main router to provide a dedicated network band for each of the UAVs to stream large amounts of data. Connections to other computers were established via Local Area Network (LAN) to the router. This setup streamlined testing and debugging by avoiding bandwidth and latency issues caused by round-robin scheduling. Force time sync based on precision time protocol (PTP) was established.

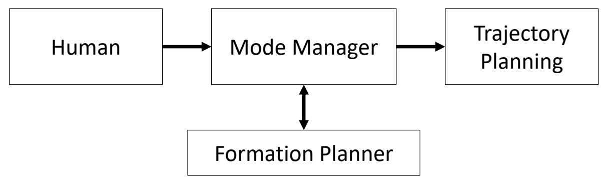


Fig. 9: The communication system architecture.

1) System Architecture

The scripts on the human operator and the UAVs communicate through TCPROS on ROS Melodic. When the human operator sends a command to the drone, this command is interpreted by the UAV's mode manager which polls the formation planner for the UAV's setpoint. The mode manager then sends this command to the trajectory planner or bypasses the formation planner when not required as shown in Fig. 9.

IV. Mode of Operations

The modes of operation were established for better human-robot collaboration. The human operator can set operation modes and assign tasks to the UAV individually or as a group through the use of an HRI device. We describe the three operation modes as follows:

A. Follow Me

Under the "Follow Me" mode, the UAV follows the human behind at an adjustable distance. For a group of UAVs, the UAVs will follow a line or a triangle formation based on the formation planning policy while following the human operator.

B. Go There

Under the "Go There" mode, the UAV(s) will navigate to a waypoint as designated by the pointing device (in the formation of multiple UAVs) while optimizing trajectories and avoiding obstacles.

C. Casualty Detection

Under the "Casualty Detection" mode, the UAV will perform a sweeping maneuver to scan for objects of interest (casualty) with its camera, stream the images back and alert the human operator.

V. Technologies for Search and Rescue Operations

A. Autonomous UAV Solutions

1) Localization and Mapping

The self-localization and mapping of the UAV were achieved through open source lidar-based simultaneous localization and mapping (SLAM) solution [7]. The robustness and accuracy of the lidar SLAM enable the real-time operation and high-fidelity map for path planning. An efficient version of the probabilistic occupancy map like the one in [8] was developed using array indexing data structure instead of octrees. The probabilistic nature of the map enables realistic modeling of the environment using noisy sensor data.

2) Formation Planning

The formation planning approach presented in this paper references the approach of [9]. The formation planning approach requires the UAVs to be kept in sync with one another. If a certain message is required to be communicated between the UAVs, the UAVs will continue publishing that message until every UAV in the team has indicated that they have received the message. In addition, if there are UAVs detected to be joining or leaving the team during the process, all of the UAVs will restart from the first step of the process.

There are 3 main steps in the formation planning approach which are summarized by the pipeline in Fig.10. The first step is to generate a convex hull, $\mathcal{C}$ that is common to all UAVs and is globally free of $\mathcal{O}$. To generate $\mathcal{C}$, each UAV exchanges its positions, $\mathbf{p}_i$ with each other. Once every UAV in the team has communicated their $\mathbf{p}_i$ to each other, a common reference start point can be determined independently by the UAVs. Using the approach in [10], each UAV can generate its own convex hull, $\mathbf{c}_i$ by expanding the line segment between by the reference start point and goal position. As each UAV may have observed different parts of the environment, each UAV's $\mathbf{c}_i$ is expected to be different. To obtain the common convex hull, $\mathcal{C}$ the UAVs exchange and intersect their individual $\mathbf{c}_i$.

The second step will be to generate a set of virtual formation positions, $\mathcal{V} = \{\mathcal{V}_1 \dots \mathcal{V}_n\}$ within $\mathcal{C}$, for the team to be assigned to. The UAVs can conform to the different formation template that has been defined. For this paper, the 3 UAVs can switch between an equilateral triangle formation and a line formation. The translation t, size s, and orientation q, i.e $z = [t, s, q]$ of a formation template f will be optimized according to the constrained optimization formulation below:

$$\begin{aligned} \mathbf{z}_f^* &= \arg\min_{\mathbf{z}} J_f(\mathbf{z}) \\ s.t.\ & \mathcal{V} \subset \mathcal{C} \\ & s \geq \frac{(2 * \text{UAV radius})}{(\text{desired formation dist.})} \end{aligned} \tag{1}$$

Each of the $\mathcal{V}_i$ is constrained to be within the global obstacle-free region, $\mathcal{C}$ and the minimum size of the formation ensures that the UAVs are at least 2 times their radius apart from each other. The following equation below defines the cost function $J_f(\mathbf{z})$:

$$J_f(\mathbf{z}) = w_t||t-g||^2 + w_s||s-\bar{s}||^2 + w_q||q-\bar{q}||^2 + c_f \tag{2}$$

In the cost function, w_t, w_s and w_q are the design weights. The cost function aims to penalize the weighted deviation to the goal g, desired size $\bar{s}$, and desired orientation $\bar{q}$. A predefined cost value for each template formation is given by c_f to quantify the preference of a template formation over another.

Lastly, each UAV will compute its assigned $\mathcal{V}_i$. The objective of the assignments is to minimize the sum of the squared traveling distance to the formation positions. This is achieved by using the Hungarian algorithm [11] to find the permutation matrix, $X \in \mathbb{R}^{n \times n}$, that minimizes the following equation:

$$\min_X \sum_{1 \leq i \leq n} \sum_{1 \leq j \leq n} x_{ij}||p_i - \mathcal{V}_j||^2 \tag{3}$$

It is expected that every UAV will compute the same assignments, X due to the deterministic nature of the Hungarian algorithm. The UAVs will then publish their assignments as an additional safeguard to confirm that there are no conflicts before moving towards their assigned $\mathcal{V}_i$.

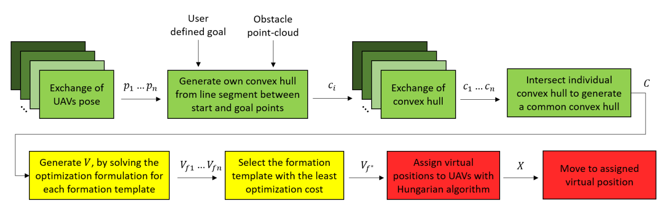


Fig. 10: Pipeline of the overall process. The 3 main steps are differentiated by the color in the order green, yellow, and red.

3) Trajectory Optimization

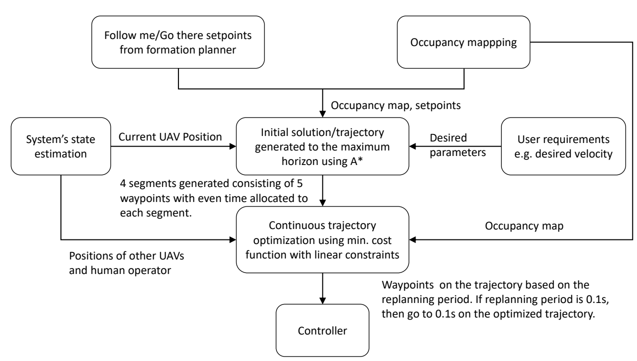


Fig. 11: Trajectory Optimization Flow Chart

Fig.11 depicts the trajectory planning and optimization process used in the system. A systematic algorithm, real-time multi-robot trajectory replanning using linear spatial separations (RLSS) [12] was heavily referenced, modified, and adapted to meet the demands of the project missions.

RLSS was selected because it does not pass specific motion primitives to the controller, enabling it to work with different controllers and leaving the generation of the primitives to the controllers themselves. The agents independently perceive the location of other UAVs in their vicinity and do not require any form of trajectory knowledge transfer from the other UAVs over the network.

In RLSS, a desired setpoint from the formation planner is issued along with a dynamic/static occupancy grid. An initial solution will then be established using A* discrete search which is run with the cost function computed from Euclidean distances between nodes. The solution is subsequently broken down into several segments (position and duration) as an initial solution for trajectory optimization.

Bezier curves are then generated to make up a B-Spline path for the entire trajectory using quadratic optimization where the cost function consists of the energy usage required from the UAV and the deviation from the initial solution. These curves are generated with the same constraints using linear spatial separation techniques derived from support vector machines as per the algorithm in [12]. The curves are protected from obstacles and preserve desired constraint requirements by a cocoon of control points generated from the optimization process. At the end of the optimization process, the trajectory is reviewed to ensure it meets the kinematic constraints and time set in the initial conditions. If not, an extension may be granted, or other means of constraint relaxation may be introduced to account for noise in dynamic environments and sensor inaccuracies.

The entire optimization sequence is done in real-time and iteratively updated at the computing frequency of ROS. This ensures that information is synchronized with the rate at which sensing information and control commands are given. Finally, after a solution in the form of a trajectory is produced, the setpoint with the timestamp equal to the optimization's update rate is referenced from the trajectory solution and fed to the position controller of the UAV.

4) Flight Controller

The UAV runs on a PX4 flight architecture [13] running on the Pixhawk 2.1 Orange for autonomous flights. The flight control scheme onboard that is used is the multi-copter position controller that takes in desired Cartesian setpoints via MAVROS which bridges the communications between the flight controller and the onboard processor that is running ROS and other algorithms. To protect the human operator during experiments, safety pilots were engaged to stand by and take over controls. Safety limits, tuning of gains, and checks on acceleration and position were also implemented along with mechanical propeller guards to minimize damage to the UAV in the event of erroneous actions from the autonomy stack during experiments.

B. UWB Localization Solution

In this section, we propose a centralized UWB-ranging SLAM which considers all UWB-ranging and odometry measurements in history to optimize the poses of multiple UAVs and a human for trajectory estimation similar to [14]. The approach is different from the conventional particle filtering-based approach [15], which relies heavily on the previous estimation for the prediction of the current pose. The use of UWB for positioning is popular due to its low cost, low power consumption, and small size [16]. The odometry of the UAVs was obtained through the lidar SLAM as mentioned before, while the human odometry was estimated through the VN-100 Inertial Measurement Unit (IMU) attached to the foot of the human.

We refer to the UAVs and humans as agents for simplicity. Formally, the goal is to estimate the path of agents k up to time T. We use x to denote the pose of the agent in SE(3), which includes the 3D position and the 3D orientation. The odometry of each agent provides the relative movement of an agent. We aim to estimate the trajectory of all agents given the odometry and UWB ranging measurements collected by all agents. This approach is referred to as the UWB-ranging SLAM. We use graph-based SLAM for implementation, which represents the pose of the agent as a node. The measurements are decoded as constraints in the graph. The graph-based SLAM aims to find the best configuration of the poses to minimize the error of all constraints through maximum likelihood estimation:

$$
\begin{aligned}
&\arg\min_{\mathbf{x}} \sum_{k=1}^{K} \sum_{m \neq k}^{K} \underbrace{\mathbf{e}(\mathbf{x}_k^1, \mathbf{x}_m^1, \Delta\mathbf{x}_{k,m}^1)^T \Omega_{k,m}^1 \mathbf{e}(\mathbf{x}_k^1, \mathbf{x}_m^1, \Delta\mathbf{x}_{k,m}^1)}_{\text{Assume known initial positions at t = 1}} + \\
&\sum_{t=2}^{T} \sum_{k=1}^{K} \sum_{m \neq k}^{K} \underbrace{\mathbf{e}(\mathbf{x}_k^t, \mathbf{x}_m^t, d(\mathbf{x}_k^t, \mathbf{x}_m^t))^T \Omega_{k,m}^t \mathbf{e}(\mathbf{x}_k^t, \mathbf{x}_m^t, d(\mathbf{x}_k^t, \mathbf{x}_m^t))}_{\text{Peer-to-peer UWB constraints}} + \\
&\sum_{t=2}^{T} \sum_{k=1}^{K} \underbrace{\mathbf{e}(\mathbf{x}_k^{t-1}, \mathbf{x}_k^t, \Delta\mathbf{x}_k^t)^T \Omega_k^t \mathbf{e}(\mathbf{x}_k^{t-1}, \mathbf{x}_k^t, \Delta\mathbf{x}_k^t)}_{\text{Odometry-based constraints}},
\end{aligned}
\tag{4}
$$

where K is the number of agents moving in the environment. $\mathbf{e}(\cdot)$ denotes the error function, which is computed based on the given poses and the constraints inferred from the observations (i.e., odometry and UWB ranging). The former is represented as the sequential odometry measurements. The latter is represented as the UWB ranging-based constraints. Constraints are additionally parameterized with a certain degree of uncertainty, which is denoted as the information matrix Ω. We apply a Huber robust kernel function (with a bandwidth of 1.0) to deal with noisy UWB-ranging measurements. In addition, we assume the initial positions of all agents are known. Therefore, we add a fixed constraint between the first nodes of all agents (see the first term in the above equation).

C. Casualty Detection

The casualty detection capability is realized by fusing a depth image of the scene with casualty detection results performed on the corresponding RGB image. The casualty detection on RGB images is achieved by running inferences on a pre-trained YOLOv3-tiny [17] CNN model. Both the depth and the RGB image are acquired by an RGB-D camera.

The expected output from the process is the 3D coordinates of detected targets in the camera frame. These local coordinates are then transformed into the global frame for the UAVs' controller for further actions.

The first step of the process is to detect targets on the RGB image by utilizing the YOLOv3-tiny model. This lightweight version is used because it is light enough to run on an edge device while detection results are relatively accurate for our purpose.

The next step in the process is to compute the coordinates of targets in the camera frame by combining the depth image and the targets' coordinates in the image frame.

Let (X, Y, Z) be the coordinates of a casualty in the camera frame and (x, y) its coordinates in the image frame. The coordinates (x, y) are known after running inferences on YOLOv3-tiny. From the acquired depth image, X is also known. So, we need to compute Y and Z to fully determine the 3D coordinates of the casualty.

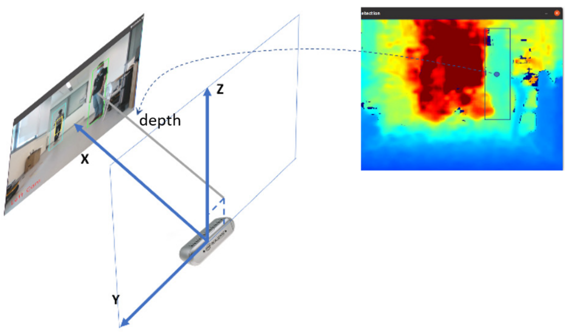


Fig. 12: Casualty's position in camera frame

Based on the diagram in Fig. 12, we can derive Y coordinate from following formula:

$$\begin{aligned} \theta &= \frac{x_0 - x_c}{W} * HFOV \\ Y &= X * \tan(\theta) \end{aligned} \tag{5}$$

in which x_0 and x_c are the x coordinates of the image center and the bounding box's center of the casualty in the image frame respectively, W is the width of the image, HFOV is the horizontal field of view of the camera.

The Z coordinate can be computed in the same way.

VI. Experiment

The experiments were carried out in a controlled indoor setting, using portable partitions to create an indoor environment that imitated typical indoor structures such as narrow passages, obstacles, and rooms. The objective of the human-robot team was to navigate through the scenario and find a simulated casualty (a mannequin). As such testing the feasibility of the different systems involved in close proximity group navigation UAVs, task allocation, and casualty detection. The experiment layout diagram is shown in Fig. 13. These experiments aim to validate the autonomy of the UAVs to follow commands given by the human operator. The network latency, CPU usage, and responsiveness to human commands are also closely monitored in these experiments.

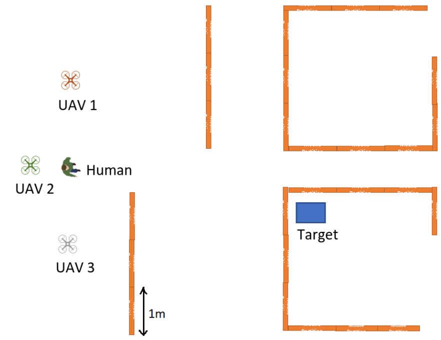
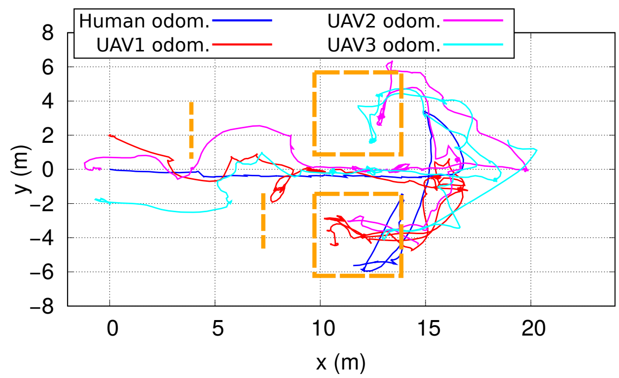
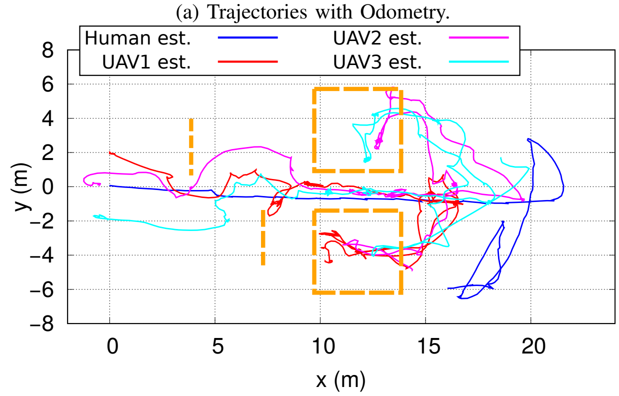


Fig. 13: Experiment layout



(a) Trajectories with Odometry.

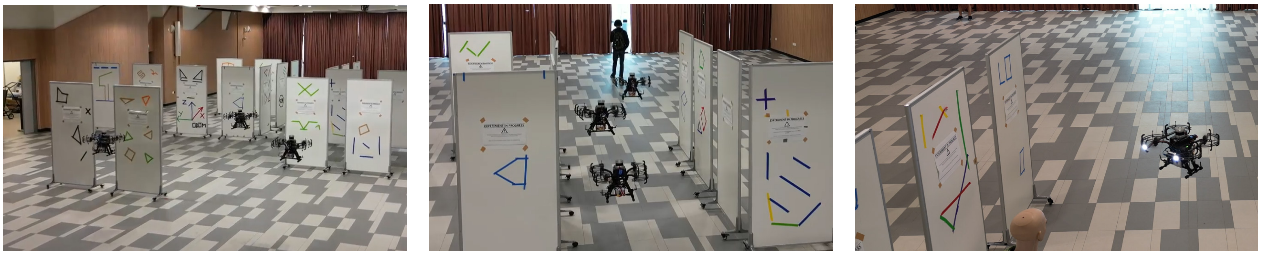


(b) Trajectories with UWB Ranging SLAM.

Fig. 14: Comparison in trajectory between Odometry and UWB Ranging SLAM. (a) Odometry Trajectory Estimate; (b) UWB Ranging SLAM Trajectory Estimate.

There are four main sequences in the experiment.

1) Initialization
   - Initialize the UAVs at starting positions.
   - Do a thorough system check to ensure everything is operational and ready.
   - UAVs proceed to take off and go into a triangle formation.
2) Navigating through a corridor
   - The human operator issues a "Follow Me" command and walks through the corridor.
   - The 3 UAVs follow the human, keeping a two-meters distance from behind.
   - From the triangle formation, the 3 UAVs go into line formation to navigate through the corridor.
3) Navigating to a specified set point



(a) Initial take-off of UAVs. (b) System navigating thru a corridor. (c) Casualty detection in action

Fig. 15: Experiment setup of indoor USAR scenario for human-robot collaboration system.

- The human operator issues a ”Go There” command to two of the UAVs to go to the room on the left.
- The human operator issues another ”Go There” command to a single UAV to go to the room on the right.

4) Casualty detection
- The room on the right contains a simulated casualty.
- The UAV then performs a ”Casualty detection” sequence, takes a picture, and sends it back to the human operator for viewing on his GUI.

The trajectories of the agents were recorded. Additionally, Fig.14b highlights the improvement of human pose estimation by UWB Ranging SLAM as compared to pure odometry shown in Fig.14a which is deteriorated due to the accumulated drift in the IMU attached to the human. This can be seen by examining the human trajectory, the human does not enter the room on the right in reality.

The experiment validates the feasibility of our robotic system of systems in an indoor environment. The UAVs successfully navigated through barriers and a corridor, went to a room, and finally detected the casualty. Both in formation and single UAV mode as shown in Fig.15.

VII. Conclusions

In this paper, we have presented our robotic system of systems as a practical solution for human-robot collaboration in USAR operations. The system leverages high levels of robot autonomy and formation planning for close proximity navigation and casualty detection. It serves as a foundation for future experimentation and the deployment of more advanced algorithms to tackle complex challenges in SAR scenarios. Although our research is focused on the indoor SAR environment, it is important to note that the system has wider applications in the SAR and surveillance domains.

Acknowledgement

The authors would like to thank Yeo Wee Hian Sean and Milven Lim for their role as safety pilots and Suryono Gunawan Ali for his role as a human operator during the experiments. The authors would also like to thank A.A.A.C Athukorala for his assistance with the project.

References

[1] L. Marconi, C. Melchiorri, M. Beetz, D. Pangercic, R. Siegwart, S. Leutenegger, R. Carloni, S. Stramigioli, H. Bruyninckx, P. Doherty, A. Kleiner, V. Lippiello, A. Finzi, B. Siciliano, A. Sala, and N. Tomatis, “The SHERPA project: Smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments,” in *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2012, pp. 1–4.

[2] G. De Cubber, D. Doroftei, D. Serrano, K. Chintamani, R. Sabino, and S. Ourevitch, “The EU-ICARUS project: Developing assistive robotic tools for search and rescue operations,” in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2013, pp. 1–4.

[3] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart, and K. Alexis, “CERBERUS in the DARPA Subterranean Challenge,” *Science Robotics*, vol. 7, no. 66, p. eabp9742, 2022.

[4] N. Hudson, F. Talbot, M. Cox, J. Williams, T. Hines, A. Pitt, B. Wood, D. Frousheger, K. Lo Surdo, T. Molnar, R. Steindl, M. Wildie, I. Sa, N. Kottege, K. Stepanas, E. Hernandez, G. Catt, W. Docherty, B. Tidd, and R. Arkin, “Heterogeneous Ground and Air Platforms, Homogeneous Sensing: Team CSIRO Data61’s approach to the darpa subterranean challenge,” *Field Robotics*, vol. 2, pp. 595–636, 03 2022.

[5] E. S. Redden and L. R. Elliott, “Robotic Control Systems for Dismounted Soldiers,” in *Human-Robot Interactions in Future Military Operations*, 2010.

[6] Stanford Artificial Intelligence Laboratory et al., “Robotic operating system.” [Online]. Available: https://www.ros.org

[7] J. Zhang and S. Singh, “LOAM: Lidar odometry and mapping in real-time.” in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.

[8] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “Octomap: An efficient probabilistic 3d mapping framework based on octrees,” *Autonomous robots*, vol. 34, pp. 189–206, 2013.

[9] J. Alonso-Mora, E. Montijano, T. Nägeli, O. Hilliges, M. Schwager, and D. Rus, “Distributed multi-robot formation control in dynamic environments,” *Autonomous Robots*, vol. 43, 06 2019.

[10] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, “Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments,” *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.

[11] S. Chopra, G. Notarstefano, M. Rice, and M. Egerstedt, “A distributed version of the hungarian method for multirobot assignment,” *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 932–947, 2017.

[12] B. Şenbaşlar, W. Hönig, and N. Ayanian, “RLSS: Real-time Multi-Robot Trajectory Replanning using Linear Spatial Separations,” *arXiv e-prints*, p. arXiv:2103.07588, Mar. 2021.

[13] L. Meier, D. Honegger, and M. Pollefeys, “Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 6235–6240.

[14] R. Liu, Z. Deng, Z. Cao, M. Shalihan, B. P. L. Lau, K. Chen, K. Bhowmik, C. Yuen, and U.-X. Tan, “Distributed ranging slam for multiple robots with ultra-wideband and odometry measurements,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 684–13 691.

[15] Z. Cao, R. Liu, C. Yuen, A. Athukorala, B. K. K. Ng, M. Mathanraj, and U.-X. Tan, “Relative localization of mobile robots with multiple ultra-wideband ranging measurements,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5857–5863.

[16] M. Shalihan, R. Liu, and C. Yuen, “Nlos ranging mitigation with neural network model for uwb localization,” in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2022, pp. 1370–1376.

[17] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv*, 2018.