

Autonomous VTOL-UAV Docking System for Heterogeneous Multirobot Team

Eduardo Narváez^{ID}, Ankit A. Ravankar^{ID}, Abhijeet Ravankar^{ID}, Takanori Emaru^{ID}, *Member, IEEE*, and Yukinori Kobayashi^{ID}, *Member, IEEE*

Abstract—Heterogeneous robot teams offer significant advantages for exploration and navigation missions in unknown environments. Notably, the air-ground robot teams have superior efficiency due to their mixed abilities and shared sensor resources useful when completing missions. However, the high energy consumption and reduced payload capability of the vertical take-off and landing (VTOL) aerial platforms limit the complete robot team’s behavior. At some point in a mission, the aircraft needs to return to the base, land, and replenish its batteries. This task restricts the team to explore the environment in-depth and to work for longer durations. To solve this problem, we present an autonomous system for docking a VTOL-unmanned aerial vehicles (UAVs) with a mobile manipulator. The central measurement-actuation unit consists of a robot manipulator mounted on a mobile platform with a visual sensor configured as an eye-in-hand device. The visual information is used to execute a stable UAV tracking to achieve air-ground robot contact firmly. This strategy allows the multirobot team to economize and potentially recover aircraft energy. It also enables the robust ground platform to protect and store the aircraft facilitating to continue the mission for a longer duration. The on-site autonomous docking can be initiated multiple times, allowing the team to explore larger areas and complete long-term missions. The proposed method has been tested in simulations and real environments with detailed experimental results.

Index Terms—Heterogeneous robot teams, multirobot systems, robot exploration, unmanned aerial vehicles (UAVs) docking.

I. INTRODUCTION

RECENTLY, the interest in heterogeneous multirobot systems for precise monitoring tasks, urban search and rescue (USAR) missions, industry applications, and space exploration has been increasing [1]. Features like fault tolerance, flexibility, and robust integration, make the robot teams truly attractive to solve distributed data collection, and remote sensing missions [2]. By choosing a multiagent

Manuscript received August 9, 2020; revised October 4, 2020; accepted November 1, 2020. Date of publication November 24, 2020; date of current version December 30, 2020. This work was supported by Japan Society for the Promotion of Science (JSPS) KAKENHI Grant JP20K04392 entitled ‘Robust SLAM by Non-Uniform UGV/UAV Groups for Large Field Management.’ The Associate Editor coordinating the review process was Shutao Li. (*Eduardo Narváez and Ankit A. Ravankar are co-first authors.*) (*Corresponding author: Ankit A. Ravankar.*)

Eduardo Narváez, Ankit A. Ravankar, Takanori Emamu, and Yukinori Kobayashi are with the Division of Human Mechanical Systems and Design, Research Faculty and Graduate School of Engineering, Hokkaido University, Sapporo 060-8628, Japan (e-mail: enarvaez@eis.hokudai.ac.jp; ankit@eng.hokudai.ac.jp; emaru@eng.hokudai.ac.jp; kobay@eng.hokudai.ac.jp).

Abhijeet Ravankar is with the School of Regional Innovation and Social Design Engineering, Faculty of Engineering, Kitami Institute of Technology, Kitami 090-8507, Japan (e-mail: aravankar@mail.kitami-it.ac.jp).

Digital Object Identifier 10.1109/TIM.2020.3039649

approach for the design, redundancy is automatically included as a central component of the system. Moreover, with cooperative behavior and parallelism, the sensing efficiency is drastically improved. In general, the heterogeneous team of robot platforms has a higher probability of finding optimal solutions for the assigned task in a reliable, faster, and efficient way than a single robot systems [3].

The tasks for the deployed robot teams consist of detailed exploration, precise navigation, and interaction within unknown environments, which in most cases, are not solvable by a single agent [4]. For monitoring unknown regions with diverse structures and obstacles, it is convenient to use heterogeneous teams that dynamically combine individual robot capabilities [5].

In this context, the heterogeneous robot teams that use aerial and ground platforms have a significant advantage as they can sense the environment from different perspectives offering an increment in redundancy. By using a distributed configuration, the sharing of resources potentially leads to an increase in general performance.

Specifically, to solve the multiple challenges of the exploration of unknown environments, unmanned aerial vehicles (UAVs) with vertical take-off and landing (VTOL) capability are very suitable. They are excellent measurement platforms, as they provide a high point of view, hovering ability, and outstanding versatility in its trajectories. On the other hand, the unmanned ground vehicles (UGVs) provide a low point of view and robust platforms that can carry large payload and be equipped with different sensor suits, actuators, and energy storage cells. Such capabilities are symbiotic to each other so that the heterogeneous team can share information and process power through a wireless channel. The information extracted from the UAV sensors can be merged with the information from the UGV sensors to generate a complete database that can be used to build a precision map of the environment for navigation and other purposes [6].

A big concern for using the VTOL-UAV is power efficiency, limited payload capacity, and weak impact protection. These factors can affect team behavior and overall mission success [7]. For this reason, it is possible to rely on the UGV robustness to counteract the UAV drawbacks as it can navigate safely through critical areas like door frames or areas with low ceilings. It can also transport enough energy storage cells to supply its tasks and potentially for many UAV overflights [8]. A specific interaction is essential for resource sharing, and evidently, physical contact between the robots is required. This contact requirement is the main

problem addressed in this work. With a stable connection mechanism, it is possible to protect the UAV and enable the robot team to explore vast environments and perform detailed inspection without needing the UAV to return to the charging station. The problem complexity lies in finding a proper attitude of the robots and an optimal trajectory while considering the robot's features and limitations, along with the environmental and dynamic constraints.

A. Related Work

Previous works on the heterogeneous robot system can be divided into two classes.

- 1) Works focused on the creation of control architectures that seek the cooperation and synchronization of robots for navigation and exploration purposes.
- 2) Works aimed toward building systems capable of solving more complex and extended missions, including physical interactions between robots in cluttered environments.

The former group's primary concern is to generate a suitable architecture where mixed data from the robot sensors can be merged, thereby creating valuable information for exploration and navigation processes and scheduled tasks to achieve a goal. For example, work [9] presents a bio-inspired pheromone deposition algorithm for cooperative exploration and mapping. In [10], a hitchhiking-based approach is adopted where resources between multiple robots are shared based on leader–follower action to save computation cost for navigation. Work in [11] presents the sharing of obstacle information in multirobot scenario with uncertainty for improved navigation and path planning. Approaches like [12] proposed a collaborative mapping with multirobot systems for navigation. Works in [13] and [14] proposed various methods to build a 3-D terrain map based on UAV and UGV sensors, all of them advancing one step forward by calculating optimal paths to guide ground robots toward a target. Similar works are presented in [15] and [16], particularly for GPS-denied environments.

Furthermore in [17], the emphasis is on creating a cooperative control scheme that uses visual information to calculate the relative localization between multiple ground robots. Similarly, study [18] uses a vision-based approach to synchronize the relative position of a dynamic target and a robot arm in a space environment. Ravankar *et al.* [19] presented multi-robot path planning for distributed charging points in map. Also, coordination methods for a group of UAVs to provide continuous coverage to a moving ground convoy is proposed in [20] and [21]. In [22], it is proposed to use visual servoing to track the UGV from UAV sensor information, while in [23], a switched cooperative control scheme to coordinate groups of the ground and aerial robots for scene exploration and localization of a dynamic target is proposed. In these cases, tasks for exploration, global guidance, and obstacle detection are improved using measurements from the UAV's high field of view.

In the second group of research works, the primary intention is to provide the heterogeneous robot teams with the ability to interact with the environment and between members. A robot team with these features can build mechanisms to share

resources and consolidate a synergic integration to provide better results in complex and long-term missions.

Maini and Sujit [24] have proposed a strategy for coordination between a refueling UGV and a quad-rotor for the exploration of large areas. Although it only presents path calculations, it improves the UAV-UGV cooperation and routing for road networks. In [25], a measurement system and algorithms were developed for vision-based aerial refueling. Several simulations were presented with techniques to solve the pose estimation of a tanker and a receiver UAV. A different approach is presented in [26], where the UGV transports the UAV to a set of target points to be deployed for precise exploration. In [27], the UAV is designed to transfer (pick and place) parcels into mobile robots for autonomous transportation in warehouses with impeded ground exploration. These works rely on visual information to calculate the relative pose of the robots and plan a safe rendezvous. They also coincide with the need to improve the physical linking process to enable resource sharing between members and temporarily treat the team as a single entity.

Many of the previous works on air–ground robots focus on the contact mechanism as a crucial task for aircraft recovery and increase overall UAV autonomy and capabilities. Designing such contact mechanisms is challenging because the team is always exposed to many external variables and demands the control strategies to be fast and robust enough to handle such variables. A standard solution that has gained significant importance in the field is VTOL-UAVs' visual assisted landing in static or mobile landing platforms. Research works in [28]–[30] developed UAV landing techniques with visual input for calculating the error between the UAV and the marker position fixed on the helipad. Moreover, when the landing zone is in motion, the control techniques and advanced strategies, as presented in [31]–[34], are required to highlight the implementation of robust tracking and final approximation for a safe landing.

In [35], a robust contact mechanism system is proposed, where the energy consumption of the whole team in reaching the target is analyzed. It is used a UAV-UGV system that is capable of negotiating with obstacle evasion, and that can also perform UAV landing and UGV lifting in disaster scenarios.

In general, the direct UAV landing proposals present control laws designed to trust mainly in the underactuated UAV controllers. Such control strategies are difficult in complex missions, as the UAV actuators are not in full control of the six degrees of freedom (DoF). This situation poses a serious risk when facing aleatory external variables [36]. Also, it demands a large landing platform to be attached to the UGV platform that can cause interference for ground navigation.

Other works such as [37]–[40] have considered the VTOL-UAV landing on surfaces subjected to six DoF variations, e.g., helipads in ships, UGVs in uneven terrain, or earthquake situations. For these cases, the systems are designed to track cartesian movements and to compensate for the angular deviation with software solutions ([37], [38]) or with actuated platform mechanisms ([39], [40]).

One exciting work summarized in [41] proposes the landing in a robot manipulator by installing a landing pad in the

end-effector of the robot arm. It performs an assisted landing with significant redundancy in a moving robot exploring the robot arm's benefits for this type of interaction.

However, the limitations of these works are that the UAV is susceptible to fall without a safe locking mechanism in the contact state. The instabilities caused by any external disturbances are not fully addressed.

As a new improvement of the techniques in the state of the art, we present in this work, a novel VTOL-UAV docking with a mobile manipulator. It is based on our previous work in [42], where we presented simulation cases that demonstrated the feasibility of the docking process with simple reactive control algorithms. In this article, we present our complete working system that was tested on a UAV-UGV team in different scenarios with further improvements such as robust tracking and locking mechanism to store the UAV. We also present the control algorithms addressing issues such as the robustness and stability of our system.

B. Contribution

This article proposes a system that can exploit the hardware and software resources of a heterogeneous robot team. A measurement and actuation platform that allows an autonomous linking process (docking) with the aerial robot is developed. A control strategy is implemented to transform the visual measurements into precise robot movements for the linking task. With our system, the air-ground robot team is capable of making safe and efficient contact for temporary storage and energy replenishment. It can overcome the limitations and enhance its ability to explore unknown areas for a longer time.

The main focus is on achieving excellent robustness and safety in the linking process. Thus, the cooperation between robots becomes natural, and the team acquires superior features to complete USAR missions. In this context, it is not desirable to rely only on the VTOL-UAV controller due to its underactuated character, as it has only four actuators to move in a six DoF space. Thus, the use of the ground robot manipulator becomes essential because it adds redundancy to the control scheme while increasing the range of action and enabling contact tasks in a larger workspace. It also becomes easier to compensate for the variations caused by air and ground external disturbances. Furthermore, the manipulator's use also allows us to secure and handle the UAV chassis and store it in an appropriate position that does not obstruct the ground sensors, becoming part of the UGV without interfering with the ground navigation.

We present a completely autonomous system that manages the robot platforms from the UAV-UGV first rendezvous, UAV tracking task to the final contact process. We propose an algorithm that fuses the reactive, predictive, and optimization approaches to support these steps. The algorithm calculates the end-effector trajectory for UAV tracking and simultaneously calculates the UAV relative pose control for contact tasks. It incorporates robotic and environmental constraints in the control loop by considering the robot platforms' limitations, e.g., robot structures, sensor placements, and motion properties while negotiating static and dynamic obstructions from the unknown environment.

We believe this is the first work providing a different perspective in the research of aerial and ground robots' usage and their integration to cooperative architectures for complex mission deployment. We believe that this can open the door for new exploration strategies of heterogeneous robot teams in unknown environments.

The remainder of this article is structured as follows. Section II contextualizes the heterogeneous robot team exploration. Section III presents a general introduction to the air-ground robot team relevant to this work, and Section IV details the design of the control algorithms and hardware devices for the UAV docking task. We also describe the experiments in simulation and real scenarios, along with their results in Section V. Finally, we present a discussion of the results obtained in both simulated and real environments with concluding remarks and future plans.

II. USAR MISSIONS WITH HETEROGENEOUS ROBOT TEAMS

A heterogeneous robot team deployed in an unknown environment is required to perform a series of navigation, exploration, and interaction tasks that are executed with priority levels defined by the mission design. In case of a search and rescue robotic operation in an indoor environment, the following is an example of a list of prioritized tasks to be accomplished by the robots:

- 1) Reach and interact with the target.
- 2) Maintain safety for all robot platforms.
- 3) Economize energy.
- 4) Extract as much information as possible from the scene.

In search and rescue operations, the process of reaching the main target could belong to a higher level of priority, while the protection of the robots can be placed at a slightly lower level. Because of the USAR applications' critical nature, different levels of priority can be set for the robot. For example, the robots in [43] are designed to provide assistance and explore unstable environments after the World Trade Center disaster, where the higher priority is to guard the stability of the scene to avoid any risk to the victims. On the other hand, the safety of the robots is not prioritized in this situation. However, in emergency response cases described in [44], the robots needed to quantify water motion, recover samples, and rapidly install a water gauge in a radioactive environment. In this case, robot safety is the top priority as it needs to return to a safe area for sample retrieving.

Furthermore, in cases where the supervision of a human operator is involved, the human agent in the loop can provide an expert opinion of the state of the whole mission and, when necessary, perform goal modification or cancellation with top priority at any time.

In other cases, the robot team's primary purpose is to explore the environment and provide sensor information for obstacle detection [6]. For that, all the team resources are used to follow trajectories to survey the scene and merge heterogeneous sensor information. In this case, the mission is not firm, and it is possible to prioritize with less rigor giving more importance to data extraction and robot protection.

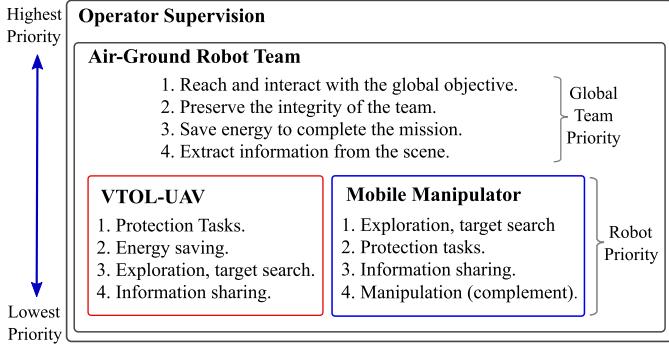


Fig. 1. Priority levels for tasks execution in the air–ground robot team.

The system developed in this work can help accomplish a mission where the main goal is to reach and interact with a *target* in an unknown environment. It is also important to extract as much information as possible from the scene with a detailed inspection approach to facilitate navigation and share essential data with future missions. If the building is large and composed of several rooms and aisles, the mission needs a long time. And if the scene is cluttered and contains aerial or ground navigation difficulties, the robots need to execute mechanisms to solve local problems and plan for an optimal way to reach the final goal.

The air–ground team proposed comprises a lightweight VTOL-UAV and a Mobile Manipulator platform (mobile robot + manipulator) in a centralized configuration. This team and the controller architecture are described in detail in Section III and IV. For this kind of air–ground robot team, it is essential to consider each robot platform’s features to establish the priority for the tasks. Each platform has to prioritize their tasks, but as part of a team, they also have a common goal and general priorities (see Fig. 1). For the mobile manipulator, the significant priority should be in following the shortest possible trajectory to reach the global *target*, and at the same time, it should be concerned in performing basic sensing of the scene for information and navigation purposes. For example, the *target* can be to reach a person, open a valve, recognize or pick up an object. The onboard sensors and actuators are then used for obstacle detection and to interact with the environment for safe navigation.

The aerial robot can cover large areas and can potentially provide sensor information with six DoF. Such information is supplied as complementary information, which means the team, although not optimal, can continue pursuing the goal even if the aircraft is not providing any data. Moreover, because of its high risk of damage and limited autonomy, the UAV is forced to execute a high priority self-protection task. It continually checks the battery level, distance with the charging station, and possible obstacle collisions in the vicinity. The UAVs’ lower priority exploration tasks can be immediately interrupted or delayed if the protection process requires to perform any operation.

The two robot platforms are in constant communication, performing full or semiautonomous tasks, but the team acts under orders from a human supervisor with topmost priority. As shown in Fig. 1, global priorities are in charge of managing

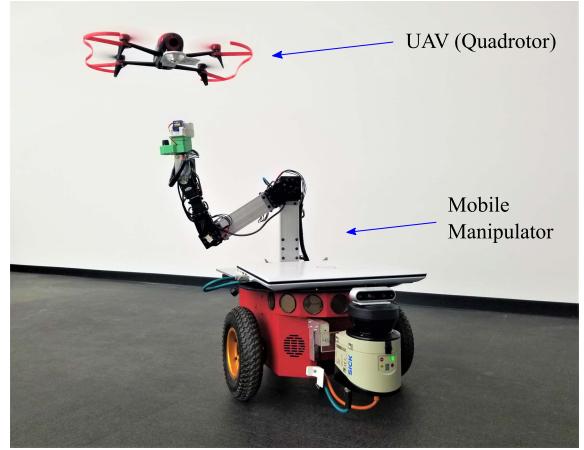


Fig. 2. Air–ground robot team.

the team as a whole, and local priorities to control each platform’s behavior. As stated earlier, the higher levels can outrank the lower levels; for example, when the UAV has some protection issue (e.g., energy deficiency or presence of an obstacle that impedes exploration), the complete team does not stop trying to reach the main *target*. The aircraft should find a way to protect itself, e.g., by reaching a charging station or calculate trajectories to avoid obstacles.

On the other hand, if the exploration and *target* pursuit are being developed correctly, the rest of the computational and hardware resources (including mobile manipulation) can be used to ensure the safety of the robots, especially of the UAV.

III. AIR–GROUND ROBOT TEAM

It is possible to use several types of multirobot systems to solve industrial, USAR, or even special missions. In this work, we propose the use of an air–ground robot team composed of a VTOL-UAV and a mobile manipulator with centralized architecture (see Fig. 2).

A. VTOL-UAV

This platform is a lightweight multirotor aircraft with excellent stability and maneuverability. The robot comes with a visual sensor, barometer, sonar, gyroscope, inertial measurement unit (IMU), and battery level sensors. The controller is in charge of linear and angular movements, specifically the *altitude*, *yaw*, *roll* and *pitch* variations. It is equipped with a docking adapter and a fiducial marker (front-bottom) to be detected with visual sensors from the ground for contact purposes.

B. Wheeled Mobile Manipulator

It is a robot with a nonholonomic wheeled base and a robot manipulator with six DoF attached to the top deck. It is designed to bring manipulation to the environment and interact with the scene with more versatility. There are many possible tool configurations for the end-effector of the robot manipulator, and it can be a gripper, a variety of actuators, or a set of sensors. For our purposes, the end-effector holds a camera that provides visual information with an eye-in-hand configuration useful for a detailed exploration when a

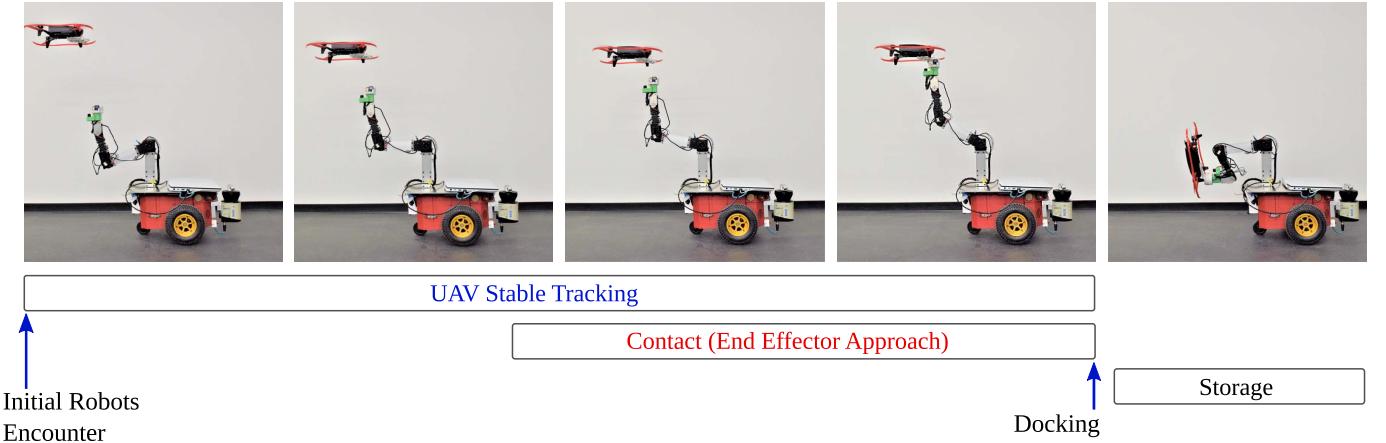


Fig. 3. General flow of docking operation.

close view of an area is required. In our case, it is used as the main measurement unit for UAV detection. Additionally, the end-effector is equipped with a docking device, designed specifically for the system proposed in this work, to lock the UAV adapter and ensure a secure UAV docking and storage.

IV. VTOL-UAV AUTONOMOUS DOCKING

The docking between the air-ground robot team is a crucial task for long-term mission planning and exploration. As shown in Fig. 3, the proposed docking operation is divided into four tasks:

- 1) Initial encounter
- 2) Tracking
- 3) Contact
- 4) Storage

The initial encounter is when the two robot platforms reach a meeting point; the end-effector searches for the UAV with the camera pointed upward. When the UAV marker enters the vision field, the UAV relative position is extracted, and the tracking task starts.

As shown in Fig. 3, the tracking task is initiated after the first encounter, which is when the UAV marker enters the vision field of the end-effector camera and follows until the final docking when the aircraft is secured in the end-effector.

A. UAV Motion Tracking

For the docking process, only the visual information is extracted with the camera. The manipulator needs to pair (align) its end-effector tip with the UAV adapter. During the final approach and actual contact, the end effector's position and orientation relative to the aircraft need to be stable and constant. For steady-state, the control system needs to consider the following:

- 1) *UAV lateral and longitudinal movements:* The camera attached at the end-effector is pointed to the UAV marker extracting its relative pose. While the UAV is in motion, the end-effector follows and plans within its workspace to keep the marker close to the center of the image.
- 2) *Self-prohibited positions:* The end-effector should move within the allowed workspace and avoid prohibited

positions. The arm joints need to be controlled within its range.

- 3) *UAV features:* If the UAV has any large displacements, for example, if it moves faster laterally than longitudinally, then the end-effector should follow such motions accordingly. Another important aspect to consider is the sensor placements. For the VTOL aircraft, the altitude sensors are pointing in a downward direction, and if they detect the mobile manipulator structure, the altitude measurement can affect and interrupt the process.

We designed a control system that considers all the previous factors to perform an optimal tracking task. It controls the UAV pose to locate in a reachable position within the manipulator workspace and uses a series of optimization, prediction, and reactive algorithms to build a *Motion Engine* that calculates proper trajectory points for the end-effector.

As shown in the complete controller scheme in Fig. 4, the UAV pose in manipulator frame P_Q^M is used to feed the manipulator *Motion Engine*, and at the same time is used for UAV pose control. The UAV pose is being controlled within \mathcal{M} coordinates, which allows it to be in a constant pose with respect to the mobile manipulator. P_Q^M is calculated by merging the end-effector position, given by the forward kinematic relation of the manipulator $\mathcal{F}(q)$, with the UAV pose in camera coordinates $P_Q^{C_A}$ and with the offset O_C caused by the position of the camera that is not located exactly in the center of the end-effector

$$P_Q^M = \mathcal{F}(q) + P_Q^{C_A} + O_C. \quad (1)$$

P_Q^M becomes the input for the *Motion Engine* block. It consists of a trajectory generator that incorporates the UAV and UGV states for optimization and motion planning. Fig. 5 details the flow of the *Motion Engine*, where the blue shading indicates the Planning and Optimization algorithms and the orange part represents the algorithm for reactive control and high-speed distribution of results. During the execution, the planning and optimization part runs nearly at $f_B = 4$ Hz, and the reactive controller part approximately twice as fast $f_O = 2f_B$. This frequency separation design is to isolate the slow performance algorithm, while using a fast process

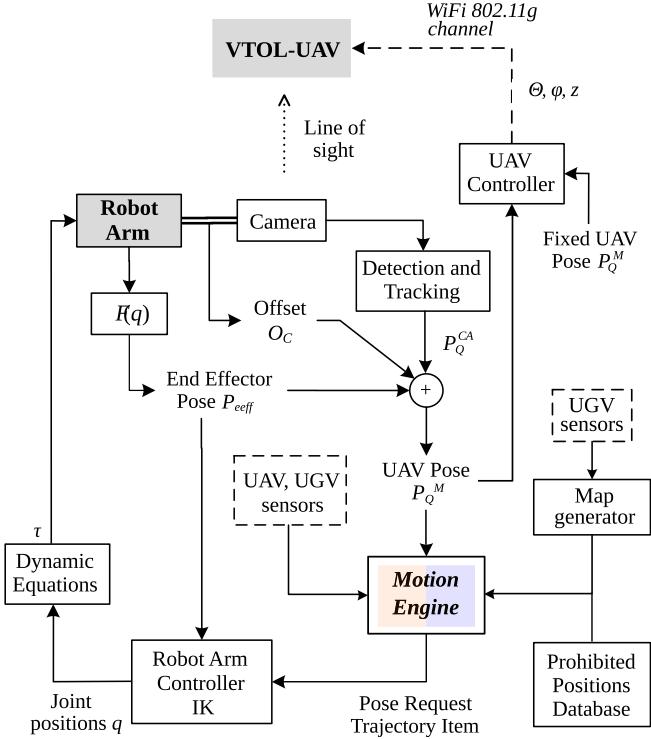


Fig. 4. General control scheme.

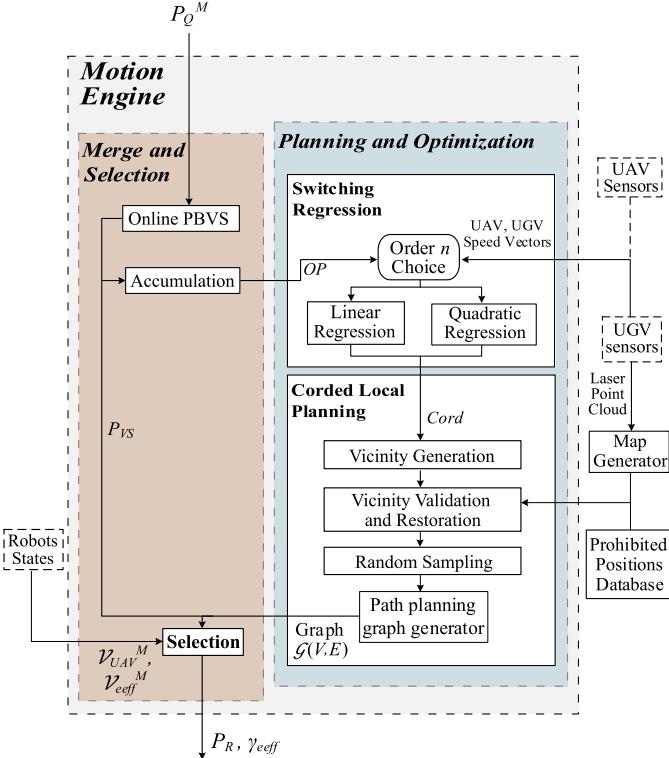


Fig. 5. Schematic of Motion Engine process.

to afford real-time control of the robot arm, ensuring to bring instructions fast enough to follow the UAV movements.

The Planning and Optimization process (blue shading in Fig. 5) is a sequential chain that starts with the calculation of a basic switching regression and a local path planning to

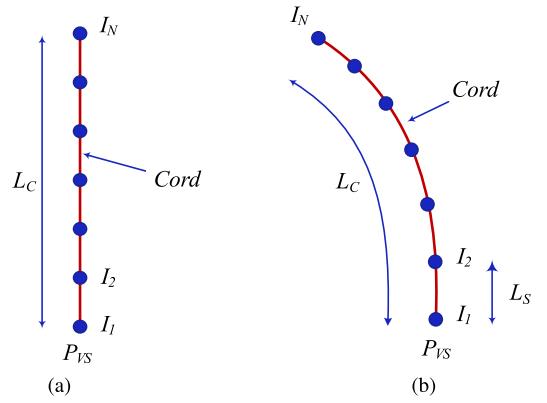


Fig. 6. Cords for (a) linear and (b) quadratic regression.

provide a dense graph, while the Merge and Selection process (orange shading in Fig. 5) is in charge of the calculation of an *Online Position-Based Visual Servoing* (Online-PBVS) [45] P_{VS} value, storage of past P_{VS} values in a list (OP), and merging with the Plannification tree and UAV+UGV states to finally provide the next optimal end-effector trajectory.

1) Switching Regression: This basic regression is for relative motion calculation. It considers the velocity differences between the mobile manipulator and the UAV to create a series of possible end-effector movements. The objective is to derive the UAV relative motion intention and project the end-effector toward the possible 6-D motion direction for tracking preservation.

The predicted movements are calculated for a short and constant range L_C (around $L_C = 15$ cm), and it is assumed that the robots will not perform complex motions for these small distances. Thus, only two important cases are necessary to implement, the linear and the quadratic regression (order $n = 1$ or $n = 2$, respectively). The algorithm will analyze the state of the robots for switching and if the UGV is moving faster than the UAV, a linear mode will be used, but if the UAV has more activity, a quadratic mode is selected.

With any of these two cord types (linear and quadratic), the result will be a connected succession of end-effector poses (*Items*), that build the *cord* and guides the motion planning, as shown in Fig. 6. The resultant *cord* will have N *items* (I_i with $i = 1, 2, \dots, N$) separated between each other with a distance equal to L_S and a total length of L_C ($L_C = NL_S$). The first *item* I_1 is filled with P_{VS} calculated in the *Select and Merge* block. The rest of the points I_i for $i = 2, 3, \dots, N$ are calculated depending on the selected order n .

$$I_i = [x_i \ y_i \ z_i] \quad (2)$$

$$y_i = f(x_i) = \alpha_0 + \sum_{k=1}^n \alpha_k x_i^k. \quad (3)$$

By using the curve fitting techniques explained in [46], it is possible to use least squares calculation and Gaussian elimination to find the coefficients α_n with the set of old P_{VS} values $OP = [x_j, y_j]$ for $j = 1, 2, \dots, N_{OP}$. The following

Algorithm 1 *Cord Calculation*

Input $P_{VS} = [x_{VS} \ y_{VS} \ z_{VS}]$, OP
Output $I_i = [x_i \ y_i \ z_i]$

- 1: $I_1 \leftarrow P_{VS}$
- 2: $\Delta x, \Delta y \leftarrow CalculateAxesVariation(OP)$
- 3: **for** $i \leftarrow 2$ **to** N **do**
- 4: $z_i \leftarrow z_{VS}$
- 5: **if** $\Delta x > \Delta y$ **then**
- 6: **for** $\delta \leftarrow 0$ **to** L_S , with small increments **do**
- 7: $x \leftarrow x_{i-1} + \delta$
- 8: $y \leftarrow a_0 + \sum_{k=1}^n a_k x^k$
- 9: **if** $(\sqrt{y^2 + x^2}) \geq L_S$ **then**
- 10: $x_i \leftarrow x$
- 11: $y_i \leftarrow y$
- 12: **else**
- 13: **for** $\delta \leftarrow 0$ **to** L_S , with small increments **do**
- 14: $y \leftarrow y_{i-1} + \delta$
- 15: $x \leftarrow a_0 + \sum_{k=1}^n a_k y^k$
- 16: **if** $(\sqrt{y^2 + x^2}) \geq L_S$ **then**
- 17: $x_i \leftarrow x$
- 18: $y_i \leftarrow y$
- 19: **return** $I_i = [x_i \ y_i \ z_i]$

matrices¹ summarize the process

$$X = \begin{bmatrix} N_{OP} & \sum x_j & \sum x_j^1 & \dots & \sum x_j^n \\ \sum x_j & \sum x_j^1 & \sum x_j^2 & \dots & \sum x_j^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum x_j^n & \sum x_j^{n+1} & \sum x_j^{n+2} & \dots & \sum x_j^{n+n} \end{bmatrix}$$

$$A = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad B = \begin{bmatrix} \sum y_j \\ \sum x_j y_j \\ \sum x_j^2 y_j \\ \vdots \\ \sum x_j^n y_j \end{bmatrix} \quad (4)$$

which means it is possible to use the equation

$$XA = B \quad (5)$$

to calculate carefully a solution for the coefficient matrix

$$A = X^{-1}B. \quad (6)$$

Once the coefficients a_n are available, the values for x_i y_i are derived by (3) with the considerations detailed in the Algorithm 1. Note that the value z_i is constant maintaining the complete *cord* in the same height, only being modified by later calculations to avoid constraints.

On the other hand, if the UAV relative pose is more or less static in M frame and the UGV is not moving in any direction, the information to perform prediction is not enough, and the regression is canceled. In this case, the cord contains a single item $N = 1$ filled with P_{VS} , and the planning will continue without main corded guidance (*cordless*).

¹The summations of (4) are for $j = 1$ to $j = N_{OP}$, where n is the selected order of the regression.

If the relative movement intention changes abruptly, a cancellation flag is created to impede the use of the cord for motion planning, the cord is replaced with a single item $N = 1$ to give time for a new pose accumulation, and a new cord is drawn in the next iteration to resume planning without delay.

2) *Corded Local Path Planning*: The behavior of the aircraft is difficult to predict in general. The computation of the *switching regression* provides a basic notion for the future relative UAV movements, but there is an exponentially growing uncertainty when the predicted distance increases. To include this factor in manipulator calculations, we propose to manage the end-effector movements along a motion planning tree expanded in a local guided region. This region or vicinity will be calculated in the spatial surroundings of the mentioned *cord*, and its volume encloses all the possible 3-D locations that have a significant probability of placing the end-effector in an optimal tracking position for near-future iterations.

The vicinity is created with sections (S_i with $i = 1, 2, \dots, N$) delimited by ellipses centered in the items I_i . The ellipses are drawn perpendicularly to the main direction of the *cord* (see Fig. 7) with Euler angles φ_i , θ_i and ψ_i , respectively, and their axes (semimajor and semiminor axes a_i and b_i , respectively) depending on the distance of the center (*Item*) to the cord origin as follows:

$$L_{os} = 1 + \sum_{k=1}^i L_s$$

$$a_i = K_a L_{os}^2$$

$$b_i = \frac{K_b L_{os}}{2}, \quad \text{with } i = 1, 2, \dots, N \quad (7)$$

where K_a and K_b are gains that determine the size of the ellipses. The semimajor axes a_i depends exponentially on the progression of the items (L_{os}) and determines the width of the ellipses perpendicular to the main direction. The semiminor axes values b_i determine the size of the ellipse in z -direction and they grow linearly. This is because the vertical position of the end-effector does not affect the tracking. If only the altitude changes in a small band, the marker position in the image is unaltered and the only uncertainty source is the scale variation.

In the case the cord has a single item ($N = 1$), the ellipse gains are doubled, $K_{sa} = 2 * K_a$ and $K_{sb} = 2 * K_b$, and the vicinity becomes an ellipsoid, see Fig. 7(c)

$$a_1 = K_{sa} L_{os}^2$$

$$b_1 = \frac{K_{sb} L_{os}}{2}$$

$$c_1 = L_s. \quad (8)$$

The next step is to confirm whether the entire volume of the vicinity is confined within the allowed manipulator workspace by checking contacts with obstacles, prohibited locations or with singularity areas. For that purpose, it is convenient to use a 3-D local map generated with UGV sensor information that is previously loaded with prohibited zones defined by the *mobile manipulator* dynamic and structural features, e.g., location of the mobile base, or areas where the end-effector will fall in cases of singularities (See Fig. 8).

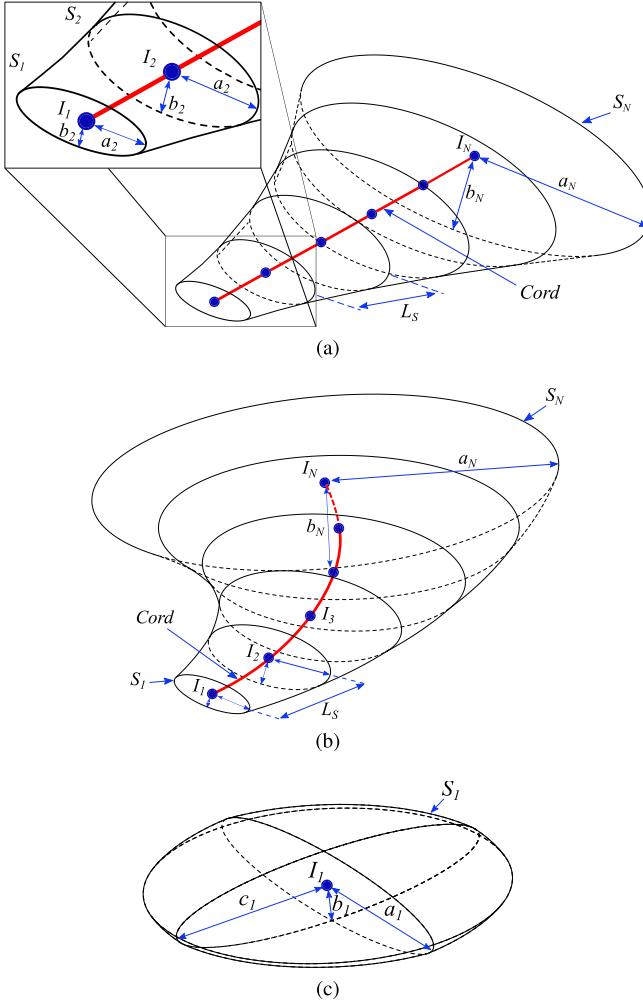


Fig. 7. Vicinities and respective cords. (a) Linear. (b) Quadratic. (c) No cord.

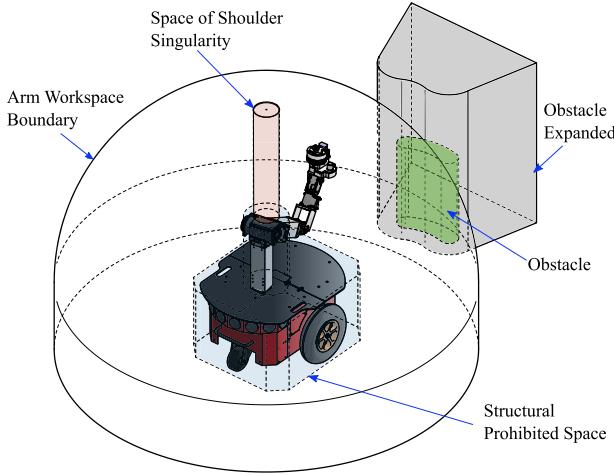


Fig. 8. Generated environment map with obstacles and prohibited areas.

In Fig. 8, the green region represents the detected obstacles and are drawn in the map as columns perpendicular to the ground expanded in polar coordinates, thereby prohibiting the end-effector from entering in the volume of the obstacle plus everything above, below, and space not seen by the sensors (gray area of Fig. 8). This consideration is taken because

the end-effector is tracking the UAV with the eye-in-hand camera, and the space between the end-effector and the aircraft cannot be obstructed. It is crucial to keep the camera field of view clear from obstacles, so the line of sight is maintained. Additionally, as the mobile robot platform only detects the faces that are facing the sensor, the obstacles are thickened for all points behind their visible faces. Thus, the obstacles are represented by forming rays drawn from the visible faces and moving away from the center of the UGV, generating obstacles expanded radially that try to compensate for the blind areas of the sensor.

The prohibited positions, e.g., singularity area or boundaries, are represented with their exact position in the map without column and radial expansion (orange and blue areas in Fig. 8). The entire group of prohibited locations in the map is thickened with a value of $a_N + T_s$ to compensate the major ellipse semimajor axis a_N and with an extra thickness T_s (see Fig. 9) so the obstacle avoidance has a safety clearance to ensure the vicinity positions will never reach the obstacle.

If any part of the vicinity touches a prohibited location in the map, the obstacle avoidance algorithm iteratively shifts the *cord* to the closest free position and smooth its curvature until the *cord* is valid and optimized as shown in Fig. 9.

After the vicinity is repaired, the *cord* does not have any contact with the obstacle, but the vicinity may intersect with the map obstacle. However, because the map contains expanded obstacles, the vicinity never intersects with the real obstacle leaving T_s length as a safety zone; all the points inside the vicinity are reachable by the end-effector.

To add an extra level of safety, the vicinity is subjected to a quick validation process which is performed by checking scattered random poses within the vicinity. If these poses are not prohibited and achieved with allowed joint angles, the vicinity has a significant probability of being valid and can be used for the planning process.

The last step consists in the generation of a connected graph for end-effector movements. There are many probabilistic approaches for path generation available in the literature. However, in this research work, we choose the Rapidly Exploring

Algorithm 2 Path Planning for End-Effector Movements

```

Input Vicinity( $S_i, N$ ),  $\mathcal{G}_0(V_0)$ 
Output  $\mathcal{G}(V, E)$ 
1: for  $i \leftarrow 1$  to  $N$  do
2:    $Rand\_Poses\_List \leftarrow InitializeSectionRandomPosesList()$ 
3:    $Rand\_Poses\_List \leftarrow RecycleVertices(V_0)$ 
4:   while  $Rand\_Poses\_List.size() < MaxSectionSamples$  do
5:      $RndPose \leftarrow RandomPoseGenerate(a_i, b_i, L_s)$ 
6:      $R\_RndPose \leftarrow Rotate(RndPose, S_i.angles)$ 
7:      $RT\_RndPose \leftarrow Translate(RotatedPose, S_i.center)$ 
8:      $Rand\_Poses\_List \leftarrow append(RT\_RndPose)$ 
9:    $Valid\_Section\_Poses \leftarrow IK\_Engine(Rand\_Poses\_List)$ 
10:  for  $j \leftarrow 1$  to  $Valid\_Section\_Poses.size()$  do
11:     $\mathcal{G}(V, E) \leftarrow append(RRT^*(Valid\_Section\_Poses[j]))$ 
12:  return  $\mathcal{G}(V, E)$ 

```

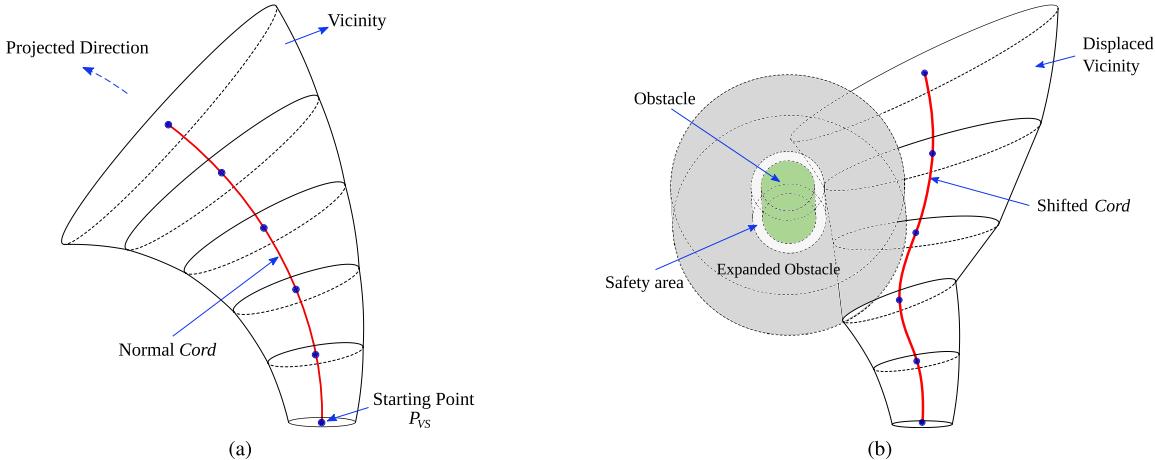


Fig. 9. Vicinies and respective *cords*. (a) First *cord* and vicinity. (b) Shifted *cord* with displaced vicinity caused by the presence of an obstacle.

Algorithm 3 IK Engine

Input Rand_Poses_List

Output ValidPoses

```

1: ValidPoses  $\leftarrow$  InitializeValidPosesList()
2: for M threads do in parallel
3:   Rand_Pose  $\leftarrow$  ExtractFirstElement(Rand_Posess_list)
4:   Joints  $\leftarrow$  InverseKinematics(Rand_Pose)
5:   isValid  $\leftarrow$  Check_Joint_Ranges(Joints)
6:   if isValid then
7:     ValidPoses  $\leftarrow$  Insert(Rand_Pose)
8: return ValidPoses

```

Random Tree Star (RRT*) algorithm proposed by [47], as it is easy to implement and can quickly calculate a proper solution for small volumes.

The graph generation algorithm will connect valid nodes or vertices V with edges E in a hierarchical graph $\mathcal{G}(V, E)$. As shown in Algorithm 2, a set of random poses are filled with recycled vertices from the previous graph and new random poses, to be later evaluated for possible addition in the graph for each vicinity section S_i .

The random poses are calculated by generating quasi-random values around the origin within an ellipse of semiaxes a_i, b_i and L_S , and are rotated and translated with their angles yaw_i, roll_i and pitch_i and center point I_i of their respective ellipses. Thus, the complete set of random poses is within the vicinity, which means that they are generated in a space free of obstacles and constraints. Moreover, there is no need for collision evaluation for each new pose added to the graph.

The next step is to confirm that each pose in the list of random poses is reachable by the end-effector. As the RRT* graph should contain a dense sampling of poses for the vicinity, it is necessary to add several hundreds of valid vertices in every iteration. To speed up the whole process, the arm model and the Jacobian matrix are pre-loaded for parallel calculation in each of the M computational threads of the inverse kinematics (*IK*) *Engine* of Algorithm 3.

Each thread computes the IK and verifies the validity of the joint angles for random poses. Together they can retrieve results quickly and comply with the required path planning and optimization frequency f_B .

Finally, the RRT* algorithm, as explained in [47] inserts each of the valid poses to the graph $\mathcal{G}(V, E)$, and connects its parent and cost with edges. This connected graph with dense sampling is used as a road map for end-effector motions, and with the help of the *selection* function, the best position to locate the end-effector for tracking purposes is calculated.

3) *Selection and Optimization*: This process is designed to work at a high frequency of f_B , providing real-time performance with an optimal end-effector location. As the UAV tracking should be preserved in an environment with static and dynamic obstacles, the end effector's movements should be fast and, if possible, acquire an anticipatory approach to be prepared in case of the risk of losing the line of sight with the aircraft. To achieve this, our approach forces the end-effector to navigate in the planning graph $\mathcal{G}(V, E)$ contained in the vicinity, which is extended toward the main motion. Thus, the end-effector has a higher probability of correctly predicting the UAV relative pose in the short term.

The selection algorithm first uses the P_Q^M pose to calculate the end-effector reactive target pose P_{VS} that locates the UAV marker in the center of the image. This calculation completes the process of Online-PBVS control. To determine how far the end-effector should move for the next step, P_{VS} is used to guarantee a reactive tracking. However, for the anticipatory approach, the velocity and acceleration vectors of the robot platforms are used to determine the general state of the tracking, which is a factor that determines how far the end-effector will travel in the graph to predict the UAV movements.

The positional angular velocities of the UAV $\mathcal{V}_{\text{UAV}}^M = [\Delta x_{\text{UAV}} \Delta y_{\text{UAV}} \Delta z_{\text{UAV}} \Delta \phi_{\text{UAV}} \Delta \theta_{\text{UAV}} \Delta \psi_{\text{UAV}}]^T \in \mathbb{R}^6$ are compared to the velocities of the end-effector $\mathcal{V}_{\text{eff}}^M = [\Delta x_{\text{eff}} \Delta y_{\text{eff}} \Delta z_{\text{eff}} \Delta \phi_{\text{eff}} \Delta \theta_{\text{eff}} \Delta \psi_{\text{eff}}]^T \in \mathbb{R}^6$, both in M frame, to obtain the relative velocity of the tracking $\mathcal{V}_{\text{tr}} \in \mathbb{R}^6$. The velocity values in \mathcal{V}_{tr} are averaged to obtain a single value that represents the quantity of movement expected in the image

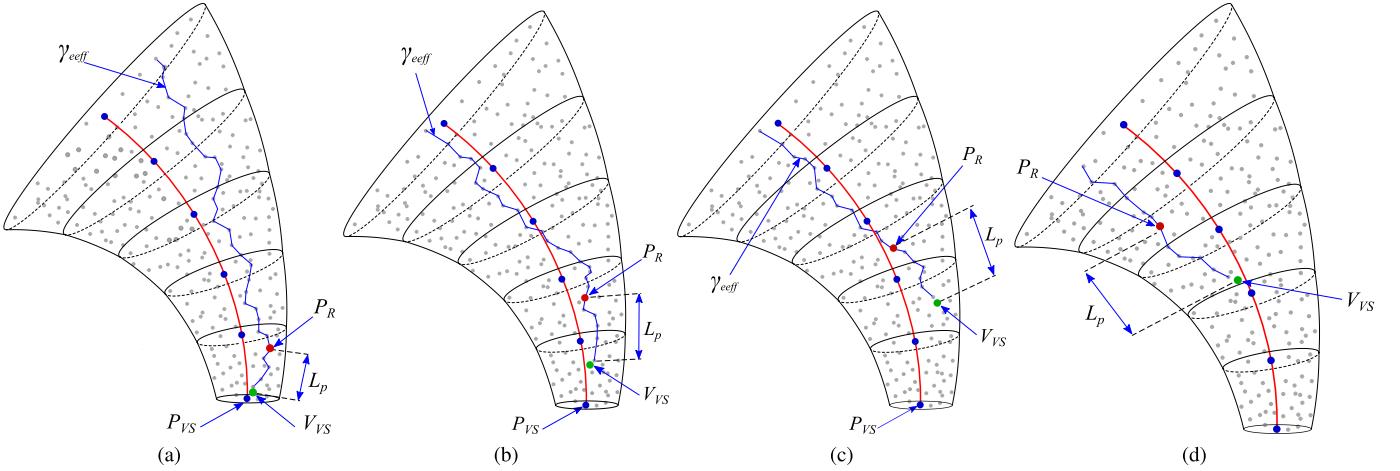


Fig. 10. Vicinies with *cords*, RRT* graph $\mathcal{G}(V, E)$, path γ_{eff} , and interest points. Representation of the path calculation for the advance of the end-effector position. (a) First iteration. (b) Second iteration. (c) Third iteration. (d) Fourth iteration.

of the end-effector camera

$$\mathcal{V}_{\text{tr}} = \overline{\mathcal{V}_{\text{eff}}} + \overline{\mathcal{V}_{\text{UAV}}} \in \text{IR}^6 \quad (9)$$

with

$$\overline{\mathcal{V}_{\text{eff}}} = \frac{K_{\text{pos}}[\Delta x + \Delta y + \Delta z] + K_{\text{ang}}[\Delta \varphi + \Delta \theta + \Delta \psi]}{6}, \\ x, y, z, \varphi, \theta \text{ and } \psi \text{ with eeff subindex} \quad (10)$$

$$\overline{\mathcal{V}_{\text{UAV}}} = \frac{K_{\text{pos}}[\Delta x + \Delta y + \Delta z] + K_{\text{ang}}[\Delta \varphi + \Delta \theta + \Delta \psi]}{6}, \\ x, y, z, \varphi, \theta \text{ and } \psi \text{ with UAV subindex} \quad (11)$$

where the gain for position velocities is double the gain for orientation variances ($K_{\text{pos}} = 2 * K_{\text{ang}}$) giving more weight to the positional differences in the calculation of the tracking state \mathcal{V}_{tr} .

Once a factor for the tracking state is ready, it is possible to derive the length of the path L_p through which the end-effector will traverse. The L_p value is equal or less than L_C and inversely proportional to \mathcal{V}_{tr}

$$L_p = K_{Lp} \frac{L_C}{\mathcal{V}_{\text{tr}}} \quad (12)$$

where K_{Lp} is a constant that transform units and adjust the extension of movement.

The final step is to calculate a connected path γ_{eff} in the RRT* graph $\mathcal{G}(V, E)$ which connects a vertex V_{vs} with a point located at L_p distance as explained in Fig. 10(a). The value of γ_{eff} is updated at f_B frequency, with origin in V_{vs} which is the closest connected vertex to the visual servoing point P_{vs} . Note that V_{vs} is also updated at f_B frequency by the *Merge and Selection* process, which increases the probability of providing a different path root every time. However, as mentioned before, if the direction of motion changes suddenly, the *Planning and Optimization* process will work in a vicinity with unique ellipsoidal section to resume in the next iteration with a complete fresh vicinity.

The final end-effector desired pose for each iteration P_R is the vertex in the path γ_{eff} located at a distance L_p from V_{vs} . This pose is optimized for tracking tasks, anticipating the direction of the main movement, and considering the prohibited areas in the workspace and the state of the robot platform.

Finally, P_R is transformed to joint goal angles q by calculating the IK equations of the arm, and motor control signals are derived by using proportional integrative derivative (PID) controllers for each of the six joints. The motors' commands are given as angle correction values that are later transformed into motor torques.

B. Contact

As indicated previously in Fig. 3, the *Contact Phase* is executed simultaneously with the *Tracking Phase* and starts with the authorization of the human operator and supervises the end-effector movement for a slow vertical approach toward the UAV. Once the end-effector is located at a smaller distance D_c from the UAV, the *docking device* starts securing sequence to attract and anchor the aircraft (refer to Section IV-C). Once the aircraft is properly docked, the *Contact Phase* is finalized.

In this phase, the end-effector is controlled in P_{RC} pose at the same location of P_R with a small change in z value at each iteration

$$P_R = [x_{\text{eff}} \ y_{\text{eff}} \ z_{\text{eff}} \ \varphi_{\text{eff}} \ \theta_{\text{eff}}] \quad (13)$$

$$P_{RC} = [x_{\text{eff}} \ y_{\text{eff}} \ z_{C_{\text{eff}}}(t_c) \ \varphi_{\text{eff}} \ \theta_{\text{eff}}] \quad (14)$$

where

$$z_{C_{\text{eff}}} = z_{\text{eff}} + k_c \ln(t_c + 1.0). \quad (15)$$

Note that $z_{C_{\text{eff}}}$ depends on the duration of the *Contact Phase* t_c , using only the positive values of the logarithm ($t_c + 1.0 >= 1$). Along with the gain k_c it generates a fast approach at the beginning but slower at the end of the *Contact Phase*.

Additionally, due to the UAV features, a new sensory constraint is added to the obstacle map. This constraint is caused by the sensor system mounted in the UAV that is pointing downward, e.g., the sonar. This sensor measures the distance to the ground and produces a cone expanded from the UAV to the ground. If this cone is included in the map, then the end-effector restricts its movements and produces safe docking without entering this area during the complete process.

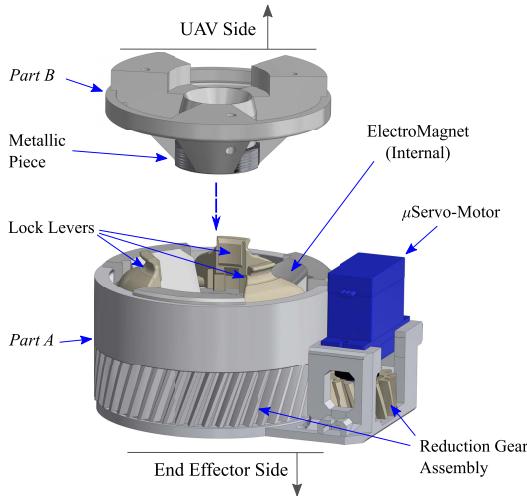


Fig. 11. Main components of the docking device.

If the tracking is abruptly interrupted by external forces, or if the camera view is occluded and motion planning is canceled, the value for $z_{c_{\text{eff}}}$ will decrease quickly to a low position to enhance the camera field of view. This ensures the increased probability of recovering the line of sight with the UAV marker and resumes the normal tracking. For safety, the contact authorization is requested after any cancellation.

After the docking process is finalized, the UAV is securely docked to the tip of the robot manipulator. For robustness and safety, the manipulator stores the aircraft chassis in a secure position for storage and transportation (see the rightmost image of Fig. 3).

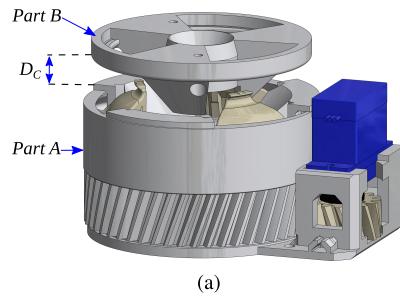
C. Docking Device

The docking device is a set of two complementary parts designed to lock the aircraft safely to the end-effector. As shown in Fig. 11, *Part A* is attached to the tip of the end-effector and contains an electromagnet and a set of levers, that are actuated by the torque of a microservomotor that is previously transformed using geared arrangement.

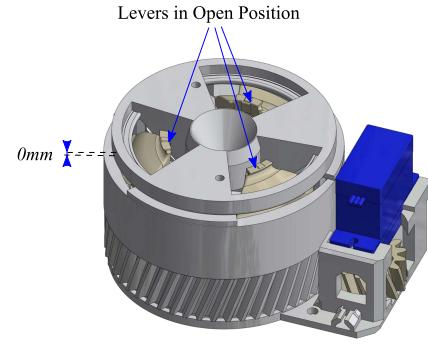
Part B is a lightweight adapter attached at the front-bottom of the UAV without interfering with its sensors. When *Part B* is aligned at a distance less than D_c with the *Part A*, the electromagnet is activated to force an initial contact ($D_c = 0 \text{ mm}$). During this initial contact, the UAV motors are still active because the electromagnetic force is not enough to fix the aircraft, especially if roll and pitch movements arise. To ensure that the aircraft is secured, the micro servo-motor is activated to move the gears and levers to couple the external ring of *Part B* firmly to *Part A*.

After the mechanical locking is achieved, the UAV motors can be turned off, and the robot manipulator is in complete control of the aircraft chassis. Please refer to the sequence in Fig. 12 for a visual explanation.

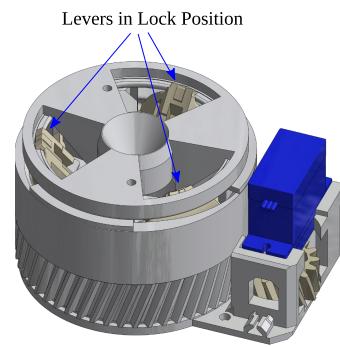
In contrast, when the UAV is taking off from the end-effector after being in the contact state, the docking device quickly releases the *Part B* in the adequate moment when the UAV motors have enough trust to hold the flight.



(a)



(b)



(c)

Fig. 12. Docking sequence. (a) *Part A* approaches *Part B* until D_c separation. (b) Electro-magnet is activated generating contact. (c) μ servo-motor is activated and the gear assembly moves the levers to lock the *Part B*.

The design considers the possible UAV movements and speed; the conical structure minimizes the effects of roll and pitch movements and allows small yaw variations. Moreover, the installed motor actuates the levers with considerable speed to quickly lock and unlock the UAV for a safe docking process.

V. SIMULATION AND EXPERIMENTS

This section presents the heterogeneous robot team with autonomous docking capability tested in simulation and real robots. Exploration strategies are described along with the details of the system that allows achieving the general goal. The results for both experiences are summarized.

The experiments are designed to prove the system's efficiency by measuring the distance traveled by each robot platform and, during tracking and contact phases, sensing the relative position of the end-effector and UAV marker. It is also possible to calculate the UAV's energy consumption during exploration and the economy that represents the docking process in the USAR mission.

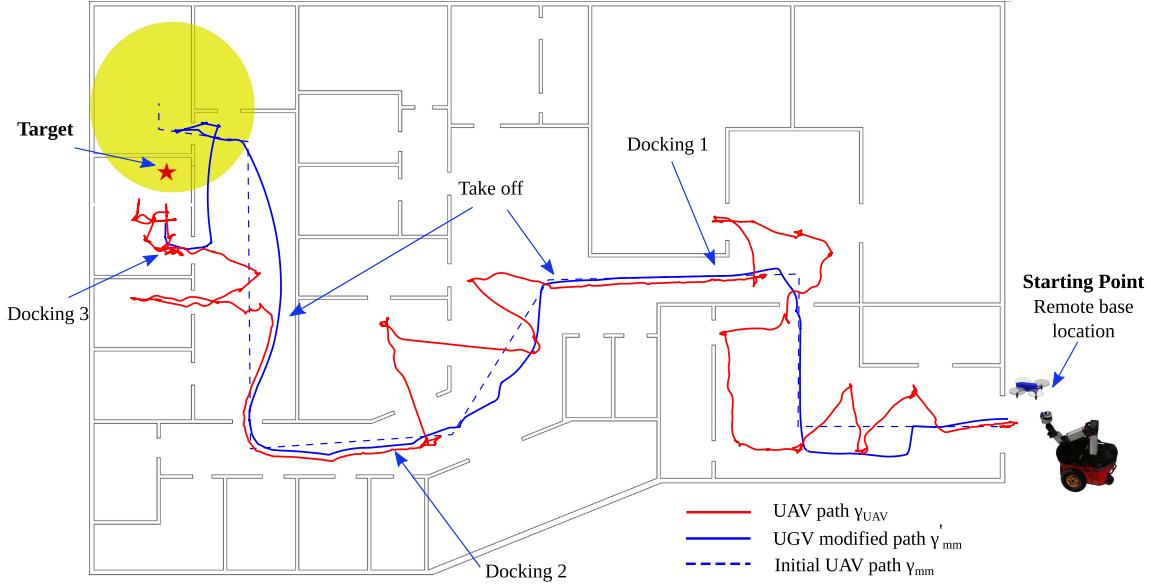


Fig. 13. Map and robot paths for simulation experiment. The yellow circle marks the initial knowledge of the possible location of the *target* and the gray areas are portions of UGV trajectory where UAV was in storage state.

For USAR missions, the indoor environment details are usually unknown by the human operator. There is no guarantee of the availability of a path that leads to the target. Moreover, there is no information about what kind of objects and obstacles are present in the scene. However, the main *target* features are well-known, and in some cases, a building map is available, and observation from the exterior is possible. The human operator can provide an initial notion of the *target* location and guide a preliminary route, as well as the general strategy for exploration.

The trajectory guidance for the UAV and UGV is assisted by the human operator for general exploration and rendezvous. Simultaneously, the *docking* process is executed as a fully autonomous task, taking control of the UAV and robot arm to achieve robust contact.

During the mission, the UAV docking process is requested by the system in the following conditions.

- 1) When battery level falls below 20%.
- 2) When the area is explored, and the current area has been surveyed with enough density in measurements.
- 3) When the operator requests the docking.
- 4) When the robot team is about to pass through a difficult area for aerial navigation.

If the conditions are suitable for UAV deployment, the human operator authorizes the UAV flight for aerial exploration to complement the ground exploration.

The experiments for simulation and real environment described below are also explained with video resources. Readers are strongly encouraged to refer to the videos provided with this manuscript that can be accessed to understand the experiments better. Table I contains the direct link to each video.

A. Simulation

We tested our approach by using a simulation experiment replicating the available robots. The platform used

TABLE I
VIDEO LINKS

Video	Experiment	URL
1	Simulated environment	https://bit.ly/Docking_Simulation
2	Real environment	https://bit.ly/Docking_Real_Env

was the Robot Operating System (ROS, Kinetic Kame) and Gazebo simulator (7.0.0) on Linux distribution (Ubuntu 16.04.06 Xenial Xerus). All computations were performed on a desktop computer with 16 GB of RAM and a CPU with a clock speed that reaches 4.2 GHz.

The robot team is deployed in an unknown environment with a mission to search for a target. The building map and robot paths are shown in Fig. 13. The approximate location of the *target* marked with the yellow circle is known. With the human operator experience, a basic path γ_{mm} is traced as a guide for ground navigation (dotted blue line).

The mobile manipulator follows the initial path γ_{mm} in direction to the possible *target* location. During the mission, γ_{mm} is modified with the presence of obstacles and structures detected by the ground and aerial sensors to finally make the UGV robot to travel across the modified γ'_{mm} path (blue solid line). Path smoothing is done to smooth out sharp corners [48]. If any of the two robots find the target, the UGV is redirected to follow the new path γ'_{mm} that leads to the new goal.

Meanwhile, the aerial robot performs a series of movements to explore different building areas. It tries to follow a path γ_{UAV} that allows covering the available zones (solid red line). If possible, every D_e distance, the UAV stops and performs yaw variations to make detailed measurements of the scene in a static position.

If at any moment, the UAV should be farther than a radius distance D_{team} measured from the ground robot, which is a safety measure considering that D_{team} can be traveled and a docking process can be executed with less than 20% of UAV energy.

TABLE II
SIMULATION TIMES

Process	Time [s]
Docking 1	22.2236
Storage 1	62.254
Docking 2	26.2709
Storage 2	88.7853
Total mission time	723.25

TABLE III
SIMULATION DATA

	Mobile Manipulator	UAV
Distance traveled [m]	68.0523	112.8383
Operation time [s]	723.25	572.1461
Energy consumed [mAh]	-	986.2606
Energy recovered [mAh]	-	185.6458

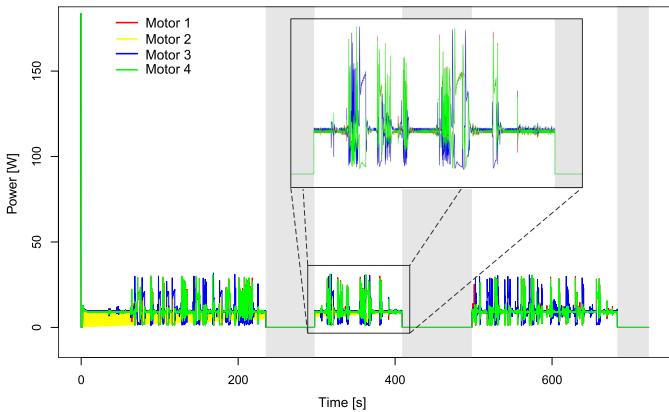


Fig. 14. Power used by the UAV motors during the simulation experiment. The gray areas represent storage processes.

The UAV exploration continues until the target is discovered, or until the system or the operator requests a *docking*. In either situation, the UAV exploration is canceled and, it proceeds to move to a meeting point where it is detected by the end-effector camera to initiate the *docking* process.

In the simulation experiment, the operator requested two *docking* processes to protect the aircraft from passing through a corridor and a door frame. The portions of the UAV path are attached to the mobile manipulator and are represented with gray shades in Fig. 13. The total time for contact and distance traveled by the robots are summarized in Tables II and III, respectively. The video of the simulation experiment can be accessed from Table I.

The power information of each aircraft motors in simulation is represented in Fig. 14. With this data, the accumulated energy consumption during the mission was calculated to draw the battery drainage process, as is shown in Fig. 15. Note that, during the time the UAV is attached to the end-effector (gray areas in the figures), the energy consumed by the aircraft is zero, and the battery level remains constant. A charging system to charge the UAV battery during contact can restore some of the energy to the aircraft resulting in more autonomy to contribute to the mission.

The decrease in the percentage of energy along with the mission is shown in Fig. 15. The blue line represents the

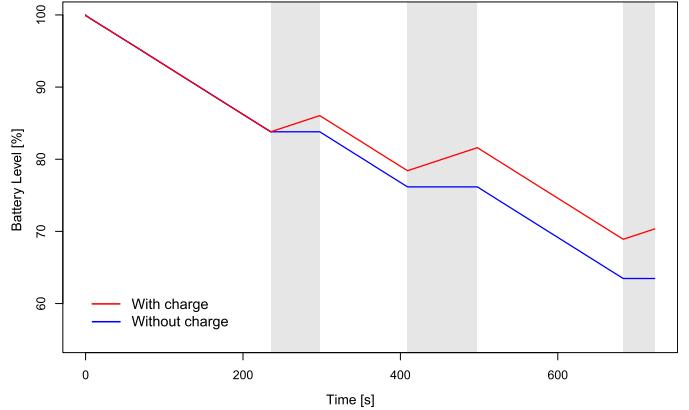


Fig. 15. UAV battery levels during the simulation. The gray areas represent storage processes.

process without charging capacity, and the red line includes an energy recovery process installed with a standard charger of 12.6 V and 3.5 A. The installed 2700 mAh battery on the aircraft was reduced to 63.47% at the end of the mission without the charging process and to 70.34% with the charging process. Using the proposed docking system, the UAV can recover near 185.64 mAh of energy (6.87% of the battery). This saved energy can potentially give a considerable amount of aircraft flight time that can be utilized for detailed inspection and other critical tasks, providing crucial measurements that increase the robot team's usefulness.

B. Experiments in a Real Environment

An overview of the experiment, along with a description of the robot platforms and sensors used, are described in this section. The mission executed is similar to the one in the simulation case but with a different environment that exhibits less debris and smaller obstacles (see Video 2 of Table I).

1) *Robotic Platforms*: The UGV used in the experiment is based on the Pioneer 3-DX robotic platform by Adept Mobilerobots. It implements differential drive motion with 10 Kg payload capacity and 7 h of runtime and is equipped with a 2-D range light detection and ranging (LiDAR) (Sick LMS111-10100) and an RGB-D camera (Realsense D435) facing the front of the platform.

A six DoF robot arm was designed from scratch with high precision Dynamixel motors (Pro and MX series) as rotational actuators for each joint. The first links of the robot arm were designed with great longitude; this conception provides a broader vertical range, essential for the end-effector approach in *Contact Phase*.

The prototype of *Docking Device* described in Section IV-C is attached at the 6th joint of the manipulator (see Fig. 16). It is fabricated with 3-D printed parts of Acrylonitrile Butadiene Styrene (ABS) material and assembled along with a standard micro servo-motor (positional rotation) and a small size electromagnet (12 V–250 N). Also, the end-effector camera (playstation (PS) Eye 640 × 480 pixels at 60 Hz) is mounted in the arrangement of the 6th joint with an offset of $O_C = [0.045, 0.01, 0.03]$ m.

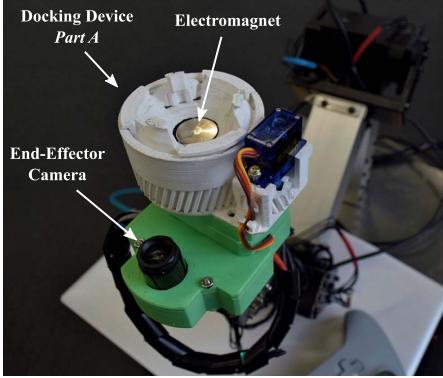


Fig. 16. End-effector, the green part corresponds to the end-effector base and camera arrange, the white portion is the Part A of the *Docking Device*.

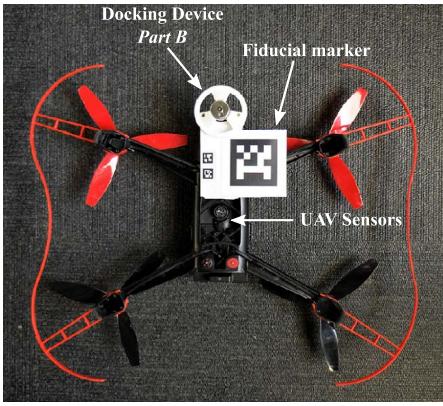


Fig. 17. Bottom view of the UAV.

The aerial robot platform is based on a commercial quadrotor, Bebop 2, by Parrot Company. It has an autonomy of 20 min (battery of 2700 mAh). It is equipped with a stabilized full high-definition (HD) camera for detailed exploration and a set of sensors for its stabilization system (IMU, barometer, accelerometer, gyroscope, vertical camera for optical flow, and ultrasound sensor). As shown in Fig. 17, the Part B of the *Docking Device* was installed in the UAV front bottom without interfering with its sensors, it was also 3-D printed with ABS material, and it contained metallic material in the center. The UAV fiducial marker was also installed at the bottom of the UAV, designed as a bundle arrangement of AprilTags [49].

For safety and stability, the aircraft speed was limited to 4 m/s in lateral displacements and 1 m/s for vertical variations, the UGV longitudinal speed to 2 m/s, and each of the robot arm joints to a maximum of 15 rpm. During the docking process, the UGV speed was limited to 0.5 m/s to avoid disturbances and to perform the docking process along with a shorter distance.

The complete robot team platforms used in the experiment are shown in Fig. 2.

2) *Architecture and Network:* The architecture of the robot system is centralized in the mobile robot manipulator. This means that a significant part of the computation process is realized by the onboard UGV computer, while the UAV computer systems are in charge only of the stabilization for its movements and self-emergency monitoring.

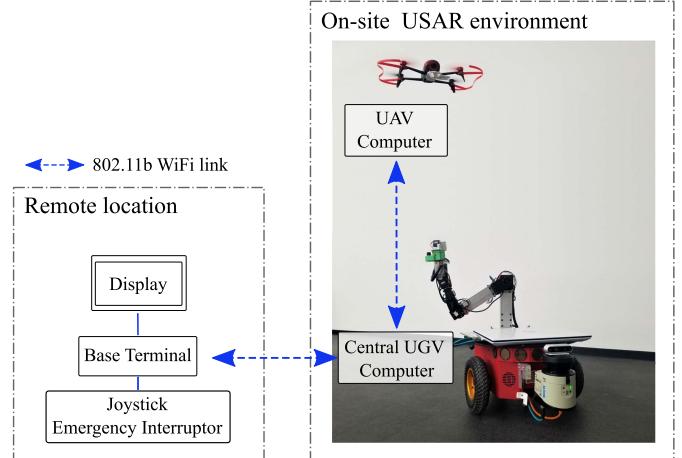


Fig. 18. Communication network between robot team and operator.

There are only supervision and teleoperation tasks available on the remote operator side, and there is no sharing of processing power. The visual information comes from the robot team, and commands are sent back. The information is communicated through Wi-Fi 802.11g channels between the robot team and the operator base and between robot platforms. Refer to Fig. 18 for a breakdown of the communication and architecture design.

3) *Experiment Design:* The robot team was deployed at the entrance of the university architecture building. The building contains rooms, long corridors, and an empty hall. The robots start exploration separately, and similar to the simulation case, the goal of the robot team is to find a known target, knowing roughly its possible location in the building.

It is assumed that the building has a particular risk or hazardous elements. Therefore, the human operator was required to supervise the robot platforms from a remote location. The aerial and ground robots were moving autonomously along a series of set-points given continuously by the operator. If necessary, it was possible to take total control of the movements of both the platforms.

4) *Results:* During the experiment, the robots followed a trajectory in direction to the given set-point. The human operator selected each set-point continuously depending on the target's direction and the zones to explore. In the UAV case, the human operator often needed to take control because it was necessary to pass over frames, avoid walls, direct the camera to the point of interest, and correct trajectories due to bad odometry. Similar attention was required for the UGV movements, especially for turning points, complicated obstacles, and path corrections.

The robot team started separately and during the exploration executed 2 autonomous docking processes and one deployment. The ordered outline of the experiment is given below.

- 1) The robot platforms started exploration separately. The UAV tried to register as many number of areas as possible while the UGV went in the direction of the potential target location γ_{mn} .
- 2) The UAV detected a door frame with some obstacles that represent danger, and the human operator also detected

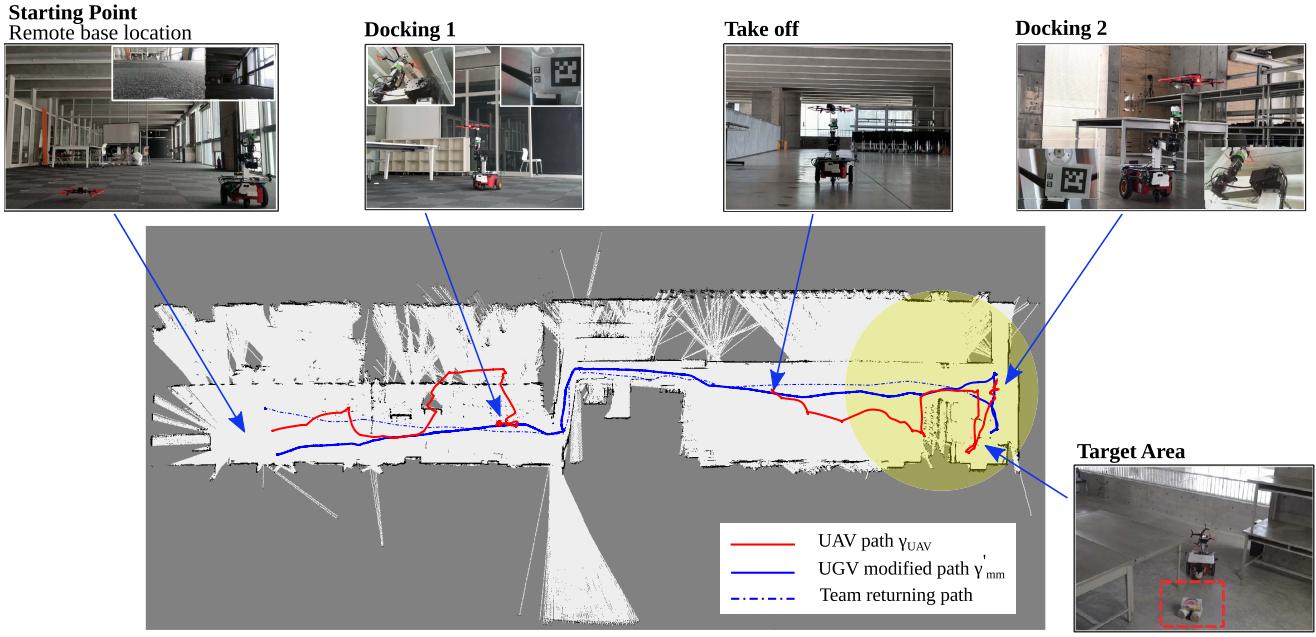


Fig. 19. Robot team paths for real robot experiment. The yellow area represents the initial operator knowledge about the location of the *target*.

TABLE IV
EXPERIMENT TIMES

Task	Time [s]
Docking 1	24.76
Storage 1	141.6
Docking 2	21.72
Storage 2	369.9
Complete mission	1110

TABLE V
EXPERIMENT DATA

	Mobile Manipulator	UAV
Distance travelled [m]	170.076	98.53
Operation time [s]	1110.0	599.25
Energy consumed [mAh]	-	1840
Energy saved [mAh]	-	1368

that the door leads to a corridor. Consequently, the first docking was requested.

- 3) Autonomous docking process executed. And the ground exploration continued.
- 4) The human operator found a large open space visible from the UGV camera and requested UAV deployment.
- 5) UAV deployed and heterogeneous exploration started.
- 6) UAV found the target and alerted the human operator.
- 7) UGV changed trajectory γ'_{mm} in direction to the target.
- 8) The human operator decided that the UAV exploration is no longer necessary and requested a second docking.
- 9) Autonomous docking process was executed.
- 10) The UGV recognized the *target* and returned to the base at the entrance of the building.

Fig. 19 shows the grid map of the environment built with the front LiDAR sensors of the UGV. The figure also displays the trajectories of the robot platforms during the experiment. In this case, the yellow area represents the possible location of the *target* where the UAV tried to explore thoroughly.

After the *target* was recognized, the robot team returned to the base with the UAV stored in a safe position. Aerial exploration was not necessary, as the team knows a path is available, and the ground sensors with the human supervision were enough to get the team back to the initial point.

Table IV summarizes the time required for the complete experiment and each docking and storage processes.

Table V shows the total distance traveled by the platforms and the energy used by the UAV with the economized energy achieved thanks to the docking process.

VI. DISCUSSION

The simulation presented in this work demonstrated the ability of the robot team to follow an original path and perform modifications according to the environmental variables. Once the docking processes were requested, the UAV reached the rendezvous points before the mobile manipulator, and the autonomous docking was executed with robustness. By repeating the docking process many times during the experiment, it demonstrated the robustness and repeatability of the approach and its potential to facilitate the exploration of large unknown environments and enable middle and long term missions.

For the experiment in a real scenario, the complete mission was performed successfully without damaging the robots. The team safely returned to the base with the stored UAV and with enough battery available. Furthermore, exploring hard to reach regions in the map was possible using the UAV such as the railings and the narrow shafts.

Fig. 20 shows the position trajectories of the marker and end-effector during the two docking processes; all the data are in manipulator coordinates. Further, in Fig. 20(a), both the docking processes were performed correctly, and the

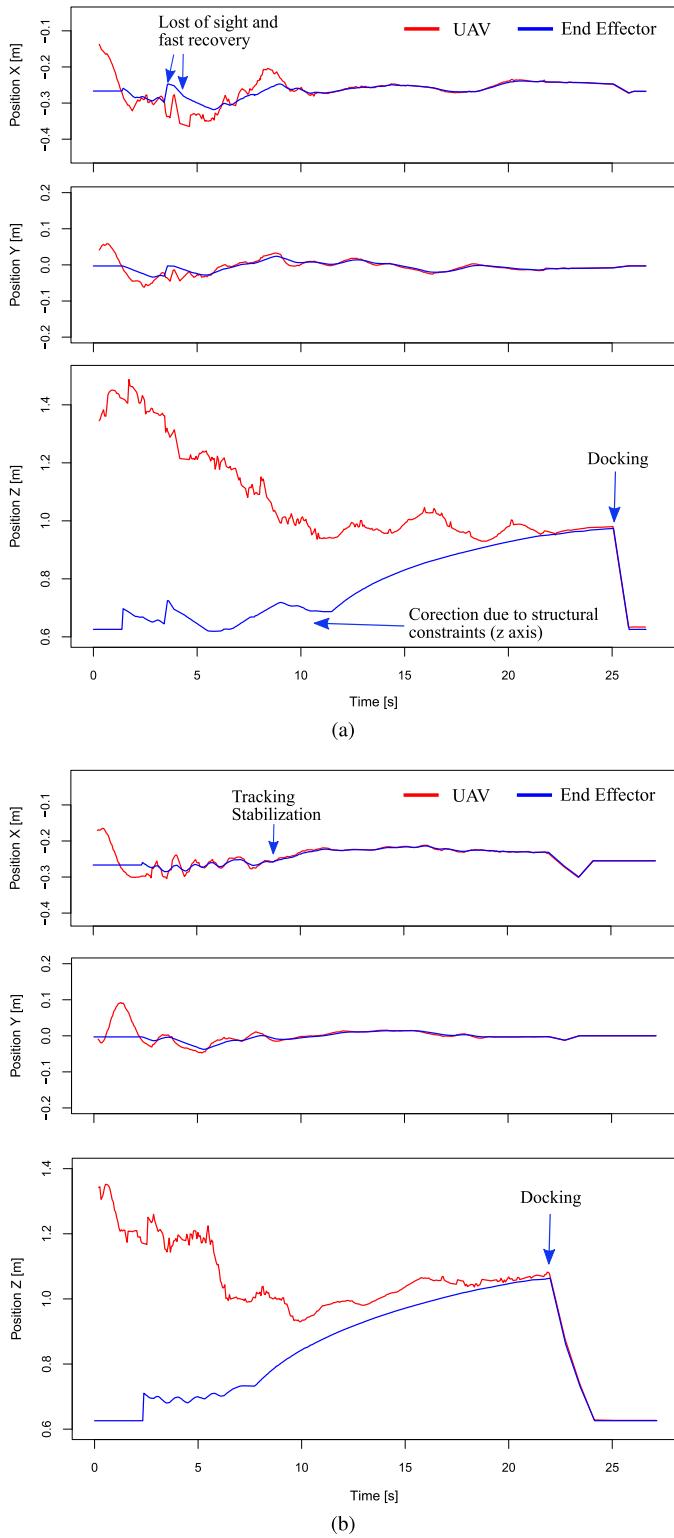


Fig. 20. UAV marker (red) and end-effector (blue) positions during docking process for real robots. (a) First docking. (b) Last docking process.

end-effector tracked the UAV marker. Although the line of sight was lost in some instant of the first docking, the tracking was quickly recovered and stabilized, and stable contact processes were achieved, as can be seen from the plots of the UAV and end-effector trajectories.

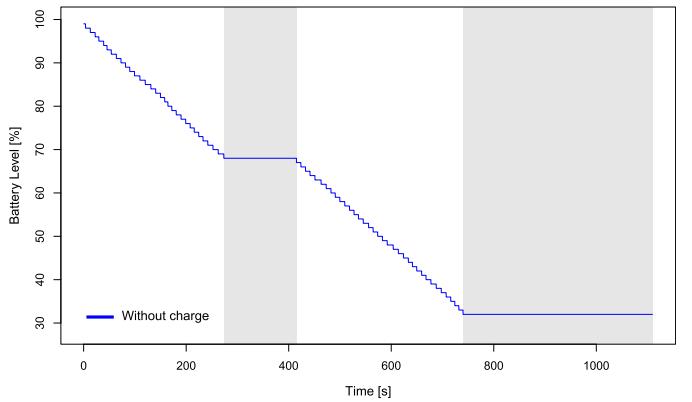


Fig. 21. UAV battery level during the experiment. The gray areas represent storage processes.

During the autonomous UAV docking processes, the system calculated viable end-effector locations to preserve the line of sight with the UAV marker, and performed a smooth motion tracking. In some instants, due to structural constraints, the system modified the local area shape to supply viable locations [see the z -direction of Fig. 20(a)].

The battery drainage during the experiment for the UAV was around 34%. However, as expected, the energy consumed by the aircraft was almost none during the storage state, as indicated by the gray zones in the plot (see Fig. 21). If we extrapolate the energy consumption, assuming no docking available, the energy consumed would be 3200 mAh approximately, which exceeds the UAV capacity of 2700 mAh.

During the experiment in the real scenario, the docking processes were performed with high accuracy and robustness. The UAV motion tracking helped maintain visual contact between the end-effector camera and the UAV marker, predicting the movements and staying one step forward while considering structural constraints. As shown in the comparisons of Fig. 20, the system calculated the end-effector locations required to follow and anticipate the movements while counteracting the delays. When the pose requests were close to structural constraints, variations in the z -axis were made to elude collision with the self-structure or avoid falling into singular positions.

The experiment proved the advantages of our system for robustness and energy management. The robot arm followed the UAV movements fast and smooth enough during tracking and final approach, compensating for the UAV's limitations. The range for available docking was also increased because the arm workspace was larger than the UGV top surface. During the *Contact Phase*, the docking device locked the aircraft with enough strength for vertical storage, allowing protection and energy-saving without affecting the ground navigation.

With our unique docking and storage function, the safety of the system is guaranteed. The saved energy allows us to do a heterogeneous exploration in hard to traverse areas of the environment that would be impossible without a docking approach. Besides, if the docking process is combined with a fast-charging device, the UAV can recover energy during storage and further increase the robot team's utility.

VII. CONCLUSION

This article developed the novel idea of VTOL-UAV autonomous docking with a mobile manipulator for air-ground multirobot systems. This approach capacitates the robot team with a mechanism to measure precisely the UAV movements for tracking and approximation and to safely link the aircraft chassis to the arm end-effector for temporary storage. The system allowed the air-ground robot team to cooperate and interact thoroughly, increasing effectiveness, and benefiting all platforms. During the USAR mission, the aerial robot can be deployed at the right moment to provide information from its useful point of view to the ground platform for better exploration. When the UAV needs protection or to save (and potentially resupply) energy, the UGV can store and transport the aircraft through unsafe areas.

The autonomous docking process with a mobile manipulator was explained in two steps: the stable UAV motion tracking and the contact process. The controller that governs the aircraft and the robot arm was also detailed, along with the reactive, predictive, and optimization methods that incorporate visual information with robotic and environmental constraints in the control loop for a robust tracking and posterior approach. Additionally, the physical contact between robots was analyzed, and a solution for the robust design of a docking device that facilitated the final contact and safe storage was presented.

With the inclusion of the robot arm for the docking process, enough redundancy and range were added to prevent failure due to environmental variations and UAV limitations. Therefore, providing a repeatable UAV docking process that increased the air-ground robot team's advantages for USAR missions.

The system was tested with simulations and real scenario experiments, proving that the docking process can be executed safely one or more times during the mission. In the tests, the robots executed exploration to find a target in the depth of a building, performing heterogeneous surveys in interest points, storing and deploying the aircraft when convenient to finally make a return of both platforms to a safe location.

For the continuation of this research work, it is convenient to test the *UAV Motion Tracking* approach for dynamic obstacles and external perturbations, and more extended missions and with multiple agents. The docking system can be effectively incorporated into different types of air-ground robot teams, and it can complement varied exploration strategies and cooperative technologies.

REFERENCES

- [1] B. B. Reed, R. C. Smith, B. J. Naasz, J. F. Pellegrino, and C. E. Bacon, "The Restore-L servicing mission," in *Proc. AIAA SPACE*, Sep. 2016, p. 5478.
- [2] M. Rossi and D. Brunelli, "Autonomous gas detection and mapping with unmanned aerial vehicles," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 4, pp. 765–775, Apr. 2016.
- [3] T. Arai *et al.*, "Advances in multi-robot systems," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 655–661, Oct. 2002.
- [4] A. Ravankar, A. Ravankar, Y. Kobayashi, and T. Emara, "Symbiotic navigation in multi-robot systems with remote obstacle knowledge sharing," *Sensors*, vol. 17, no. 7, p. 1581, Jul. 2017.
- [5] S. Balakirsky *et al.*, "Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from RoboCup rescue," *J. Field Robot.*, vol. 24, nos. 11–12, pp. 943–967, 2007.
- [6] M. Garzón, J. Valente, D. Zapata, and A. Barrientos, "An aerial-ground robotic system for navigation and obstacle mapping in large outdoor areas," *Sensors*, vol. 13, no. 1, pp. 1247–1267, Jan. 2013.
- [7] J. Delmerico *et al.*, "The current state and future outlook of rescue robotics," *J. Field Robot.*, vol. 36, no. 7, pp. 1171–1191, Oct. 2019.
- [8] D. Falanga, A. Zanchettin, A. Simovic, J. Delmerico, and D. Scaramuzza, "Vision-based autonomous quadrotor landing on a moving platform," in *Proc. IEEE Int. Symp. Saf., Secur. Rescue Robot. (SSRR)*, Oct. 2017, pp. 200–207.
- [9] A. Ravankar, A. A. Ravankar, Y. Kobayashi, and T. Emara, "On a bio-inspired hybrid pheromone signalling for efficient map exploration of multiple mobile service robots," *Artif. Life Robot.*, vol. 21, no. 2, pp. 221–231, Jun. 2016.
- [10] A. Ravankar, A. Ravankar, Y. Kobayashi, Y. Hoshino, C.-C. Peng, and M. Watanabe, "Hitchhiking based symbiotic multi-robot navigation in sensor networks," *Robotics*, vol. 7, no. 3, p. 37, Jul. 2018.
- [11] A. Ravankar, A. A. Ravankar, Y. Kobayashi, and Y. Kobayashi, "On sharing spatial data with uncertainty integration amongst multiple robots having different maps," *Appl. Sci.*, vol. 9, no. 13, p. 2753, Jul. 2019.
- [12] Y. Yue *et al.*, "A multilevel fusion system for multirobot 3-D mapping using heterogeneous sensors," *IEEE Syst. J.*, vol. 14, no. 1, pp. 1341–1352, Mar. 2020.
- [13] P. Kim, L. C. Price, J. Park, and Y. K. Cho, "UAV-UGV cooperative 3D environmental mapping," in *Proc. Comput. Civil Eng.*, Jun. 2019, pp. 384–392.
- [14] J. Delmerico, E. Mueggler, J. Nitsch, and D. Scaramuzza, "Active autonomous aerial exploration for ground robot path planning," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 664–671, Apr. 2017.
- [15] J. H. Kim, J.-W. Kwon, and J. Seo, "Multi-UAV-based stereo vision system without GPS for ground obstacle mapping to assist path planning of UGV," *Electron. Lett.*, vol. 50, no. 20, pp. 1431–1432, Sep. 2014.
- [16] H. Qin *et al.*, "Autonomous exploration and mapping system using heterogeneous UAVs and UGVs in GPS-denied environments," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1339–1350, Feb. 2019.
- [17] L. Rosa, M. Cognetti, A. Nicastro, P. Alvarez, and G. Oriolo, "Multi-task cooperative control in a heterogeneous ground-air robot team," *IFAC-PapersOnLine*, vol. 48, no. 5, pp. 53–58, 2015.
- [18] M. E. Stieber, M. McKay, G. Vukovich, and E. Petriu, "Vision-based sensing and control for space robotics applications," *IEEE Trans. Instrum. Meas.*, vol. 48, no. 4, pp. 807–812, Aug. 1999.
- [19] A. Ravankar *et al.*, "Multi-robot path planning for smart access of distributed charging points in map," in *Artificial Life and Robotics*. Japan: Springer, 2020, pp. 1–9, doi: [10.1007/s10015-020-00612-8](https://doi.org/10.1007/s10015-020-00612-8).
- [20] X. Chu Ding, A. R. Rahmani, and M. Egerstedt, "Multi-UAV convoy protection: An optimal approach to path planning and coordination," *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 256–268, Apr. 2010.
- [21] H. G. Tanner and D. K. Christodoulakis, "Decentralized cooperative control of heterogeneous vehicle groups," *Robot. Auto. Syst.*, vol. 55, no. 11, pp. 811–823, Nov. 2007.
- [22] J. Gomez-Avila, C. Lopez-Franco, A. Y. Alanis, N. Arana-Daniel, and M. Lopez-Franco, "Ground vehicle tracking with a quadrotor using image based visual servoing," *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 344–349, 2018.
- [23] H. G. Tanner, "Switched UAV-UGV cooperation scheme for target detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 3457–3462.
- [24] P. Maini and P. B. Sujit, "On cooperation between a fuel constrained UAV and a refueling UGV for large scale mapping applications," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2015, pp. 1370–1377.
- [25] H. Duan and Q. Zhang, "Visual measurement in simulation environment for vision-based UAV autonomous aerial refueling," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 9, pp. 2468–2480, Sep. 2015.
- [26] F. Ropero, P. Muñoz, and M. D. R-Moreno, "TERRA: A path planning algorithm for cooperative UGV-UAV exploration," *Eng. Appl. Artif. Intell.*, vol. 78, pp. 260–272, Feb. 2019.
- [27] B. Arbanas *et al.*, "Aerial-ground robotic system for autonomous delivery tasks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 5463–5468.
- [28] T. Merz, S. Duranti, and G. Conte, "Autonomous landing of an unmanned helicopter based on vision and inertial sensing," in *Experimental Robotics IX*. Berlin, Germany: Springer, 2006, pp. 343–352.
- [29] B. Miller, A. Miller, A. Popov, and K. Stepanyan, "UAV landing based on the optical flow videonavigation," *Sensors*, vol. 19, no. 6, p. 1351, Mar. 2019.

- [30] K. Y. Scheper and G. C. de Croon, "Evolution of robust high speed optical-flow-based landing for autonomous MAVs," *Robot. Auton. Syst.*, vol. 124, Feb. 2020, Art. no. 103380.
- [31] Y. Bi and H. Duan, "Implementation of autonomous visual tracking and landing for a low-cost quadrotor," *Optik-Int. J. Light Electron Opt.*, vol. 124, no. 18, pp. 3296–3300, Sep. 2013.
- [32] K. E. Wenzel, A. Masselli, and A. Zell, "Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle," *J. Intell. Robotic Syst.*, vol. 61, nos. 1–4, pp. 221–238, Jan. 2011.
- [33] J. Roldán, P. García-Aunon, M. Garzón, J. de León, J. del Cerro, and A. Barrientos, "Heterogeneous multi-robot system for mapping environmental variables of greenhouses," *Sensors*, vol. 16, no. 7, p. 1018, Jul. 2016.
- [34] P. R. Palafox, M. Garzón, J. Valente, J. J. Roldán, and A. Barrientos, "Robust visual-aided autonomous takeoff, tracking, and landing of a small UAV on a moving landing platform for life-long operation," *Appl. Sci.*, vol. 9, no. 13, p. 2661, Jun. 2019.
- [35] T. Petrovic, T. Haus, B. Arbanas, M. Orsag, and S. Bogdan, "Can UAV and UGV be best buddies? Towards heterogeneous aerial-ground cooperative robot system for complex aerial manipulation tasks," in *Proc. 12th Int. Conf. Inform. Control, Automat. Robot. (ICINCO)*, vol. 1, Jul. 2015, pp. 238–245.
- [36] B. J. Emran and H. Najjaran, "A review of quadrotor: An underactuated mechanical system," *Annu. Rev. Control*, vol. 46, pp. 165–180, Jan. 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1367578818300932>, doi: [10.1016/j.arcontrol.2018.10.009](https://doi.org/10.1016/j.arcontrol.2018.10.009).
- [37] P. Vlantis, P. Marantos, C. P. Bechlioulis, and K. J. Kyriakopoulos, "Quadrotor landing on an inclined platform of a moving ground vehicle," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 2202–2207.
- [38] S. A. Conyers, N. I. Vitzilaios, M. J. Rutherford, and K. P. Valavanis, "A mobile self-leveling landing platform for VTOL UAVs," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 815–822.
- [39] A. Campos, J. Quintero, R. Saltarén, M. Ferre, and R. Aracil, "An active helideck testbed for floating structures based on a stewart-gough platform," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 3705–3710.
- [40] Y. Huang, Z. Zheng, L. Sun, and B. Zhu, "Disturbance observer-based saturated control for a quadrotor landing on a vessel," in *Proc. 13th IEEE Int. Conf. Control Automat. (ICCA)*, Jul. 2017, pp. 419–424.
- [41] V. Lippiello, R. Mebarki, and F. Ruggiero, "Visual coordinated landing of a UAV on a mobile robot manipulator," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Oct. 2013, pp. 1–7.
- [42] E. Narváez, A. A. Ravankar, A. Ravankar, Y. Kobayashi, and T. Emaru, "Vision based autonomous docking of VTOL UAV using a mobile robot manipulator," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Dec. 2017, pp. 157–163.
- [43] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 33, no. 3, pp. 367–385, Jun. 2003.
- [44] K. Nagatani *et al.*, "Emergency response to the nuclear accident at the Fukushima Daiichi nuclear power plants using mobile rescue robots," *J. Field Robot.*, vol. 30, no. 1, pp. 44–63, Jan. 2013.
- [45] A. C. Sanderson and L. E. Weiss, "Adaptive visual servo control of robots," in *Robot Vision*. Berlin, Germany: Springer, 1983, pp. 107–116, doi: [10.1007/978-3-662-09771-7_7](https://doi.org/10.1007/978-3-662-09771-7_7).
- [46] S. C. Chapra *et al.*, *Numerical Methods for Engineers*. Boston, MA, USA: McGraw-Hill, 2010.
- [47] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
- [48] A. Ravankar, A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, "Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges," *Sensors*, vol. 18, no. 9, p. 3170, Sep. 2018.
- [49] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4193–4198.



Eduardo Narváez received the B.S. degree in electronic sciences from Nariño University, Pasto, Colombia, in 2014, and the M.S. degree in robotics and automation from the Technical University of Madrid, Madrid, Spain, in 2016. He is currently pursuing the Ph.D. degree with the Laboratory of Robotics and Dynamics, Graduate School of Engineering, Hokkaido University, Sapporo, Japan.

His research interests include visual sensory systems, cooperative robotics, robot navigation, and artificial intelligence.



Ankit A. Ravankar received the M.S. and Ph.D. degrees from Hokkaido University, Sapporo, Japan, in 2012 and 2015, respectively.

Since 2016, he has been an Assistant Professor with the Faculty of Engineering, Department of Human Mechanical Systems and Design Engineering. His research interests include mobile robot navigation, simultaneous localization and mapping (SLAM), computer vision, multirobot systems, artificial intelligence, and machine learning.

Dr. Ravankar was a recipient of the Ministry of Education, Culture, Sports, Science and Technology-Japan (MEXT) Scholarship by the Government of Japan for his graduate studies.



Abhijeet Ravankar received the M.S. and Ph.D. degrees from the University of Aizu, Aizuwakamatsu, Japan, and Hokkaido University, Sapporo, Japan, in 2011 and 2017, respectively.

He worked with Panasonic Corporations, Osaka, Japan, briefly from 2011 to 2014 as a Computer Vision Engineer. He is currently working as an Assistant Professor with the Department of Mechanical Engineering, Kitami Institute of Technology, Kitami, Japan. His research interests include robot navigation, multirobot systems, computer vision, and artificial intelligence.



Takanori Emaru (Member, IEEE) received the M.E. and Ph.D. degrees in electrical engineering from Hokkaido University, Sapporo, Japan, in 1998 and 2002, respectively.

He was a Research Fellow of the Japan Society for the Promotion of Science, University of Electro-Communications, Chofu, Japan, from 2004 to 2006. He was an Assistant Professor with Osaka Electro-Communication University, Neyagawa, Japan, from 2006 to 2007. He is currently an Associate Professor with Hokkaido University.

His research interests include the areas of robotics, navigation, sensor, and nonlinear signal processing.



Yukinori Kobayashi (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in mechanical engineering from Hokkaido University, Sapporo, Japan, in 1981, 1983, and 1986, respectively.

He is currently the President with the National Institute of Technology, Tomakomai College, Tomakomai, Japan, and a Visiting Professor with Hokkaido University. His research interests include vibration control of flexible structures, control problem of robots having flexibility, path planning and navigation of mobile robots, vibration analysis, and nonlinear vibrations of continuous systems.