Mixed-Initiative Planning and Execution for Multiple Drones in Search and Rescue Missions

Giuseppe Bevacqua, Jonathan Cacace, Alberto Finzi, Vincenzo Lippiello
DIETI, Università degli Studi di Napoli Federico II
via Claudio 21, 80125, Napoli, Italy
{g.bevacqua, jonathan.cacace, alberto.finzi, lippiell}@unina.it

Abstract

We present a mixed-initiative planning and execution system for human multi-drones interaction during search and rescue missions. The proposed system should allow a single operator to supervise and orchestrate the operations of a set of UAVs by means of a natural multimodal communication. In particular, we consider the task of searching for missing persons in a real-world alpine scenario. In this context, we assume that the human operator is an alpine rescuer, involved in the scene and co-located with the drones, hence not fully dedicated to the robotic platforms, but only able to provide sparse and sketchy interventions. This scenario requires a framework that supports adjustable autonomy, from explicit teleoperation to a complete autonomy, and an effective and natural mixed-initiative interaction between the human and the robotic team. In this paper, we illustrate the domain and the overall framework discussing the system at work in a simulated case study.

Introduction

We present a mixed-initiative system for multiple drones suitable for search and rescue activities in a real-world alpine scenario. This work is framed within the SHERPA project [She, 2013; Marconi et al., 2012]. Differently from typical human-multidrones interaction scenarios [Cummings et al., 2007; Ollero et al., 2005; Landén, Heintz, and Doherty, 2012; Bitton and Goldberg, 2008; Malasky et al., 2005], in this work we assume a human operator that is co-located with the robots and not fully dedicated to their supervision and control. In this context, the human level of involvement in supporting the robots behavior is not ensured: as a member of the rescue team involved in the search and rescue activities, the human operator might be capable to directly operate the robots, or involved in a specific task, hence only able to provide sketchy and sparse inputs. This scenario requires a framework that supports adjustable autonomy, from explicit teleoperation to a complete autonomy for the robots, and an effective and natural mixed-initiative interaction between the human and the robots [Murphy et al., 2000].

The framework presented in this work should allow a single human operator to supervise and orchestrate the operations of a set of UAVs by means of a natural multimodal communication (using gestures, speech, joypad, tablet interface, etc.) supported by adjustable autonomy. In the proposed approach, we assume a high-level supervisory system that can compose and execute structured robotic tasks while the human rescuer can provide interventions when necessary. These interventions range from abstract task assignments for the multi-drone system (e.g. new areas to explore, search strategies definition, paths to follow, etc.) to navigation adjustments (e.g. deviations from planned paths or trajectories) or precise maneuvering of single robots (e.g. inspection of cluttered environments). More specifically, the proposed human-robot interaction framework combines a multimodal interaction module with a layered mixed-initiative supervisory system. The latter is composed of a multirobot supervisory system interacting with single robot supervisors. For each supervisor, the executive control cycle is managed by a BDI (Belief Desire Intention) system [Ingrand, Georgeff, and Rao, 1992] that orchestrates task planning, switching, decomposition, and execution. The robotic activities are represented as hierarchical tasks which are continuously instantiated and supervised by the executive system depending on the environmental events and the human requests. In this setting, the operator is allowed to continuously interact with the supervisory systems at different levels of abstraction (from high-level tasks assignment/switching to path/trajectory adjustments) while these human interventions are interpreted, monitored, and integrated exploiting the planning and execution control loops. Indeed, following a mixed-initiative planning and execution approach, these interventions can be associated with system reconfigurations which are managed by replanning activities [Finzi and Orlandini, 2005; Carbone et al., 2005; Sellner et al., 2006; Brenner and Nebel, 2009]. However, in our setting, different planning/replanning engines are strictly intertwined in order to address mission, path, and control constraints [Cacace et al., 2015]. In order to evaluate the effectiveness of the proposed system, we designed a simulated rescue and search case study where a human operator interacts with a set of UAVs in order to accomplish typical searching tasks [NATO, 1988; Bernardini, Fox, and Long, 2014] in an alpine scenario.

Search and Rescue Mission with UAVs

Following standard guidelines for search and rescue [NATSAR, 2011; CSAR, 2000; NATO, 1988] and theory of optimal search [Stone and Kettelle, 1989], we assume the following search phases for the rescue mission: (1) define the search area for the targets; (2) define sub-areas for assignment of search pattern; (3) assign specific search patterns to cover each sub-area; (4) define a sequence for the search patterns execution; (5) execute the chosen sequence of patterns, marking the positions of the victims found. During the execution of the mission each of these steps can be dynamically rearranged by the human expert depending on the context. In particular, the human experts should be able to refine the search areas and the associated priority value depending on the development of the mission and the new information gathered. Analogously, if the exploration is supported by UAVs, these areas can be dynamically assigned/reassigned to robots. For this purpose, we introduce some primitives to set exploration paths and areas and the associated search methods (see Table 1). In our setting, a search path is represented by a set of waypoints $p = \{(x_1, y_1), \ldots, (x_n, y_m)\}$ in the 2D map, instead, a search area is specified by a center and a radius $a = \langle (x, y), r \rangle$ (more complex search areas can be easily included). Search areas and paths are also associated with a priority value P_i depending on the estimated probability of finding targets in that area. The search areas can be assigned with an exploration method that instantiates one of the search patterns suggested by the NATO search and rescue manual for helicopter search [NATO, 1988] (here extended to drones as in [Bernardini, Fox, and Long, 2014]):

- *Sector Search (SS)*: that covers the center of the search area and permits a view of the search area from many angles (Figure 1, A).
- *Parallel Track Search (PTS)*: used for a uniform search coverage if the search area is large and the approximate location of the survivor is known (Figure 1, B).
- *Creeping Line Search (CLS)*: used when the search area is narrow and the probable location of the survivor can be on either side of the search track (Figure 1, C).
- *Expanding Square Search (ESS)*: used when the search area is small and the position of the survivor is known within a close limit (Figure 1, D).

In our setting, we assume that each pattern can be instantiated by assigning an area of search and a specific step of expansion (or an angle in the case of SS). We introduce a cost function $C_a(a, sp, u)$ that estimates the cost of the search pattern sp applied to the search area a for the drone u, analogously a cost function $C_p(p, u)$ is to assess the cost of a search path p for the drone u. In this context, once a set of search areas $A = \{a_1, \ldots, a_n\}$ and search paths $P = \{p_1, \ldots, p_m\}$ have been specified by the human expert (step (1) and (2)), that human operator should interact with the autonomous system in order to assign and instantiate the exploration tasks to the drones (step (3) and (4)) and then monitoring and orchestrating the execution (step (5)). Notice that these search assignments may be rearranged depending on the current state of the mission and the drones along with their capabilities.

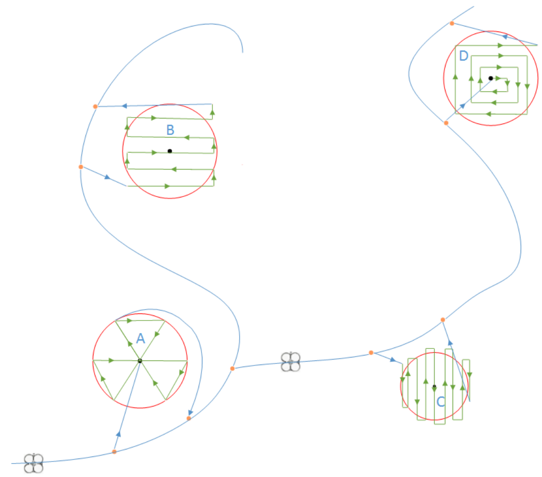

Figure 1: Exploration strategies for two drones searching the environments.

HRI Architecture

The human operator should interact with the robots in a simple and intuitive manner, focusing the cognitive effort on relevant and critical activities (e.g. visual inspection, precise maneuvering, etc.) while relying on the robotic autonomous system for routinized operations and behaviors (task decomposition, path planning, waypoint navigation, obstacle avoidance, etc.). In this context, the robotic architecture must be capable of managing different control modes: *Autonomous*, i.e. the robot can plan and execute a complex task without the human support; *Manual*, i.e. each robot can be directly teleoperated by a human; *Mixed-Initiative*, i.e. the user can execute some operations, while the autonomous system reacts or reconfigures itself accordingly. For this purpose, we designed a modular architecture suitable for supervising and orchestrating the activities of both groups of robots and single robots. The operator should be capable of interacting with the system using different modalities (joypad, gestures, speech, tablet, etc.) at different levels of abstraction (task, activity, path, trajectory, motion, etc.). These continuous human interventions should be suitably and reactively integrated in the robotics control loops providing a natural and intuitive interaction.

The architecture of the HRI system presented in this paper is depicted in Figure 2; in the following we illustrate each component.

Multimodal Interaction. The *multimodal module* allows the operator to interact with the robots using speech, gestures, joypad, tablet, etc.. The integration of different modalities permits a natural, flexible, and robust communication between the human and the system. The *speech* modality is used to control the robot both in mixed-initiative and manual control. We focused on instructions concerning movement, selection, and exploration commands (a subset of these can

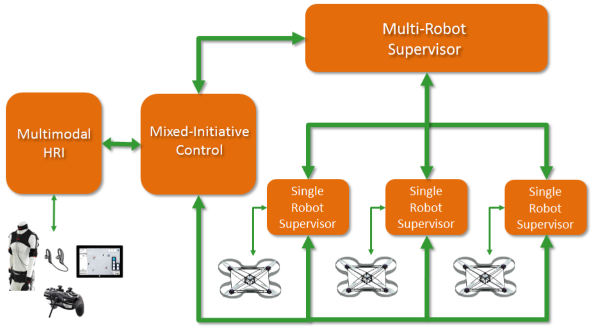

Figure 2: The overall HRI architecture.

be found in Table 2 and Table 1). Gesture-based communication (e.g. pointing, directional signals, etc.) may be used to complete navigational commands with deictic communication (e.g. *go-there*) during proximity interaction with the co-located drones. Joypad-based control is mostly used to manually teleoperate the robots or to adjust the execution of specific tasks. For instance, the operator is allowed to modify the robot speed, orientation or elevation through the joypad, without changing the robot task. A *display/tablet* allows the operator to keep the states of the robot, its current task and textual/graphical feedback on the environment and the operative state. Some informations, like quick notifications, may be sent to headphones, while more complex data have to be displayed. For example, a suitable map of the environment is used to select the areas and path to explore and the waypoints to reach (see Figure 6). Informations coming from the different channels have to be integrated to produce a single interpretation for a task, command, or query; for this purpose, we rely on a multimodal interaction framework based on a late fusion approach [Rossi et al., 2013].

Multi Robot Supervisory System. The *Multi Robot Supervisory System* (MRS) is to delegate tasks to the *Single Robot Supervisory System* (SRS) and monitor their execution with respect to multi-robot integrity, resource, and mission constraints. In particular, in our context the MRS should complete and delegate the abstract, incomplete, and sketchy tasks provided by the operator. For example, the operator may only specify a set of areas to be explored without specific assignments for the single drones or assign a task that cannot be accomplished by a drone, given its current state and equipment. In particular, for each robot the MRS should track the pose, the tasks, subtasks, and actions under execution and power/battery information. Particular tasks are also associated with additional information, like the path followed or the particular region a robot is monitoring. On the other hand, the robots have to make decisions alone without continuously asking confirmations or details to the human. For this reason, the system should be able to delegate and monitor simple, but abstract commands, like $ScanArea$ and $SearchPath$, which are then decomposed in detailed subtasks. Complex delegation system for UAVs are provided in the literature (e.g. [Landén, Heintz, and Doherty, 2012]), since in this work our focus is on mixed-initiative human-robot interaction, we will rely on a simple, but reactive MRS managed by a BDI (Belief Desire Intention executive system) [Rao and Georgeff, 1991] executive system (see Figure 3, upper layer), implemented by a PRS [Ingrand, Georgeff, and Rao, 1992] engine, that interacts with a hierarchical task planner [Erol, Hendler, and Nau, 1994; Montreuil et al., 2007]. Notice that the BDI paradigm is particularly suited for our mixed-initiative system because it provides a flexible, reactive, and adaptive executive engine that is also intuitive for the human.

Single Robot Supervisory System. The SRS can continuously receive tasks from both the MRS and the Operator. The interaction with the latter is mediated by the *Mixed Initiative Control* (MIC) module that supervises the coherence of the human behavior with respect to the robotic behavior at different levels of abstraction (multi/single-agent task, path, trajectory); moreover, it manages the communication between the robots and the human (e.g. task accepted/refused, task accomplished, failures notifications, human decision request, etc.). This communication should be suitably filtered depending on the task and the human operative state, for this purpose the deployment of a multimodal dialogue manager is envisaged [Lucignano et al., 2013], however, in this work we will assume a simpler approach where the notifications are provided to the user in a rule-based fashion (depending on the task and the current state of the operator). The SRS (see Figure 3, second and third layer) is subdivided into two layers: the *High-level Supervisory Control* layer (HLS) which is responsible for user interaction, goal management, task/path planning and execution monitoring, while the *Low-level Supervisory Control* (LLS) layer that manages the low-level execution of the action primitives. Analogously to the MRS, also the HLS is orchestrated by a BDI-based executive system interacting with a hierarchical task planner for task decomposition. In this case, the executive system interacts also with a path planner to instantiate navigation commands and search strategies. (more details are provided in the section about mixed-initiave planning and execution). The executive engine provides goal management, task decomposition tracking the current high-level environmental and executive state. Moreover, it manages any interrupts of the active plan. In this setting, the operator can interact at any time by sending new goals or interrupting the current action execution. Once a complete plan is generated, its execution is managed by the *Plan Supervisor*. The low-level *Primitive Supervisor* receives from the *Plan Supervisor* a list of micro actions associated with a sequence of waypoints, each tagged with constraints, i.e.: minimal distance from obstacles and maximum velocity. These constraints along with the micro operation are then used by the *Control Manager* to select the right controller (e.g. trading-off velocity and precision). Given the controller and the waypoints, the *Trajectory Planner* can then generate and monitor the control trajectory.

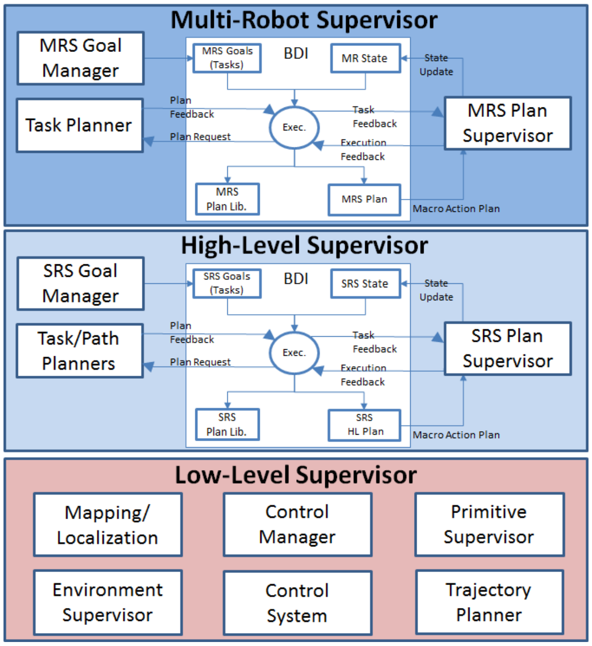

Figure 3: Multi Robot and Single Robot Supervisory Systems.

Mixed Initiative Planning and Execution

In the architecture presented above, the human is allowed to continuously interact with the system at different levels of abstraction. The system supports smoothly sliding from fully autonomous control to fully teleoperated control and vice versa; this interaction is integrated into a continuous planning and execution process that reconfigures the robots activities according to the human intentions and the operative state. In the following, we provide some details about this process.

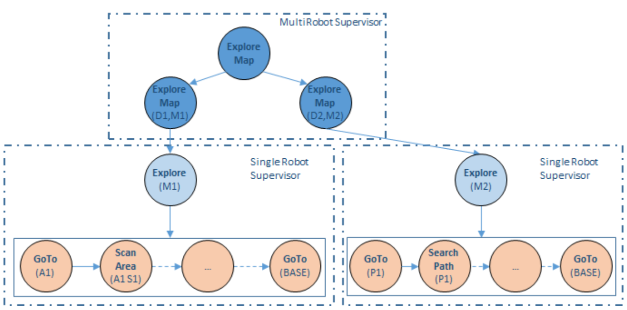

Figure 4: Example of a hierachical plan where the MRS task decomposition is then refined by the SRSs.

Task level interaction. At the higher level of abstraction, the operator can specify the high-level tasks the robot has to perform. Each task is hierarchically represented and can be decomposed into a set of operations and commands that are sent to the lower-level supervisory system. The executive cycle of both the MRS and SRS are managed by a PRS engine [Ingrand, Georgeff, and Rao, 1992] that provides goal management and task decomposition; moreover it tracks the current high-level environmental and the executive state handling any interrupts of the active plan. The operator is integrated in this loop and can interact at any time by sending new goals, changing tasks, or interrupting the current action execution. Depending on the task, the executive system can also call a Hierarchical Task Planner to complete or optimize the task decomposition process. In particular, we rely on the Human Aware Task Planner (HATP) framework [Montreuil et al., 2007], a SHOP-like Hierachical planner [Nau et al., 2001] that can explicitly represent the human interventions. Note that the hierarchical planning paradigm - in combination with the BDI framework - is particularly suited for this domain since it allows the user to monitor and modify the plan at different abstraction levels (task, activity, path, trajectory, motion, etc.) supporting both situation awareness and an intuitive interaction with the generated plan structure. In our setting, the HATP planner is mainly invoked for the resolution of abstract tasks like, e.g. $ExploreMap$, $ScanAreas(A)$, $SearchPath(P)$, etc. (a list of possible tasks/subtasks in our domain can be found in Table 1). Each task can be further specified with the explicit assignment of the parameters, e.g. the robot r and the area a to be scanned can be explicitly provided (i.e. $ScanArea(r, a)$) along with the p search pattern (i.e. $ScanArea(r, a, p)$). Notice that, if the UAV is not explicitly defined by the operator, the MRS system should generate the assignments trying to maximize the overall mission reward (see Figure 4); otherwise, the task can be directly provided to the SRS of a specific robot, in this case the MSR should only check for constraints violations. In the HATP, each operation A^a_k for a robot a can associated with a duration D^a_k and a cost function C^{ctxt}_k. For instance, in our scenario, the estimated cost of scanning an area a with the search pattern sn for a robot u, evaluated in the current context, is the sum of the cost of reaching the area $C_r(u, a)$ and the cost of scanning it $C_a(a, sp, u)$ minus the reward gathered for the area exploration which is proportional to the associated priority P_a. As far as multi-robot constraints are concerned, HATP allows us to define specific *social rules* associated with a cost for their violation $\langle S_k, P^{ctxt}_k \rangle$. In our domain, we only penalize plans where robots explore the same areas or paths and unbalanced distributions of the search effort for the available drones. Therefore, each plan P is associated with a cost:

$$Cost(P) = \Sigma_{a_i \in P} C^{ctxt}_{a_i} + \Sigma_{s_k \in P} P^{ctxt}_{s_k},$$

where a_i is an action of the plan P, s_k is a social rule. In this context, the planner should provide a feasible plan P associated with the minimal cost obtained before a specific timeout. Indeed, the executive system invokes the planner providing a latency; if a solution cannot be generated within the planning latency a default task is executed to recover from the plan failure. In our case, the timeout is defined by context- and task-based rules, for instance if a robot is landed or hovering the planning latency can be extended (up to 5 sec. in our tests), instead, during the flight it can be reduced (max 1 sec. in our tests), otherwise, if the mission time or the energy is below a suitable threshold only a re-

Task	Description
SetSearchArea	Add/delete/modify a search area
SetSearchPath	Add/delete/modify a search path
SetSearchPattern	Change the exploration method
SearchPath	Search along the path
ScanArea	Scan the area with a pattern
ExploreMap	Explore the map
GoTo	Move towards a direction or an area
AbortMission	abort the overall mission

Table 1: List of mission level tasks.

Command	Description
Up	Take off or increase of the altitude
Land	Move the robot to the ground
Down	Decrease the altitude
Left	Move the robot to the left
Right	Move the robot to the right
Forward	Move the robot ahead
Backward	Move the robot to the back
Away	Move the robot away from the target
Closer	Move the robot towards the target
Faster	Increase the speed of the robot
Slower	Decrease the speed of the robot
Go	Move the robot to a specific position
Rotate	Rotate the robot with a specific angle
Switch On/Off	Turn on/off the UAV engines
Brake	The robot brakes

Table 2: List of navigation commands.

active recovery behavior is allowed. More complex policies can be easily introduced and assessed.

Path and Trajectory level interaction. The primitive tasks introduced above (e.g. $Explore(a, u, p)$, $GoTo(a)$, etc.) are associated with drone movements to be suitably planned and executed. We deploy an RRT^* algorithm [Karaman and Frazzoli, 2011] for the generation of obstacle-free paths in the 3D space. Given the waypoints and constraints (proximity of the obstacles) provided by the path planner, a trajectory planner generates a trajectory in terms of position, velocity, acceleration, and jerk. The trajectory is generated exploiting a 4-th order spline concatenation method that preserves continuous acceleration. In the proposed architecture, this trajectory can be directly modified by the human interventions, i.e. the trajectory planner can be continuously invoked to adjust the current trajectory. Indeed, in the mixed-initiative mode, autonomous and human contributions are composed to obtain only one position command. This way, the human interventions can move the robot away from the planned trajectory. However, when the human intervention is released the autonomous mode is enabled and the robot is gradually brought towards the planned trajectory without the need of replanning. More specifically, we assume that, in the mixed initiative mode, the operator can control the robot in velocity. In this setting, the human generates a relative position command $h_C(t) = (x_{m_t}, y_{m_t}, z_{m_t})$ that is added to the $a_C = (x_t, y_t, z_t)$ which is generated by the trajectory planner. The h_C function is calculated as follows:

$$h_C(\text{t}) = \begin{cases} h_C(t-1) + human(t) \text{ if } mixed = ON \\ h_C(t-1) + \Lambda(t)\ otherwise \end{cases}$$

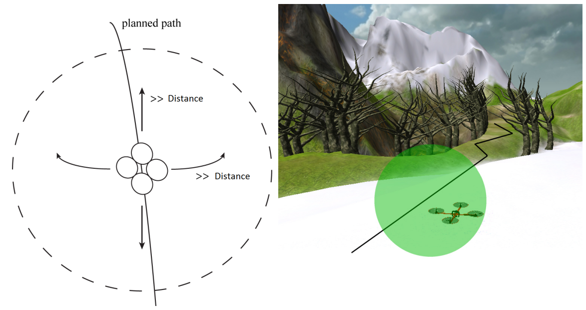

Figure 5: Spherical envelope for trajectory adjustments without replanning.

where $human(t)$ represents the control reference generated by the human operator (through the joypad, gestures, voice, etc.) at time t while $\Lambda(t)$ is a linear function that increases or decreases the value of $h_C(t)$. It is used to drive the $h_C(t)$ towards the one provided by the autonomous control when the joypad is released (see [Cacace, Finzi, and Lippiello, 2014] for an analogous approach). Moreover, we assume that the human operator can move the robot within a spherical region centered in the current planned position (see Figure 5). This sphere represents the context-dependent workspace of the user operator. A replanning process (analogous to the one used for obstacle avoidance) is started when the human operator moves the robot out of this sphere. In this case, the autonomous system generates another path and trajectory to reach the next waypoint. Additional details can be found in [Cacace, Finzi, and Lippiello, 2014]. Note that path and trajectory replanning can also elicit task replanning if the conditions associated with the execution of the current tasks are not valid anymore (e.g. preconditions, energy, resource, and time constraints); these consistency conditions are continuously assessed by the PRS executive systems (single and multi-robot).

In Table 2, we can find some examples of commands that the operator can provide to the system in a multimodal manner (joypad, speech, gestures, etc.) to interact with the robots or to directly controlling them. Note that these commands can be sketchy and context-dependent, for instance, if the robot is in *idle state* the *Up* is for *take off*, otherwise it will increase the UAV altitude. The navigational commands (*left, right, forward, backward*) are robot dependent (e.g. *left* moves the robot to the left side of its camera) and can be abstract (the actual movement can be instantiated by the system) or more specific (e.g. *left* $1m$). The *faster/slower* commands change the robot speed during the execution of a command (they have no effect if the robot is idle); each invocation of these commands will increase/decrease the actual speed to a given percentage up to a limit value. The *go* command moves the robot towards a specific location associated either to coordinates stored on the map or to a symbolic location (either already provided in the map or marked by the operator during the mission execution).

Figure 6: Specification of search paths and areas in a simulated environment (tablet interface). The red segments are possible paths followed by missed hikers, darker areas are associated with a higher priority value.

Simulation and Evaluation

A simulated alpine scenario has been defined with different configurations in order to evaluate the system performance. In this section, we illustrate the scenario and some initial tests we carried out in order to assess the system at work during a typical search and rescue mission in the three modalities: manual control, mixed initiative, and fully autonomous. In this setting, we started to consider only joypad and tablet based interactions.

Platform. We assume a set of simulated quadrotors with the specification of the Asctec Pelican (flight time 20 min., max. airspeed 16 m/s, max. climb rate 8 m/s, max. payload 650 g, etc.) equipped with standard sensors. The overall software system has been developed in ROS under linux uduntu 12.04. The environment has been simulated using the Unity3D game engine.

Environment. The simulated scene, depicted in Figure 6, includes several situations in which an hiker might have lost its way. We considered both summer and winter features. The scene comprises missed hikers and some associated items, either lost by the hikers or irrelevant objects. These objects can help the operator in the search operations (clues), but also divert him/her away from the right direction.

Test and scenario description. The performance of the system has been evaluated considering three control modalities. In the first test the system works in the autonomous mode, lacking any interaction with the operator. In the second test case, the operator may only use the joypad, while the high-level supervisory system is disabled both at the multi-robot and at the single robot level. In this scenario, when one of the robots is not directly operated it waits in the hovering state. In the third case, the operator is supported by the overall system and can work in the mixed initiative control mode. The main aim here is to illustrate the mixed initiative framework at work in a typical rescue scenario comparing its performance with respect to the ones obtained in the other two modalities. In Figure 6, we illustrate the map provided to the user at the beginning of each test: it represents the search environment where paths, areas, and likelihood values of finding survivors are represented. The interface employed for the tests is depicted in Figure 8, here the video streaming of the controlled robot is full screen, while smaller windows show the video streaming of the other robot and the environmental map with the robots positions. Messages from each robot are also provided though the interface; the information available to the user during the tests depends on the control mode as illustrated in Table 3.

Each test starts with the robots positioned in a fixed point and should end in that position (see Figure 7). During the tests, we assumed a perfect positioning system and a perfect object/human detection system when the target is in the camera field of view (50-degree) within a range of $30m$.

In the autonomous case, we assume that the mission is planned by the MSR-level task planner at the start and then reactively adjusted by the autonomous system during the exploration, depending of the detected objects. When a relevant object is detected, the robot is to replan in order to

Figure 7: Simulated environment: starting point (base) for the two drones in our tests.

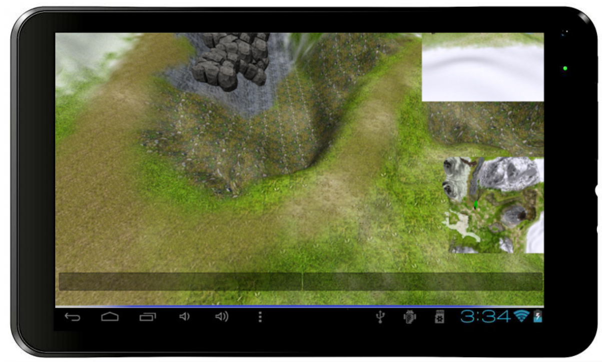

Figure 8: User interface during the simulated mission in the mixed initiative mode. The camera streaming of the controlled drone is full screen, the other drone camera and the environmental map are the smaller windows on the right. Text messages for the two drones are illustrated at the bottom.

Test Type	Autonomous	Manual	Mixed-Initiative
Information available	- Num. of survivor - Loc. of areas	- Num. of survivor - Loc. of areas	- Num. of survivor - Loc. of areas
Messages Shown	- None	- Time elapsed - Battery warning - Survivor conf.	- Salient object - Survivor alert - Exploration start/end - Time elapsed - Failure request.
Control	- Autonomous	- Teleoperated	- Mixed-Initiative

Table 3: Information available to the user during the tests.

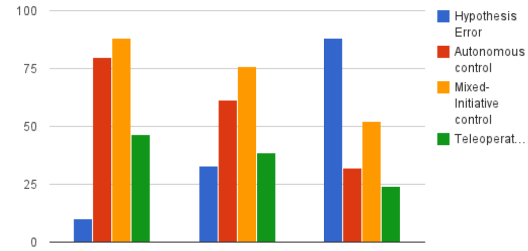

Figure 9: Percentage of success in target detection with respect to the control modalities.

explore the surrounding area with a predefined scan path. Analogously, when the battery power falls below a suitable threshold, the robot should replan in order to come back to the initial position.

In the teleoperated mode, the robots are directly controlled through a tablet and a joypad used to define direction, speed, orientation of the robot and of the camera. During the tests, the user receives only two messages: the confirmation of the effective survivor sighting and a warning about the battery level, if it drops below a fixed threshold.

In the *mixed initiative* scenario, the user can interact with the MRS and SRS during the mission using the tablet and the joypad. The graphical information provided by the tablet interface is similar to the one of the *teleoperated* case, but additional textual information is provided (see Table 3). Analogously to the *autonomous* mode, also in this case the mission is planned in advance, but the human is allowed to provide interventions at the plan and the trajectory level. On the other hand, the system alerts the operator when salient clues are detected by a robot. The operator can then inspect the clues and decide whether to check the area. The operator can always inspect the current operative and environmental state: including robots position, speed, state, task, plan. These notices can be supplied through different channels, like audio notifications or messages on a tablet. In this setting we decided for the text notification on the tablet.

Test set-up. In our tests, we considered the scenario depicted in Figure 6 to be explored by 2 robot. The mission goal is to find 15 persons within 10 minutes. The testing area is a simulated environment of $160 \times 140\ m^2$ with 9 areas to be explored and 9 paths. 9 clues (one for victims) and 21 irrelevant objects (distractors) are randomly distributed on the environment. We defined 3 different dispositions of survivors and objects within the scene. In the first case (test A), all the targets are located inside the areas and paths with a uniform distribution. In the second (test B) and third case (test C), the 66% and 13% of the targets is located inside the areas/paths with a uniform distribution, while the remaining are uniformly positioned in the rest of the scene. Each target can be associated with 1 clue which is positioned within a range of $20m$. In the teleoperation and mixed-initiative modes, each modality has been executed 12 times, by 4 users (3 tests for each mode after 2 session of training).

Results. In Figure 9, we illustrate the percentage of survivors found in the three control modalities with respect to the different test cases (A, B, C). As expected, the teleoperated mode is not effective, indeed not only the parallel search of the two drones cannot be exploited in this case, but also the lack of task/path guidance reduces the overall situation awareness, hence the number of correct detections is significantly lower than in the other modes. On the other hand, since we assume a reliable human/object detection system, the autonomous mode is very effective when the initial hypothesis is accurate. In this case, the two robots can find about 80% of survivors, scanning all the areas. However, the success rate rapidly drops when the initial hypothesis becomes less accurate; indeed, the autonomous system is not flexible enough to diverge from the planned activities. Instead, the mixed-initiative mode seems more effective than the autonomous mode for each of the test cases and this advantage seems emphasized when the initial hypothesis becomes wrong. Indeed, in the worst case, the mixed-initiative mode behaves significantly better (57.78%) than the autonomous (32.25%) and the teleoperated ones (24.12%). The significance of these results is illustrated in the Tables 4, 5 and 6 where we compare the mixed-initiative, autonomous, and teleoperated results for each of the 3 tests cases. The comparison of the mixed-initiative performance in the three testing scenarios can be found in Table 7.

Targets	Min	Max	Mean	Std	t-test
Mixed-Initiative	12	14	12.00	0.84	0.1009915
Autonomous	11	12	11.25	0.5	
Survivors	**Min**	**Max**	**Mean**	**Std**	**t-test**
Mixed-Initiative	12	14	12.00	0.84	0.0001655
Teleoperation	6	8	7	0.71	

Table 4: Mixed-initiative mode vs. automous and teleoperated mode (test A).

In the mixed-initiative mode, we also analyzed the human interventions for the different cases. In Table 8 we can ob-

Targets	Min	Max	Mean	Std	t-test
Mixed-Initiative	11	12	11.40	0.55	0.034469
Autonomous	7	13	9.20	2.28	
Survivors	**Min**	**Max**	**Mean**	**Std**	**t-test**
Mixed-Initiative	11	12	11.40	0.55	<.0001
Teleoperation	4	7	5.8	1.3	

Table 5: Mixed-initiative mode vs. automous and teleoperated mode (test B).

Targets	Min	Max	Mean	Std	t-test
Mixed-Initiative	7	9	7.80	0.84	<.0001
Autonomous	2	3	2.6	0.55	
Survivors	**Min**	**Max**	**Mean**	**Std**	**t-test**
Mixed-Initiative	7	9	7.80	0.84	<.0001
Teleoperation	3	5	3.6	0.89	

Table 6: Mixed-initiative mode vs. automous and teleoperated mode (test C).

Targets	Min	Max	Mean	Std	t-test
Mixed-Initiative (test A)	13.2	12	12.00	0.84	0.0019205
Mixed-Initiative (test B)	11	12	11.40	0.55	
Survivors	**Min**	**Max**	**Mean**	**Std**	**t-test**
Mixed-Initiative (test B)	11	12	11.40	0.55	<.0001
Mixed-Initiative (test C)	7	9	7.80	0.84	
Survivors	**Min**	**Max**	**Mean**	**Std**	**t-test**
Mixed-Initiative (test A)	13.2	12	12.00	0.84	0.000459
Mixed-Initiative (test C)	7	9	7.80	0.84	

Table 7: Mixed-Initiative performance w.r.t. the accuracy of the initial hypothesis.

serve that, as expected, the percentage of the time spent in teleoperation (joypad usage) increases with the complexity of the domain, indeed, when the initial hypotheses are wrong the user should intensify the interventions diverging from the planned activities. This is directly correlated with the increment of the low-level interventions (e.g. trajectory corrections or teleoprated search) and task replanning episodes. The latter can be directly invoked by the user or indirectly elicited by external events (e.g. object detection) or constraint violations (e.g. low energy, resource conflicts, etc.). Note that the high-level interventions seem more sparse because are usually associated with strategic decisions (e.g. new areas to be explored). Notice also that since the operator can easily provide direct adjustments during the mixed-initiative mode without provoking replanning, the time spent in teleoperation remains high for each of the cases analysed in Table 8.

		Test A	Test B	Test C
Joypad usage (%)	**Mean**	30.67%	43.07%	53.04%
	Std	1.53	2.64	2.34
Low Level Int.	**Mean**	5.4	6.4	6.8
	Std	1.34	2.19	1.92
Task Replanning	**Mean**	10.0	12.0	13.8
	Std	0.83	1.87	3.49

Table 8: Human interventions and task replanning episodes during the 3 test scenarios in the mixed initiative mode.

Conclusion

In this paper we presented a mixed-initiative planning and execution system for human multi-drones interaction during search and rescue missions in an alpine environment. The peculiar aspect of the domain is the presence of a human operator which is not fully dedicated to the drones control, but operative in the rescue scenario, hence only able to provide sketchy and sparse input to the robots. This interactive scenario requires a flexible mixed-initiative framework that supports multimodal HRI and sliding autonomy. The proposed system allows the user to interact with the robotic team at different levels of abstraction, from abstract multirobot task assignment to direct teleoperation of specific robots. The main aim of the paper was the presentation of the novel domain along with the overall framework focusing on planning and execution capabilities. A simulation environment has been also developed to assess the effectiveness of the proposed mixed-initiative system. In the initial tests presented in this paper we illustrated the system at work during a typical mission comparing its performance with respect to the obtained employing direct teleoperation and a simple autonomous system. An extensive evaluation of the proposed framework is left as a future work.

Acknowledgment

The research leading to these results has been supported by the FP7-ICT-287513 SAPHARI and FP7-ICT-600958 SHERPA projects.

References

Bernardini, S.; Fox, M.; and Long, D. 2014. Planning the behaviour of low-cost quadcopters for surveillance missions. In *Proc. of ICAPS 2014*.

Bitton, E., and Goldberg, K. 2008. Hydra: A framework and algorithms for mixed-initiative uav-assisted search and rescue. In *Proc. of CASE 2008*, 61–66.

Brenner, M., and Nebel, B. 2009. Continual planning and acting in dynamic multiagent environments. *Journal of Autonomous Agents and Multiagent Systems* 19(3):297–331.

Cacace, J.; Finzi, A.; Lippiello, V.; Loianno, G.; and Sanzone, D. 2015. Aerial service vehicles for industrial inspection: task decomposition and plan execution. *Appl. Intell.* 42(1):49–62.

Cacace, J.; Finzi, A.; and Lippiello, V. 2014. A mixed-initiative control system for an aerial service vehicle supported by force feedback. In *Proc. of IROS 2014*, 1230 – 1235.

Carbone, A.; Finzi, A.; Olandini, A.; Pirri, F.; and Ugazio, G. 2005. Augmenting situation awareness via model-based control in rescue robots. In *Proc. of IROS 2005*, 3699–3705. IEEE.

CSAR. 2000. *Canadian National Search and Rescue Manual*. Department of National Defence.

Cummings, M.; Bruni, S.; Mercier, S.; and Mitchell, P. J. 2007. Automation architecture for single operator, multiple uav command and control. *The International Command and Control Journal* 1(2):1–24.

Erol, K.; Hendler, J.; and Nau, D. S. 1994. Umcp: A sound and complete procedure for hierarchical task-network planning. In *Proc. of AIPS-1994*, 249–254.

Finzi, A., and Orlandini, A. 2005. Human-robot interaction through mixed-initiative planning for rescue and search rovers. In *Proc. of AI*IA 2005*, 483–494.

Ingrand, F.; Georgeff, M.; and Rao, A. 1992. An architecture for real-time reasoning and system control. *IEEE Expert: Intelligent Systems and Their Applications* 7(6):34–44.

Karaman, S., and Frazzoli, E. 2011. Sampling-based algorithms for optimal motion planning. *Int. J. Rob. Res.* 30(7):846–894.

Landén, D.; Heintz, F.; and Doherty, P. 2012. Complex Task Allocation in Mixed-Initiative Delegation: A UAV Case Study. In *PRIMA 2010*, volume 7057. 288–303.

Lucignano, L.; Cutugno, F.; Rossi, S.; and Finzi, A. 2013. A dialogue system for multimodal human-robot interaction. In *Proc. of ICMI 2013*, 197–204.

Malasky, J.; Forest, L. M.; Khan, A. C.; and Key, J. R. 2005. Experimental evaluation of human-machine collaborative algorithms in planning for multiple uavs. In *IEEE International Conference on Systems, Man and Cybernetics*, 2469 – 2475.

Marconi, L.; Melchiorri, C.; Beetz, M.; Pangercic, D.; Siegwart, R.; Leutenegger, S.; Carloni, R.; Stramigioli, S.; Bruyninckx, H.; Doherty, P.; Kleiner, A.; Lippiello, V.; Finzi, A.; Siciliano, B.; Sala, A.; and Tomatis, N. 2012. The sherpa project: Smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments. In *Proceedings of SSRR-2012*.

Montreuil, V.; Clodic, A.; Ransan, M.; and Alami, R. 2007. Planning human centered robot activities. 2618–2623. Proc. of SMC 2007.

Murphy, R. R.; Casper, J.; Micire, M.; and Hyams, J. 2000. Mixed-initiative control of multiple heterogeneous robots for urban search and rescue. *IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans* 19972003.

NATO. 1988. *ATP-10 (C). Manual on Search and Rescue. Annex H of Chapter 6*.

NATSAR. 2011. *Australian National Search and Rescue Manual*. Australian National Search and Rescue Council.

Nau, D. S.; Muoz-Avila, H.; Cao, Y.; Lotem, A.; and Mitchell, S. 2001. Total-order planning with partially ordered subtasks. In *Proc. of IJCAI 2001*, 425–430.

Ollero, A.; Lacroix, S.; Merino, L.; Gancet, J.; Wiklund, J.; Remuss, V.; Perez, I. V.; Gutirrez, L. G.; Viegas, D. X.; Benitez, M. A. G.; Mallet, A.; Alami, R.; Chatila, R.; Hommel, G.; Lechuga, F. J. C.; Arrue, B. C.; Ferruz, J.; de Dios, J. R. M.; and Caballero, F. 2005. Multiple eyes in the skies: architecture and perception issues in the comets unmanned air vehicles project. *IEEE Robot. Automat. Mag.* 12(2):46–57.

Rao, A. S., and Georgeff, M. P. 1991. Deliberation and its role in the formation of intentions. In *Proc. of UAI*, 300–307.

Rossi, S.; Leone, E.; Fiore, M.; Finzi, A.; and Cutugno, F. 2013. An extensible architecture for robust multimodal human-robot communication. In *Proc. of IROS 2013*, 2208–2213.

Sellner, B.; Heger, F.; Hiatt, L.; Simmons, R.; and Singh, S. 2006. Coordinated multi-agent teams and sliding autonomy for large-scale assembly. *Proceedings of the IEEE* 94(7).

2013. Sherpa, eu collaborative project ict-600958. http://www.sherpa-project.eu/sherpa/.

Stone, L., and Kettelle, J. 1989. *Theory of Optimal Search*. Topics in Operations Research Series. I N F O R M S: Institute for Operations Research & the Management Sciences.