Journal of Intelligent & Robotic Systems (2019) 94:669–686
https://doi.org/10.1007/s10846-018-0899-0

Investigating Human-Robot Teams for Learning-Based Semi-autonomous Control in Urban Search and Rescue Environments

A. Hong[1] · O. Igharoro[1] · Y. Liu[1] · F. Niroui[1] · G. Nejat[1] · B. Benhabib[2]

Received: 8 October 2017 / Accepted: 26 June 2018 / Published online: 9 August 2018

Abstract

Teams of semi-autonomous robots can provide valuable assistance in Urban Search and Rescue (USAR) by efficiently exploring cluttered environments and searching for potential victims. Their advantage over solely teleoperated robots is that they can address the task handling and situation awareness limitations of human operators by providing some level of autonomy to the multi-robot team. Our research focuses on developing learning-based semi-autonomous controllers for rescue robot teams. In this paper, we specifically investigate the influence of the operator-to-robot ratio on the performance of our proposed MAXQ hierarchical reinforcement learning based semi-autonomous controller for USAR missions. In particular, we propose a unique learning-based system architecture that allows operator control of larger numbers of rescue robots in a team as well as effective sharing of information between these robots. A rigorous comparative study of our learning-based semi-autonomous controller versus a fully teleoperation-based approach was conducted in a 3D simulation environment. The results, as expected, show that, for both semi-autonomous and teleoperation modes, the total scene exploration time increases as the number of robots utilized increases. However, when using the proposed learning-based semi-autonomous controller, the rate of exploration-time increase and operator-interaction effort are significantly lower, while task performance is significantly higher. Furthermore, an additional case study showed that our learning-based approach can provide more scene coverage during robot exploration when compared to a non-learning based method.

Keywords Urban search and rescue · Multi-robot rescue teams · Semi-autonomous control · Operator-to-robot ratio

1 Introduction

In Urban Search and Rescue (USAR), mobile robots can effectively explore disaster environments with minimum a priori knowledge about the locations of victims and scene layout [1, 2]. The majority of past robotic USAR missions, however, have been based on the utilization of teleoperated single robots [2–5]. Operators of such robots have, typically, experienced perceptual difficulties in trying to understand the 3D cluttered environments via remote visual feedback [2]. Furthermore, the (single) rescue robots have experienced task-handling limitations.

Recently, researchers have considered multi-robot teams for a number of applications, including material transportation [6], reconnaissance and surveillance [7–13], inspection and manipulation [14], and USAR missions [4, 15, 16]. It has been claimed that increased efficiency and system robustness can be achieved through robot redundancy. However, one of the main challenges that an operator may face in controlling a multi-robot team in teleoperation mode is the simultaneous control of multiple robots while juggling between their respective tasks and keeping situational awareness (SA) of all the robots. Furthermore, time constraints on these tasks can lead to operator fatigue and decrease in efficiency [17]. Thus, it has been recommended that such multi-robot teams be given some level of autonomy [4].

The objective of this paper is, thus, to promote the use of a novel controller with MAXQ hierarchical reinforcement learning (HRL) deliberation for improved supervision and control of multi-robot teams in USAR environments. This is achieved by investigating the operator-to-robot ratio

✉ F. Niroui
farzad.niroui@mail.utoronto.ca

[1] Autonomous Systems and Biomechatronics Laboratory, Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Rd, Toronto, Ontario, M5S 3G8, Canada

[2] Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Rd, Toronto, Ontario, M5S 3G8, Canada

when using this novel controller. It is conjectured that the controller's efficiency improvement over teleoperation is amplified as the robot team's size increases.

MAXQ HRL provides a USAR controller with the ability to effectively allocate subtasks to robots in order to complete the overall mission. The approach provides two primary advantages, when compared to methods currently in use: (*i*) the operator can handle a greater number of robots, without significant performance loss, due to the controller's ability to only request human assistance when a robot is stuck or when there is uncertainty in human identification; and, (*ii*) less interaction effort is required by the operator to control the team.

The overall goal of our learning-based control architecture is to reduce interaction effort—the amount of time the operator is required to interact with the team [18], while improving task performance [19]. Past controllers have mainly been shown to control operator-to-robot teams with up to 1:12 ratio and require high-level decisions be made by only the human operators (e.g., [6, 20–26]). Our architecture, in contrast, uses HRL to learn both high-level and low-level tasks from the USAR environment in order to increase performance in autonomous exploration and victim identification.

2 Multi-robot Teams for USAR

Our research in the area of robot-assisted USAR has focused on the development of learning-based semi-autonomous controllers and exploration planners [27, 28] for single robots, as well for non-cooperating [29] and cooperating [30] multi-robots teams. These controllers allow operators and robots to share the important tasks of exploring unknown cluttered USAR scenes and searching for victims. However, as noted by others and us, in order to effectively implement such semi-autonomous controllers for cooperative multi-robot teams, the impact of increased numbers of robots on system performance must be investigated [15, 20, 31–33], as well as the effect of task automation on robot team performance [21–26, 34–37].

2.1 Operator-to-Robot Ratios

The issue of operator-to-robot ratio in USAR environments, for teleoperated robots, has been investigated by numerous researchers. In [15], for example, it was shown that task performance increases as operator control increases from four to eight robots, but subsequently decreases as the number of robots is further increased from eight to twelve, when participants were asked to teleoperate robots in a simulated USAR environment.

In [31] and [32], the use of live-streaming versus asynchronous video displays, when operators were teleoperating multi-robot teams, was compared in a simulated USAR environment. With respect to overall performance, the two approaches were similar for all groups of robots, with the peak number of victims being found with eight robots.

Multi-robot team compositions of two operators and 24 robots were investigated in [20], using both teleoperation and autonomous path planning control modes. It was shown that the team structure has no significant effect on the number of victims found, however, teleoperation exploration time may be longer when each operator individually controls twelve robots.

In [33], a study was conducted to compare several different operator-to-robot configurations of teleoperated multi-robot USAR teams in a simulated environment. The results showed that the effectiveness of task sharing increases with an increase in the operator-to-robot ratio.

One of the many challenges a human operator could face in USAR environments is the simultaneous control of multiple robots. Thus far, in the literature, it has been shown that human operators can effectively control up to eight robots in a teleoperation mode before losses in performance occur [15, 31, 32]. In order to increase the number of robots that a single operator can handle, task automation techniques have been proposed.

2.2 Task Automation for Multi-robot Teams

In order to increase effectiveness of team performance in USAR missions, numerous high-level and low-level task automation methods have been investigated [21–26, 34, 38–41]. In [34], the foraging tasks of a multi-robot team were investigated to determine which tasks can be automated to reduce operator workload. Experiments were conducted in a simulated USAR environment with four, eight and twelve robots. Two conditions were tested: where operators had full control over each team of robots, and where operators had independent control only over the exploration or perceptual search subtasks. Results showed better performance when the subtasks were individually performed by the operators, rather than full control, especially, with increased number of robots. The overall results supported the automation of the robot exploration tasks.

In [21], low-level task autonomy was considered. Namely, a semi-autonomous controller with robot path planning and navigation autonomy was compared to teleoperation for a 2:24 operator-to-robot configuration in a simulated environment. Operators were able to find more victims using the semi-autonomous controller. Furthermore, there was a substantial advantage for using autonomous planning when operators shared control of all 24 robots.

Techniques have also been developed to assist operators in forming and managing robot teams [22] as well as determining or adapting search strategies during USAR missions [23–26]. For example, in [22], a team management framework was presented to account for lost, failed, and new robots in a heterogeneous multi-robot team deployed in disaster zones. The framework coordinated tasks between heterogeneous robots, and helped a team reshape itself. Usage of the framework in a simulated environment showed that it can increase environment coverage and the number of victims identified when compared to the case where the robot tasks and team sizes were fixed.

With respect to search strategies, techniques have been used to 1) provide a list of potential strategies to an operator [24], 2) interpret operator strategies to provide direct commands to the robot team [25], 3) provide feedback to an operator with respect to which actions he/she should take while implementing USAR tasks [23], or 4) allow the robots to complete the tasks on their own and ask for operator assistance when needed, i.e., when a robot needed help navigating through rubble piles or was stuck [26]. Simulation and experimental results have shown that missions were completed in less time [23, 25] and covered more terrain [23, 26] when compared to teleoperation or fully autonomous control.

Managing and controlling a team of robots to explore cluttered environments while searching for victims can be difficult for autonomous controllers. The aforementioned work has shown that different high-level and low-level tasks can be automated to aid with such missions. However, humans still need to be directly involved to supplement the lack of perceptual autonomy and to use brute force when needed. Therefore, semi-autonomous controllers have been shown to help reduce workload and, thus, increase the number of robots an operator can handle. The increased number of robots in a team has been shown to improve USAR mission performance in both exploration efficiency and victim identification. These controllers, though, have mostly been shown to manage less than 1:12 operator-to-robot team ratios. In order to further increase the number of robots an operator can control, the integration of AI-based techniques should be considered to assist in determining which tasks should be automated and when during deployment.

2.3 AI-Based Approaches for Semi-autonomous Control

Markov Decision Process (MDP) based techniques have been proposed for semi-autonomous control in applications including multi-agent UAV navigation [35, 36] and single-vehicle path planning [37]. In [35] and [36], a Mixed Markov Decision Process (MIMDP) approach was utilized, which was created from two MDP models—one for the autonomous system and one for the supervision unit. The approach allowed the autonomous system to decide what actions to take and when the supervision unit should be requested and control transferred to it. The MIMDP approach, however, was only implemented for a single UAV problem, with experiments focusing on confirming the making of requests by the agent to an operator. In [36], a human-help provider MDP was presented for the control of UAVs by providing three different help requests to operators ranging from critical to non-critical. The system was tested with 1-15 UAV agents and 1-3 operators to determine how many requests from the agents were treated by the operators.

In [37], a Transfer of Control Partially Observable Markov Decision Process (TOC POMDP) was used for the task of semi-autonomous path planning between a driver and a vehicle. Namely, the TOC POMDP was used to determine the transfer of control actions. The capability of the system was compared to both human drivers and autonomous vehicles using data from the Open Street Map in order to have a vehicle reach a goal destination. After 100 trials, the TOC POMDP system always reached the goal destination in similar time as the human driver, while the autonomous system took longer and at times failed to reach the goal.

The USAR problem addressed herein differs from the problems discussed above in that it involves more than just navigation or path planning. Namely, the problem at hand comprises three subtasks: exploration, victim identification, and navigation in cluttered environments. This significantly increases the state space of the USAR problem, limiting the use of traditional modeling techniques, such as MDPs, POMDPs, or DCOPs (Dynamic Distributed Constraint Optimization Problems). The main drawback of these traditional techniques is that they often fail to scale up to large numbers of subtasks and agents.

Furthermore, both MDPs and POMDPs suffer from the *curse of dimensionality*, where state parameter dimensionality can increase exponentially with team size [42]. MAXQ, on the other hand, utilizes a hierarchical organizational structure by decomposing an overall task into a finite set of subtasks recursively, where each subtask is modelled as a MDP. MAXQ uniquely supports temporal, subtask and state abstraction, which can significantly reduce the number of state variables needed and can speed up the overall learning process for real-world problems [43]. Also, it has fewer constraints on its policies (i.e., mapping of states to possible actions) and, thus, requires less prior knowledge about the environment.

In contrast to existing controllers, our hierarchical learning semi-autonomous controller manages task allocation between robots and human operators effectively, while learning from the cluttered USAR environment to increase performance in exploration and victim identification. Thus, allowing higher robot-to-operator ratios without significant performance loss. The controller is integrated with a user interface that allows for sharing of information as well as tasks with the operator. To the authors' knowledge, no such learning approaches have been used for multi-robot semi-autonomous control for cluttered urban search and rescue applications.

3 MAXQ Hierarchical Reinforcement Learning for Multi-robot Teams

As abovementioned, herein, we propose the use of a MAXQ HRL based semi-autonomous controller, developed in our lab [27–30, 44, 45], to increase the performance of multi-robot and human operator teams. The fundamental principle of MAXQ is to decompose the decision-making problem modeled as an MDP, M_0, into a finite set of smaller and easier to resolve subtasks, $M_1, M_2, \cdots, M_n$, and to derive the optimal policies for these subtasks in order to achieve a hierarchical optimal policy for the overall task, M_0. The purpose of MAXQ learning is to determine this hierarchical optimal policy in order to maximize the expected cumulative reward for M_0, defined as the action-value function, namely the Q function. For every subtask, M_p, a policy, π_p, that maps all possible states of the subtask to a child task is defined. The child task can be either a primitive action or another subtask to execute. Subsequently, the hierarchical optimal policy, π, is the set containing all the policies for all subtasks. More details on MAXQ learning can be found in [43].

The MAXQ task hierarchy for the multi-robot USAR problem at hand is presented in Fig. 1. Herein, sub-scenes are defined as isolated regions of the USAR environment. The *Root* task represents the overall USAR problem of scene exploration and victim identification. Cooperation is achieved by providing the robots with their own copy of the task hierarchy while sharing the same common *Root* task.

The *Root* task is divided into five different hierarchical subtasks: *Search Sub-scene* (SSS_i, where i represents the index of the sub-scene), *Navigate to Unvisited Regions* (NUR), *Victim Identification* (VI), *Navigate* (NG), and *Human Control* (HC). The purpose of the SSS_i subtask is to allocate rescue robots to the different sub-scenes to explore. The primitive action $Exit\,Sub\text{-}scene(ESS)$ is used to terminate the SSS_i subtask and guide a robot out of a sub-scene. Once robots are deployed into their designated sub-scenes, the NUR and VI subtasks are used to allow them to cooperatively explore unvisited regions within these sub-scenes and identify potential victims, respectively. For the NUR subtask, the primitive action $Standby$ is used to end the exploration of a sub-scene for all the corresponding robots in that sub-scene. For the VI subtask, when a victim is identified, the primitive action Tag is executed to tag the victim's location. While exploring the environment in a sub-scene and looking for victims, the lower-level NG subtask can be called from either the NUR and VI subtasks to allow a robot to perform local navigation and obstacle avoidance, which utilizes 2D grid map information of the robot's surrounding cells [27]. The primitive actions θ, F, and B, allow a robot to rotate, move forward, and move backward, respectively. For example, the NUR subtask can call the NG subtask to locally navigate unexplored regions, while the VIsubtask can call the NG subtask to move towards a potential victim.

The HC subtasks are used to request for human assistance and allow the operator to intervene when a robot cannot execute any of the aforementioned subtasks autonomously. The *Low Level HC* subtask allows an operator to take over control of the NG subtask, and the *High Level HC* subtask allows an operator to take control of the NUR and VI subtasks. In order to minimize the workload of the user, MAXQ only requests for human assistance of a subtask when required (i.e., when the robot is stuck or there is uncertainty in victim identification).

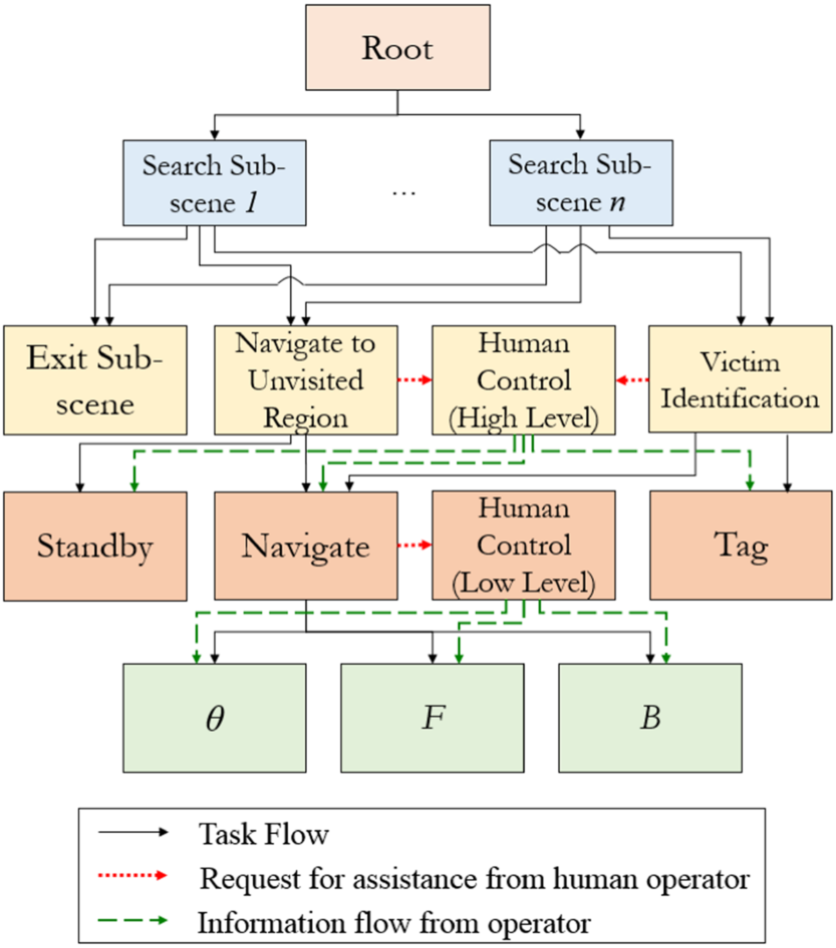
The MAXQ state function of the *Root* task is defined as $S_{Root}(V, S_S, M_G)$: V represents the presence of potential victims; S_S denotes the sub-scene status (i.e., unexplored, being explored, or explored); and, M_G is a collection of 2D occupancy maps of USAR sub-scenes.

For the SSS_i subtask, the state function is defined by $S_{SSS_i}(V_i, L_{Robot}, M_{G,i}, A_{O,SSS_i})$, where $L_{Robot} = \{L^1_{Robot}, L^2_{Robot}, \cdots, L^k_{Robot}\}$ denotes the robots' locations within the same sub-scene SS_i with respect to the starting location of the first robot, which is defined as the origin of the local coordinate frame, and number of robots deployed in the same sub-scene is depicted by k. $M_{G,i}$ is the 2D occupancy map of the sub-scene obtained by merging 2D maps generated by each individual robot, j, deployed into the same sub-scene. $A_{O,SSS_i} = \left\{A^1_{O,SSS_i}, \cdots, A^{j-1}_{O,SSS_i}, A^{j+1}_{O,SSS_i}, \cdots, A^k_{O,SSS_i}\right\}$ represents the other robots' actions/subtasks.

The state function of NUR is defined as $S_{NUR}(L_{Robot}, M_{G,i}, A_{O,NUR,i})$, where $A_{O,NUR,i} = \left\{A^1_{O,NUR,i}, \cdots A^{j-1}_{O,NUR,i}, A^{j+1}_{O,NUR,i}, \cdots, A^k_{O,NUR,i}\right\}$ represents the other robots' actions/tasks while cooperatively executing the NUR subtask with $Robot^j$. A direction-based exploration strategy based on frontiers is implemented to effectively explore a sub-scene utilizing the 3D cluttered terrain information of the environment in this subtask [27].

The state function of the *VI* subtask is $S_{VI}(\mathrm{L}^j_{V/Robot}, M^j_{G,i})$, where the potential victim's location

Fig. 1 MAXQ task hierarchy [27]


Root
Search Sub-scene *1*
...
Search Sub-scene *n*
Exit Sub-scene
Navigate to Unvisited Region
Human Control (High Level)
Victim Identification
Standby
Navigate
Human Control (Low Level)
Tag
θ
F
B
Task Flow
Request for assistance from human operator
Information flow from operator

is marked as $L^j_{V/Robot}$ in the scene. When the primitive action Tag is executed, the victim's location is tagged within $M^j_{G,i}$.

The state function of the *NG* subtask is $S_{NG}(C^j_l, D^j_E, D^j_{xy}, L_{V/Robot})$, where $Robot^j$'s surrounding cells C^j_l, $l = 1$ to 8, can be categorized according to the depth profile information D^j_{xy} of the rubble pile in the robot's surrounding environment; the desired exploration direction (determined by NUR) is depicted by D^j_E. The primitive actions, rotate $Robot_j$ by an angle (θ), and move $Robot^j$ forward (F) or backwards (B), are determined by the status of the robot's surrounding cells and sent to the robot's low-level controller to execute into motion commands.

MAXQ decomposes the *Root* task into a finite set of subtasks or primitive actions recursively. In a MAXQ task hierarchy, the possible states of each task are mapped to a child (either a primitive action or another subtask) through a policy π. In the proposed MAXQ task hierarchy, the Q value (action-value function) for the *Root* task is defined as follows [44]:

$$Q\,(Root, s, SSS_i) = V\,(SSS_i, s) + C(Root, s, SSS_i), \quad (1)$$

where $V\,(SSS_i, s)$, the projected value function of executing the SSS_i subtask in state s; and $C(Root, s, SSS_i)$, the completion function representing the discounted cumulative reward of executing the SSS_i subtask, can be defined as:

$$V\,(SSS_i, s) = Q\left(SSS_i, s, \pi_{SSS_i}(s)\right), \text{ and} \quad (2)$$

$$C\,(Root, s, SSS_i) = \sum\nolimits_{s' \in S_{Root}, N} \left\{P_{Root}\left(s', N|s, SSS_i\right)\gamma^N \right. \\ \left. Q\left(Root, s', \pi_{Root}(s')\right)\right\}, \quad (3)$$

where $\pi_{SSS_i} \in \{ESS, NUR, VI\}$ and $\pi_{Root} \in \{SSS_1, \cdots, SSS_n\}$ represent the policies for the SSS_i subtask and *Root* task, respectively. S_{Root} is the state function of the *Root* task, γ is the discount factor and N denotes the number of transition steps from state s to the next state s'. P_{Root} is the probability transition function for the *Root* task.

The action-value function of the *Root* task is recursively decomposed into the summation of action-value functions of its subtasks. For example, the action-value function for SSS_i can be further decomposed as follows:

$$\begin{aligned} Q(\mathrm{SSS}_i, s, ESS) &= V(\mathrm{ESS}, s) + C(\mathrm{SSS}_i, s, ESS), \\ Q(\mathrm{SSS}_i, s, NUR) &= V(\mathrm{NUR}, s) + C(\mathrm{SSS}_i, s, NUR), \\ Q(\mathrm{SSS}_i, s, VI) &= V(\mathrm{VI}, s) + C(\mathrm{SSS}_i, s, VI), \end{aligned} \tag{4}$$

where $V(ESS, s)$, $V(NUR, s)$, and $V(VI, s)$ are the projected value functions and $C(SSS_i, s, ESS)$, $C(SSS_i, s, NUR)$ and $C(SSS_i, s, VI)$ are the completion functions. It should be noted that *ESS* is a primitive action and its projected value function is defined by:

$$V(ESS, s) = \sum\nolimits_{s'} P(s'|s, ESS) R(s'|s, ESS), \tag{5}$$

where P and R represent the probability transition function and the expected reward function, respectively. The action-value functions for the remaining subtasks can be defined in a similar manner.

When multiple robots are deployed to search the exact same sub-scene, each robot first has its own action-value functions and receives rewards for its own contribution to the relevant subtasks. This information is, then, utilized with similar information from the other robots in the sub-scene in order to determine the overall action-value function for the corresponding subtask. Cooperative learning occurs by each robot considering the actions of the other robots while updating its own projected value and completion functions. For example, when a sub-team of k rescue robots $\left\{Robot_i^1, \cdots, Robot_i^k\right\}$ cooperatively search sub-scene i, the projected value function and completion function for $Robot_i^j$ are defined by:

$$V^j\left(SSS_i, s, A_{o,SSS_i}\right) = Q^j\left(SSS_i, s, A_{o,SSS_i}, \pi_{SSS_i}\right), \tag{6}$$

$$\begin{aligned} &C^j\left(Root, s, A_{o,Root}, SSS_i\right) \\ &= \sum\nolimits_{s' \in S,N} \left\{ \begin{matrix} P_{Root}\left(s^{'}, N \,\middle|\, s, SSS_i\right) \gamma^N \\ Q^j\left(Root, s', A_{o,Root}, \pi_{Root}(s^{'})\right) \end{matrix} \right\}, \end{aligned} \tag{7}$$

where $A^l_{o,SSS_i} \in \{NUR, VI, ESS\}$ and $A^l_{o,Root} \in \{SSS_1, \cdots, SSS_n\}$, with $(l = 1, \cdots, k; l \neq j)$.

The projected value functions and completion functions are updated on-line. The projected value function for $Robot_i^j$ executing primitive action ESS in state s is updated by:

$$V_{t+1}^j\left(ESS, s, A_{o,SSS_i}\right) = (1-\alpha) \cdot V_t^j\left(ESS, s, , A_{o,SSS_i}\right) + \alpha \cdot \frac{r}{k}, \tag{8}$$

where α is the learning rate, t and $t+1$ represent the time instants when the corresponding value and completion function are determined, and r represents the total real-valued reward for searching sub-scene SS_i. Herein, $A^l_{o,SSS_i} = ESS(l = 1, \cdots, k; l \neq j)$ since all the robots cooperatively searching the same sub-scene terminate the subtask simultaneously.

The completion function $C^j(SSS_i, s, A_{o,SSS_i}, NUR)$ for $Robot_i^j$ is defined by:

$$\begin{aligned} C_{t+1}^j(\mathrm{SSS}_i, s, A_{o,SSS_i}, NUR) = {} & (1-\alpha) \cdot C_t^j(SSS_i, s, A_{o,SSS_i}, NUR) \\ & + \alpha \cdot \gamma^N \cdot \{C_t^j(SSS_i, s', A_{o,SSS_i}, \alpha^*) \\ & + V_t^j(\alpha^*, s', \hat{A}_{o,SSS_i})\}, \end{aligned} \tag{9}$$

where $\hat{A}_{o,SSS_i}$ represents the subtasks that the other robots will execute in state s'. The greedy action a^* is defined as:

$$\begin{aligned} a^* = {} & argmax_{a'} \left\{ \tilde{C}_t^j\left(SSS_i, s', \hat{A}_{o,SSS_i}, a'\right) \right. \\ & \left. + V_t^j\left(a', s', \hat{A}_{o,SSS_i}\right) \right\}, \end{aligned} \tag{10}$$

where $a' \in \{ESS, NUR, VI\}$. $\tilde{C}_t^j$ is a completion function only used to determine the locally optimal policy for subtask SSS_i, which is updated by:

$$\begin{aligned} \tilde{C}_{t+1}^j(SSS_i, s, A_{o,SSS_i}, \alpha) = {} & (1-\alpha) \cdot \tilde{C}_t^j(SSS_i, s, A_{o,SSS_i}, \alpha) \\ & + \alpha \cdot \gamma^N \cdot \{\tilde{R}^j(s') \\ & + \tilde{C}_t^j(SSS_i, s', \hat{A}_{o,SSS_i}, \alpha^*) \\ & + V_t^j(\alpha^*, s, \hat{A}_{o,SSS_i})\}, \end{aligned} \tag{11}$$

where the pseudo-reward $\tilde{R}^j\left(s'\right)$ indicates how desirable the terminal state s' is.

The action-value function for each robot in the sub-team is, then, used to determine the action-value function for all robots in the sub-team:

$$\begin{aligned} & Q^{sub-team}\left(SSS_i, s, \pi_{SSS_i}\right) \\ & = \sum_{j=1}^{k} \left\{ Q^j\left(SSS_i, s, A_{o,SSS_i}, \pi_{SSS_i}\right) \right\}. \end{aligned} \tag{12}$$

MAXQ recursively executes the learned policies for each subtask in the task hierarchy shown in Fig. 1. Due to state abstraction, the task hierarchy can be scaled-up to address problems with large numbers of sub-scenes and robots as well as different size environments.

The reward system used herein for MAXQ is based on our previous work, Table 1, [30]. Positive rewards are given to encourage transitions from a robot's current state to desirable states. Negative rewards are given when a transition is made from a robot's current state to an undesirable state. The reward values are chosen based on two criteria: (1) the rewards should encourage transitions from the robot's current state to desirable states, and to avoid transitions to undesirable states, and (2) potential benefits and costs should be used to determine the magnitudes of the rewards in order to promote convergence to optimal policies. For example, successfully exiting an explored

Table 1 MAXQ transition rewards for multi-robot USAR [30]

Subtask	Robot state transition	Reward
Root	The mission is completed successfully	+100
Search Sub-scene	Exit a sub-scene after it has been successfully explored	+50
Search Sub-scene	Exit a sub-scene when there are still accessible unknown cells.	−10
Navigate to Unvisited Regions	Exit into Standby after exploring all unvisited regions in the sub-scene	+10
Navigate to Unvisited Regions	Exit into Standby when there are still accessible unvisited regions	−10
Victim Identification	Tag a victim correctly	+10
Victim Identification	False identification by tagging an object that is not a victim	−10
Navigate	Move into an unvisited region in the desired global exploration direction	+15
Navigate	Avoid an obstacle	+10
Navigate	Collide with an obstacle, a victim or another robot in the team	−20
Navigate	Repeatedly revisit an explored region	−1
Human Control	Human Control is requested when necessary	+10
Human Control	Human Control is unnecessarily requested	−10

sub-scene is given a positive reward of +50. However, if the sub-scene is exited prior to all accessible unknown cells being explored, a negative reward of −10 is given.

The MAXQ semi-autonomous controller was trained for multi-robot teams. The training results for a team of ten robots are presented in Fig. 2. A single trial consists of a group of robots exploring one environment with multiple sub-scenes. An episode is a single primitive action for one robot. The results show convergence after approximately 407,697 episodes (20 trials). During the USAR experiments, the trained model is updated on-line to adapt to new unknown simulated USAR environments. Therefore, through learning, MAXQ determines what subtasks to select and what actions to take in specific states in order to maximize the cumulative rewards for the overall task.

Our implementation of the semi-autonomous controller allows the operator to provide more than one primitive action when the controller requests for help. Actions by the operator are recorded into the existing 2D occupancy map. When the operator hands back control over to the semi-autonomous controller, the latter learns from the information gathered by the operator. This provides a robust approach in searching USAR scenes with human assistance.

3.1 An Example Scenario

Let us consider the scenario where one robot $Robot^j$ is deployed to search one sub-scene at the top-level. Also, let us assume that in State s, some sub-scenes have been explored or are being explored by one or more robots while the remaining sub-scenes are unexplored, e.g., Sub-scene SS_i is unexplored. Following the learned policy for the *Root* task, Subtask SSS_i is selected. In order to compute the value function for the *Root* task in the aforementioned State s, the action-value function for executing Sub-task SSS_i is calculated. Upon the completion of the Sub-task SSS_i, the robot is given a real-valued reward of +50 when it exists the sub-scene. When all sub-scenes have been searched by the robot team, a pseudo-reward of +100 is given to the overall team at the *Root* task.

While the robot executes the Sub-task SSS_i, one of the lower-level sub-tasks, NUR or VI, is selected according to the learned policy for the sub-task SSS_i. For example, when

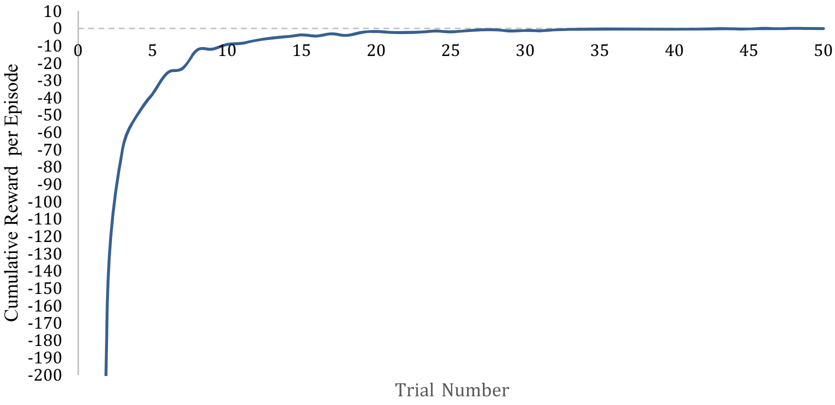


Fig. 2 Cumulative reward received per episode in each trial for 10 robots

unvisited regions are detected in the sub-scene, the Sub-task NUR is selected. The robot is given a real-valued reward of +10 for completing the Sub-task NUR and exiting into *Standby*.

In order to execute the Sub-task NUR, the lower-level Sub-task $Navigate$ is called. Let us assume that the cells immediately surrounding the robot consist of a non-climbable cell in front, and open cells to the right, left and back. The map of the environment indicates that there are still unvisited unknown cells in the region to the right and left of the robot. In such a scenario, the primitive action rotate is selected since the robot will receive a real-valued reward of +19 (+10 for avoiding an obstacle, +10 for moving into an unvisited region in the desired global exploration direction, and -1 for taking the rotation action within an already visited cell), which is higher than the potential rewards for move forward (−25 for colliding with an obstacle) and move backwards (+9, including +10 for avoiding an obstacle and −1 for taking the move back action into an already visited cell).

4 Proposed Multi-robot Single-Operator System Architecture

Our proposed system architecture for semi-autonomous control of a multi-robot team is shown in Fig. 3. The system encompasses both a user interface and a MAXQ HRL-based deliberation layer. In the teleoperation mode, the MAXQ HRL-based deliberation layer is not present, and the user interface is used to directly control the robots individually.

4.1 Robot Sensors

Each robot is equipped with four 3D sensors used to provide depth information about its surroundings. This information is used to classify the terrain as open space, climbable, or non-climbable obstacles, as well as to build a map of the cluttered environment. Each robot has an inertial navigation system (INS) for tracking the robot's position and orientation within the environment, and a 2D camera that provides video streaming to the operator and is also used for victim identification by the robot. In semi-autonomous mode, victim identification is implemented by analyzing 2D images provided by the camera using a skin-detection method [45]. The victim's location is, then, tagged on the map. In teleoperation mode, victim identification is achieved by the operator using the 2D video stream.

4.2 Mapping

The mapping module receives 3D information of the USAR environment from the 3D sensors mounted on the robot, and uses this information to classify the terrain. Accessible regions are classified as open or climbable obstacles. Inaccessible regions are classified as non-climbable obstacles. Terrain classification is accomplished by fitting a plane to the 3D data using a least-squares method. The slope of the plane is used to determine whether regions are traversable (i.e., climbable or non-climbable). This module also uses the information from the INS to localize the team of robots within the map.

A 2D occupancy grid map is used to represent terrain information as well as the locations of the victims in the environment. Grid cells are labeled as

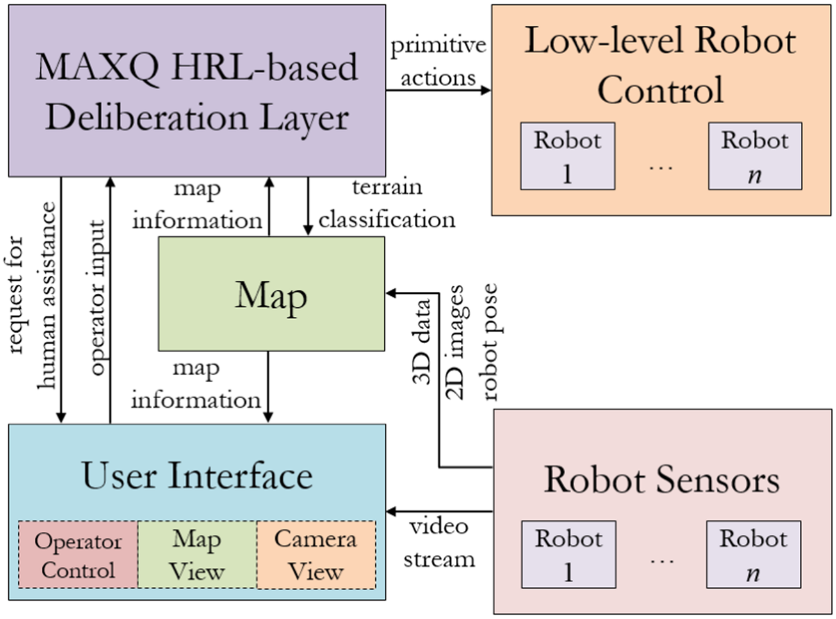


Fig. 3 System architecture for multi-robot rescue team in semi-autonomous mode

open, climbable, non-climbable, and victim cells. The accessibility of a cell is determined by the terrain properties of the USAR environment (more details are provided in [27]). This approach allows detailed mapping of 3D cluttered environments by providing information about the traversability of the cells the robots are exploring. The 2D occupancy map information is sent to both the semi-autonomous controller and the user interface. The global map of the USAR scene can be viewed by combining together all the individual sub-scene maps generated by each robot in the USAR environment.

In teleoperation mode, terrain classification is not available. A 2D occupancy grid is provided, only consisting of visited regions and victim locations.

4.3 MAXQ HRL-Based Deliberation Layer

The objective of introducing the MAXQ HRL technique (as described in Section 3) into the Deliberation Layer is to have the robot team learn from its own experiences and those of an operator in order to effectively perform tasks in USAR environments [3]. By introducing the MAXQ HRL technique, a rescue robot team can cooperatively learn and determine which tasks should be executed at a given time, and decide whether a rescue robot or an operator should carry out those tasks to achieve optimal performance. The root mission task is decomposed into subtasks that are executed sequentially by individual robots in order to search the sub-scenes that they are assigned to. However, since each robot has a copy of the task hierarchy, the team as a whole implements these subtasks in parallel, as we proposed in [30].

4.4 User Interface

A user interface was developed for handling communication between the operator and the multi-robot team in both semi-autonomous and teleoperation modes, Fig. 4. The interface comprises three main modules: (1) multi-robot operator control (bottom of interface), (2) map view (top right corner of interface), and (3) 2D camera view (top left corner of interface).

The operator control input module handles various user inputs from an XBOX gamepad that are used to teleoperate the robots. These control inputs include moving a robot forward/backward, turning the robot, tagging a victim, and switching between different robots in a team. In addition, the operator can also return control back to a robot in the semi-autonomous mode.

The map view module receives map information from the map (generation) module and displays the map of already visited regions and victim locations in the sub-scenes. The camera view output module displays a live video stream for each individual robot. For the semi-autonomous mode, the user interface also alerts the operator when a robot needs human assistance. Such directing of attention has been shown to improve HRI performance [46].

The robot team view also provides the status of the robots in the team. The operator is limited to controlling one robot at a time. The team view also indicates whether a particular robot is being controlled manually (MAN) via teleoperation or by the semi-autonomous (AUTO) controller. The user interface also provides the operator with an occupancy grid map of the USAR scene, in addition to the camera view of the robot.

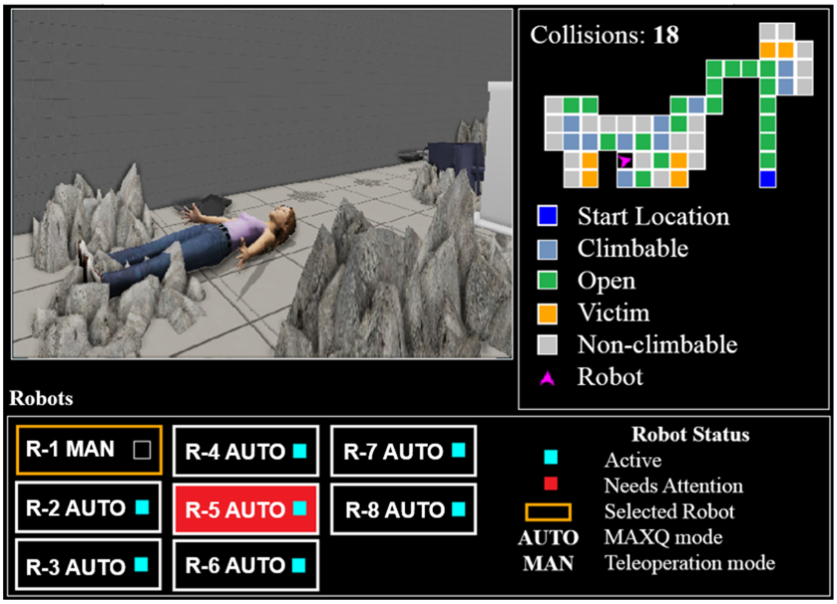


Fig. 4 User interface for human operator

Our interface follows an adaptive interface design. Namely, adaptive interfaces can automatically make decisions regarding the need for automation as well as change the degree of automation in order to decrease operator workload and the number of required human operator interventions, and increase situational awareness [47–49]. In semi-autonomous mode, the user interface is interconnected with the MAXQ HRL-based deliberation layer. The team of robots moves autonomously until a robot requires human assistance. During operator input, the robot control status switches to MAN mode to indicate that the robot is under the operator's control. In teleoperation mode, the MAXQ HRL-based deliberation layer is not present. The user interface directly sends operator inputs to control the team of robots. The operator has full control over each individual robot in the team.

4.5 Low-Level Robot Control

In the semi-autonomous mode, the primitive actions (i.e., move forward, move backward, turn) from the MAXQ HRL-based deliberation layer are converted into motion commands for the team of robots. In teleoperation mode, the operator has direct control of the motion of the robots and the low-level controller is used to process operator commands into low-level motion commands.

4.6 USARSim

USARSim was used as our 3D simulation environment [50]. The USARSim platform was used together with the Unreal Developer's Kit (UDK) game engine [51]. This provided the capability to create realistic unstructured USAR environments consisting of both rubble and victims. Herein, rubble is defined by concrete piles and overturned furniture. Our aforementioned system architecture contains a library of extensive functions for communicating with USARSim.

USARSim was utilized due to its high-fidelity simulation capabilities of USAR robots and environments, which can be used to investigate HRI and multi-robot coordination [52]. In particular, USARSim has already been validated for its ability to provide accurate models of robot geometry and kinematic design, sensors, and environments [53]. Furthermore, USARSim allows multiple robots to be used concurrently [54], and has been used extensively in USAR mission studies, which require operators to search for victims while exploring an environment with multiple robots [20, 31, 55–57]. Therefore, in this work, USARSim has been used to accommodate the large groups of robots (up to 20 robots) and varying environment sizes we are considering and to ensure repeatability across our different test conditions.

4.7 Software Implementation

The software components include the UDK game engine running USARSim, Multi-Robot Operator Team (MROT), and the MAXQ HRL-based deliberation program, Fig. 5. We integrated MROT, our custom multi-robot remote control application for USARSim, with the MAXQ program, our semi-autonomous controller, to build an effective command console for operators monitoring multiple semi-autonomous robots in real-time within USARSim. With our implementation, operators can, at a glance, monitor the status of all robots in the team. Operator intervention is made seamless by detecting when an operator intends to take control of a robot, or when the MAXQ program requests assistance. Following an operator intervention, control is transferred back to the MAXQ program with a single input.

5 Experiments in Simulated Environments

The experiments consisted of operators controlling teams of robots in the simulated environment of USARSim to investigate the influence of the operator-to-robot ratio on the performance of the HRL-based semi-autonomous controller as well as on the full teleoperation control of the robots. We used a combination of effectiveness and efficiency metrics to evaluate the controller [58]. Namely, the performance metrics used were: (1) percentage of scene coverage, (2) percentage of victims found, (3) number of robot team collisions in the environment, and (4) total exploration time.

For our experiments, we also measured the human-performance metrics of Interaction Effort (IE), using operator control time to show an operator's effort in controlling a robot team [18, 58], and Task Performance (*TP*) to provide an overall metric for mission performance (exploration and victim identification) [19]. *TP* enables the identification of bottlenecks in the system (i.e., robot team size), in which performance can be negatively impacted [59], while reducing *IE* can directly improve the effectiveness of the human-robot team [18].

5.1 Procedure

Twenty-one people participated in the trials, ranging from 23 to 36 years in age ($\mu = 25.9$, $\sigma = 3.6$). All participants were engineering students. None had prior experience with controlling a USAR robot, however, they had varying expertise in playing 3D video games, ranging from none to little experience (43%), as well as moderate to more experienced (57%).

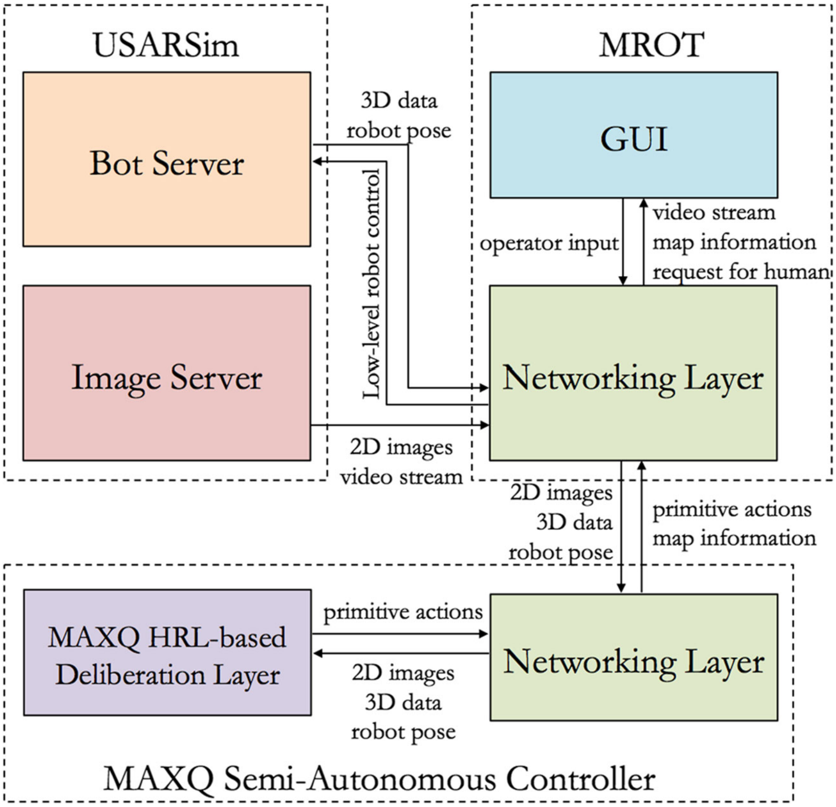


Fig. 5 Software schematic diagram

Each trial consisted of having the participant control a team of 5, 10, 15, and 20 robots in both semi-autonomous mode and teleoperation mode, respectively. Namely, each participant controlled four different team configurations for each mode (total of 8 trials per participant). The USAR scenes used for the experiments occupied 288 m^2, 544 m^2, 944 m^2, and 1184 m^2 for the 5, 10, 15, and 20 robot teams, respectively. Each USAR scene was divided into smaller sub-scenes. The overall size of each sub-scene varied, ranging from 32 to 80 m^2, the amount of clutter ranged from 60% to 75% of the overall sub-scene, and the number of victims ranged from one to four.

Figure 6 provides several examples of rubble piles and victim configurations within the sub-scenes. The Pioneer P3AT mobile robotic platforms were used, which contained four 3D sensors located to scan the front, left, right, and back of each robot for terrain classification, an INS, and a 2D camera with 320 × 240 pixel resolution.

A counter-balance approach was used; half of the participants started each trial in teleoperation mode, and the remaining in semi-autonomous mode. After finishing the trials in each mode, participants switched to the next mode. To avoid carry-over effects there was an average of three days of rest between the two modes. The robot team size configurations in each mode were randomized for each participant (e.g., 5-20-15-10, 15-20-5-10). Each participant had ten minutes of training with respect to the gamepad control inputs and the user interface prior to the experiments. The objective for the operator was to explore the USAR environment to cover as much area as possible and to identify as many victims within the overall environment with no time limits. They were told to explore the environment by traversing the cluttered terrain. After the experiments were completed, the participants were asked to complete a 5-point Likert questionnaire (5—Strongly Agree, 1—Strongly Disagree) based on their experiences.

5.2 Results and Discussion

A (one-tailed) statistical power analysis was first conducted to confirm that the sample size was sufficient with respect to the performance metrics. We obtained powers greater than 0.99 ($p < 0.05$) for all the metrics.

During semi-autonomous operations, all robots in the team worked in parallel and asked for assistance from the operator only when required. Thus, for example, we expected some increase in exploration time with increased environment size and robot team size. In teleoperation mode, however, the hypothesis was that a more significant increase in exploration time would occur with increased robot team size since each individual robot requires the operator's attention at all times. Table 2 shows the average values and

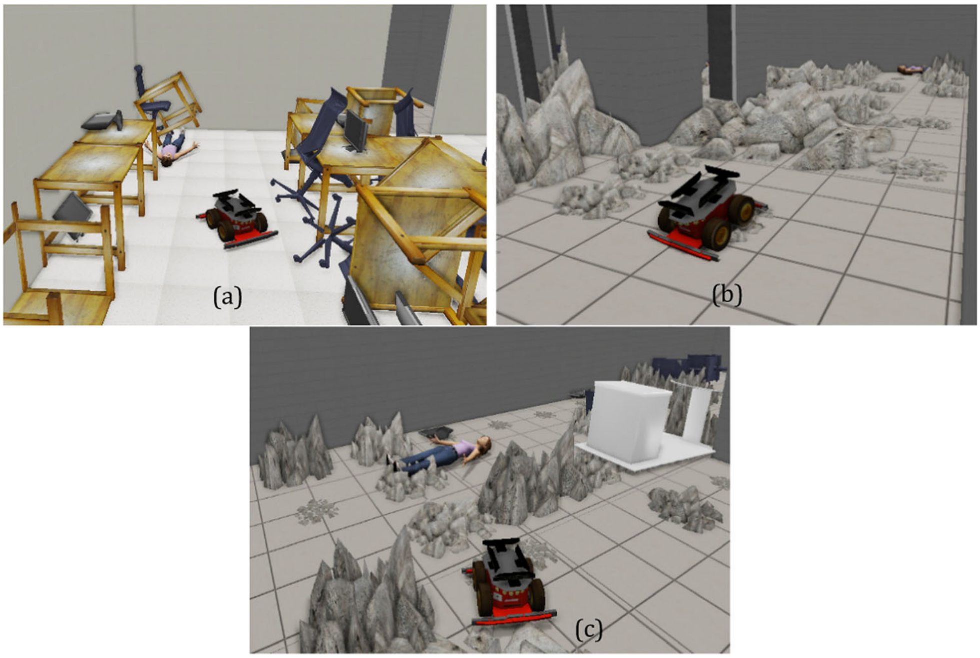


Fig. 6 Example sub-scenes: **a** with overturned furniture, **b** with climbable and non-climbable concrete piles, and **c** a combination of both furniture and concrete piles

ranges for the collected performance metrics for the experiments.

Figure 7 shows the relationship between the number of robots in a team and the average exploration time. A second-order-polynomial least-squares fit was utilized for both the semi-autonomous and teleoperation cases. As expected, in both the semi-autonomous and the teleoperation modes, the total exploration time increases as the number of robots increases for larger USAR scenes. However, the rate of increase is significantly higher for teleoperation as we had hypothesized. This mode also results in a greater number of robot collisions with the environment. It should be noted that collision avoidance is not one of the primary tasks in USAR missions, and therefore the operators were not explicitly asked to avoid obstacles. We postulate, if we had told the operators to avoid obstacles, the number of

Table 2 Varying robot-team size performance metrics

Exploration	# of Robots on	Average metric value (range for all participants)				
mode	a team	# of Victims	% of Scenes explored	# of Collisions	% of Victims found	Total exploration time (s)
Semi-Autonomous	5	12	100 (100–100)	0.3 (0–7)	100 (100–100)	101 (76–162)
	10	20	100 (100–100)	0.5 (0–4)	100 (100–100)	129 (90–192)
	15	30	100 (100–100)	6.2 (1–15)	100 (100–100)	184 (109–344)
	20	37	100 (100–100)	9.5 (0–27)	100 (100–100)	195 (126–284)
Tele-operation	5	12	93 (81–100)	10.8 (0–48)	98 (83–100)	319 (168–811)
	10	20	88 (76–100)	17.8 (0–57)	98 (90–100)	516 (277–1082)
	15	30	89 (77–100)	28.4 (4–77)	98 (93–100)	916 (552–1688)
	20	37	88 (81–97)	55.1 (16–174)	97 (92–100)	1405 (737–3386)

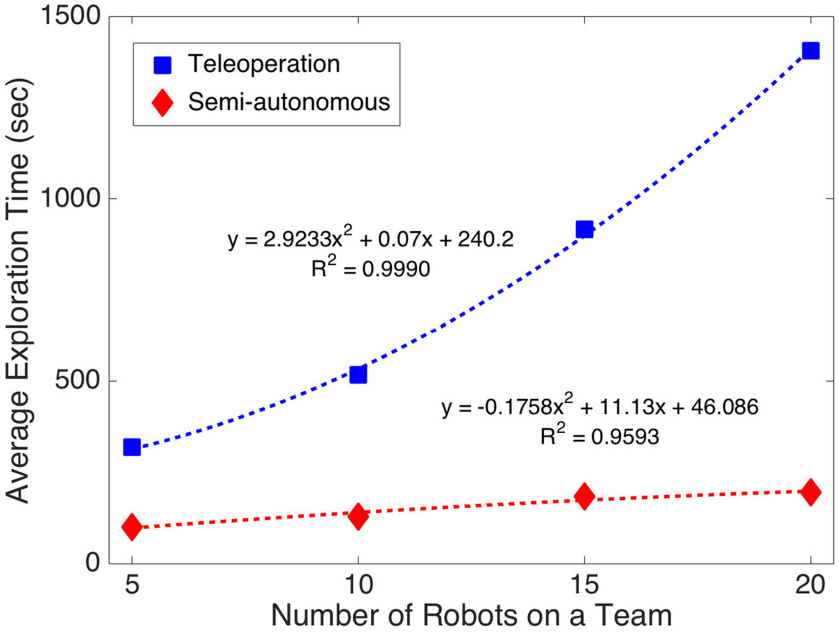


Fig. 7 Total exploration time for all participants controlling 5, 10, 15, and 20 robots in both control modes

collisions could potentially decrease, but in return the total exploration time would increase as they would be more cautious while moving the robots in the environment [60].

For easier comparison, when linear least-squares was utilized, the slopes were determined to be about 6.7 s/robot versus 73.2 s/robot, with a confidence level of more than 94.64%. Namely, the results confirm the difficulty an operator would face when trying to control a large team of rescue robots, exploring the scenes sequentially. Furthermore, when using the semi-autonomous controller, robot teams were able to explore 100% of the scenes and identify all the victims while minimizing the number of collisions they had in the environment.

Operator interaction effort, IE_h, for each robot team size for participant h was defined as:

$$IE_h = \frac{O_{th}}{E_{th}}, \tag{13}$$

where O_{t_h} represents the total time Participant h was controlling the robots, and E_{t_h} represents the total exploration time for that participant.

In order to determine the overall task performance, a weighted linear combination of the effectiveness metrics was used. In particular, task performance, TP_h, for Participant h, was defined as:

$$TP_h = w_1 S_h + w_2 C_h + w_3 V_h, \tag{14}$$

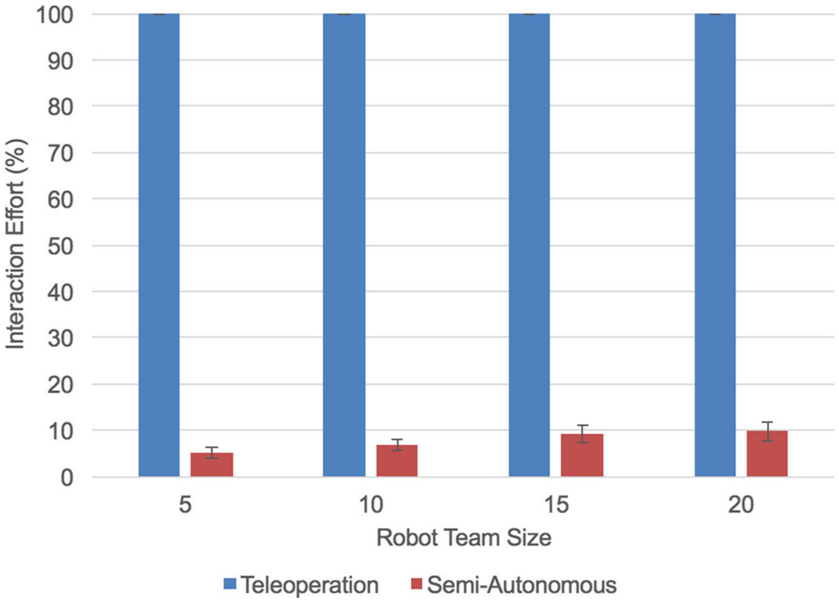


Fig. 8 Interaction effort between control modes

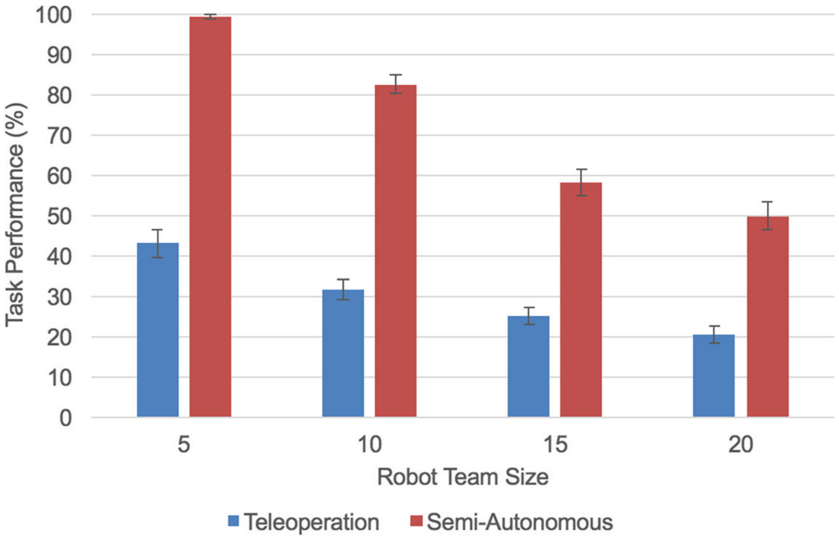


Fig. 9 Task performance between control modes

where S_h is the percentage of scene explored, C_h is the number of collisions, V_h is the percentage of victims found and w_1, w_2, and w_3 are performance weights $\left(\sum w_i = 1\right)$:

$$S_h = \frac{S_{t_h}}{\max S_{t_h}}, \tag{15}$$

$$C_h = \frac{\max C_{t_h}/C_{t_h}}{\max\left(\max C_{t_h}/C_{t_h}\right)}, \tag{16}$$

$$V_h = \frac{V_{t_h}}{\max V_{t_h}}. \tag{17}$$

Above, S_{t_h}, C_{t_h}, and V_{t_h} represent the percentage of scenes explored, number of collisions, and percentage of victims found over E_{t_h}, respectively. The average operator IE and TP per robot team size is presented in Figs. 8 and 9, respectively.

An analysis of variance (ANOVA) test was performed to determine statistical significance for all performance metrics. The results showed that semi-autonomous mode was significantly better compared to teleoperation mode in all performance metrics regardless of the robot team size: (1) percentage of scene exploration, $F\,(1, 160) = 279.0$, $p < 0.001$; (2) percentage of victims found, $F\,(1, 160) = 40.42$, $p < 0.001$; (3) total exploration time $F\,(1, 160) = 201.7$, $p < 0.001$; and (4) total number of collisions, $F\,(1, 160) = 77.79$, $p < 0.001$.

Table 3 Comparison of learning, non-learning and teleoperation for exploration

Exploration mode	# of Robots on a team	% of Scenes explored
Learning-based	1	100
	4	100
Non-Learning based	1	24
	4	61
Tele-operation	1	44
	4	93

With respect to human performance metrics, statistical significance was also determined between the control modes regardless of the robot team size for: (1) IE, $F\,(1, 160) = 12740$, $p < 0.001$; and, (2)

Table 4 Post-Experiment Questionnaire

Statement	Mean	SD
1. With the information provided, I was able to visualize the layout of the environment in:		
a. Teleoperation mode.	4.5	0.5
b. Semi-autonomous mode.	3.9	1.1
2. I had a difficult time monitoring all of the sensory information in:		
a. Teleoperation mode.	2.4	1.4
b. Semi-autonomous mode.	2.6	1.2
3. The user interface was easy to use.	4.4	0.5
4. I had confidence in the robots performing their tasks in semi-autonomous control.	3.8	0.7
5. I had an easy time controlling all the robots in:		
a. Teleoperation mode.	3.5	1.4
b. Semi-autonomous mode.	4.3	0.9
6. I felt stressed during:		
a. Teleoperation mode.	3.1	1.0
b. Semi-autonomous mode.	1.9	1.0
7. In general, the semi-autonomous mode was more stressful for me.	2.1	1.1
8. I had a better overall experience in:		
a. Teleoperation mode.	3.1	1.2
b. Semi-autonomous mode.	4.1	0.7
9. Given a choice, I would choose manual teleoperation over semi-autonomous control.	1.7	0.9

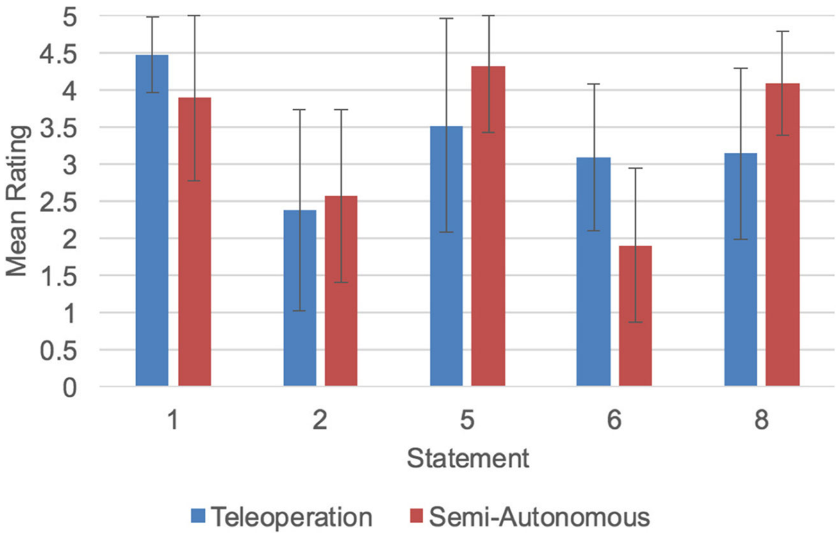


Fig. 10 Mean ratings of post-experiment questionnaire directly comparing the teleoperation and semi-autonomous modes

TP, $F(1, 160) = 506.2$, $p < 0.001$. Namely, both interaction effort and task performance were significantly better for the semi-autonomous mode compared to the teleoperation mode. We also found no significant correlation between 3D video games experience of the participants and TP, $(\rho = -0.1,\ p = 0.527)$.

From Table 2, when comparing performance across the robot team sizes using the learning-based semi-autonomous controller, one can note that regardless of the number of robots on the team, 100% scene coverage and victim identification is obtained. The increase in total exploration time can be due to the fact that the robots are exploring larger scenes (i.e., 288 m^2 for 5 robots and 1184 m^2 for 20 robots). The decrease in *TP* is due primarily to this increase in exploration time and the number of collisions. However, the rate of decrease slows down as the number of robots on the team increases. On the other hand, *IE* between the team sizes only slightly increases.

5.3 Exploration with and Without Autonomy and Learning

In order to further investigate the impact of learning and autonomy on robot exploration, we conducted a separate experiment in USARSim with a single robot and a group of four robots in a 96 m^2 highly cluttered USAR scene. The experiment compared our learning-based MAXQ technique with: (1) an existing autonomous non-learning technique for direction-based exploration [26], and (2) teleoperation. Thirteen participants from the same demographics were used for this test condition. The results are presented in Table 3.

The percentage of scene coverage for this scene, with one robot, was 100% using MAXQ, 24% using the non-learning technique, and 44% with teleoperation. With a team of four robots, the scene coverage percentage was 100% using MAXQ, 61% using the non-learning technique, and 93% with teleoperation. In the case of the non-learning technique, the lower scene coverage was due to the robots getting into situations where they became trapped in corners or rubble piles, whereas when using MAXQ, the robots were able to avoid such situations through learning. With respect to teleoperation, when comparing the 4 robot team, the operator required, on average, 4.8 times the exploration time (for 93% scene coverage) than MAXQ, which is consistent with the results discussed above.

5.4 Post-Experiment Questionnaire

The results from the 5-point Likert questionnaire showed that participants had a better overall experience using the semi-autonomous controller (with a mean of 4.1 for this question versus a mean of 3.1 for teleoperation) and felt less stressed during the USAR mission (with a mean of 1.9 versus a mean of 3.1 for teleoperation), Table 4. Participants preferred the semi-autonomous controller over solely teleoperated robots due to the task handling capabilities of the former—with no apparent a priori design bias toward either mode of operation (e.g., Questions 1 and 2). The standard deviation for each question is also presented in Table 4 to show the variability of the questionnaire results [61]. The participants' responses for all questions are fairly consistent with no major deviations. Figure 10 provides a graphical representation of the direct comparison of the statements 1, 2, 5, 6, and 8 for both the teleoperation and semi-autonomous modes.

6 Conclusions

This paper investigated the influence of the operator-to-robot ratio on the performance of our unique system architecture for using a learning-based semi-autonomous controller to aid an operator in a multi-robot team USAR. Experiments showed that operator performance improved significantly when aided by our MAXQ learning-based semi-autonomous controller. With the MAXQ semi-autonomous controller, operators can cover more area in a shorter time, and exhibit greater patience in exploration. Our semi-autonomous controller architecture proved to be more effective when the operator is controlling a large robot team compared to pure teleoperation. In addition, operator interaction effort decreased significantly when compared to teleoperation. A case study comparing our approach to an autonomous non-learning based method for exploration also showed that our MAXQ semi-autonomous method provided more scene coverage in cluttered environments. In conclusion, learning-based semi-autonomous controllers have the potential to provide both multi-robot system (e.g. coverage, time) and human operator (e.g. task performance, interaction effort) performance improvements over teleoperated control of large groups of robots. These controllers can be used for exploration, and person and object identification tasks in cluttered environments. The learning capabilities also allow robots to adapt to new and unknown environments.

Acknowledgments This work was funded by the Natural Science and Engineering Research Council of Canada (NSERC) and the Canada Research Chairs (CRC) Program.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Casper, J., Micire, M., Murphy, R.: Issues in intelligent robots for search and rescue. In: Proceedings SPIE Unmanned Ground Vehicle Technology II, pp. 292–302 (2000)
2. Casper, J., Murphy, R.: Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. IEEE Trans. Syst. Man Cybern. B **33**(3), 367–385 (2003)
3. Liu, Y., Nejat, G.: Robotic urban search and rescue: a survey from the control perspective. J. Intell. Robot. Syst. **72**(2), 147–165 (2013)
4. Wong, C., Seet, G., Sim, S.: Multiple-robot systems for USAR: key design attributes and deployment issues. Int. J. Adv. Robot. Syst. **8**(1), 85–101 (2011)
5. Gatsoulis, Y., Virk, G.S., Dehghani-Sanij, A.A.: On the measurement of situation awareness for effective human-robot interaction in teleoperated systems. J. Cogn. Eng. Decis. Mak. **4**(1), 69–98 (2010)
6. Adams, J.A.: Multiple robot/single human interaction: effects on perceived workload. Behav. Inf. Technol. **28**(2), 183–198 (2009)
7. de Visser, E., Parasuraman, R.: Adaptive aiding of human-robot teaming effects of imperfect automation on performance, trust, and workload. J. Cogn. Eng. Decis. Mak. **5**(2), 209–231 (2011)
8. Chen, J.Y.C., Durlach, P.J., Sloan, J.A., Bowens, L.D.: Human-robot interaction in the context of simulated route reconnaissance missions. Mil. Psychol. **20**(3), 135–149 (2008)
9. Breslow, L.A., Gartenberg, D., McCurry, J.M., Trafton, J.G.: Dy7namic operator overload: a model for predicting workload during supervisory control. IEEE Trans. Hum. Mach. Syst. **44**(1), 30–40 (2014)
10. McKendrick, R., Shaw, T., de Visser, E., Sager, H., Kidwell, B., Parasuraman, R.: Team performance in networked supervisory control of unmanned air vehicles effects of automation, working memory, and communication content. Hum. Factors **56**(3), 463–475 (2014)
11. Fincannon, T.D., Evans, A.W., Phillips, E., Jentsch, F., Keebler, J.: The influence of team size and communication modality on team effectiveness with unmanned systems. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, pp. 419–423 (2009)
12. de Visser, E., Kidwell, B., Payne, J., Lu, L., Parker, J., Brooks, N., Chabuk, T., Spriggs, S., Freedy, A., Scerri, P., Parasuraman, R.: Best of both worlds design and evaluation of an adaptive delegation interface. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, pp. 255–259 (2013)
13. Cummings, M.L., Bertucelli, L.F., Macbeth, J., Surana, A.: Task versus vehicle-based control paradigms in multiple unmanned vehicle supervision by a single operator. IEEE Trans. Hum. Mach. Syst. **44**(3), 353–361 (2014)
14. Prewett, M.S., Saboe, K.N., Johnson, R.C., Coovert, M.D., Elliot, L.R.: Workload in human-robot interaction: a review of manipulations and outcomes. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, pp. 1393–1397 (2009)
15. Velagapudi, P., Scerri, P., Sycara, K., Wang, H., Lewis, M., Wang, J.: Scaling effects in multi-robot control. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2121–2126 (2008)
16. Hsieh, M.A., Cowley, A., Keller, J.F., Chaimowicz, L., Grocholsky, B., Kumar, V., Taylor, C.J., Endo, Y., Arkin, R.C., Jung, B., Wolf, D.F., Sukhatme, G.S., MacKenzie, D.C.: Adaptive teams of autonomous aerial and ground robots for situation awareness. J. Field Robot. **24**(11), 991–1014 (2007)
17. Zhang, K., Xiaobo, L.: Human-robot team coordination that considers human fatigue. J. Adv. Robot. Syst. **11**(6) (2014)
18. Olsen, D.R., Goodrich, M.A.: Metrics for evaluating human–robot interactions. In: Proceedings of the PERMIS (2003)
19. Gatsoulis, Y., Virk, G., Dehghani-Sanij, A.: On the measurement of situation awareness for effective human-robot interaction in teleoperated systems. J. Cogn. Eng. Dec. Mak. **4**(1), 69–98 (2010)
20. Gao, F., Cummings, M.L., Solovey, E.T.: Modeling teamwork in supervisory control of multiple robots. IEEE Trans. Hum. Mach. Syst. **44**(4), 441–453 (2014)
21. Lewis, M., Wang, H., Chien, S.Y.: Process and performance in human-robot teams. J. Cogn. Eng. Decis. Mak. **5**(2), 186–208 (2011)
22. Gunn, T., Anderson, J.: Dynamic heterogeneous team formation for robotic urban search and rescue. J. Comp. Syst. Sci. **81**(3), 553–567 (2015)
23. Rosenfeld, A., Agmon, N., Maksimov, O., Azaria, A., Kraus, S.: Intelligent agent supporting human-multi-robot team collaboration. In: Proceedings of the AAMAS Workshop ARMS (2015)

24. Nourjou, R., Smith, S.F., Hatayama, M., Szekely, P.: Intelligent algorithm for assignment of agents to human strategy in centralized multi-agent coordination. J. Softw. **9**(10), 2586–2597 (2014)
25. Chen, J.Y.C., Barnes, M.: Supervisory control of multiple robots: effects of imperfect automation and individual differences. Hum. Factors: J. Hum. Factors Ergon. Soc. **54**(2), 157–174 (2012)
26. Vilela, J., Liu, Y., Nejat, G.: Semi-autonomous exploration with robot teams in urban search and rescue. In: IEEE International Symposium on Safety, Security, and Rescue Robotics, pp. 1–6 (2013)
27. Doroodgar, B., Liu, Y., Nejat, G.: A learning-based semi-autonomous controller for robotic exploration of unknown disaster scenes while searching for victims. IEEE Trans. Cybern. **44**(12), 2719–2732 (2014)
28. Niroui, F., Sprenger, B., Nejat, G.: Robot exploration in unknown cluttered environments when dealing with uncertainty. In: Proceedings of the IEEE International Symposium on Robotics and Intelligent Sensors (2017)
29. Liu, Y., Nejat, G., Doroodgar, B.: Learning-based semi-autonomous control for robots in urban search and rescue. In: Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, pp. 1–6 (2012)
30. Liu, Y., Nejat, G.: Multirobot cooperative learning for semiautonomous control in urban search and rescue applications. J. Field Robot. **33**(4), 512–536 (2016)
31. Brooks, N., Scerri, P., Sycara, K.: Asynchronous control with ATR for large robot teams. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, pp. 444–448 (2011)
32. Wang, H., Lewis, M., Chien, S., Velagapudi, P.: Scaling effects for synchronous vs. asynchronous video in multi-robot search. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, pp. 364–368 (2009)
33. Sato, N., Matsuno, F., Yamasaki, T., Kamegawa, T., Shiroma, N., Igarashi, H.: Cooperative task execution by a multiple robot team and its operators in search and rescue operations. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1083–1088 (2004)
34. Lewis, M., Wang, H., Chien, S.Y.: Choosing autonomy modes for multirobot search. Hum. Factors **52**(2), 225–233 (2010)
35. Mouaddib, A., Zilberstein, S., Beynier, A., Jeanpierre, L.: A decision-theoretic approach to cooperative control and adjustable autonomy. In: Proceedings of the European Conference on Artificial Intelligence, pp. 971–972 (2010)
36. Côté, N., Canu, A., Bouzid, M., Mouaddib, A.: Humans-robots sliding collaboration control in complex environments with adjustable autonomy. In: Proceedings of the IEEE/WIC/ACM International Conference Web Intelligence and Intelligent Agent Technology, pp. 146–153 (2012)
37. Wray, K.H., Pineda, L., Zilberstein, S.: Hierarchical approach to transfer of control in semi-autonomous systems. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 517–523 (2016)
38. Zhang, K., Collins, E.G., Shi, D.: Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction. J. Auton. Adapt. Syst. 7(2) (2012)
39. Zhang, K., Collins, E.G., Barbu, A.: An efficient stochastic clustering auction for heterogenous robotic collaborative teams. J. Intell. Robot. Syst. **72**(3–4), 541–558 (2013)
40. Zhang, K., Collins, E.G., Barbu, A.: A novel stochastic clustering auction for task allocation in multi-robot teams. In: Proceedings of the IEEE/RSJ International Conference on Intelligent and Robotic Systems, pp. 3300–3307 (2010)
41. Zhang, K., Collins, E.G., Shi, D., Liu, X., Chuy, O.: A stochastic clustering auction (SCA) for centralized and distributed task allocation in multi-agent teams. Distrib. Auton. Robot. Syst. **8**, 345–354 (2009)
42. Zhang, Z., Littman, M., Chen, X.: Covering number as a complexity measure for POMDP planning and learning. In: AAAI Conference on Artificial Intelligence, pp. 1853–1859 (2012)
43. Dietterich, D.: Hierarchical reinforcement learning with MAXQ value function decomposition. J. Artif. Intell. Res. **13**, 227–303 (2000)
44. Liu, Y., Nejat, G., Vilela, J.: Learning to cooperate together: a semi-autonomous control architecture for multi-robot teams in urban search and rescue. In: IEEE International Symposium on Safety, Security, and Rescue Robotics (2013)
45. Doroodgar, B., Nejat, G.: A hierarchical reinforcement learning based control architecture for semi-autonomous rescue robots in cluttered environments. In: IEEE International Conference on Automation Science and Engineering, pp. 948–953 (2010)
46. Chien, S., Wang, H., Lewis, M.: Effects of alarms on control of robot teams. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, pp. 434–438 (2011)
47. Oppermann, R.: Adaptive User Support: Ergonomic Design of Manually and Automatically Adaptable Software. CRC Press, Boca Raton (1994)
48. Parasuraman, R., Galster, S., Squire, P., Furukawa, H., Miller, C.: A flexible delegation-type interface enhances system performance in human supervision of multiple robots: empirical studies with RoboFlag. IEEE Trans. Cybern. **35**(4), 481–493 (2005)
49. Clare, A., Cummings, M., How, J., Whitten, A., Toupet, O.: Operator object function guidance for a real-time unmanned vehicle scheduling algorithm. J. Aerosp. Comput. Inf. Commun. **9**(4), 161–173 (2012)
50. Sourceforge. USARSim. [Online]. http://sourceforge.net/projects/usarsim/
51. Epic Games Inc. UDK. [Online]. http://www.unrealengine.com/udk
52. Lewis, M., Wang, J., Hughes, S.: USARSim: simulation for the study of human-robot interaction. J. Cognitive Eng. Decis. Mak. **1**(1), 98–120 (2007)
53. Wang, J., Lewis, M., Hughes, S., Koes, M.: Validating USARsim for use in HRI Research. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, pp. 457–461 (2005)
54. Carpin, S., Lewis, M., Wang, J., Balakirsky, S., Scrapper, C.: USARSim: a robot simulator for research and education. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1400–1405 (2007)
55. Balaguer, B., Balakirsky, S., Carpin, S., Visser, A.: Evaluating maps produced by urban search and rescue robots: lessons learned from RoboCup. J. Auton. Robot. **27**(4), 449–464 (2009)
56. Shiroma, N., Chiu, Y., Sato, N., Matsuno, F.: Cooperative task execution of a search and rescue mission by a multi-robot team. J. Adv. Robot. **19**(3), 311–329 (2005)
57. Calisi, D., Farinelli, A., Iocchi, L., Nardi, D.: Multi-objective exploration and search for autonomous rescue robots. J. Field Robot. **24**(8-9), 763–777 (2007)
58. Steinfeld, A., Fong, T., Kaber, D., Lewis, M., Scholtz, J., Schultz, A., Goodrich, M.: Common metrics for human-robot interaction. In: Proceeding of the ACM SIGCHI/SIGART Conference on Human-robot Interaction, pp. 33–40 (2006)
59. Donmez, B., Pina, P.E., Cummings, M.L.: Evaluation criteria for human-automation performance metrics. In: Performance Evaluation and Benchmarking of Intelligent Systems, ch. 2, pp. 21–40 (2009)
60. Tubre, T.C., Collins, J.M.: Jackson and Schuler (1985) Revisited: a meta-analysis of the relationships between role ambiguity, role conflict, and job performance. J. Manag. **26**(1), 155–169 (2000)

61. Barde, M.P., Barde, P.J.: What to use to express the variability of data: Standard deviation or standard error of mean Perspect. Clin. Res. **3**(3), 113–116 (2012)

A. Hong received his B.A.Sc. degree in Engineering Science (Aerospace) from the University of Toronto, Toronto, Canada, in 2014. He also received his M.A.Sc. degree in mechanical engineering in 2016 from the University of Toronto. His research focussed on building emotion recognition architectures for social bi-directional human-robot interaction, and improving human-robot interaction during multi-robot urban search and rescue. Alex is currently pursuing entrepreneurship, where he is engaging K-12 students around the world in learning robotics. When he has spare time, he enjoys bouldering and competing in Men's Artistic Gymnastics.

O. Igharoro received his B.A.Sc. in Mechanical Engineering from the University of Toronto. For his M.Eng. degree he worked in the Autonomous Systems and Biomechatronics Laboratory at the university with Dr. Nejat focusing on robot control architecture and graphical user interface design.

Y. Liu received the B.E. and M.A.Sc. degrees from the University of Science and Technology, Beijing, China, and Beijing University of Posts and Telecommunications, Beijing, China, in 1999 and 2002, respectively. He is pursuing the Ph.D. degree in the Autonomous Systems and Biomechatronics Laboratory at the University of Toronto. His current research interests include learning-based semi-autonomous control of multi-robot teams in urban search and rescue environments.

F. Niroui received his B.A.Sc. in Mechatronics Engineering from the University of Waterloo, Waterloo, Canada, in 2016. He is currently pursuing his M.A.Sc. degree in the Autonomous Systems and Biomechatronics Laboratory (ASBLab) under the supervision of Dr. Nejat at the University of Toronto, Toronto, Canada. His research interests include robot navigation and exploration within cluttered environments and the application of machine learning in robotics.

G. Nejat received her B.A.Sc. and Ph.D. degrees in mechanical engineering from the University of Toronto, Toronto, Canada, in 2001 and 2005, respectively. She is an Associate Professor and the Director of the Autonomous Systems and Biomechatronics Laboratory (ASBLab) in the Department of Mechanical and Industrial Engineering at the University of Toronto. She is also the Canada Research Chair in Robots for Society. Prof. Nejat is also the current Director of the Institute for Robotics and Mechatronics (IRM) at the University of Toronto, and an Adjunct Scientist at the Toronto Rehabilitation Institute. Her research interests include robot perception, human-robot interaction, semi-autonomous and autonomous control, and intelligence of assistive/service robots for search and rescue, exploration, healthcare, and cooperative applications.

B. Benhabib P.Eng., has been a Professor in the Department of Mechanical Engineering at the University of Toronto since 1986. His research interests are in the area of design and control of intelligent autonomous systems. His research in the past 30 years was supported by over 100 Masters and Doctoral Students, as well as a large number Postdoctoral Fellows and Research Engineers. Their combined effort has resulted in over 350 international journal and conference publications, as well as several book chapters. He is also the author of the book *Manufacturing: Design, Production, Automation, and Integration*.