Proceedings Paper:

Miyauchi, G. orcid.org/0000-0002-3349-6765, Lopes, Y.K. and Gross, R. orcid.org/0000-0003-1826-1375 (2023) Sharing the control of robot swarms among multiple human operators: a user study. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Proceedings. 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 01-05 Oct 2023, Detroit, Michigan, USA. Institute of Electrical and Electronics Engineers (IEEE) , pp. 8847-8853. ISBN 9781665491914

https://doi.org/10.1109/IROS55552.2023.10342457

Sharing the Control of Robot Swarms Among Multiple Human Operators: A User Study

Genki Miyauchi[1], Yuri K. Lopes[2], Roderich Groß[1]

Abstract— **Simultaneously controlling multiple robot swarms is challenging for a single human operator. When involving multiple operators, however, they can each focus on controlling a specific robot swarm, which helps distribute the cognitive workload. They could also exchange some robots with each other in response to the requirements of the tasks they discover. This paper investigates the ability of multiple operators to dynamically share the control of robot swarms and the effects of different communication types on performance and human factors. A total of 52 participants completed an experiment in which they were randomly paired to form a team. In a 2×2 mixed factorial study, participants were split into two groups by communication type (direct vs. indirect). Both groups experienced different robot-sharing conditions (robot-sharing vs. no-robot-sharing). Results show that although the ability to share robots did not necessarily increase task scores, it allowed the operators to switch between working independently and collaboratively, reduced the total energy consumed by the swarm, and was considered useful by the participants.**

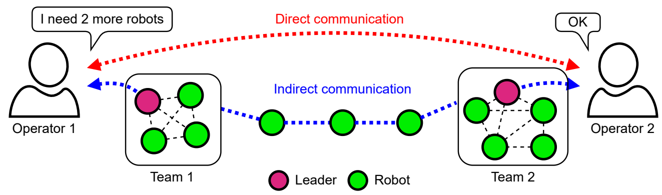


Fig. 1. Illustration of two human operators each controlling a subset of the robot swarm (e.g. a team) via a leader agent. Other robots maintain connectivity between the two teams. An operator can request robots from the other operator either directly (i.e. verbally communicating) or indirectly (i.e. sending a message through the swarm).

I. INTRODUCTION

Robot swarms are large groups of loosely coupled robots that accomplish tasks using local interaction [1]. With their inherent robustness and scalability, robot swarms are a promising technology for assisting in complex missions from search-and-rescue to infrastructure inspection and repair. As the push to bring robot swarms into the real world continues, it is important to consider how we leverage the autonomy of swarms to support humans. Recent advances in human-swarm interaction have shown that it is possible for a single human operator to control a swarm of robots using methods such as teleoperation [2], proximal interactions via gestures [3], [4], [5], augmented reality [6], and virtual reality systems [7].

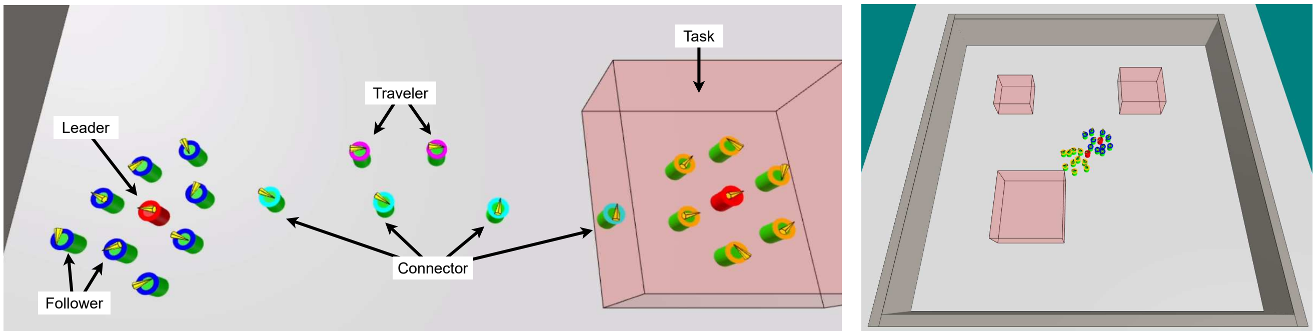
When a robot swarm splits into smaller sub-swarms, it becomes challenging for a single operator to control the sub-swarms without exceeding their own cognitive capacity [8]. Furthermore, if working alongside the robots, a single operator will not always be able to supervise all sub-swarms through proximal interaction, as each sub-swarm could be working at a different location. These problems could be addressed by having multiple operators who each interact with a different sub-swarm. This would allow the operators to share the required cognitive load to potentially achieve more complex tasks and improve productivity.

Introducing multiple operators requires careful design of the system. One factor affecting system performance is the relationship between humans and robots [9]. Prior research on multi-operator control of robot swarms can often be categorized into either (1) statically allocated robots (i.e. a subset of robots are assigned to each operator) or (2) shared robots (i.e. all operators can interact with all robots at any time) [10]. Statically allocated robots are beneficial when the tasks can be completed by the operators independently [11], [12], but do not have the flexibility to swap robots between the initially allocated teams. Shared robots give individual operators some flexibility on how many robots to use on a given task [13], [14], but operators may struggle from "diffusion of responsibility" due to all the robots being shared [15].

Another factor that affects the performance is communication, which is often considered a prerequisite for operators to cooperate effectively [16]. Poor communication quality can hinder this cooperation as it becomes difficult to maintain a shared mental model of the situation. In addition, excessive communication can become an overhead for the operators, resulting in increased cognitive workload [17]. Implicit or indirect communication (e.g. via a computer interface) can be effective when undertaking highly interdependent tasks, provided that the members have a sufficient understanding of the situation [18].

A further factor is the user interface. Achieving a high level of transparency is important, especially when operators are using the swarm as a shared resource [19]. Information such as the robots' position and internal states can be helpful

[1] G. Miyauchi and R. Groß are with the Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, United Kingdom {g.miyauchi, r.gross}@sheffield.ac.uk
[2] Y. K. Lopes is with the Department of Computer Science, Santa Catarina State University, Joinville, Brazil yuri.lopes@udesc.br
* This document is issued within the frame and for the purpose of the OpenSwarm project. This project has received funding from the European Union's Horizon Europe Framework Programme under Grant Agreement No. 101093046. Views and opinions expressed are however those of the author(s) only and the European Commission is not responsible for any use that may be made of the information it contains.
* G. Miyauchi is supported by a Departmental Scholarship from The University of Sheffield.



(a) (b)

Fig. 2. Overview of simulation scenario. (a) *Leaders* and *workers* are represented as red and green cylinders respectively; yellow arrows represent their headings. The workers can assume three roles (indicated by the color of their LED ring): *followers* (blue or orange) are part of a team, *connectors* (cyan) maintain connectivity among teams, and *travelers* (magenta) switch from one team to another. The task to complete is shown as a red transparent box. (b) Overhead view of an example arena containing three tasks. Tasks with a larger area require more followers to complete. Once a task is completed, it is removed from the arena and a new task appears at a random position.

in understanding the state of the swarm. However, it may not be possible to obtain such information from all the robots of the swarm, and in practical scenarios, the operators may have to work with the information that is available locally on a single robot. Having such restricted situational awareness can have a detrimental effect on the operators' ability to interact with a swarm [20], making it challenging for them to collaborate.

In [21], we proposed a framework based on supervisory control theory [22], [23], [24] whereby a swarm of robots solves spatially distributed tasks while maintaining overall connectivity. The swarm was accompanied by two leader agents who directed the robots towards specific task areas. However, the behavior of the leader agents was hard-coded: no human operators were involved.

In this paper, we conduct a user study to examine the ability of multiple human operators to dynamically share the control of robot swarms. To understand if the sharing of robots can be used effectively, we compare the performance of a pair of operators in completing a set of spatially distributed tasks with and without the ability to share robots. As sharing robots requires some level of coordination between the operators, we also investigate how either direct or indirect communication affects performance (see Fig. 1). To conduct the user study, we extend our framework [21] with a novel user interface that allows each operator to drive the respective leader agent of their team while being provided with its local camera feed, and request from or send robots to the other team. Our study is in contrast to previous work on multi-operator control of robot swarms as it (i) considers the case that the operators have dedicated teams and can exchange robots between them and (ii) restricts each operator's situational awareness by providing only a local view of the environment.

The paper is structured as follows. Section 2 describes the system setup, user interface, and experimental design. Section 3 presents the results obtained from the experiments, followed by a discussion in Section 4. Section 5 concludes the paper.

II. METHODOLOGY

A. Scenario

Our study investigates the ability of two human operators to control a swarm of robots to complete a set of tasks scattered across a bounded environment with no obstacles. We employ the robot behaviors from our previous work on connectivity-preserving robot swarms [21]. In this framework, each robot chooses its actions based on a formal model designed using supervisory control theory [22], [23], [24]. We consider two types of agents: leader and worker. Each operator controls a *leader*, allowing them to move through the environment and provide instructions to the robots. In a real-world scenario, the leader agent could either be the operator themself, a portable device carried by the operator, or a robot controlled remotely by the operator.

The *workers* are robots that are capable of completing tasks. At any moment in time, each worker is assigned to at most one leader. Throughout the mission, the workers autonomously maintain connectivity between the two teams by forming a robot chain. The workers can either assume the roles of *follower*, *connector*, or *traveler* as seen in Fig. 2a. Here, we summarize the robots' behavior in each role.

Follower behavior. A worker assigned to a leader is a follower. A follower performs a flocking motion based on virtual forces. It is attracted towards the leader and other followers within the same team and repulsed from any other leaders or workers. When the leader and itself are both inside the same task area, it is considered to be working on that task. When the distance to the other team increases, the followers will leave the team, one at a time, to become a connector to maintain connectivity between the two teams by forming a robot chain.

Connector behavior. A worker maintaining the robot chain is a connector. They allow the two teams to exchange information with each other. Unlike in [21], the connectors minimize the length of the chain by maintaining a straight line between the two teams. This is done by each connector finding the distance and angle to its two adjacent neighbors in the chain and moving to the middle point between these two neighbors. If the two teams move toward each other, the robot chain becomes shorter; the connector closest to the team leaves the chain to join the team.

Traveler behavior. The leaders are able to send a specified number of their followers to the other leader. When a follower receives a signal from its leader to join the other team, it becomes a traveler. A traveler moves along the robot chain until it reaches the other team to become a follower of that team. This allows the operators to rebalance the number of followers in each team.

Each task requires a specific number of followers to complete, which is visible to the operator when inside the task area. When outside, the operator can make a guess, as the task areas have lengths and widths of 0.4 m, 0.5 m, 0.6 m, 0.8 m, or 1.0 m, for tasks requiring 1, 3, 6, 9, or 12 followers, respectively. This resembles a search-and-rescue scenario in the real world; a rescuer might be able to predict the size of a task, but may only find out the exact number of robots or resources needed to accomplish it after arriving at the mission site. To complete a task, the leader and its followers must remain inside the task area for a certain duration. The time required to remain inside the task area increases linearly to the number of followers needed to complete it, and having an excess number of followers does not make the task complete faster. In the extreme cases considered—single-robot and 12-robot tasks—the followers need to remain for 5 s and 30 s, respectively. Completing a task awards the pair of operators a score equal to the number of followers required to complete it. Once a task is completed, it is removed and a new task appears at a uniformly random location without overlapping with existing tasks.

The operators' joint objective is to score as many points as possible during the trial. The operators must guide their followers to the task areas, while also monitoring the number of followers in their team to ensure they have enough to complete the tasks. The full robot controller models can be found in the supplementary material [25].

B. Robot and Simulation Platform

The operators interact with the simulated robots running in the ARGoS simulator [26]. The simulation consists of 2 leaders and 20 workers in a 4 m×4 m arena (see Fig. 2b). We use the e-puck [27] for our study, which is a mobile differential-wheeled robot with a diameter of 0.07 m. We use a maximum speed of 8 cm/s. The e-puck has eight proximity sensors with a range of 0.1 m distributed around its body. We assume a range-and-bearing system with a range of 0.8 m to communicate with neighboring robots.

C. User Interface

The operators interact with the simulated robot swarm through a custom graphical user interface based on Webviz [28], a web interface plugin for ARGoS [26]. Webviz uses a client-server architecture, allowing multiple users to simultaneously interact with the same simulation from different devices (see Fig. 3).

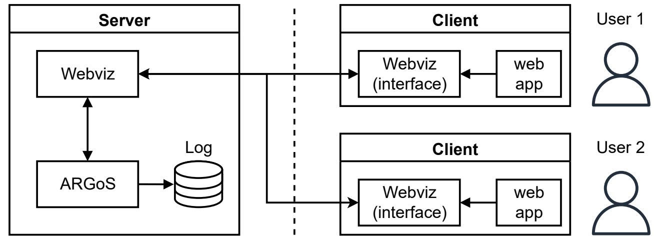


Fig. 3. System overview. A user accesses the shared swarm simulation through a simple web-based app. When the user interacts with the interface, the inputs are sent to the server and reflected in the ARGoS simulation. The client-server architecture is realised using Webviz.

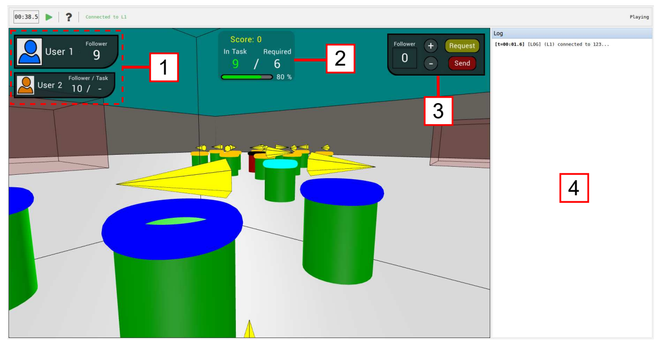


Fig. 4. The interface used for the experiment. The operator is shown a first-person perspective of the leader situated in the simulated arena. The interface displays (1) the team information panel, (2) the task information panel, (3) the request-and-send panel, and (4) the log panel. No overhead view or mini-map of the arena is provided to the user.

Fig. 4 shows the interface presented to a user. It shows a first-person view of the environment from the leader's perspective. The interface consists of four panels that support the user in completing the tasks; *team information panel*, *task information panel*, *request-and-send panel*, and *log panel*. No overhead view of the environment is provided. The users can drive the leader inside the simulation using a keyboard.

Team information panel. This panel provides information for the user to determine whether it or its partner has enough followers to complete their respective tasks. It is placed top-left in the interface and shows information about the two teams. The panel consists of two parts, which display the number of followers in the user and partner's team, respectively. In addition, if the partner is inside a task area, it also displays the size of that task (i.e. the number of followers needed) next to the partner's follower count.

Task information panel. This panel provides information about the current task and is placed top-center in the interface. When a user is inside a task area, it displays the size of the task, the number of followers currently inside the task area, and a progress bar that fills up when the followers are performing the task. The team score is also displayed at the top, which updates when either of the users completes a task. If the user moves outside of the task area, information about that task will no longer be displayed until the user re-enters the task area. Any progress made towards a task will remain even if a user exits the task area before it is completed.

Request-and-send panel. This panel is used to request or send followers to the partner and is placed top-right in the interface. A user can specify the number of followers using the plus and minus buttons and then press either the request or send buttons to confirm the action. If the `Request` button is selected, a message containing the specified number of followers will be sent to the partner via the robot chain. If the `Send` button is selected, the specified number of followers from the user's team will begin to travel to the partner's team. This allows the users to explicitly share their followers without needing to communicate with each other verbally.

Log panel. This panel displays the request and send messages received from the partner as well as any messages sent by the user, in chronological order. It is placed on the right side of the interface. By monitoring the log panel, a user can keep track of their partner's request or confirm that their partner has sent followers to their team.

D. Experiment Design

Our experiment followed a 2×2 mixed factorial design [29], where the robot-sharing condition (robot-sharing [RS] vs. no-robot-sharing [NRS]) was the within-subjects factor and the communication type (direct [DIR] vs. indirect [IND]) was the between-subjects factor. Within-subject evaluations assessed whether the dynamic sharing of robots affected the performance and human factors. Between-subject evaluations compared whether differences in communication style affected the performance and human factors.

In the RS condition, the operators were able to request or send robots to each other using the interface. If an operator found that it did not have enough followers to execute the task, they could request followers from their partner. In the NRS condition, the operators could not share robots. This meant some tasks required more robots than were available in a single team. In such cases, the task could still be completed if both teams entered the task area and the sum of followers reached the required number of followers for the task.

In DIR communication, the operators verbally communicated with their partners throughout the trial. They were seated nearby but were unable to see each other's computer screens. To share robots, an operator verbally requested followers from their partner. Only the `Send` button was displayed as the operators directly asked their partner for followers. In IND communication, the operators were not aware of who their partners were (and could not see their respective screens) because they were randomly paired. They were not allowed to verbally communicate: they could only request a specific number of followers via the interface. The operator that received a request, either verbally or through the interface, then decided how many followers to send.

1) Performance Measures: Performance measures include *task score*, *distance traveled*, *team separation*, and *robots shared*. These data were obtained from the simulation logs, which recorded the position and state of every robot and task, and all inputs received from both operators. Task score is the points obtained during the trial. If a task was being performed when the trial ended, we awarded partial points for the proportion of the task that was completed. Distance traveled is the total distance moved by the leaders and workers during a trial. It is used here as a proxy for the total energy consumed by the swarm. Team separation is the average distance between the two teams throughout the trial. The separation was calculated by finding the geometric center of both teams and taking the distance between the two points. This tells us how spread out the two teams were during the trial. For the RS condition, we recorded the number of followers shared between the teams.

2) Subjective Questionnaires: Subjective data were obtained individually through a paper-based questionnaire after each trial. The participant's subjective workload was obtained using the NASA-TLX [30] on a 7-point Likert scale. Subjective situational awareness was obtained using the Situational Awareness Rating Technique (SART) [31] on a 7-point Likert scale. We also asked participants about how well they understood their partner or the robots' actions, the usability of the interface, and whether they found the ability to share robots useful on a 7-point Likert scale. The questionnaire concluded with open-ended questions asking about what their strategy was in completing the tasks and any additional comments they had about the experiment. The questionnaires used in the experiment can be found in the supplementary material [25].

3) Participants: The study received ethical approval from The University of Sheffield. All participants were staff and students within the university. A total of 52 participants (34 males, 18 females) completed the experiment. All participants were adults over 18 and belonged to one of the age groups below (*Under 20*: 1, *20-29*: 28, *30-39*: 14, *40-49*: 5, *50-59*: 3, *60 & Over*: 1).

4) Procedure: First, the participants completed a preliminary questionnaire that asked about their previous gaming experience. Next, the experimenter gave an introduction on how to use the interface and explained the participant's goal in the trials. This was followed by a short training session for the participants to familiarize themselves with the robots' behaviors and the functionality of the interface.

For the main trials, participants were randomly paired to work together. The pair of participants each interacted with the robot swarm through the interface to score as many points as possible until the trials ended. The robot-sharing condition was controlled by enabling or disabling the request-and-send panel in the interface. The communication type was controlled by assigning the pair to either DIR or IND communication. Participants completed two trials to experience both robot-sharing conditions (RS and NRS) within their assigned communication type (DIR or IND). The order of the robot-sharing conditions were counterbalanced across participants to minimize any learning effect. Each trial lasted for 10 minutes. After each trial, the participants individually completed a post-trial questionnaire. Participants took a short break before moving on to the second trial. The overall experiment took around 75 minutes. An accompanying video is included which illustrates the user study, the simulation, and the user interface.

TABLE I

SUMMARY OF PERFORMANCE MEASURE AND QUESTIONNAIRE RESULTS. N IS THE NUMBER OF SAMPLES AND SD IS THE STANDARD DEVIATION.

Measure	Cond.	Comm.	N	Mean	SD
Task Score	RS	DIR	14	100.39	16.84
		IND	12	92.25	16.71
	NRS	DIR	14	106.31	13.97
		IND	12	99.20	16.19
Distance Traveled (m)	RS	DIR	14	10.28	1.44
		IND	12	10.51	1.22
	NRS	DIR	14	10.79	1.59
		IND	12	11.54	1.29
Team Separation (m)	RS	DIR	14	1.42	0.21
		IND	12	1.48	0.14
	NRS	DIR	14	0.96	0.17
		IND	12	1.00	0.18
Robots Shared	RS	DIR	14	22.79	11.19
		IND	12	25.33	7.23
	NRS	DIR	—	—	—
		IND	—	—	—
Workload	RS	DIR	27	3.62	1.10
		IND	24	3.56	1.04
	NRS	DIR	28	3.42	1.00
		IND	24	3.18	1.09
Situational Awareness	RS	DIR	28	4.44	0.78
		IND	23	4.49	0.76
	NRS	DIR	28	4.36	0.65
		IND	24	4.14	0.72

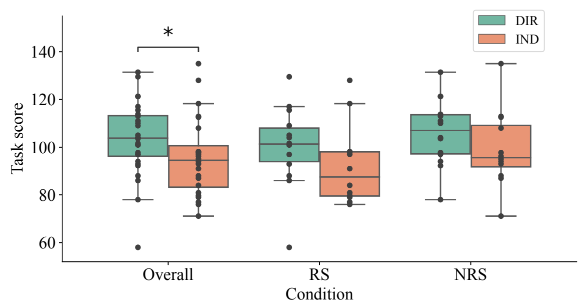
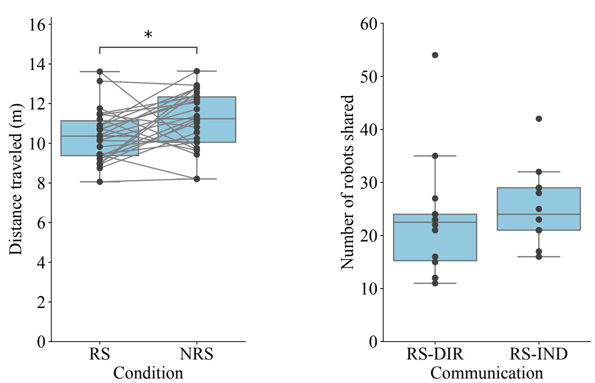


Fig. 5. Comparisons of task score by robot-sharing condition and communication type. Statistically significant results ($p < 0.05$) are indicated with an asterisk ($*$).



(a) (b)

Fig. 6. Comparisons of (a) the total distance traveled by the robots against the robot-sharing condition and (b) the total number of robots shared between the two teams in the RS condition against the communication type. Paired samples are joined by a line. Statistically significant results ($p < 0.05$) are indicated with an asterisk ($*$).

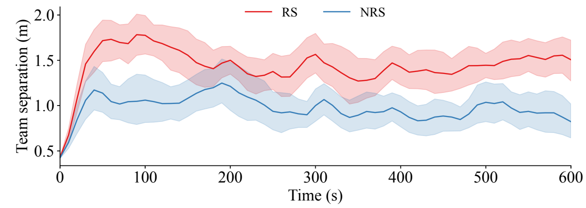


Fig. 7. Average separation between the two teams under different robot-sharing conditions. The solid lines each represent the mean across 26 trials and the transparent regions represent the 95% confidence intervals.

III. RESULTS

We performed quantitative and qualitative analyses to examine the effect of the robot-sharing capability and communication type between two operators. Quantitative analyses were based on the simulation logs, which recorded the states and positions of robots and tasks, the points scored, and the user inputs to the interface at each time step. Qualitative analyses were based on questionnaire responses, which asked the participants to rate their experience on a set of scales. Table I summarizes the results. Two-way mixed analysis of variance (ANOVA) were conducted to examine the effect of the two factors. Results are reported as significant when $p < 0.05$ and marginally significant when $p < 0.08$.

A comparison between the communication types indicates that there was a significant difference in task score ($F(1, 23) = 5.36, p = 0.030$). Fig. 5 shows that participants in the DIR communication scored more points ($M = 103.35, SD = 15.48$) than those in the IND communication ($M = 95.73, SD = 16.48$). No significant difference was found in the points scored between the robot-sharing conditions ($F(1, 23) = 2.72, p = 0.112$).

The ability to share robots had a significant effect on the total distance traveled by the robots ($F(1, 23) = 4.53, p = 0.044$). Fig. 6a shows the total distance traveled in the RS and NRS conditions. Since each pair of participants experienced both conditions, a line is drawn to show the paired data. On average, participants in the RS condition ($M = 10.39, SD = 1.32$) traveled less than in the NRS condition ($M = 11.14, SD = 1.48$), and therefore consumed less energy. No significant effect was found on the distance traveled between communication types ($F(1, 23) = 0.62, p = 0.441$).

The ability to share robots also had a significant effect on team separation ($F(1, 23) = 86.2, p < 0.0001$). Fig. 7 shows that participants in the RS condition ($M = 1.45, SD = 0.18$) were more likely to stay apart from the other team than in the NRS condition ($M = 0.98, SD = 0.18$) throughout the trial. There was no reportable difference in team separation between communication types ($F(1, 23) = 0.98, p = 0.333$).

Within the RS condition, we examined if the communication type affected the number of robots shared between the participants. Fig. 6b shows that there was no significant difference in the number of robots shared ($F(1, 24) = 0.46, p = 0.505$).

To compare the overall workload, we calculated the mean

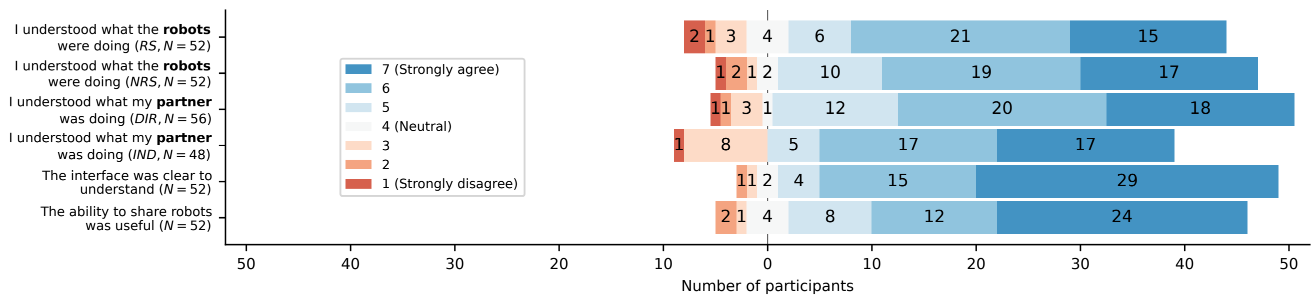


Fig. 8. Post-trial questionnaire responses. Participants were asked to rate how well they understood their partner and the robots' actions, the interface, and whether they found the ability to share robots useful after the trial.

of the raw scores from the six categories in the NASA-TLX questionnaire. The ability to share robots had a marginally significant effect on the global workload ($F(1, 49) = 3.82, p = 0.056$). Participants in the RS condition ($M = 3.59, SD = 1.06$) reported a slightly higher global workload than in the NRS condition ($M = 3.31, SD = 1.04$). The overall situational awareness was also calculated using the mean of the raw scores from the nine categories in SART. The participants' global situational awareness was not affected by either communication type ($F(1, 49) = 0.18, p = 0.673$) or the ability to share robots ($F(1, 49) = 3.09, p = 0.085$).

IV. DISCUSSION

In all trials, every pair of participants were able to complete tasks and dynamically share robots, despite undergoing only a short training period (i.e. up to 10 minutes). Fig. 8 shows that the lack of direct communication caused the percentage of participants understanding their partner's actions to drop from 89.3% to 81.3%. Despite the lack of direct communication and understanding of who their partners were, on average, participants in the IND communication were able to score 92.6% of the points achieved by those using DIR communication. Overall, these results are in line with previous work, where the operators benefit from direct communication for coordinating their actions.

The ability to share robots (i.e. RS condition) decreased the total distance traveled by the robots, and hence the energy consumed by the swarm. This could be attributed to participants more likely focusing on different tasks at the same time. If the partner requested some followers, the participant could simply send the requested amount of followers, or even more if they choose to do so. By contrast, when participants were unable to share robots (i.e. NRS condition), they had to bring their teams into the same area if the task required many followers. This resulted in increasing the average distance traveled by each robot in the NRS condition.

Participants in the NRS condition were also hesitant to tackle larger tasks compared to those in the RS condition. Some participants reported that when they did not have the required number of followers to complete a task, they preferred to move on to another task when the other team was far away. This meant the teams in the RS condition were often exploring new task areas earlier on, as their ability to share robots provided added flexibility to meet the task requirements. This could be crucial during search-and-rescue missions.

Although the teams were faster at reaching the tasks in the RS condition, the task scores were slightly better in the NRS condition. This could be explained by the larger team separation seen in the RS condition. The larger team separation meant there were fewer followers in each team. This resulted in some participants spending more time waiting for the requested followers to arrive instead of working on the tasks. The best performing participants were those who prioritized tasks that required the same or slightly fewer followers than their current number of followers in the team. These participants were able to minimize the waiting time and obtain higher scores. The limiting factor in this study was the number of workers available to the participants compared to the size of the tasks. Increasing the number of workers may have increased the performance in the RS condition.

It was expected that participants in the RS condition would experience a higher workload than those in the NRS condition due to the addition of the robot-sharing capability. While this trend was observed, results indicated only a marginal significance, where the RS condition had a slightly higher workload. This suggests that the dynamic sharing of robots between operators could be incorporated into human-swarm systems without having a significant effect on operator workload. According to the questionnaire responses in Fig. 8, most participants found the ability to share robots useful, even among those that scored fewer points in the RS condition. Several participants expressed that they found the ability to share robots useful not only because they could adjust the number of followers in the two teams, but because it allowed them to work more independently on the tasks and potentially increase their performance by executing tasks in parallel. This behavior to work independently matches with the increased team separation observed in the RS condition.

V. CONCLUSION

We conducted a user study to investigate the effects of dynamically sharing robots between two human operators on task-related performance measures and human factors. The operators were each provided only a local, first-person view of the simulated environment, and were accomplishing tasks, starting with preassigned teams of robots. They were able to request and share robots with each other either by communicating directly (i.e. verbally) or indirectly via simple messages that passed through the robot swarm. Our findings show that, despite the short training received, the operators were able to dynamically share robots under limited situational awareness. Although sharing robots did not necessarily increase task scores, it provided the operators the flexibility to work independently or collaboratively, reduced the energy consumed by the swarm (i.e. the total distance traveled), and was considered useful by the operators. Regarding the communication type, a decrease in task scores was noted when verbal communication was prohibited. Further training could help operators understand when are effective moments to request or send robots to further improve task scores. For future work, we plan on conducting the user study with physical robots to understand whether these findings generalize to more realistic scenarios.

References

[1] M. Birattari, A. Ligot, D. Bozhinoski, M. Brambilla, G. Francesca, L. Garattoni, D. Garzón Ramos, K. Hasselmann, M. Kegeleirs, J. Kuckling, F. Pagnozzi, A. Roli, M. Salman, and T. Stützle, "Automatic Off-Line Design of Robot Swarms: A Manifesto," *Frontiers in Robotics and AI*, vol. 6, 2019.

[2] P. Walker, S. A. Amraii, N. Chakraborty, M. Lewis, and K. Sycara, "Human control of robot swarms with dynamic leaders," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 1108–1113.

[3] J. Nagi, A. Giusti, L. M. Gambardella, and G. A. Di Caro, "Human-swarm interaction using spatial gestures," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 3834–3841.

[4] J. Alonso-Mora, S. Haegeli Lohaus, P. Leemann, R. Siegwart, and P. Beardsley, "Gesture based human - multi-robot swarm interaction and its application to an interactive display," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5948–5953.

[5] B. Gromov, G. Abbate, L. M. Gambardella, and A. Giusti, "Proximity human-robot interaction using pointing gestures and a wrist-mounted imu," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8084–8091.

[6] S. O. Sachidanandam, S. Honarvar, and Y. Diaz-Mercado, "Effectiveness of augmented reality for human swarm interactions," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 11 258–11 264.

[7] I. Jang, J. Hu, F. Arvin, J. Carrasco, and B. Lennox, "Omnipotent virtual giant for remote human-swarm interaction," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 2021, pp. 488–494.

[8] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 9–26, 2016.

[9] H. Yanco and J. Drury, "Classifying human-robot interaction: An updated taxonomy," in *2004 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, vol. 3, 2004, pp. 2841–2846.

[10] S. Nagavalli, M. Chandarana, K. Sycara, and M. Lewis, "Multi-operator gesture control of robotic swarms using wearable devices," in *Proceedings of the Tenth International Conference on Advances in Computer-Human Interactions (ACHI)*. IARIA, 2017.

[11] T. D. Fincannon, A. W. Evans, F. Jentsch, E. Phillips, and J. Keebler, "Effects of sharing control of unmanned vehicles on backup behavior and workload in distributed operator teams," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 53, no. 18, pp. 1300–1303, 2009.

[12] J. R. Grosh and M. A. Goodrich, "Multi-human management of robotic swarms," in *International Conference on Human-Computer Interaction (HCII)*. Springer, 2020, pp. 603–619.

[13] F. Gao, M. L. Cummings, and E. T. Solovey, "Modeling teamwork in supervisory control of multiple robots," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 4, pp. 441–453, 2014.

[14] J. Patel and C. Pinciroli, "Improving human performance using mixed granularity of control in multi-human multi-robot interaction," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020, pp. 1135–1142.

[15] M. Lewis, H. Wang, S.-Y. Chien, P. Scerri, P. Velagapudi, K. Sycara, and B. Kane, "Teams organization and performance in multi-human/multi-robot teams," in *2010 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2010, pp. 1617–1623.

[16] S. L. Marlow, C. N. Lacerenza, J. Paoletti, C. S. Burke, and E. Salas, "Does team communication represent a one-size-fits-all approach?: A meta-analysis of team communication and performance," *Organizational Behavior and Human Decision Processes*, vol. 144, pp. 145–170, 2018.

[17] J. MacMillan, E. E. Entin, and D. Serfaty, "Communication overhead: The hidden cost of team cognition." in *Team Cognition: Understanding the Factors That Drive Process and Performance.* American Psychological Association, 2004, pp. 61–82.

[18] R. Rico, M. Sánchez-Manzanares, F. Gil, and C. Gibson, "Team implicit coordination processes: A team knowledge-based approach," *Academy of Management Review*, vol. 33, no. 1, pp. 163–184, 2008.

[19] K. A. Roundtree, M. A. Goodrich, and J. A. Adams, "Transparency: Transitioning from human-machine systems to human-swarm systems," *Journal of Cognitive Engineering and Decision Making*, vol. 13, no. 3, pp. 171–195, 2019.

[20] G. Kapellmann-Zafra, N. Salomons, A. Kolling, and R. Groß, "Human-robot swarm interaction with limited situational awareness," in *Swarm Intelligence*, ser. Lecture Notes in Computer Science. Springer, 2016, pp. 125–136.

[21] G. Miyauchi, Y. K. Lopes, and R. Groß, "Multi-operator control of connectivity-preserving robot swarms using supervisory control theory," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 6889–6895.

[22] P. Ramadge and W. Wonham, "Supervisory control of a class of discrete event processes," *SIAM Journal on Control and Optimization*, vol. 25, no. 1, pp. 206–230, 1987.

[23] Y. K. Lopes, S. M. Trenkwalder, A. B. Leal, T. J. Dodd, and R. Groß, "Supervisory control theory applied to swarm robotics," *Swarm Intelligence*, vol. 10, no. 1, pp. 65–97, 2016.

[24] ——, "Supervisory control of robot swarms using public events," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 7193–7199.

[25] G. Miyauchi, Y. K. Lopes, and R. Groß, "Online supplementary material," 2022. [Online]. Available: https://doi.org/10.15131/shef.data.21088627

[26] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, 2012.

[27] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, vol. 1, no. 1, 2009, pp. 59–65.

[28] J. Patel, P. Sonar, and C. Pinciroli, "On multi-human multi-robot remote interaction: A study of transparency, inter-human communication, and information loss in remote interaction," *Swarm Intelligence*, vol. 16, no. 2, pp. 107–142, 2022.

[29] C. D. Wickens, J. D. Lee, Y. Liu, and S. E. Gordon Becker, *An introduction to human factors engineering*, 2nd ed. Upper Saddle River, NJ: Prentice Hall.: Pearson Education, 2004.

[30] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (task load index): Results of empirical and theoretical research," in *Advances in Psychology*. Elsevier, 1988, vol. 52, pp. 139–183.

[31] R. M. Taylor, "Situation awareness rating technique (SART): The development of a tool for aircrew systems design," in *Situational Awareness in Aerospace Operations (AGARD-CP-478)*. NATO-AGARD, 1990, pp. 3/1–3/17.