Towards Human Control of Robot Swarms

Andreas Kolling
University of Pittsburgh
School of Information Sciences
Pittsburgh, USA
akolling@pitt.edu

Steven Nunnally
University of Pittsburgh
School of Information Sciences
Pittsburgh, USA
smn34@pitt.edu

Mike Lewis
University of Pittsburgh
School of Information Sciences
Pittsburgh, USA
ml@sis.pitt.edu

ABSTRACT

In this paper we investigate principles of swarm control that enable a human operator to exert influence on and control large swarms of robots. We present two principles, coined *selection* and *beacon* control, that differ with respect to their temporal and spatial persistence. The former requires active selection of groups of robots while the latter exerts a passive influence on nearby robots. Both principles are implemented in a testbed in which operators exert influence on a robot swarm by switching between a set of behaviors ranging from trivial behaviors up to distributed autonomous algorithms. Performance is tested in a series of complex foraging tasks in environments with different obstacles ranging from open to cluttered and structured. The robotic swarm has only local communication and sensing capabilities with the number of robots ranging from 50 to 200. Experiments with human operators utilizing either *selection* or *beacon* control are compared with each other and to a simple autonomous swarm with regard to performance, adaptation to complex environments, and scalability to larger swarms. Our results show superior performance of autonomous swarms in open environments, of selection control in complex environments, and indicate a potential for scaling beacon control to larger swarms.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics - operator interfaces

General Terms

Human Factors, Measurement, Experimentation

Keywords

Human-robot interaction, metrics, evaluation, multi-robot system

1. INTRODUCTION

In recent decades the number of mobile robots deployed in field applications has risen dramatically. Their usage offers obvious advantages of reduced costs, removing humans from harms way, or enabling entirely new applications that were previous impossible, especially when combining many such robots into a comprehensive system. The tasks and missions already carried out by large robot teams range from search, exploration, rescue, surveillance, pursuit, up to deploying infrastructure. The domains of application are equally diverse and range from low-cost warehouse security to interplanetary exploration. New developments in commodity hardware which serve as low cost replacements for otherwise expensive sensing or motion capabilities promise to further accelerate the trend towards deploying large teams of mobile robots. This trend, however, poses a challenge for the control of such systems. Currently, most large robotic systems are controlled by multiple operators, often remote controlling the robotic devices. For larger systems with more robots and low cost hardware such a control approach is not practical. While autonomy is already playing a vital role, even for powerful systems, in the form of tools, such as mappers, path planners, monitoring and detection systems, it will also play an increasingly important role for the control of robotic systems with a very large number of robots, so called swarms. An increased usage of autonomous methods, however, poses a challenge to allow human operators to exert control on such robot systems. Enabling human operators to control robot swarms with hundreds of robots is still an open problem. Currently, multi-robot approaches generally scale to at most ten's of robots per operator even when using state of the art mapping, path planning, target detection, and coordination algorithms to alleviate the load on the operator.

Much of the recent work on swarms is focused on algorithms for autonomous swarms solving specific problems. The specificity in terms of the working assumptions render one of the key advantages of a robot swarm moot, namely its supposed flexibility and wide range of potential applications. It is envisioned that swarms will be capable of assisting in a number of complex problems and the space of possibilities is vast. For this reason, enabling the control of swarms by a human operator who could interact with the autonomy and adapt to specific challenges in a variety of conditions is crucial. The goal of this paper is to investigate principles of control for large swarms and to determine how humans perform in controlling swarms uisng implementations of these principles for a complex foraging task in a variety of challenging environments. Foraging tasks are a useful formal model of many practical applications such as search and rescue or surveillance. To solve the foraging task we provide the operator with methods developed for the control of autonomous swarms that enable basic capabilities such as optimal deployment and connectivity maintenance, but we are placing these algorithms into a new, more complex context, in which former guarantees of performance do not hold anymore. The operator's task is then to utilize these methods, mitigate their limitations, and perform a set of complex mission in environments with different types of obstacles. In the following we will shortly present related work in Section 2 including some details on the algorithms used in this paper. This is followed by Section 3 which discusses the two principles of control, selection and beacon control. Section 4 presents the missions and simulation platform used for the experiments. Finally, we close with a presentation and discussion of our results in Section 5 and a conclusion in Section 6.

2. RELATED WORK

The literature on human control of swarms is rather sparse and we present the few works that are related to ours. On the other hand, the literature on autonomous swarm behavior is too vast to be reviewed here and we shall focus only on a very small subset relating to the algorithms from control theoretic approaches that we utilize in our testbed.

One of the earlier contributions on human control of large robot systems is made in [5] in which some of the main challenges for supervisory control of swarms are discussed. The author calls for further research to address the development of new swarming technology and the lack of understanding of supervisory control of such systems, particularly with respect to the interaction of autonomy with operators. One of the first to consider practical challenges in controlling swarms of robots is [11]. Therein experiences with a swarm of 112 robots are presented. The focus is kept on hardware and the development of software for the swarm. A similar approach is taken in [10]. The presented software tool ROBOTRAK addresses hardware and software problems with regard to the control of robot swarms, especially programmability. Centered around another practical application [1] considers swarm control in the context of searching for a radiation source. It proposes yet another architecture based on a set of desirable features and presents a simple test system in which operators successfully aid a swarm in locating radiation sources. Also focused on a particular task, in [7] a team of Unmanned Aerial Vehicles (UAVs) is controlled by a single operator using behavior-based controls. These controls enable UAVs to perform a surveillance mission semi-autonomously while the operator generates a mission plan. Similarly as in [6] much of the direct control is based on a leader-following approach and the operator can choose to teleoperate individual UAVs. Much of the above work is related to practical obstacles from a robotics perspective in the design, programming, and deployment of swarms rather than the direct controllability of a swarm by an operator and no comprehensive user studies have yet been attempted.

Another and slightly more general approach is taken in [9]. Therein the authors use so called *physicomimetics*, i.e. the simulation of physical forces, to control a swarm. Two basic forms of control are distinguished. The first is a top-down control for which an operator sets global swarm parameters and the system learns to adjust the parameters of individual robots to achieve the desired global parameter. The second is a bottom-up approach in which virtual agents are used to modify the behavior of existing agents through interaction instead of directly setting their parameters. For both approaches a learning method is proposed to either set the parameters or determine the placement of virtual agents. The application considered is the defense of a resource against an attacker. The physicomimetic approach promises to be an intuitive control paradigm due to the force metaphors borrowed from physics with which operators should be familiar. Since no comprehensive user studies have been attempted this claim remains to be validated.

While the above considerations are worthwhile our focus is rather on principles of control that enable the executing of a variety of missions with the operator injecting mission-specific knowledge into the system. Especially the integration of simple as well as complex robot behaviors, which may be autonomous algorithms, into a system controlled by a human operator is a major goal. For this purpose we now shortly introduce some related work on the autonomous control of robotic networks. In particular, we are considering the problems of rendezvous, coverage and deployment, and connectivity maintenance. An algorithm for the rendezvous problem for distributed robotic networks was introduced in [3]. It assumes open and uncluttered environments and that every robot can obtain the location of its neighbors. An algorithm for optimally distributing a network of robots in an open environment is given in [4]. An additional algorithm that deploys robots in environments that are non-convex and simply-connected is presented in [8], although therein the goal is to simply cover the entire space while in [4] an optimal cover given a sensor deprecation with respect to distance is computed. A rigorous formalization and unifying framework of much of the above is presented in [2]. While this work is rigorous in terms of the theory the working assumptions are rather strict and often violated in practical application. Yet, these algorithms make ideal candidates for our robot behaviors and we are going to utilize the connectivity maintenance, rendezvous, and deployment algorithms. Connectivity maintenance allows us to guarantee that all robots form a connected communication network at all times and computes a set of admissible control inputs that satisfy this constraint. This is crucial for swarms with only local communication. Different communication networks can be chosen which lead to different communication constraints. Rendezvous algorithms guarantee that a set of robots can find agreement over a location at which they get together which could provide a useful functionality for users in the presence of obstacles. The more complex algorithm is the optimal deployment which relies on the robots to compute a Voronoi Diagram and move towards their respective centroid. Using this algorithm a user can achieve optimal deployment in open spaces without manually dispersing robots.

3. APPROACH

In this paper we explore two basic approaches for controlling the robot swarm. We shall call these *selection* and *beacon* control. Both are based on setting the modes of robots in order to control their behavior. The set of modes available to the operator can be customized and the modes for our experiments are described in Section 4. Modes can range from trivial behaviors, such as stop, to complex autonomous behaviors, such as optimal deployment. The first control approach, selection control, allows the operator to select a subset of robots by drawing a rectangular marquee. This is similar to most computer programs and games in which operators can select units in this manner and provide custom instructions to the selected units or area. Most computer users are familiar with such selection tools. Once a subset is selected, the robots can be instructed to switch to one of the available modes which they will continue to execute. A selection is persistent until a new selection is made. The second control approach, beacon control, is based on a different paradigm and inspired by prior work on biological and autonomous swarms. Here an operator modifies the environment for the swarm by placing a beacon that influences nearby robots. A beacon has a location, a range and an associated mode. All robots within its range switch to the associated mode. In contrast to selection control, which requires an active selection, robots are passively influenced by the closest beacons once they get into its range. Beacons are persistent until they are removed by an operator and multiple beacons can be present in an environment. Swarm phenomena such as leader or predator models can be simulated by placing beacons that attract or repel nearby robots.

Both control approaches can in theory enable an operator to exert very precise control, even over individual robots, by using very small and frequent selections or by placing many small range beacons. The key difference between selection and beacon control is their temporal and spatial persistence and the resulting active or passive selection of robots. More precisely, a beacon is spatially persistent since it influences robots within a given range but it is not temporally persistent since robots can enter or exit a beacon's range as time passes. In contrast, selection control influences the same robots regardless of how much time has passed since selecting a particular group. Yet, the location of this group may have changed in the meantime and hence selection control is not spatially persistent. As a consequence both control control approaches enable very different strategies with different degrees of effort and complexity. Consider a scenario in which robots have to perform a sequence of tasks when at a certain location, such as moving on a safe path around an obstacle. By using beacons any robot entering at any time will perform this sequence of tasks. Selection control, on the other hand, requires the operator to select robots once they reach the location to give them the appropriate instructions. But if only one group of robots has to execute a sequence of task at a location, then selection control is expected to allow better control since all robots remain under control of the operator despite their continuous movement.

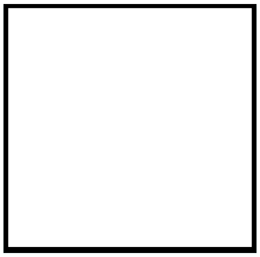
In addition, there is also a very practical difference between selection and beacon control. Selection control requires knowledge about the location of the robots to associate them to an area designated for selection. Beacon control, on the other hand, can be implemented by broadcasting a targeted signal within an area and thereby influencing all robots within that area without knowledge of their location. Additional variations and extensions of two basic control approaches are readily envisioned. One could consider selections or beacons that influence a subset of robots or other spatial relationships, e.g. selection by heading instead of location. In all these cases the key differences between selection and beacon control in terms of persistence and passive vs. active selection remain. But there are also entirely different methods to exert influence on a swarm of robots that are not considered here. One other such method is the setting global parameters that influence the otherwise autonomous behavior of an entire swarm, arguably the most indirect form of influence. The properties of such methods will depend even more heavily on type of autonomous behavior and swarm hardware. The primary purpose of this paper is to investigate the above control paradigms, but we also compare the performance of human operators with a baseline given by a simple autonomous swarm that solves the same task as the operators. This part is described in further detail in the next section.

4. METHODS

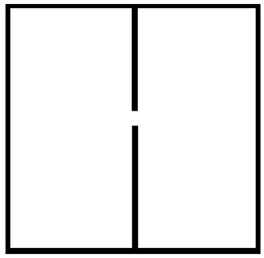
In order to explore how operators might control large swarms using the control principles discussed in the previous sections we developed a simple testbed in NetLogo [12]. NetLogo is a simulation platform suitable for modeling interactions between large sets of agents. The system was implemented using NetLogo's built-in language as well as Java extensions for computing Voronoi Diagrams and Delaunay graphs. Five different environments as shown in Fig. 1 were used. The size is set to 400 by 400 patches. Each patch in the user display is two pixels wide and high. The user interface for control is shown in Fig. 2.

Robots are placed randomly in the top right corner of each environment. In order for the operator to see the location of a robot and send it instructions it has to be connected to the communication network rooted at a base station. The base station is also placed in the top right corner. In addition to obstacles the communication links of a robot also constrain its motion. To maintain connectivity only motion that does not break an existing communication link is permitted. Robots can communicate with each other when within communication range r_c and when no obstacles are blocking their line-of-sight. The operator can choose whether robots have to maintain all communication links or a subset of these by choosing one of the following communication graphs: r_c-disk graph, r_c-limited Delaunay graph, r_c-limited Gabriel graph, or minimum spanning tree (see [2] for formal definitions). The r_c-disk graph is a graph given by embedding all robots into the environment and connecting all that are within line-of-sight and range r_c. All other graphs are subgraphs of the r_c-disk graph. More precisely, the r_c-limited Delaunay graph is the intersection between the Delaunay graph of all robots and the r_c-disk graph. Similarly, the r_c-limited Gabriel graph is the Gabriel graph of all robots intersected with the r_c-disk graph. The minimum spanning tree is computed from the r_c-disk graph by considering Euclidean distance as weights.

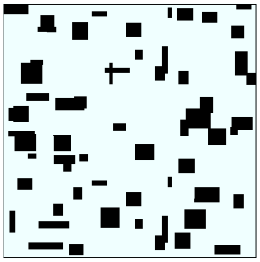
The minimum spanning tree imposes the fewest constraints on motion since it contains the fewest possible communication links while the disk graph contains all possible links. Robots that cannot communicate with the base station are invisible to the operator and wander around the environment choosing a new random direction upon collision. They are, however, still subject to communication constraints due to their local communication links to other robots and they form a separate local connected network. The base station imposes an additional motion constraint on its closest robot to ensure that at least one robot stay within its range. In our missions all robots are initially in a connected network and are expected to remain connected. In more realistic set-



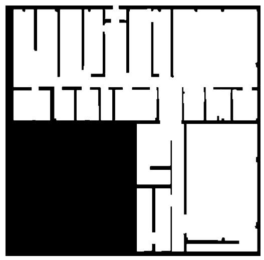
(a) Map 1: Open environment



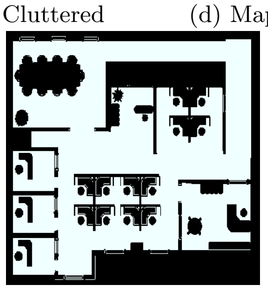
(b) Map 2: Two rooms



(c) Map 3: Cluttered



(d) Map 4: Structured



(e) Map 5: Cluttered and structured

Figure 1: Five test environments (a), (b), (c), (d), and (e). Obstacles are black and free space is white.

tings than our simulations, however, noise and other factors can still lead to communication loss.

For maps 1 to 4 the map is known by the operator and drawn in the user interface. For map 5 no map is given in advance and the operator has to explore the environment. To facilitate this a trail for each robot is drawn. Robots can be instructed to be in one of the following modes:

1. Stop: robots stop at their current position;
2. Come: robots move towards a target location;
3. Rendezvous: robots execute the rendezvous algorithm;
4. Deploy: robots execute the deployment algorithm;
5. Random: robots move with a new random heading after every collision;
6. Heading: robots synchronize their heading and move into the same direction;
7. Leave: robots move away from a target location

Robots in the above modes receive the following colors respectively: grey, blue, yellow, green, turquoise, purple and pink. All robots start in the random mode. In all modes, except deploy and random, robots slide along obstacles when these obstruct the desired direction of motion. Robots collide with each other when within a distance of two patches

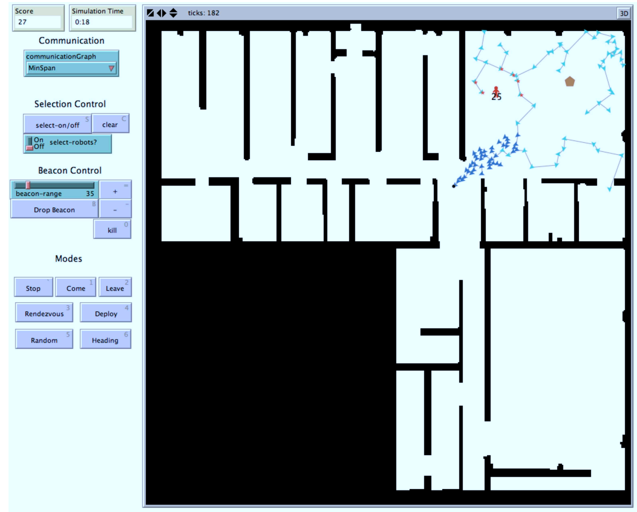


Figure 2: The simple user interface used for the experiments. Robots are small arrows, communication links are grey edges, the basestation is brown, and persons with information are red with their information value displayed in black. Robots currently within information range r_i of a person receive a red "*". Beacon and selection control only see their respective control panel.

in the environment. All robots move with the same maximum speed of 5 patches per second. For the beacon control the operator can place, move, set the mode of, change the range of, and remove beacons. The heading mode requires an additional mouse click to determine the heading. For the selection control the operator can select a rectangular set of robots, clear the current selection, and set the mode of all robots in the current selection. The come, leave and heading modes require an additional click to determine the target location or direction. Operators can generate complex behaviors from these modes. For example, behaviors similar to leader-follower behaviors can be approximated by using rendezvous and come modes.

4.1 Missions

Based on the five maps from Fig. 1 we presented the operator with one training scenario and seven missions. The task is a classical foraging task framed as information collection. Participants are instructed that their robots have to collect information from persons that appear randomly throughout the map. Each person has a different amount of information to be collected and each robot close enough to a person collects at a standard rate. The amount of information a person has is displayed in the user interface. Once all information is collected from a person it disappears. Persons that appeared with no robots within sensing range r_s are not visible on the display until a robot comes within sensing range. A new person appears with probability $\frac{1}{4}$ at a random location and with an information value sampled uniformly from $[1, 50] \subset \mathbb{N}$. On average 6.25 information units appear per second. The information of persons not within information range r_i of robots decays at a rate of 0.5 units per second. Robots within information range r_i of a person collect the information from it at the rate c (between 0.1 and 0.4 units per second). Information collected by robots is added to the operator's score. There are four different robot team configurations as seen in Table 1 that determine the capabilities of individual robots. The training scenario takes place in map 3 with robot configuration 2 and lasts 25 minutes. It precedes the seven missions each lasting five minutes. On average 1,875 information units spawn for every mission. In order to collect all this information and reach the maximum possible score robots need to cover the entire environment to quickly find every new person and exactly $\frac{5}{16}$ of all robots need to collect information at all times. Team configurations are chosen so that swarms with different numbers of robots retain a similar overall capability.

Robot Configuration	Robots	r_c	r_s	r_i	c
1	50	60	60	30	0.4
2	100	40	40	20	0.2
3	150	30	30	15	0.1333
4	200	25	25	15	0.1

Table 1: The settings for each robot configuration.

Table 2 shows the map and team configuration for each mission. Participants in missions 3 to 6 are controlling a variable number of robots from 50 to 200. The assignment of robot configurations in these missions is balanced across both maps (3 and 4) and w.r.t. whether a participant first has a larger number of robots in a map. All other missions have the standard configuration 2 with 100 robots.

Mission	Map	Robot Configuration
1	1	2
2	2	2
3	3	1|3|2|4
4	3	3|1|4|2
5	4	2|4|1|3
6	4	4|2|3|1
7	5	2

Table 2: The configuration and map used for every mission. Missions 3 to 6 have four possible and counter-balanced sequences of configurations that participants are assigned to.

Finally, we also used a simple autonomous swarm to solve the foraging task. The autonomous swarm leaves all robots in the random mode unless they are currently within information range r_i of a person in which case they stop. The chosen communication graph is the minimum spanning tree and robots are subject to the same communication constraints on motion as for human operators. Note that emulating this kind of autonomy is already difficult for an operator due to the large number of persons that spawn within five minutes, i.e. 75 persons on average. For beacon control an operator would have to place a beacon at every target they see and set the range to r_i and the mode to stop, then set the beacon to random once the information is collected and remove the beacon. This leads to an average total of 225 actions. For selection control all robots that get into information range of a target would need to be selected immediately and then stopped. This leads to a potentially even very larger number of actions since robots can enter the range of a person at different times. In this case the number of actions also increases with the number of robots rather than the number of persons.

To collect data we recruited 32 participants from the campus of the University of Pittsburgh, most of which were graduate and undergraduate students. The autonomous swarm ran through all missions eight times. We also tested the system with two experienced operators that contributed to the design of the interface to obtain a baseline for the scores human operators could achieve. The results of these experiments are presented in the next section.

5. RESULTS AND DISCUSSION

In this section we present and discuss the results of our experiments. In particular we address the following questions:

- Do selection, beacon and autonomous control perform differently?
- What impact do more complex environments have on performance?
- How do participants make use of the available modes?
- How do the control methods scale to larger swarms?

The average scores for participants and autonomous swarms for all maps and missions in robot configuration 2 (100 robots) is shown in Fig. 3[1]. A two-way analysis of variance of the score across maps and control conditions revealed an interaction between maps and control condition ($p < 0.001$***), an effect of control condition ($p < 0.001$***) and an effect of maps ($p < 0.001$***). Comparing only beacon and selection control no such interaction is present ($p = 0.7944$) but the effect of control condition and maps remain. In Fig. 3 this becomes apparent when looking at the steep drop of the autonomous swarm from map 1 to maps with obstacles. Excluding map 1 leads to no significant interaction between all control conditions and maps 2 to 5 ($p = 0.3907$) as well as no significant effect of maps ($p = 0.5761$). The effect of the control condition remains significant ($p < 0.001$***). On maps with obstacles the average scores for control conditions selection, beacon, and autonomous are 719, 590, and 695 respectively. Here selection control performs best. On all maps these averages become 779, 627, and 847 and autonomous swarms perform best overall due to the high scores in map 1. This suggests that human operator are generally poor at solving foraging tasks with swarms, not beating the simplest form of autonomy, but can adapt to complex environments. Table 3 shows results from running the experiment with two experienced operators that contributed to programming the system. These provide a rough indicator for the scores that are achievable by human operators with some experience. Note that despite the added experience the high performance of autonomous swarm is difficult to replicate with beacon control. The experienced operator with selection control achieves scores close to the autonomy in map 1 and can also maintain high scores in environments with obstacles (see Table 3 missions 3 to 7).

[1]The scores reported are the actual scores participants achieved. Normalizing the scores to show the collected fraction of all information that spawned leads to the same results, i.e. the influence of randomness in spawning the information is marginal and only has an impact on very small samples such as the experienced operators.

Mission	1	2	3	4	5	6	7
Expert (S)	1317	1075	1075	888	1140	991	822
-normalized	72%	64%	53%	39%	60%	48%	50%
Expert (B)	1078	884	519	903	664	654	556
-normalized	55%	44%	35%	45%	38%	40%	30%

Table 3: Scores from two experienced operators using selection (S) and beacon control (B). Missions 3,4,5 and 5 were tested with 50, 150, 100, and 200 robots respectively. Normalized scores indicate the percentage of points collected of the actually spawned information and allows for a better comparison of this small sample.

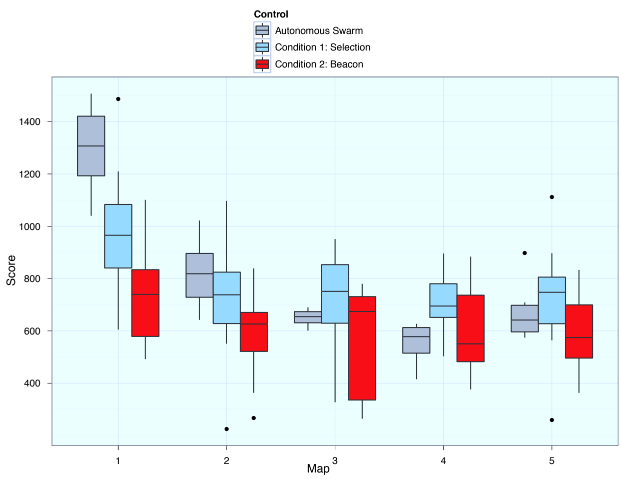


Figure 3: A box plot of the score across participants for maps 1 to 5. Note that these scores only include missions with 100 robots and participants have either only map 3 or only map 4 with 100 robots, i.e. the sample size is reduced for map 3 and 4.

Participants using different control methods also utilized different robot modes with different frequency as seen in Fig. 4. An explanation for this difference can be found in the impact the operator mode instructions have on the team. A mode instruction here is either a switch of mode for a selected set of robots or a beacon, affecting all nearby robots. Let us call the mean number of robots influenced by a mode instruction *mode impact*. For selection control mode impact relates to the size of the selections and for beacon control it relates to the number of robots influenced by a beacon. Selection and beacon control differ significantly w.r.t. mode impact as seen in Table 4 with a mode impact of 10 robots per instruction for selection control and 30 robots for beacon control. The number of instructions given differs as well with 56 for selection control and 39 for beacon control. This leads to an overall higher number of robot mode switches for beacon control at 1027 robot mode switches per mission due to the higher mode impact. Some of these switches can be attributed to robots in the random mode getting close to a beacon, explaining some of the difference in Fig. 4. For beacon control the large mode impact has a statistically significant ($p < 0.001$***) negative impact on score while for selection control mode impact does not have any effect.

The correlation between score and mode impact for beacon control is -0.45. Conversely, the number of instructions has a marginally significant ($p = 0.0897$) positive impact on score for selection control and no impact for beacon control. The correlation between the number of instructions and score is 0.414 for selection control. This is an indication that increased activity of the user, i.e. more mode instructions, helps more for selection control and that in beacon control many of the induced robot mode switches actually impede performance. On another note, operators seem not to exploit the rendezvous algorithm and rather adapt to the presence of obstacles manually and achieve rendezvous with the come mode.

	Selection	Beacon
Score***	779	627
Operator mode switches***	56	39
Robot mode switches***	428	1027
Mode impact***	10	30

Table 4: A comparison of selection and beacon control across all missions with 100 robots.

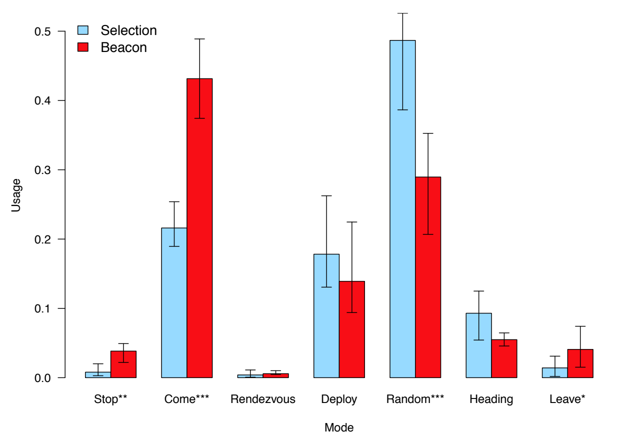


Figure 4: The figure shows the usage of each mode as the proportion of robots in that mode across the entire mission. The mean usage across all mission is shown. Arrows indicate the smallest and largest mean usage across all mission.

The above results all refer to missions with 100 robots. In the following we shall investigate difference with regard to changing robot configurations. These are available for maps 3 and 4. The main questions here relate to scalability, i.e. the ability to control larger teams of robots of similar overall capability but with individual robots being far less capable. The scores across different robot swarm sizes are shown in Fig. 5. A multiple analysis of variance of score across control condition (autonomous, selection, beacon), maps (3,4), and number of robots (50, 100, 150, and 200) revealed a significant impact of the number of robots ($p < 0.001$***) but no significant interactions nor effects of control condition.

One would expect that the autonomous swarm is not affected by increasing the number of robots if the overall capabilities were similar. In fact, for the autonomous swarm

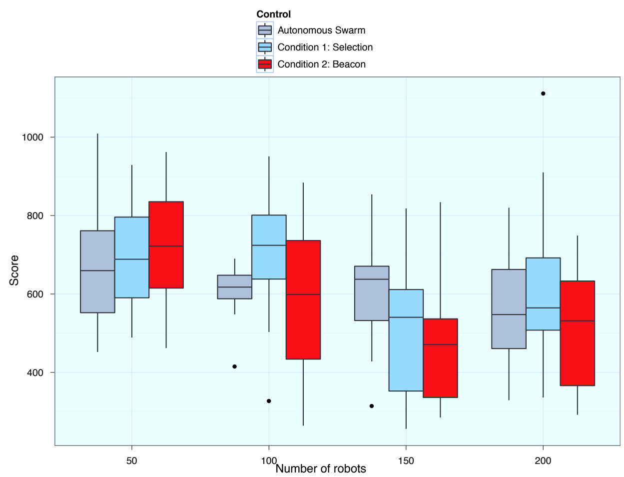


Figure 5: Score of maps 3 and 4 across different robot configurations with 50,100,150, to 200 robots across control conditions.

there is no significant difference across the number of robots for the score ($p = 0.5457$) confirming the similar capabilities. For selection and beacon control we do, however, have a significant effect of the number of robots ($p < 0.01^{**}$ and $p < 0.001^{***}$ respectively). A simple linear regression for selection and for beacon control gives $b = -0.8144$ ($t(62) = -2.073$, $p < 0.05^{*}$) with an intercept of $a = 733.6875$ for selection control and $b = -1.4689$ ($t(62) = -3.894$, $p < 0.001^{***}$) with an intercept of a=759.4062 for beacon control, both shown in Fig. 6. Both show a downward trend in score with an increasing number of robots, but less so for selection control. Considering the relatively stable performance of the autonomous swarm we can conclude that the increased difficulty of instructing the robots in a larger swarm impede performance.

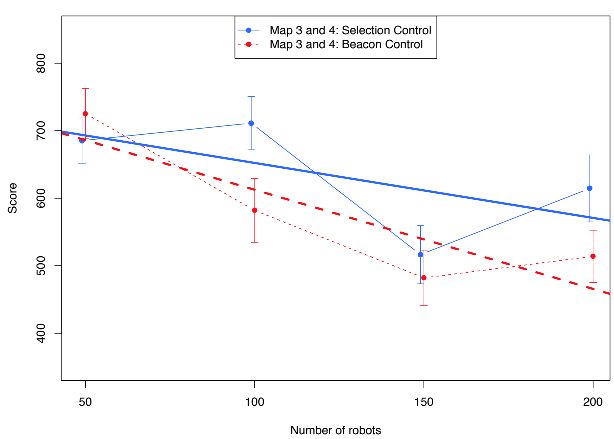
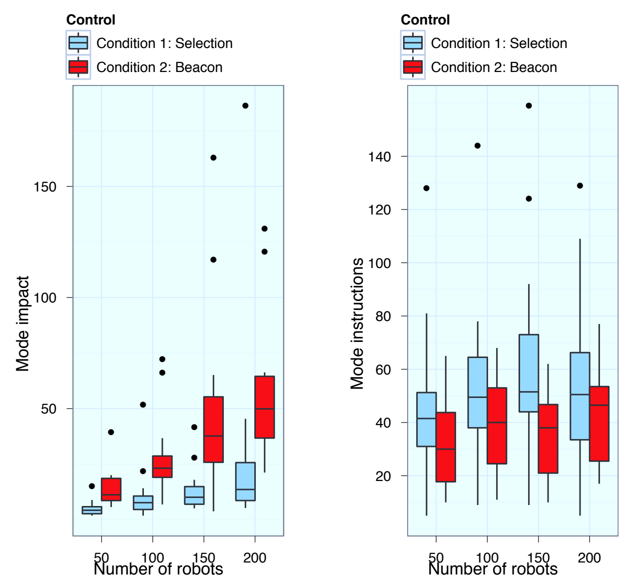


Figure 6: The mean score across different robot configurations with 50,100,150, and 200 robots for selection and beacon control. The standard error is shown with whiskers and the regression line as a thicker line.

While it is expected that larger swarms are more challenging to control we can also investigate whether and how operators adopt to these larger swarms, e.g. by increasing the frequency of mode instructions, number of beacons or size of selections, i.e. increasing the mode impact. Fig. 7 shows that only the mode impact but not the number of instructions scales with the number of robots. Hence, operators are not adapting directly to the larger swarm with increased activity but affect more robots with each mode instruction. Hence, each selection and each beacon influence more robots. In principle, this would suggest that the control methods both scale to the larger swarms and the detriment in performance is due to the reduces per-robot precision in the control. But looking closer at the correlation between activity and score gives us a more detailed picture. For selection control the number of mode instructions correlates with score by 0.1140633, 0.4676364, 0.370013, and 0.7896718 for 50, 100, 150, and 200 robots respectively. For larger swarms, increased activity is hence rewarded with better scores. For beacon control the correlations with score are 0.227069, 0.1523191, 0.3296887, and 0.08906478 for 50, 100, 150, and 200 robots, respectively. Beacon control shows no clear tendency for increased rewards for increased activity. This suggests that the two types of control do indeed scale differently and the performance impediment in beacon control is not mitigated by increased activity.



(a) (b)

Figure 7: (a) Mode impact for varying numbers of robots. (b) Number of mode instructions for varying numbers of robots.

6. CONCLUSION

We presented an investigation of two principles of human control for robot swarms, selection and beacon control and showed how these perform on a set of foraging missions in environments with different complexity. The key differences between these two principles are their spatial and temporal persistence and the resulting active or passive influence on the robots swarm, enabling different control strategies. Our results showed that novice human operators perform better with selection control. Both types of control enabled human operators to adapt to environments with complex obstacles and their drop in performance is less than that of a simple autonomous swarm that performs better than human operators in open environments. In fact, the different types of maps, two rooms, cluttered, structured, or blind with cluttered and structured obstacles, impeded performance similarly. Overall, the influence of the operator to adapt to these environments was successful despite human operators being generally worse at controlling large swarms for foraging tasks. Supporting the capabilities of human operators to adapt to complex environments with improved autonomy could combine the best of both.

One major problem in controlling swarms is scaling to larger number of robots and hence larger environments and tasks. We observed a stronger correlation between activity and scores in larger swarms for selection control. For beacons there was no such increased correlation. We conjecture that this is due to the fact that beacon control is indeed more scalable when used to its full potential. The strategic placement of beacons becomes more important as the swarm gets larger and mere activity alone does not improve performance. While it is more difficult to use and learn, beacon control seems to have a reduced dependence on activity, a crucial factor for scaling to very large swarms. Further work, possibly with extensive training of participants, might well show that beacon control can perform well and scale, despite our current results showing the opposite.

The missions in this paper relate to tasks that require the distribution of a swarm in an environment and coordinating the motion of a large number of robots. Some examples of such tasks are the establishment of an ad-hoc network infrastructure, the transport of assets to target locations, exploration, and mapping. But the two principles of control could also be investigated under different conditions, types of swarms, and tasks. In addition, further autonomous algorithms and modes could also be considered. It may well be that beacon control approaches work for better for particular modes and tasks while selection control performs better for others. Our study provides a starting point for further such investigations.

7. REFERENCES

[1] S. Bashyal and G.K. Venayagamoorthy. Human swarm interaction for radiation source search and localization. In *Swarm Intelligence Symposium, 2008. SIS 2008. IEEE*, pages 1–8. IEEE, 2008.

[2] F. Bullo, J. Cortés, and S. Martínez. *Distributed control of robotic networks*. Applied Mathematics Series. Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[3] J. Cortés, S. Martínez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8):1289–1298, 2004.

[4] J. Cortés, S. Martínez, T. Karatasand, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20:243–255, 2004.

[5] ML Cummings. Human supervisory control of swarming networks. In *2nd Annual Swarming: Autonomous Intelligent Networked Systems Conference*, 2004.

[6] XC Ding, M. Powers, M. Egerstedt, and R. Young. An optimal timing approach to controlling multiple UAVs. In *American Control Conference, 2009. ACC'09.*, pages 5374–5379. IEEE, 2009.

[7] X.C. Ding, M. Powers, M. Egerstedt, S. Young, and T. Balch. Executive decision support. *IEEE Robotics & Automation Magazine*, 16(2):73–81, 2009.

[8] A. Ganguli, J. Cortes, and F. Bullo. Distributed coverage of nonconvex environments. In V. Saligrama, editor, *Networked Sensing Information and Control (Proceedings of the NSF Workshop on Future Directions in Systems Research for Networked Sensing, May 2006, Boston, MA)*, pages 289–305. Lecture Notes in Control and Information Sciences, Springer Verlag, 2007.

[9] Z. Kira and M.A. Potter. Exerting human control over decentralized robot swarms. In *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*, pages 566–571. IEEE, 2009.

[10] M. Li, A. Alvarez, F. De Pellegrini, B. Prabhakaran, and I. Chlamtac. ROBOTRAK: a centralized real-time monitoring, control, and coordination system for robot swarms. In *Proceedings of the 1st international conference on Robot communication and coordination*, page 37. IEEE Press, 2007.

[11] J. McLurkin, J. Smith, J. Frankel, D. Sotkowitz, D. Blau, and B. Schmidt. Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots. In *AAAI spring symposium*, 2006.

[12] U Wilensky. Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL, http://ccl.northwestern.edu/netlogo/.