Human Interaction Through an Optimal Sequencer to Control Robotic Swarms

Huao Li*
School of Computing and Information
University of Pittsburgh
huao.li@pitt.edu

Jaeho Bang*
Robotics Institute
Carnegie Mellon University
jaeho.bang@gmail.com

Sasanka Nagavalli
Robotics Institute
Carnegie Mellon University
snagaval@andrew.cmu.edu

Changjoo Nam
Robotics Institute
Carnegie Mellon University
cjnam@cmu.edu

Michael Lewis
School of Computing and Information
University of Pittsburgh
ml@sis.pitt.edu

Katia Sycara
Robotics Institute
Carnegie Mellon University
katia@cs.cmu.edu

***Abstract*—The interaction between swarm robots and human operators is significantly different from the traditional human-robot interaction due to unique characteristics of the system, such as high cognitive complexity and difficulties in state estimation. In this paper, we concentrate on a method for conveying input from the operator to the swarm. Previous research has shown that control through switching between behaviors offers the greatest flexibility but is particularly difficult for human operators. A recently developed method for finding optimal sequences for composing behaviors offers a potential tool for aiding human operators controlling swarms through behavior switching. This paper compares participants performing a navigation task with and without the availability of an optimal sequencing aid. Results show that the task of preplanning a sequence of behaviors and durations is more difficult for participants than switching between executing behaviors to navigate. Users who used the aid frequently created shorter paths than infrequent users and the control group. In the trials that the aid was used, participants tended to generate more complicated sequences and achieve the first attempt more rapidly, than trials in which the aid was not used.**

***Index Terms*—Human-robot interaction, swarm robotics, multi-robot systems**

I. Introduction

Robot swarms use numerous simple robots to accomplish complex tasks. Global behaviors emerge from local interaction between the individual robots. These behaviors [1] which include but are not limited to flocking (consensus on heading), deployment (dispersion to Voronoi centers), and rendezvous (movement to consensus location), allow the swarm to achieve a diversity of tasks despite the individual robots' limited capability. A key advantage of robot swarms is that the system is robust to individual robot failures and does not require centralized planning or control. Real world applications proposed for robot swarms include area coverage, search and rescue operations, agriculture, and military operations. Ideally one might wish to synthesize sophisticated local control laws that would allow the swarm to accomplish complex tasks without further intervention. Unfortunately this is not yet feasible for anything beyond the simplest tasks and would not allow human interaction unless we were able to synthesize and swap out controllers in real time.

Today, swarm behavior can practically be influenced by humans in three basic ways [2] 1) by changing control law parameters [3], for example, changing the heading of a flocking swarm, 2) by switching between control laws [4], switching from flocking to rendezvous to concentrate swarm to pass a narrow aperture, for example, or 3) by altering the swarm's environment [5], herding the swarm with controlled robots, for example. In each case the human control signal needs to be conveyed in some way such as broadcast or propagation with some indication of priority. These factors have been shown to effect the rate of convergence to consensus (switch in behavior or parameter) by up to three orders of magnitude [6]. Each approach has its drawbacks. Control through an altered [7] or virtual environment [8] is indirect and much slower in response, although it allows flexibility in influencing behavior. Control through parameters can be rapid and direct but has less flexibility. Control through switching among behaviors is both direct and can tailor behavior flexibly. It implicitly subsumes control through parameters because default or human specified parameters must be provided for a new behavior to execute. While researchers have shown operators can use behavior switching to guide swarms in tasks such as target search [3] there is great variability and undoubtedly large inefficiencies in their choice and timing of behaviors.

Because the state of a swarm executing a behavior is constantly changing, its proximity to converging states of potential succeeding behaviors is also changing, making some points in time more advantageous for switching between behaviors than others. Researchers have proven that this phenomena, they term neglect benevolence, exists for all LTI (linear time invariant) systems [4] and that human participants can learn to approximate optimal switch times between swarm behaviors although their responses almost always precede the optimal

*These authors have equal contributions to this paper.

This research has been sponsored in part by AFOSR Award FA9550-15-1-0442 and an NSERC PGSD scholarship.

time [9]. While control of a library of behaviors and the ability to specify their parameters provide an extraordinary degree of control over a swarm, humans have difficulty in recognizing swarm states [10] and predicting the effects of their interventions [11] as well as the timing of their actions [9], making unaided control difficult and inefficient. The problem of behavior composition is extremely difficult even for a computer because of the large space of alternative behaviors and durations that must be searched. The problem is reminiscent of RRT (rapidly-exploring random tree) path planning for which researchers have found [12]–[14] substantial gains from allowing human interaction to steer the algorithm through restricted regions reducing the space that must be searched.

In this paper we present an attempt to apply this approach to computationally mediated control of a simulated swarm. A study [15] has recently developed methods for optimally composing sequences of swarm behaviors based on search techniques similar to the A* algorithm. Given an initial state, goal state, and objective function (such as minimizing distance traveled) the algorithm can determine an optimal sequence of behaviors and durations to reach the goal. Because of the size of the search space, the algorithm is only suitable for very small problems unless either the sequence of behaviors or their durations can be fixed. In the reported experiment participants were allowed to specify behavior durations while the algorithm returned optimal sequences of behaviors matching the given durations.

II. Method

A. Swarm Behaviors

The behavior library employed in this experiment is based on Nagavali et al's work [15], which consists of following concrete behaviors:

1) *Flock*: All robots flock in a direction based on their current initial positions and orientations.
2) *Flock East*: Robots flock in the positive x-axis direction of the map.
3) *Flock West*: Robots flock in the negative x-axis direction of the map.
4) *Flock North*: Robots flock in the positive y-axis direction of the map.
5) *Flock South*: Robots flock in the negative y-axis direction of the map.
6) *Rendezvous*: All Robots move toward each other.
7) *Antirendezvous*: All robots move away from each other.

The three kinds of emergent behavior being composed in this experiment were generated by different weightings associated with responses within three concentric bands: repel, align, and cohere (attract), as shown in Fig1. For Flocking, all three responses were active with highest weight given to align. To constrain the flocking direction, a vector parameter is applied in the four biased flocking behaviours. Rendezvous, convergence to a consensus location, was achieved through a high weight on cohere with a slight weight on repel to prevent collisions. In Antirendezvous, moving swarm apart, only repel had non zero weight.

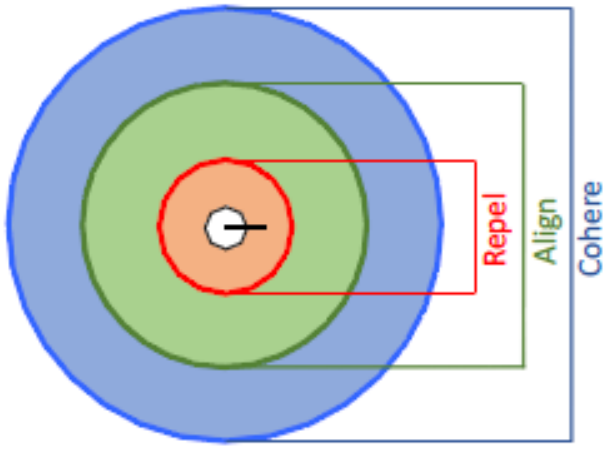

Fig. 1. Response regions.

B. Behavior Composition

The sequencer generates the optimal sequence of behaviors for a user supplied set of durations. The algorithm generating the optimal sequences is based on [15] which solves the following problem: Given the current state x_0 and a set of goal states χ_g, a behavior library of swarm B, time horizon t_f, set of duration times T, a swarm behavior execution cost function $C(\cdot)$ and heuristic cost-to-goal estimation function $H(\cdot)$, find a sequence of swarm behaviors selected from B that respectively last T to ensure that the swarm is in one of the goal states in χ_g at the end of the time horizon t_f. If the algorithm cannot find such a swarm behavior sequence, it returns an empty sequence ($\varnothing$) to indicate failure.

Latency of response is a key usability problem for this approach because of the extensive search required for optimal behavior composition. To improve user experience the program was parallelized through GPU programming in three ways: multiple states corresponding to valid sub-sequences were expanded simultaneously, robot positions were calculated in parallel, and all possible behaviors in the behavior library are searched and expanded together. These efforts reduced latencies for problems of the size studied from 10-30 seconds to 2-8 seconds, just below the 10 second limit needed to maintain user attention [16].

C. Task

The navigation task requires participants to control a swarm of robots to reach the destination area without colliding with obstacles or going outside the map. Users control the composition of behaviors by entering a sequence of behaviors and their durations in a two column list. When executed this sequence of switched behaviors defines the paths which would be taken by the robots. In the control condition the user must enter both behaviors and durations. In the aid available condition the participant may either specify both behaviors and durations as in the control or enter only behavior durations in the first column and allow the aid to return an optimal sequence of

behaviors matching those durations. After receiving the input sequence, the system computes the trajectory of the swarm and visualizes the result. Participants were instructed to try their best to generate a valid and complete sequence within the given time. A valid path was defined as robot trajectories that are within the map boundary, and do not collide with obstacles. A complete path was one in which all robot trajectories reach the goal region. Participants were encouraged to continue searching for improved sequences to minimize the sum of distances traveled by robots even after attaining a valid and complete sequence.

D. Testing Maps

Twenty testing maps were generated, each with a different arrangement of obstacles and goal regions. The number of obstacles per map ranged from 1 to 10, and the size of obstacles was varied. The goal region size was kept constant but its location changed among the maps. Initial robot positions were also changed between maps but the number of robots (10) and their initial orientation (facing east) were kept constant. A pilot test was run to validate the difficulty and appropriateness of maps. Three maps were found to be too hard for participants to finish within the given time and were excluded. The remaining 17 maps were separated into a training set (5 maps) and test set (12 maps).

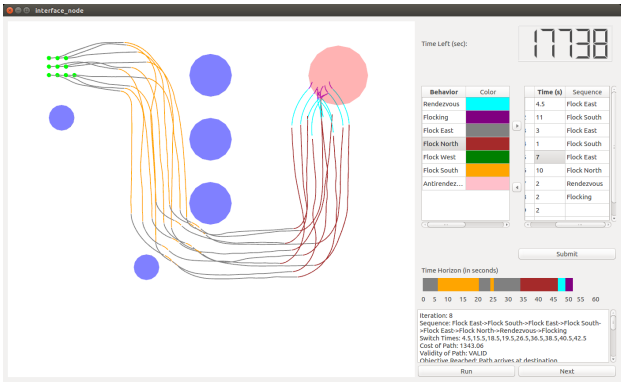

Fig. 2. Sequencing Interface.

E. Sequencer Interface

The sequencer interface (Fig.2) allows users to control the swarm of robots to finish a navigation task by generating behavior sequences. A complete sequence command consists of an array of concrete behaviors selected from the given library and the corresponding duration time of each behavior. On the panel to the right is the behavior list and sequence editor where users could define and alter the behaviors and their durations. The user could add or remove behaviors from the sequence by inserting new behaviors at the bottom of the list and moving them up to the desired position or deleting a behavior causing subsequent behaviors to move up in the list. Durations could be changed by typing a new value into their textbox.

On the panel to the left is a map of the swarm's workspace. Small green circles correspond to swarm robots' initial positions, blue circles correspond to obstacles, and the red circle corresponds to the destination area that all robots must reach. Lines correspond to the trajectory taken by each robot, while colors of the line correspond to the behavior during particular segments. These colors correspond to the list of behaviors and durations on the panel to the right and the timeline near the bottom of the panel. A timer at the top of the panel shows the remaining time. When the timer reaches 0, the trial is terminated and the next trial begun.

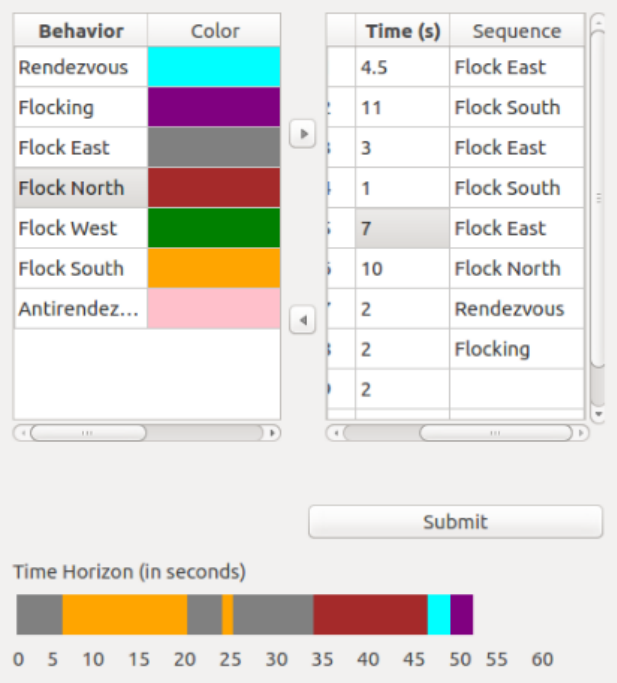

Fig. 3. Behavior List and Sequence Editor.

After entering the duration, selecting the behavior sequence manually or with help of the aid, and pressing the Submit button, the user can visualize the duration sequence on a progress bar-like time horizon at the bottom of the sequencing panel (Fig.3). Each color corresponds to the behavior color of the corresponding path segment. In aided cases where the user has entered more durations than the available sequence, the time horizon is displayed as black. If the user input is invalid (eg. more behaviors given than durations or the sum of durations exceeds the maximum simulation time which is set to 60 seconds for the experiment), a message indicating the type of error is presented. When an input is valid and the Run button is pressed, the system calculates the cost, validity of path, completeness of path, and the remaining behavior sequence if this has not been fully given and displays the paths on the map. Validity and completeness were highlighted in the labels as green if valid, red if invalid, and black when first initialized. All of this information is displayed in the console on the lower right which also provides a history of commands. The Next button located to the right of the Run

button allows the user to move to the next map even if there is time remaining. To prevent the user from pressing this button accidentally, pressing the Next button invokes a popup that asks once more whether the user wants to move to the next map. A window at the bottom of the screen retains history of the trial allowing the user to view earlier sequences.

III. Experiment

a) Participants: Twenty-one paid participants (average age $= 22.95 \pm 1.72$, 6 female participants) were recruited from the University of Pittsburgh and Carnegie Mellon University communities. All of the participants reported substantial video game experience. Eleven participants were randomly assigned to the aid-available group while the other ten participants were unaided.

b) Procedure: To evaluate the feasibility of human control of a swarm through prespecified behavior composition and the benefit of aiding this form of control through optimum sequence generation, we conducted an experiment comparing a behavior composition control with an aid-available experimental condition. Demographic information including age, gender, education level and video game experience were collected at the beginning of the session followed by a a 10-minute introduction. The experimenter walked through the initial training map demonstrating the task and the use of the interface. Participants solved the next 4 training maps with experimenter assistance to familiarize themselves with the system and task. Participants in the aid-available condition were instructed in both modes of interaction but encouraged to use the aid on the practice maps. Participants in the unaided group received additional feedback at the end of each training map showing the optimal sequence of behaviors given their last sequence of durations. After the fixed order training map set, participants were given 12 test maps in counter balanced order with initial and goal regions indicated and instructed to assemble a sequence of behaviors and duration times that were valid and complete. They were encouraged to optimize the sequence after the first successful attempt, during the time remaining in the 4 minute trial. Trails ended at 4 minutes or when the user decided to terminate. After the completing the experiment, participants filled out the System Usability Scale [17]. The entire session lasted approximately an hour.

IV. Results

No significant difference on the total path length was found between the aided group and unaided group (Aided: 1013.23 ± 45.5, Unaided: 1056.73 ± 28.0, $p = .437$). There was also no difference in average time to completion (Aided: 175.50 ± 7.0s, Unaided: 183.52 ± 6.3s, $p = .615$) and completion rate (Aided: $25.76 \pm 4.7\%$, Unaided: $23.33 \pm 4.9\%$). No difference was found in completed maps with 74.2% completion in the aided group and 76.6% in the unaided condition. Unaided participants also reported marginally better usability ($75.2 \pm 0.99 vs. 67.3 \pm 0.85, p = .074$) for the system than those with access to the aid.

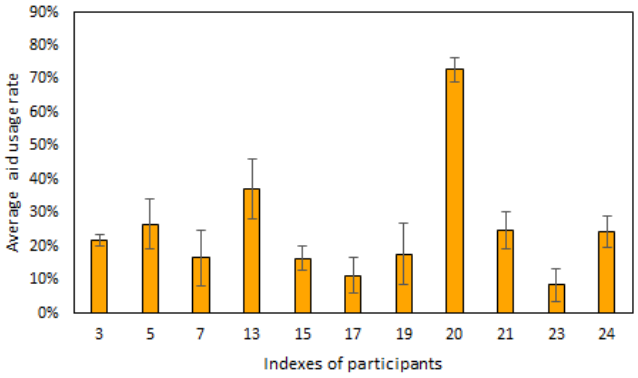

Fig. 4. Usage rates for participants in Aid-Available group.

However, defining the algorithm usage rate of each participant as (the number of sequences generated by the algorithm)/(the number of all the sequences generated), the average usage rate in the aided group was only 25.3% (SD=16.9%). We found that because the low rate of use many of the participants in the aid-available condition used the aid only rarely as shown in Fig. 4. The poor usage rate may reveal usability issues, but on the other hand, it may also bias our evaluation of the effectiveness of the aid algorithm itself. Alternately, excluding the trials in which the algorithm was not used from the data of the aided group and incomplete trails from both aided and control groups, allows a different kind of comparison between aided and unaided trials (Fig. 5).

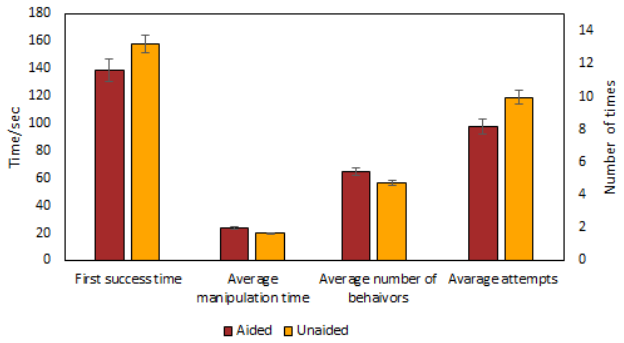

Fig. 5. Performance metrics in aided and unaided trials.

Under these conditions the aided group needed more time to generate a single sequence than the control group (23.90 vs. 19.83, $t(154.37) = 3.631, p < .001$), but the average number of behaviors in each sequence was also higher (5.42 vs. 4.71, $t(158) = 3.631, p < .001$). The aided group made fewer attempts (8.17 vs. 9.55, $t(212) = 2.824, p = .005$) in generating sequences and achieved their first successful attempt more rapidly (138.5s vs. 157.9s, $t(182.87) = 1.873, p = .063$) often in a single trial.

V. Discussion

The poor completion rate and subjective reports indicate the current sequenced path planning process is neither efficient nor natural. Participants found it awkward when they were requested to provide unanchored and arbitrary parameters and received feedback in the form of paths which might fail to reach the goal (incomplete) or a message indicating them to be invalid. Participants may prefer real-time interaction methods for switching between behaviors as in [10], [11] rather than preplanning a complete behavior sequence which places greater demands on working memory and decision making. We are considering adding intermediate goal functions to shorten the feedback loop and make the process more controllable for users.

Our results show that participants were mostly unwilling to use the sequencing algorithm and found it less usable than generating the sequences manually. There are several possible reasons why the aid may be difficult to use: a) If the input duration time is not long enough for swarms to reach the target area, the algorithm generates a feasible path with the shortest distance between the endpoint and final target. This can lead to a local optimum in which the swarm leaves obstacles between it and the target, while ignoring a valid longer path. From the users' perspective, this reflects unreliability because the algorithm appears to ignore obvious obstacles in its path. b) Due to the computational limitations, the aided algorithm may have long response times of up to 8 seconds when generating the sequence, which is neither time-efficient nor pleasant to use, c) For easy maps with one explicit feasible path, manual or auto-generated sequences are almost identical since all approach the optimal solution. Using the aid effectively for hard maps with multiple obstacles which might benefit most from human-machine cooperation requires an efficient strategy such as defining the first few behaviors and durations to constrain the path direction, and letting the algorithm fill out the rest. However, only a few participants in the aided group developed such strategies resulting in the lopsided distribution of frequent and infrequent algorithm users shown in 4.

Providing access to an aiding algorithm influenced participants' performance in sequence generation. Comparing trials with and without algorithm use, both frequent and infrequent algorithm users arrived at their first valid and complete trajectory more rapidly and produced more complex paths with a larger number of behaviors. However, no performance improvement in trajectory length was found. Those facts show that the aid algorithm we used in the study could efficiently help participants to generate a feasible sequence for swarms to finish the task, but not significantly contribute to the sequence optimization.

Another interesting finding is that for frequent users who had developed effective strategies for interacting with the algorithm, path lengths were reduced as well. Here we divided participants by their usage rate of the aid algorithm, and defined them as frequent users (users in the aided group who used algorithm more than 18%) and infrequent user (remaining users in the aided group + control group). A possible explanation for the difference between frequent and infrequent algorithm users may lie in the distinction between exploration and optimization stages of the task. When a user begins forming a sequence, the algorithm always produces valid trajectories (within the map and without collision with obstacles) for a given set of duration times, which is hard to guarantee in manual composition. So the use of the algorithm should decrease users' workload at this stage, and lead to earlier discovery of a feasible and complete solution because only feasible solutions are being considered. After the first feasible and complete trajectory is generated, users need to trim the sequence to shorten its path length to improve performance. Because the algorithm only supplies a duration for the final segment, without strategies, the aided group has little direct assistance for changing intermediate duration times or behaviors. As a consequence, while they find a complete and valid path in fewer attempts, these users have difficulty in improving paths further and generate fewer paths. Frequent users, by contrast, who have developed effective strategies, such as progressing from smaller to larger sets of durations to refine the path, are able to achieve shorter path lengths than other groups.

In future research, we plan to explore approaches that allow incremental construction and modification of paths while preserving optimality of composition. We additionally plan to expand our approach to other task settings such as area coverage, in which finding a feasible trajectory is much easier than in navigation, but harder for optimization.

References

[1] F. Bullo, J. Cortes, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.

[2] 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.

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

[4] S. Nagavalli, L. Luo, N. Chakraborty, and K. Sycara, "Neglect benevolence in human control of robotic swarms," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6047–6053.

[5] A. Kolling, K. Sycara, S. Nunnally, and M. Lewis, "Human swarm interaction: An experimental study of two types of interaction with foraging swarms," *Journal of Human-Robot Interaction*, vol. 2, no. 2, 2013.

[6] S. A. Amraii, P. Walker, M. Lewis, N. Chakraborty, and K. Sycara, "Explicit vs. tacit leadership in influencing the behavior of swarms," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2209–2214.

[7] M. A. Goodrich, B. Pendleton, S. Kerman, and P. Sujit, "What types of interactions do bio-inspired robot swarms and flocks afford a human?" *Proc. Robot.: Sci. Syst. VIII*, pp. 105–112, 2013.

[8] 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*. IEEE, 2009, pp. 566–571.

[9] S. Nagavalli, S.-Y. Chien, M. Lewis, N. Chakraborty, and K. Sycara, "Bounds of neglect benevolence in input timing for human interaction with robotic swarms," in *Proceedings of the tenth annual acm/ieee international conference on human-robot interaction*. ACM, 2015, pp. 197–204.

[10] P. Walker, M. Lewis, and K. Sycara, "Characterizing human perception of emergent swarm behaviors," in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 002 436–002 441.
[11] ——, "The effect of display type on operator prediction of future swarm states," in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 002 521–002 526.
[12] A. Griner, "Human-rrt collaboration in unmanned aerial vehicle mission path planning," Ph.D. dissertation, Massachusetts Institute of Technology, 2012.
[13] M. Taïx, D. Flavigné, and E. Ferré, "Human interaction with motion planning algorithm," *Journal of Intelligent & Robotic Systems*, vol. 67, no. 3-4, pp. 285–306, 2012.
[14] S. S. Mehta, C. Ton, M. J. McCourt, Z. Kan, E. A. Doucette, and W. Curtis, "Human-assisted rrt for path planning in urban environments," in *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 941–946.
[15] S. Nagavalli, N. Chakraborty, and K. Sycara, "Automated sequencing of swarm behaviors for supervisory control of robotic swarms," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2674–2681.
[16] J. Nielsen, *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
[17] J. Brooke *et al.*, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.