Levels of Automation for Human Influence of Robot Swarms

Phillip Walker, Steven Nunnally and Michael Lewis
University of Pittsburgh

Nilanjan Chakraborty and Katia Sycara
Carnegie Mellon University

Autonomous swarm algorithms and human-robot interaction (HRI) have both attracted increasing attention from researchers in recent years. However, HRI has rarely extended beyond single robots or small multi-robot teams. While one of the benefits of robot swarms is their robust capabilities and the ability of their distributed algorithms to deal autonomously with the complex interactions amongst swarm members, there is undoubtedly a need for humans to influence such swarms in some circumstances—especially when these swarms are operating in unknown or hostile environments. In this paper, we approach the problem of human-swarm interaction (HSI) using previous research in levels of automation (LOAs) in HRI. We create a target searching task whereby the swarm can operate at two different levels of autonomy: an autonomous dispersion algorithm, or user-defined goto points. We investigate what environmental conditions are conducive to different amounts of human influence, and at what point further human intervention has a detrimental effect on the swarm's performance. The results show that for complex environments containing numerous obstacles and small passageways, there is indeed a need for some human influence; however, after a certain point, further influence causes performance degradation.

INTRODUCTION

Swarms are made up of numerous small, relatively inexpensive robots with basic sensing and locomotive capabilities. By themselves, these individual robots are not able to perform many tasks of interest; however, as a group they can be used to perform certain tasks much more robustly and quickly than a more expensive individual robot or multi-robot team. While the distinctions between swarms and multi-robot systems are still not perfectly defined, for the purposes of this paper, the primary distinction is one of control complexity. Lewis, Wang, & Scerri (2006) use three levels of command complexity to describe the possible workloads faced by a human operator when controlling a robot team. For instance, when performing a simple waypoint following task, control of a robot team is typically $O(n)$, as each additional robot places an increasing demand on the operator because it must be coordinated with all of the pre-existing robots. When the operator must deal with the interactions between the robots, increasing the number causes an exponential increase in the control requirements for the human, making control $O(>n)$. Treating the robots as a swarm alleviates this problem however, because the control algorithm handles the robot interactions, and thus scales to large numbers of robots with little extra effort, making control $O(1)$.

Such control algorithms are typically distributed algorithms that exhibit emergent behaviors based on local interactions, thus allowing the swarm to act as a unified group rather than a collection of distinct agents. Approaches to swarm robotics come from the bio-inspired (Goodrich, Sujit, Kerman, & Pendleton 2011) and physicomimetic (Spears & Spears, 2012) views, as well as amorphous and spatial computing (Khalsa, 2011; Bachrach, McLurkin, & Grue 2008). While these algorithms are powerful for controlling large groups of robots in a decentralized manner and can display a wide range of behaviors, they are difficult to predict and control after the swarm is deployed. A primary goal of this paper is to address this concern in a variety of obstacle-filled environments with a swarm performing a target searching task.

Related Works

Human-swarm interaction (HSI) requires a different approach than human-robot interaction (HRI) because much of swarm control is necessarily autonomous and distributed (specifically the robot interactions). Research in swarm robotics has focused primarily on improving the autonomous control algorithms and the hardware of the robots, ignoring the human control element. A few recent studies have investigated how humans can effectively influence swarms. Kolling, Nunnally, & Lewis (2012) investigated the ability of the human to use various motion control algorithms and communication graphs to accomplish simple tasks, and in Walker, et. al. (2012) and Nunnally, et. al. (2012), the authors looked more closely at a specific target searching task to investigate communication effects—specifically bandwidth and latency—on HSI. While communication is an important aspect when dealing with swarms due to the limited communication abilities available to simple robots, we must also investigate how human operators perceive a swarm and its performance, and how they decide when and how often to act. This second problem is of particular importance, because the manner and frequency with which an operator interacts with the swarm could have a significant impact on the ability of the human-swarm system to accomplish tasks (Walker, et. al. 2012).

In general, human-robot and human-swarm interaction is accomplished through supervisory control (Sheridan & Verplank 1978; Sheridan, 1992; Sheridan, 2002). Although

supervisory control represents a general approach to interaction with automation, this paper only considers supervisory control of robotic applications. Supervisory control in robotics describes a human-robot system in which the robot executes the decision-making and control of tasks in a semi-autonomous manner—requiring intermittent monitoring and control on the part of the human operator. Sheridan & Verplank (1978) proposes a 10-point level of automation (LOA) scale to characterize the degrees of autonomy possible for human-machine systems, ranging from a system where the machine has full autonomy to one where the human controls everything. This scale has been used and modified extensively to describe and evaluate levels of automation for a number of supervisory control systems on differing robotic platforms (Riley, 1989; Kaber & Endsley, 1997; Ruff, Narayanan, & Draper, 2002; Proud, Hart, & Mrozinski , 2003).

In recent years, researchers have extended the idea of LOAs to a second dimension that specifies the task that is being performed. To give a simple example, one human operator supervising multiple unmanned aerial vehicles (UAVs) to explore a given area for targets may want the dispersion and paths of the UAVs to be automated, yet retain control over what is marked as a target and what each of the UAVs will do upon target detection. On the four-stage information-processing model defined in (Parasuraman, Sheridan, & Wickens, 2000), this corresponds to an action implementation stage LOA of 7 or higher (depending on the details of the implementation), and an information analysis stage LOA of 4 or lower. This second dimension allows HRI system designers to vary the LOA depending on the current operation of the robot (Endsley, 1999; Parasuraman, Sheridan, & Wickens, 2000; Miller & Parasuraman, 2003; Miller & Parasuraman, 2007). In Coppin & Legras (2012), the authors introduce the autonomy spectrum, which extends the two-dimensional LOA model by allowing different user-selectable modes at each control task corresponding to different possible LOAs for the same task. Furthermore, their model includes predefined pathways between different LOA combinations at each stage, corresponding to the different possible methods of operation.

This research has provided promising results towards a better theory of how human operators might influence individual robots or small robot teams operating under different LOAs. However, while research in levels of automation in HRI has been primarily concerned with determining which LOAs are appropriate for human operators, there has been little research investigating how a user perceives and interacts with a swarm operating under autonomous or semi-autonomous algorithms. Most algorithms for swarm behavior, such as rendezvous or dispersion, only give provably correct outcomes once many assumptions or simplifications are met (such as a static environment with no obstacles). In most real world cases these assumptions are not met, and thus we need human operators to observe the swarm as it changes and decide when the swarm is operating according to the operator's desired goals and when the human should override the autonomous control and provide corrective input. However, while human supervisory control of swarming

robots may be necessary in these cases, it is unknown how much control is needed, and after what point further operator influence will degrade the performance of the swarm. In other words, what is the best balance between automation and direct human control? This paper investigates this question in the context of a swarm performing a simple foraging task in several environment types.

Hypotheses and Contributions

In this paper, we study the problem of human control of swarm behavior and investigate the proper balance between automation and direct human control. Our approach of influencing the swarm as a whole is distinguished from earlier work in human-multi-robot control, in which individual robots are sequentially controlled. We hypothesize that, in more open environments, a higher amount of automation will suffice in a simple target searching task. However, in environments with dense obstacles, and especially those which contain small passageways and complex obstacle arrangements, a higher level of human influence may improve swarm performance, but only up to a point, after which performance will degrade.

The next section will describe the task that was given to the participants, including a description of the simulation used, the algorithms under which the robots operated, and the overall design of the experiment. The subsequent section will present and discuss the results of the study, followed by a section concluding the paper and presenting possible avenues of future research.

TASK DESCRIPTION

Our study is designed to investigate how humans use different levels of intervention to control a swarm in a target searching task, and whether or not increased human interaction causes a decrease in swarm performance after a certain point.

The Environment

We use four different environments of size 100x100 meters, each containing 100 randomly placed targets initially unknown to both the user and the swarm. Each environment contains a different number and arrangement of obstacles: two with random obstacle placement at high and low densities, one with a more structured obstacle placement (an office building floor plan), and one open control environment (Figure 1).

We use Stage v. 3.2.2 (Gerkey, Vaughan, & Howard, 2003) to simulate the environment, the targets, and a swarm of 100 differential drive P2AT robots. The GUI and robot controllers are implemented using the Robot Operating System (ROS) (Quigley, et. al., 2009). The interface receives the positions and target information from each robot and displays them for the user in a large viewing window, and allows the user to input commands using the mouse and buttons on a side panel. The robots are displayed as small circles, with a heading line indicating the direction each robot is pointing. When the swarm sees a target, it will light up the

color of that target, indicating to the operator that a target is within range of that robot. Each robot is equipped with a proximity sensor with a range of four meters (we use a color blob finder in the ROS implementation as the proximity sensor), which is used to identify the obstacles, targets, and other robots nearby.

Swarm Behavior

The human operator is in charge of moving the swarm around the environment to locate targets that are dispersed randomly throughout the open space. Two different commands are provided to maneuver some or all of the swarm: a *disperse* and a *goto* command. These two modes of operation correspond to high and low levels of autonomy, respectively, with the *disperse* mode being equivalent to a 7 on Sheridan's LOA scale (the system executes autonomously and informs the human), and *goto* mode being equivalent to a 2 (the human chooses from a set of possible actions). The *goto* mode is not fully in the operator's control (a LOA value of 1), because the user cannot control the interactions between the selected robots—only their global goal direction. We will now discuss the *disperse* and *goto* commands in detail.
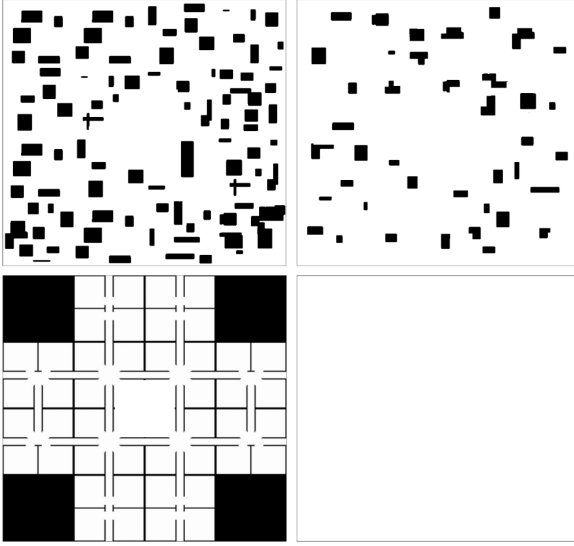

Figure 1. Each of the four environments used in the study: the dense random environment (top left), sparse random (top right), structured (bottom left), and control (bottom right).

Disperse. Each robot performs the *disperse* command by using the blob finder to observe all obstacles, targets, and other robots within 4 meters of the robot. Obstacles and other robots contribute a repulsive force vector scaled linearly by their distance to the robot (Equation 1). Targets have an attractive force vector toward them, which always has a magnitude of 1 (Equation 2).

$$\langle f_{x,repel}, f_{y,repel} \rangle = (\sum_{i=0}^{N} (r-d)*\langle x_i, y_i \rangle)/N \qquad (1)$$

$$\langle f_{x,attract}, f_{y,attract} \rangle = (\sum_{i=0}^{N} \langle x_i, y_i \rangle)/N \qquad (2)$$

In the above equations, *r* represents the maximum range of the

blob finder, *d* the Euclidean distance to the object (as reported by the blob finder), *N*, the number of obstacles returned by the blob finder, and x_i and y_i the *x* and *y* position of the obstacle *i* in the robot's coordinate frame. The attractive forces from the target will always take precedence over the repulsive forces from obstacles and other robots, so long as the robot is not within 1 meter of a repulsive object, in which case the repulsive force takes precedence. Only unidentified targets exhibit an attractive force—identified targets are treated as obstacles. A target is marked as identified once at least 2 robots view it simultaneously. We used this requirement to simulate the need for robustness in target identification, as swarm robots will likely have less reliable sensors on-board with which to identify targets. Requiring multiple simultaneous viewings would help overcome the false reportings of any one sensor. If the robot is currently attracted towards a target, its assigned linear velocity is the scalar component of the force vector if the robot is more than 1 meter from the target, otherwise it is 0 (meaning the robot will wait next to the target until it is identified, or until the user directs the robot elsewhere). The magnitude of the angular velocity is given in Equation 3.

$$v_a = atan2(f_{y,attract}, f_{x,attract}) \qquad (3)$$

If there are obstacles within 1 meter of the robot, or no unidentified targets within range of the blob finder, then the repulsive force takes precedence. In this case, the linear velocity is set to the max speed of *0.4m/s*, and the magnitude of the angular velocity is given by Equation 4.

$$v_a = atan2(f_{y,repel}, f_{x,repel}) \qquad (4)$$

Goto. At any point during the mission, the human may choose to select some subset of the robots to perform a *goto* command. When the operator issues a *goto* command, each robot receives the point specified by the user and sets its angular velocity using Equation 5 and its linear velocity to a constant *0.4m/s*.

$$v_a = atan2(f_{y,goto}, f_{x,goto}) - \theta_r \qquad (5)$$

In Equation 5, x_{goto} and y_{goto} are coordinates of the *goto* point in the world coordinate frame, and θ_r is the robot's heading. Once the robot is within 1 meter of the *goto* point, it automatically switches to performing a *disperse* command. In this manner, the operator can let the swarm disperse to explore and cover new areas autonomously, or override the automation and tell subgroups to move to a location along a specific path. The *goto* command allows the user more control over the behavior of the robots; however, it requires more operator attention, especially if the operator wishes to issue several sequential *goto* commands, say to move part of the swarm around a large obstacle.

Experimental Design

Twenty participants (8 men and 12 women) were recruited from the University of Pittsburgh and surrounding areas to participate in the study. Initially, each participant received instructions on the task—both how to control the swarm and mark targets—and an explanation of how the *disperse* and *goto* algorithms work. After instruction, they were allowed 10 minutes to train themselves with operating

the swarm and using the interface. Following this, each participant had 10 minutes to search each of the four environments in Figure 1, presented in a random order. In all conditions, the participants began with the swarm of 100 robots positioned randomly in a 20x20 meter box at the center. Participants were told that their goal was to find as many targets as possible within the time given.

RESULTS AND DISCUSSION

During the experiments, each robot's position and current mode (*goto* or *disperse*) was logged each second. We used number of targets marked as the dependent variable in this study and the measure of success for each participant in each environment. A one-way ANOVA indicates that, overall, there was a significant difference between environments in terms of the number of targets marked ($F = 19.15$, $p < .001$). Table 1 gives the differences between environments.

Each participant was assigned to one of three equal-sized groups for analysis based on how much of the time their robots spent in *goto* mode (as opposed to *disperse* mode). The members of the swarm in *goto* mode are being directed by the operator to specific locations in the environment, and thus are operating at the lower level of autonomy than those members in *disperse* mode. Post hoc analysis showed that these three groups gave the clearest picture of how much human control was useful, and where the performance dropoff occurred. In the structured environment, the three groups were *low* use of *goto* commands (robots spent on average 37.6 to 91.6% of the time in *goto* mode), *medium* (91.6 to 98.1%), and *high* (98.1 to 100.0%). Results from a one-way ANOVA indicate that in the structured environments, there was a significant difference between these groups ($F = 10.15$, $p = .001$), with the *medium* *goto* group marking significantly more targets (mean $M = 48.67$) than the *low* ($M = 23.43$, $p < .001$) or *high* ($M = 37.43$, $p = .062$) groups (Figure 2). There were no significant differences between the three groups in the dense random ($M_l = 43.14$, $M_m = 62.5$, $M_h = 57.43$, $F = 0.41$, $p = .671$), sparse random ($M_l = 65.14$, $M_m = 76.83$, $M_h = 60.71$, $F = 0.58$, $p = .568$) or control ($M_l = 71.57$, $M_m = 80.33$, $M_h = 73.71$, $F = 0.33$, $p = .727$) environments. In the results above, M_l, M_m, and M_h represent the mean number of targets identified in each of the *low*, *medium*, and *high goto* groups, respectively. We chose to group the participants afterwards as opposed to placing them in predefined groups beforehand because we were unsure at what point further human intervention would lower performance. These results indicate that a significant amount of human intervention was necessary in the structured environments (around 91 to 98%), however, full human control was less successful.

	Random Sparse	Structured	Control
Random Dense, $M = 53.95$	.020*	.002*	< .001*
Random Sparse, $M = 67.10$	-	< .001*	.159
Structured, $M = 35.90$	-	-	< .001*
Control, $M = 74.95$	-	-	-

Table 1. The p-values for each of the environment pairs compared by mean number of targets found. Significant differences are indicated by an asterisk.

Further analysis shows that the most successful operators use the *goto* mode to break their swarms up into several subgroups, each larger than or equal to the threshold needed to mark a target, and directed those groups to explore separate areas of the map. Subgroups were represented by connected components of the sensing graph of the robots, whereby two robots were connected if they were within sensing range of each other. A linear regression on the data shows that operators with a higher number of connected components of sufficient size (i.e. larger than the target threshold) found more targets on average ($r^2 = .096$, $p = .005$, see Figure 3). Here, r^2 is the correlation coefficient of determination for the linear model. The data also show that, when grouped into equal-sized *low*, *medium*, and *high* groups by number of connected components, the *high* group maintained their higher number of connected components throughout the experiment more consistently, staying on average at 70.0% of the maximum number of connected components attained throughout the experiment (standard deviation $\sigma = 0.059$). This was significantly higher than the *low* ($M = 55.5\%$, $\sigma = 10.5\%$, $p = .023$) and *medium* ($M = 60.8\%$, $\sigma = 7.5\%$, $p = .019$) groups.
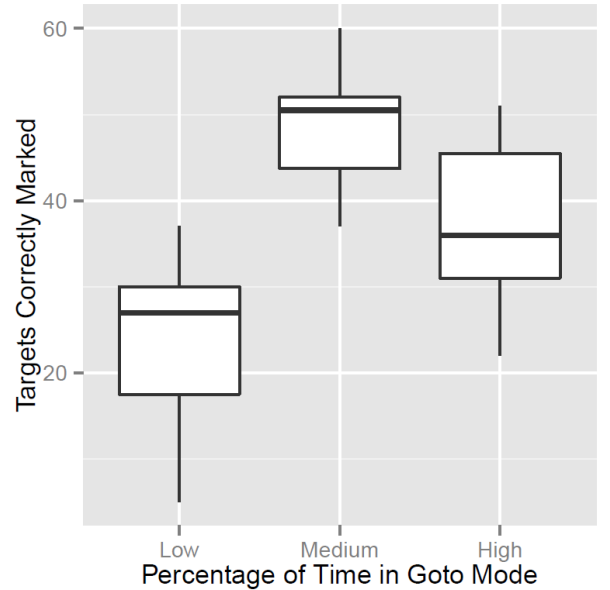

Figure 2. The number of targets marked by the each of the *low*, *medium* and *high* groups of participants in the structured environment. The groups were determined by the average time the robots spent in *goto* (non-autonomous) mode.

These results indicate that operators who either rely on the automation too much by leaving the swarm in *disperse* mode, or who take manual control nearly all the time, do worse than those who strike a balance between high and low automation. This result was only evident in the structured environment, as this environment required some intervention to move subgroups of the swarm into different areas and through the doorways and hallways, but too much intervention caused robots to get stuck in corners, likely causing operators to spend more time switching between subgroups than searching for targets.

In the more open and less structured environments (i.e. the dense and sparse random and control maps), differences in the usage of the autonomous dispersion algorithm and manual control via *goto* commands had no effect on the number of targets identified. This is likely because there was little downside to manually controlling the swarm, as there are no corners or confined spaces in which the swarm could get stuck. The most successful operators overcame the disadvantages of reducing inter-robot distances through use of the *goto* command by breaking the swarm up into subgroups that searched different areas of the map.
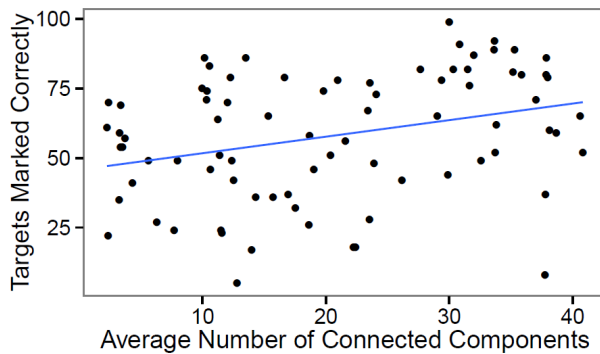

Figure 3. The number of targets marked based on the average number of connected components that were larger than the target identification threshold. Each point represents one participant in one condition.

CONCLUSIONS AND FUTURE WORK

This study provides support for the idea that a balance between levels of automation is necessary in HSI. In structured environments, some intervention was needed to properly disperse a swarm and discover targets throughout the map; however, when the operator took too much control and never allowed the automation to operate, performance declined. We also demonstrated that in less structured environments, participants were able to adopt one of two strategies to find targets successfully: either leave most of the control of swarm movement to the autonomous dispersion algorithm, or manually break the swarm up into subgroups to explore different areas of the map.

Future work should investigate separating operators into predefined LOA groups for other swarm tasks, including mapping and target identification tasks with realistic sensor models, to determine if the optimal amount of human influence is similar to this study. Other work could also improve the dispersion algorithm to allow for adequate coverage even when there are significant obstacles in the environment. Finally, future research should also investigate how humans perceive the ability of autonomous algorithms to perform certain tasks, and whether they believe their own manual influence improved or could improve swarm performance.

AKNOWLEDGEMENTS

REFERENCES

Bachrach, J., McLurkin, J., & Grue, A. (2008). Protoswarm: a language for programming multi-robot systems using the amorphous medium abstraction. *Proceedings of the 7th international joint conference on autonomous agents and multiagent systems, 3,* 1175–1178.

Coppin, G. & Legras, F. (2012). Autonomy spectrum and performance perception issues in swarm supervisory control. *Proceedings of the IEEE, 99,* 590–603.

Endsley, M. (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics, 42,(3),* 462–492.

Gerkey, B., Vaughan, R., & Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. *Proceedings of the 11th international conference on advanced robotics*, pp. 317–323.

Goodrich, M. Sujit, P., Kerman, S., Pendleton, B. (2011). Enabling human interaction with bio-inspired robot teams: Topologies, leaders, predators, and stakeholders. *Brigham Young University, Tech. Rep.*

Kaber, D. & Endsley, M. (1997). Out-of-the-loop performance problems and the use of intermediate levels of automation for improved control system functioning and safety. *Process Safety Progress, 16(3),* 126–131.

Khalsa, K. (2011). Realistic simulation of spatial computers and robot swarms. *Ph.D. dissertation, University of Colorado, Boulder.*

Kolling, A., Nunnally, S., & Lewis, M. (2012). Towards human control of robot swarms. *Proceedings of the 7th international conference on human-robot interaction*, pp. 89–96.

Lewis, M., Wang, J., & Scerri, P. (2006). Teamwork coordination for realistically complex multi robot systems. *NATO Symposium on Human Factors of Uninhabited Military Vehicles as Force Multipliers,* 2006.

Miller, C. & Parasuraman, R. (2003). Beyond levels of automation: An architecture for more flexible human-automation collaboration. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 47(1),* 182–186.

Miller, C. & Parasuraman, R. (2007). Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 49(1),* 57–75.

Nunnally, S. , Walker, P., Kolling, A., Chakraborty, N., Lewis, M., Sycara, K., & Goodrich, M. (2012). Human influence of robotic swarms with bandwidth and localization issues. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 333–338.

Parasuraman, R., Sheridan, T., & Wickens, C. (2000). A model for types and levels of human interaction with automation," *IEEE International Conference on Systems, Man and Cybernetics, 30(3),* 286–297.

Proud, R., Hart, J., & Mrozinski, R. (2003). Methods for determining the level of autonomy to design into a human spaceflight vehicle: a function specific approach. *DTIC Document, Tech. Rep.*

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. (2009). Ros: an open-source robot operating system. *ICRA Workshop on Open Source Software, 3(3.2).*

Riley, V. (1989). A general model of mixed-initiative human-machine systems. *Human Factors and Ergonomics Society Annual Meeting Proceedings, 33(2),* 124–128.

Ruff, H., Narayanan, S., & Draper, M. (2002). Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple simulated unmanned air vehicles. *Presence: Teleoperators & Virtual Environments, 11(4),* 335–351.

Sheridan, T. (1992). *Telerobotics, automation, and human supervisory control.* The MIT press.

Sheridan, T. (2002). *Humans and automation: System design and research issues.* John Wiley & Sons, Inc., 2002.

Sheridan, T. & Verplank, W. (1978). Human and computer control of undersea teleoperators," *DTIC Document, Tech. Rep.*

Spears, W., & Spears, D. (2012). *Physicomimetics: physics-based swarm intelligence.* Springer-Verlag New York Inc.

Walker, P. M., Kolling, A., Chakraborty, N., Nunnally, S., Sycara, K., & Lewis, M. (2012). Neglect benevolence in human control of swarms in the presence of latency. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3009–3014.