2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)
Bari, Italy. October 6-9, 2019

Moderating Operator Influence in Human-Swarm Systems

C. Chace Ashcraft, Michael A. Goodrich, and Jacob W. Crandall

Abstract— **In human-swarm systems, human input to a robot swarm can both inhibit desirable swarm behaviors and allow the operator to properly guide the swarm to achieve mission goals. Indeed, the way that control is shared between the human operator and the inherent collective robot behaviors determines in large part the success of the human-swarm system. In this paper, we seek to understand how to design human-swarm systems that effectively moderate human influence over a robot swarm. To do this, we implement a simulated swarm system based on honeybees, and study how interacting with this swarm using various methods of moderating human influence impacts the success of the resulting human-swarm system. Our results demonstrate that moderating human influence is essential to achieving effective human-swarm systems, and highlight the need for future work in determining how to better moderate human influence in human-swarm systems.**

I. Introduction

Robotic swarms have great potential use in many practical applications. Swarm technology is based on having simple robots perform complex tasks through local sensing and communication between robots. The complex or intelligent behavior is said to *emerge* through the interactions between the swarm robots rather than through a centralized controller. Because the emergent behaviors are often the result of complex interaction rules that are not well understood, it is difficult to design algorithms that ensure that the swarm will exhibit the desired behavior during deployment, especially for previously unanticipated environments and scenarios.

Human interaction with the swarm can potentially provide the necessary flexibility for the swarm to adapt to less-understood environments and unanticipated scenarios [1], [2]. However, human interaction with a robot swarm presents a new challenge for command and control. As the size of the swarm increases, control must become more focused on the swarm as a whole rather than on the individuals of the swarm [3]. This may seem obvious considering humankind's limited capacity to multitask [4], [5], but how to share control with a swarm is less obvious. Winfield and Nembrini state that "A distinguishing characteristic of distributed systems based upon swarm intelligence is that they have no hierarchical command and control structure, and hence no common mode failure point or vulnerability" [6]. Ironically, the thing that makes swarms resilient, their decentralization, is what makes human control over them difficult. By adding a human operator to manage the swarm, one adds an element of centralization and potentially restricts the autonomy of each robot, which is usually the source of the swarm's robustness and emergent behavior. In the case of low-level interactions, it can be challenging for people to give inputs that produce the desired effect on the swarm.

In this paper, we consider the problem of designing the human-swarm system such that control over the swarm's behavior is effectively shared by the human operator and the robots' autonomy. We propose an interaction scheme design to keep the amount of influence the human has over the swarm at a moderated level, ensuring the operator has sufficient control over the swarm without overriding the swarm's emergent behavior. In so doing, we hope to allow the swarm to take advantage of useful human input and ignore detrimental input. We then implement a specific human-swarm system to investigate our theory, and analyze the system via simulation and user study. Our results confirm the importance of moderating operator influence based on (a) human skill with the swarm and (b) the information they are provided, but also show that future work is needed to learn how to adequately moderate human influence in human-swarm systems.

II. Related Work

Kolling *et. al.* published a survey in 2016 summarizing work in human-swarm interaction [2]. In this survey, they categorize swarm models into four categories: Bioinspired, Control-theoretic, Amorphous Computing, and Physics-inspired. Biological systems are likely some of the earliest inspirations for swarms, and are readily tied to the term *swarm*. Perhaps one of the most popularly implemented and studied model, at least in simulation, is Couzin *et. al.*'s model [7], which has been used to display a variety of interesting spacial behaviors. It has also been used to study human-swarm interaction [8], [9].

The model [10] we use in this paper is bioinspired, but is different from most swarm models. The behavior we are interested in is less the spatial positioning of the swarm than their ability to accomplish a certain task [11], [12], [13]. Sumpter published several studies on biological swarms and collective behavior, as well as principles for engineering bioinspired swarm systems [14], [15].

In this paper, we attempt to combine and build upon two topics of related work in the context of humans interacting with large, task-based robot swarms: *Neglect Benevolence* and *Shared Control*. Neglect Benevolence is the property that the system's performance will improve by having the human delay input [16]. This is opposed to *Neglect Tolerance* [4], which regards how long a system can continue to perform sufficiently well without human input. Nagavalli *et. al.* ([16]) showed that allowing a human operator of a simple swarm

*The authors are with the Computer Science Department, Brigham Young University, Provo, UT, USA, and can be reached at cc.ash@byu.edu, mike@cs.byu.edu, and crandall@cs.byu.edu, respectively.

to provide input to the swarm too early can result in a sub-optimal outcome.

We also build upon the concept of shared control in human-swarm systems. Shared control refers to attempts to improve the performance and capability of a human-machine system by balancing the intent of the operator with the sensors and algorithms that run the machine [17], [18]. For example, Crandall and Goodrich showed that sharing control during teleoperation of a single robot increased performance of the human-robot team, reduced the amount of attention the robot needed to function correctly, and was easier to use than manual teleoperation. In 2014, Brown, Jung, and Goodrich studied shared control with a bio-inspired swarm based on Couzin's model [19].

Finally, while much research in swarms is done using techniques such as graph theory and differential calculus, we have chosen to use an agent-based model. Agent-based models are often used to examine or study the effects of simple rules or behaviors on a large set of robots interacting in an environment [20], especially when more rigorous methods of analysis become computationally prohibitive (see also examples in [21]).

III. Shared Control in Human-Swarm Systems

In this section, we discuss the impact of human influence on the swarm, and propose human influence as a primary consideration in the design of human-swarm systems.

A. Terminology

There are three, not necessarily distinct, parties of interest integral to the development and deployment of robot swarms: the *Problem Holder*, the *Designer*, and the *Operator*. The Problem Holder is the person or group of people who define the purpose of the swarm, fund the Designer, and most likely employ the Operator. The Designer is the person or group responsible for the design and implementation of the robot swarm of interest, and the Operator is the person or group that interacts with the swarm during its deployment.

Another important term for this work is *influence*. Influence is the ability of the Operator to alter the behavior of the robots in the swarm. In the context of complex distributed systems, a general definition of influence is difficult to derive. Therefore, in this work, we will only analyze influence comparatively, i.e. we will claim that one interaction scheme provides an operator with more influence than another if it gives the operator greater ability to dictate the actions of the members of the swarm. This imprecise definition is less than ideal, but is sufficient for this work.

B. A Theory

Due to the complexity of swarm systems and the sometimes inherent communication limitations present in such systems, it is often difficult for a human to understand the state of the swarm at any particular time, let alone how to interact with it in order to drive it to some desired state. Thus, while it may be desirable to have a human operator in a swarm system, it is challenging to know how best make use of the human. For example, allowing an operator to have high influence over the swarm with little training may cause the operator to hinder or block the desirable emergent behaviors that make the swarm useful.

We propose that there are three primary methods to overcome this problem:

1. Training: Overcome challenges and problems with swarm interaction by making the operator an efficient and knowledgeable user of the swarm.
2. User interface design: Make the user interface sufficiently intuitive and easy to use such that human users can easily know what to do and how to do it.
3. User interaction design: Design the interaction method in order to take advantage of useful human input while moderating the effect of bad input.

While all three approaches can improve the performance of human-swarm systems, we focus on the third category: user interaction design.

We argue that a primary consideration when designing a human-swarm interaction scheme should be the influence the Operator has over the swarm. At a high level, the Problem Holder has a purpose for the swarm, as well as some measure of performance, but depends on the Designer and Operator to fulfill that purpose. The Designer can design the swarm to maximize the performance function given by the Problem Holder under various assumptions, which will most likely be broken at various times during deployment. The Operator can assist the swarm in maximizing its performance by providing information unavailable to the swarm's sensors, providing high-level reasoning the swarm is unable to perform, or overcoming issues caused by the violation of swarm design assumptions. However, if the Operator has too much influence over the robot's actions, it can override the swarm's collective behaviors, which will decrease the swarms performance and robustness. Thus, a balanced level of influence can potentially allow the swarm to take advantage of good input while being unaffected by detrimental input, as in the conceptual illustration in Figure 1.

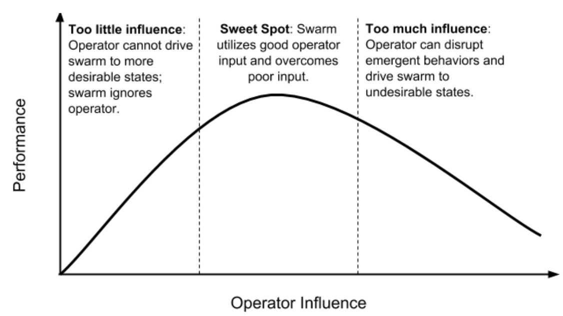


Fig. 1. Conceptual relationship between Operator influence and swarm task performance for some task and swarm.

While the *sweet spot* defining desirable levels of human influence is contingent on the swarm system, environment, and specific operator knowledge and capabilities, some general principles for balancing human influence would be desirable. Thus, in the remainder of this paper, we explore the impact of human influence on swarm performance. We also propose and evaluate *Feedback Based Dynamic Influence*, the idea that Operator influence should change over time based on feedback from the swarm and Operator, as a general design principle of human swarm systems. By allowing influence to change based on feedback from the swarm and Operator, the Designer can maintain balanced influence. Exactly how the influence should vary based on the feedback will depend on the swarm, task, and possible feedback, but as the Designer will be designing and implementing all three, these are variables the Designer can control for.

We explore these design principles, via simulation and user study, in the context of a particular robot swarm patterned after honey bees.

IV. A Robot Swarm

To begin to study how operator influence impacts a swarm's performance, we implemented a simulated swarm designed to solve the best of N problem. Our swarm is patterned after a hub-based colony, a swarm that revolves around a central location or hub. Example biological hub-based colonies include ants, bees, and termites.

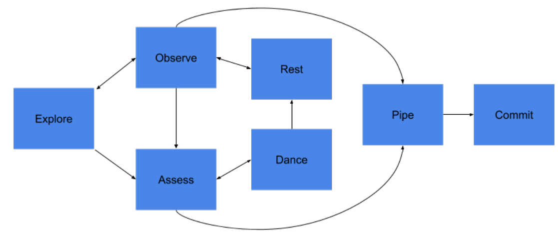
A. Honeybee Model

The swarm we examine in this work is adapted from the honeybee model defined by Nevai *et. al* [10] for nest site selection [22], [23], [24]. Our implementation of this swarm is an agent-based model with robots following the behavior described in Figure 2. In this system, each simulated robot is assumed to be simple and inexpensive. As a consequence, we assume that each robot is capable only of simple behaviors, short-range communication, limited signal processing, and no real-time GPS tracking.

The task this model attempts to accomplish is high quality site selection in a large environment in limited time. In this task, each potential site in the environment has a site quality. A robot assesses this quality when it visits the site, and the swarm dynamics then over time allow the swarm to select a particular site in the environment. Practical applications of this model could include any task that requires selecting a single location out of many in a large environment, where site quality can be assessed by a robot. Further applications could be plausible by making small modifications to the swarm, but for simplicity, we work with the model as stated. We assume that the Problem Holder's purpose for this swarm is to select the best site in the environment, and so we use that as our primary measure of performance.

B. Characteristics of the Robot Swarm

To better understand the swarm, we evaluate it's ability to select the best site in the environment under various circumstances. In particular, we exam (a) how the number robots in the swarm and (b) the distribution of potential sites in the environment impacts the swarms ability to find the site with the highest quality.



Explore	The robot randomly explores the environment for a finite time, seeking potential nest sites.
Observe	The robot returns to the hub (if not there already) and randomly moves about the hub watching for dancers and pipers.
Assess	A robot in this state is attempting to assess the quality of a site. This may come from the robot discovering a site during exploring or by observing a dancer advertise a site.
Dance	After a robot discovers a site and assesses it itself, it returns to the hub to communicate its findings to the rest of the colony through a "dance." Real honey bees perform what is called a *waggle dance*, but we only simulate its effect.
Rest	Biological bees need to rest, and we assume robot bees will need to charge or something similar. Regardless, having robots simply go to the hub and do nothing for a period of time seems important to the total dynamics of nest selection.
Pipe	When the number of robots assessing a site exceeds a threshold, robots begin to pipe. In real bees this is thought to be done by bees vibrating their wings at a certain frequency around the bees at the hub. This "warms them up" and they start doing the same.
Commit	When all robots at the hub are piping, the collective concludes that it has made a choice, and the whole colony moves to the site that was piped for.

Fig. 2. Honey bee model state transition diagram and table adapted from the honey-bee model defined by Nevai *et. al* [10].

1) Quantity Increases Performance: We experimented with various numbers of robots in the swarm in two environments with equidistant sites and various site qualities. Experiments consisted of 100 simulations for each environment/robot-number combination. The results are given in Figures 3 and 4. These results illustrate that having more robots increases the likelihood the swarm selects the best site. However, more robots also increase the time it takes for the swarm to make a choice. This is because more robots search the environment more thoroughly and with more redundancy, but more robots make it more difficult for the convergence criteria of the swarm to be satisfied.

2) Effects of Site Distribution: The distribution of sites throughout the environment strongly affects the performance of the swarm. Three particular characteristics of site distribution that we have noticed regard the number of sites in the environment, the distance the sites are from the hub, and if sites are *blocked* (by line of site to the hub) by other sites.

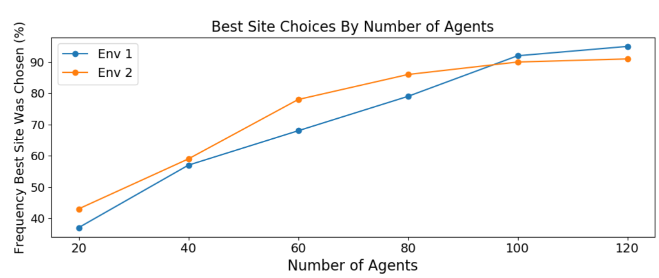


Fig. 3. Site choices for two different environments, with equidistant sites of various qualities, and six difference numbers of robots.

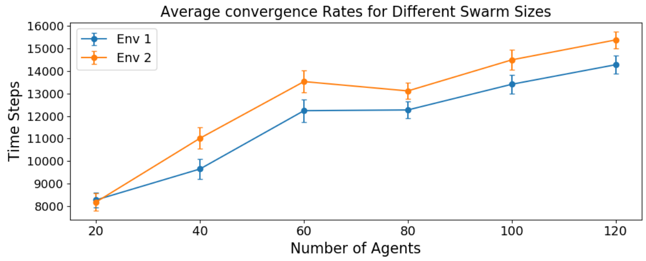


Fig. 4. Convergence rates for two environments and six different numbers of robots.

We tested the swarm in environments with 6, 12, 24, and 48 equidistant sites with qualities evenly distributed between 0.01 and 0.9, with 50 robots, and with 100 simulations each. The results, given in Table I, show a steady increase in convergence time and a steady decrease in best site selection as the number of sites increases.

TABLE I

SWARM PERFORMANCE AS THE NUMBER OF SITES INCREASE.

# Sites	Best Site Selection	Avg Converge Rate (in time steps)
6	65%	12534
12	60%	19797
24	48%	26170
48	39%	33201

We also varied the distances of sites from the hub. In one environment we placed 20 sites evenly distributed around the hub and at equidistance from the hub. In a second environment, we moved the best site 150% farther from the rest of the sites. A swarm with 100 robots exhibited a 52% decrease in best-site selection in the second environment compared to the first, and a 49% increase in average convergence time. We did a similar test with the environment in Figure 5, with and without the (yellow) *blocking* sites, which obscure the higher-quality sites from the hub. This makes the higher-quality sites more difficult for the robots to find since a robot returns to the hub once it encounters a site to report its quality. In all, the blocking sites reduced the frequency of best site selection from 95% to 55%, and increased the average convergence time by 26%.

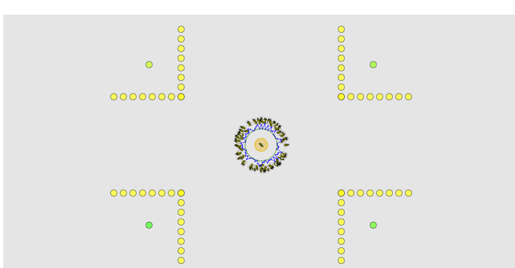


Fig. 5. Environment with higher-quality (green) sites obscured from the hub by lower-quality (yellow) sites.

These initial results illustrate that this robot swarm, with a sufficient number of robots, often identifies the best site in the environment under ideal environmental conditions. However, when the swarm has lower numbers of robots or the environmental conditions are not ideal, the swarm often fails to find the best site.

V. HUMAN INTERACTION WITH THE ROBOT SWARM

Given that the robot swarm does not always find the best site in the environment on its own, a human operator that either possesses some knowledge of the environment or knowledge of the swarm's weaknesses could potentially help the swarm improves its performance. To study the role of operator influence on the Operator's ability to help the swarm, we developed a user interface that allows an operator to interact with the same. This interface has two components, a GUI display, which displays the status of the robot swarm to the Operator, and an input mechanism.

A. GUI Display

The GUI display consists of a 2D representation of the simulated environment which provides the Operator with information about the environment and the robot swarm. A screen shot of this display is shown in Figure 6. Potential sites are displayed as colored circles, with the color of the site indicating its quality. Site qualities are real numbers in the left open interval $(0, 1]$, which map to colors between dark red (low-quality close to 0) and dark green (high quality).

The robots' hub is the yellowish circle at the center of the environment (called the "hub"). The robots begin at the hub and then begin exploring in random directions. They then return, report discovered sites, and transition states according to Figure 2. The locations of robots outside the hub are unknown. To give the Operator information about where robots are, the number of robots leaving and entering the hub in each direction is shown via the *radial display* surrounding the hub. Discovered sites are displayed via smaller colored circles, called *robot markers*, similar to potential sites. The *best site indicator* is a purple circle with a site quality printed next to it that displays the best site reported for a given 1,000 time step window.

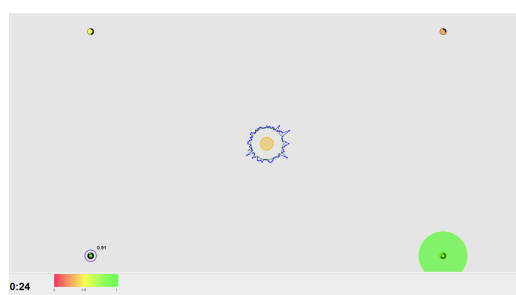


Fig. 6. An example of the UI with a simple environment, consisting of four sites equidistant from the hub. Robot markers, the radial display, the best site indicator, and a user-placed beacon are also displayed.

B. Input Mechanism – Beacons

The Operator places *beacons* on the GUI display corresponding to a location in the environment the Operator would like the robots to explore. Placed beacons are displayed on the GUI as green transparent circles, with the size of the circle being the beacon's radius of effect. We consider two different types of beacons that exhibit different levels of influence, referred to as *Attractor 0* and *Attractor 1*:

- **Attractor 0**: With probability 0.8, exploring robots that enter the radius of effect of a beacon will be attracted towards the center.
- **Attractor 1**: Same as Attractor 0, but also causes some observing robots in the hub to explore in the direction of the beacon. These robots ignore any potential site outside of a beacon's radius of effect.

Attractor 1 provides the Operator with more influence than Attractor 0, as it alters more robot behaviors. The difference in influence becomes especially obvious when noticing that, later in the simulation, exploring robots are very rare due to the assessing and recruiting processes. Therefore, Attractor 0 tends to only have influence towards the beginning of the simulation, while Attractor 1 maintains some influence until the swarm converges.

C. Simulation Results

Two examples illustrate the trade-offs associated with giving the Operator influence over the swarm using Attractors 0 and 1. The first example deals with the environment shown in Figure 6. In this scenario, the Operator receives (potentially false) information about the site qualities from an external source, and relies on this information to (stubbornly) push the robots toward the best site according to this information (disregarding information provided by the swarm as it explores the environment). The results, given in Figure 7, show that Attractor 1 provides enough influence for the Operator to push the swarm to select any site in the environment the Operator desires, regardless of its actual quality. On the other hand, Attractor 0 allows the swarm to more frequently choose the best site even when the Operator tries to push the swarm towards a less desirable site.

Fig. 7. Site selection results with the Operator consistently placing beacons on a single site for the whole simulation.

The second scenario deals with the environment shown in Figure 8, which contains hazards, called *traps*, surrounding the highest-quality site. These traps destroy robots that enter them. We then simulated the effect of the Operator attempting to attract robots to the high-quality site behind the traps using the two beacon types. This behavior is possible if the Operator is unaware of the traps, but has information about potential sites.

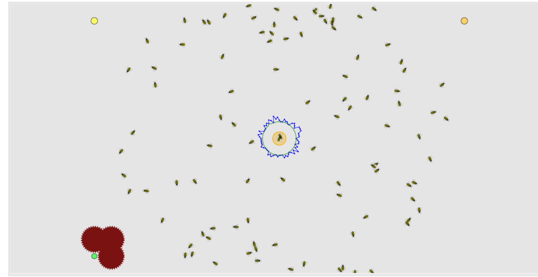


Fig. 8. Environment with hazards (traps – red circles) around the highest quality site.

For Attractor 0, no more than 21% of the robots were lost to traps, while with Attractor 1 at least 80% were lost. Depending on the convergence criteria, this implies that lower influence (Attractor 0) allows the swarm to choose one of the other sites, but higher influence (Attractor 1) allows the Operator to destroy too many robots for the swarm to converge. The traps violate an implicit environmental safety assumption of the Designer, and poor Operator input with higher influence worsened the consequences over the same Operator behavior with lower influence. Thus, it seems that the lower-influence setting may be preferred.

D. Increasing Performance By Moderating Influence

In context of the previously defined parties of interest, we can interpret the previous results as the Problem Holder desiring a swarm that discovers and selects the highest quality site in an environment, the Designer using a modified honeybee model to develop a swarm to do that, and the Operator using beacons to assist the swarm and increase its performance. However, through poor decision-making or lack of information, the Operator decreases swarm performance.

We attempt to modify influence using *Feedback Based Dynamic Influence* by implementing software that monitors robot information as they enter and exit the hub, and then dynamically adjusts what actions the Operator can take and how the robots respond to them. We call this software the *Influence Verification and Adjustment Module*, or *IVAM*.The IVAM only allows a certain number of robots to leave the hub in a given direction when called by Attractor 1, until robots leaving in that direction return to the hub. The number of robots allowed to leave in a direction depends on the distance from the hub the beacon is placed, and ranges from 1 to 5 robots. In preliminary experiments, the IVAM successfully provided Attractor 0 performance in the case of poor Operator input, while maintaining Attractor 0 and Attractor 1 performance for good input.

VI. User Study

To both evaluate the IVAM and to further investigate the impact of operator influence on the performance of the robot swarm, we conducted a user study.

A. Experimental Design

In this study, we evaluated how *influence* and *information* impact the performance of the robot swarm. By displaying incorrect or incomplete site information, we force the human to either rely more on feedback from the swarm or potentially provide input that is detrimental to swarm performance. Thus, varying information quality allows us to evaluate human performance under various practical conditions.For each independent variable, we designate three possible values. The levels of influence we considered were **High**, **Low**, and **IVAM**. The three levels of information, which refers to the accuracy of the site information given to the user on the GUI display, were **Perfect**, **Missing**, and **Mislabeled**. Descriptions of the different levels for each variable are given in Table II. The values of influence vary between subjects, while the values for information vary within subjects. The result is a 3x3 user study.

TABLE II

DESCRIPTION OF INFORMATION SETTINGS.

Influence	**Description**
High	8 beacons may be placed simultaneously.
Low	1 beacon may be placed at a time.
IVAM	1 beacon may be placed at a time, and IVAM is enabled.
Information	**Description**
Perfect	The information displayed to the user is the same as the true environment.
Missing	Some sites are not displayed to the user, this always includes the best quality site in the environment. Qualities of sites shown are accurate.
Mislabeled	All site locations are shown accurately, but most or all site qualities are incorrectly displayed.

In the study, each user experienced three simulated deployments, each in a different environment and with different information. The order of the environments was consistent between users, but the level of information was fully counterbalanced across the study. Environments were designed to be difficult for the swarm to find the best site, but included human-recognizable patterns that the human could potentially use to assist the swarm. Each simulation ran for approximately 16 minutes. If the swarm did not converge before the time limit, it was recorded as a failure to converge. Various other parameter settings for the simulations are given in Table III. We recruited 36 participants for 108 data points, or 12 data points per influence-information pair.

TABLE III

CONSTANT SIMULATION PARAMETERS ACROSS ALL VARIABLES. NOTE THAT ts STANDS FOR "TIME STEPS," AND s FOR "SECONDS."

Number of Robots	100
Beacon Type	Attractor 1
Beacon Radius	75
Beacon Duration	500 ts ($\approx 7.1s$)
Probability Robots Ignore a Beacon	0.2

The primary performance measure, designated by the Problem Holder, is the percentage of trials in which the best site was chosen. If the performance measures previously described are higher for the IVAM influence type than for the others, this suggests that the IVAM appropriately moderates Operator influence for our human-swarm system. Otherwise, the IVAM is providing too much or too little influence to the Operator, and our theory and implementation will require further examination.

B. Simulation Results

Prior to the user study, we tested IVAM using two different simulated operator behaviors. We refer to the two operators as *AI 1* and *AI 2*, and assign them the following behaviors:

- AI 1 (poor input): Places beacons on the best site given in the initial information until convergence or the end of the simulation.
- AI 2 (better input): Places beacons on the best site given in the initial information, but then updates site information based on feedback from robot markers and the best site indicator. Places beacons on the best known site until convergence or the end of the simulation, and slightly slows beacon placement after 7.1 minutes.

Swarm performance for the three environments without human input is provided in Table IV, and results for the simulated operators are given in Tables V and VI.

TABLE IV

FREQUENCY THAT SWARM CHOSE BEST SITE SANS OPERATOR INPUT.

Environment	Ratio	Probability
1	39/100	0.39
2	38/100	0.38
3	26/100	0.26
Overall	103/300	0.34

TABLE V

SIMULATED USER STUDY RESULTS FOR AI1.

Infl\Info	Perfect	Missing	Mislabeled	Totals
High	30/30 (1.0)	0/30 (0.0)	0/30 (0.0)	30/90 (0.33)
Low	30/30 (1.0)	0/30 (0.0)	1/30 (0.03)	31/90 (0.34)
IVAM	29/30 (0.97)	7/30 (0.23)	11/30 (0.37)	47/90 (0.52)

TABLE VI

SIMULATED USER STUDY RESULTS FOR AI2.

Infl\Info	Perfect	Missing	Mislabeled	Totals
High	20/30 (0.67)	11/30 (0.37)	10/30 (0.33)	41/90 (0.46)
Low	22/30 (0.73)	6/30 (0.2)	10/30 (0.33)	38/90 (0.42)
IVAM	28/30 (0.93)	11/30 (0.37)	18/30 (0.6)	57/90 (0.63)

The results suggest that the IVAM can help maintain or increase swarm performance compared to the other two influence types, which is a good indication that the IVAM algorithm is balancing influence as desired. These results are encouraging, and further motivate our user study.

C. User Study Results

We recruited 36 people to participate in the study. Out of all participants, 30 were students, 20 were male, 16 were female. 10 of the students were in Computer Science or Computer Engineering, while the rest came from various other fields ranging from open major to Physiology and Developmental Biology. User ages ranged from 18 to 53.

The results differ substantially from those observed in the simulated study, as IVAM did not outperform the other influence types. In the case of Perfect information, it seems that

IVAM was too restrictive, as Low influence outperformed it. For both imperfect information types, High influence performed the best. This suggests that High influence is the best choice for imperfect information, Low is for Perfect information, and IVAM should not be used at all. Although the IVAM influence was effective with simulated operators, it ultimately failed with human operators. The results suggest that the correct amount of influence varies at least based on the accuracy of the information provided to the user, and the user's ability with the swarm.

TABLE VII

USER STUDY RESULTS.

Infl\Info	Perfect	Missing	Mislabeled	Totals
High	6/12 (0.5)	8/12 (0.67)	11/12 (0.92)	25/36 (0.69)
Low	11/12 (0.92)	4/12 (0.33)	7/12 (0.58)	22/36 (0.61)
IVAM	9/12 (0.75)	4/12 (0.33)	5/12 (0.42)	18/36 (0.5)

Average beacon usage is provided in Table VIII, and average convergence times (non-converging cases omitted) in Table IX. Unsurprisingly, High influence had the highest beacon use, though usage was well below the maximum usage of 67 beacons per minute, suggesting that human users were often conservative with beacon use. We also see that Low influence, while performing at least as well as IVAM, did so with fewer beacons on average. Convergence times are surprising similar for each influence type, but also show High influence converged faster on average with perfect information, and Low with imperfect information.

TABLE VIII

AVERAGE BEACON PER MINUTE USAGE FOR INFLUENCE-INFORMATION PAIRS.

Infl\Info	Perfect	Missing	Mislabeled	Overall
High	18.87	22.89	16.81	19.5
Low	4.92	5.09	4.80	4.94
IVAM	5.89	5.76	5.41	5.69

TABLE IX

CONVERGENCE TIME RESULTS BY INFLUENCE TYPE WITH NON-CONVERGENCE CASES OMITTED. UNITS ARE IN MINUTES.

Influence	Min	Max	Avg	StDev
High	2.7	14.5	6.8	3.1
Low	3.1	12.9	6.4	2.5
IVAM	3.1	13.3	6.8	2.5

Over the three environments considered, the swarm converged to the best site about 34% of the time in the absence of human input (Table IV). Table VII shows that the human-swarm system chose the best site with substantially greater frequency. This suggests that the human operators, on average, improved the swarm's performance.

D. Statistical Analysis

We use the GLIMMIX procedure from the SAS statistical software to examine the statistical significance of our results. The results for the Type III Test of Fixed Effects are provided in Table X. From these results, we observe that influence was not statistically significant alone, but information was, and that there is an interaction effect between influence and information. Therefore, influence should not be considered independently from information, but we may consider influences within each information type or between influence-information pairs.

TABLE X

TYPE III TEST OF FIXED EFFECTS RESULTS.

Effect	Num DF	Den DF	F Value	Pr > F
Influence	2	66	1.26	0.2906
Information	2	66	3.55	0.0344
Infl*Info	4	66	3.02	0.0237

Out of the 36 possible pairwise comparisons, only eight show statistically significant differences (i.e. $p \leq 0.05$). Those significant comparisons are given in Table XI.

TABLE XI

STATISTICALLY SIGNIFICANT DIFFERENCES (USING $p = 0.05$) BETWEEN INFLUENCE-INFORMATION PAIRS. NOTE THAT H=HIGH, L=LOW, I=IVAM, P=PERFECT, MS=MISSING, AND ML=MISLABELED.

(Infl, Info) 1	(Infl, Info) 2	t	Pr > t
(H, ML)	(H, P)	2.04	0.0453
(H, ML)	(I, ML)	2.23	0.0289
(H, ML)	(I, MS)	2.50	0.0149
(H, ML)	(L, MS)	2.51	0.0145
(I, ML)	(L, P)	-2.26	0.0271
(I, MS)	(I, P)	-2.01	0.0486
(I, MS)	(L, P)	-2.53	0.0139
(L, MS)	(L, P)	-2.63	0.0107

E. Non-convergence Analysis

Table XII shows that a significant number of failures to choose the best site were caused by the swarm failing to converge in the allotted time. We have observed two main causes of non-convergence in this human-swarm system. First, operators can prevent the swarm from meeting its convergence threshold by consistently placing beacons over many sites or over a single site, particularly with High influence. Such behavior overrides the swarm's inherent emergent behavior. Second, the swarm can also become perpetually split between two similar quality sites. When the operator does not have enough influence, she or he cannot drive the swarm to choose either one.

From these observations and the results shown in Table XII, we draw two conclusions. The first is that, while human operators sometimes realize that too much input is a bad thing for this swarm, they sometimes do not. This is a problem the IVAM could potentially solve. However, in our studies, the IVAM provided participants with too little influence to push the swarm out of an equilibrium in which the robots assessed two sites of similar quality (and hence failed to select either site). This further suggests that the IVAM failed to appropriately balance influence.

F. Qualitative Analysis

Reviewing user behaviors also shows the utility of allowing the operator to place multiple beacons (High influence)

TABLE XII

NON-CONVERGENCE RATES GIVEN VARIATIONS IN INFLUENCE AND INFORMATION FOR TWO SIMULATED OPERATORS (AI 1 AND AI-2) AND OUR HUMAN PARTICIPANTS.

Infl\Info	Perfect	Missing	Mislabeled	Totals
		AI 1		
High	0/30	0/30	10/30	10/90
Low	0/30	0/30	12/30	12/90
IVAM	0/30	3/30	5/30	8/90
		AI 2		
High	10/30	7/30	8/30	25/90
Low	8/30	4/30	5/30	17/90
IVAM	1/30	3/30	0/30	4/90
		Human		
High	4/12	2/12	1/12	7/36
Low	0/12	0/12	1/12	1/36
IVAM	1/12	3/12	2/12	6/36

and the disadvantage of having the swarm make the final choice. Users given High influence were much more liberal with their beacon use, and were more apt to explore the environment and confirm the displayed site information, increasing the likelihood of choosing the best site. However, users seemed to spend most of the simulation convincing the swarm to converge to a site, while our results suggest that giving the human the option to make the final choice would have saved a significant amount of time with minimal change to site choice performance.

VII. CONCLUSIONS AND FUTURE WORK

We have considered the problem of sharing control between a human operator and the inherent swarm behavior in human-swarm systems. In particular, we have argued that an important design parameter of these systems is the degree of influence the human has on the swarm. Our results demonstrate that both too little and too much human influence can result in degraded swarm performance. As such, we have considered how to dynamically alter influence during system deployment.

While early results with simulated operators suggested that IVAM was dynamically balancing influence appropriately, which supported our theory, later results with human operators suggest that IVAM fails to do so. However, the results reinforce the notion that balancing influence between the swarm and the human operator is an important part of human-swarm interaction design, as Low influence produced better performance when information was good, and High when information was poor. Future work should include IVAM redesign and retesting, and determining how to effectively moderate human influence in human-swarm systems.

ACKNOWLEDGMENT

This research was sponsored by the US Army Research Lab under grant DOD ARL W911NF-16-2-0226 and by the Office of Naval Research (ONR), under grant number N000141613025. The views and conclusions contained herein are those of the authors only and should not be interpreted as representing those of ONR, the U.S. Navy or the U.S. Government. A special thanks to Mike Barnes and Eric Holder at ARL for feedback on experimental design and applications of swarm models.

REFERENCES

[1] 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.

[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. Scharre, "Robotics on the battlefield part II," *Center for New American Security*, 2014.

[4] D. R. Olsen Jr and S. B. Wood, "Fan-out: measuring human control of multiple robots," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2004, pp. 231–238.

[5] D. R. Olsen and M. A. Goodrich, "Metrics for evaluating human-robot interactions," in *Proceedings of Performance Metrics for Intelligent Systems*, vol. 2003, 2003, p. 4.

[6] A. F. Winfield and J. Nembrini, "Safety in numbers: fault-tolerance in robot swarms," *International Journal of Modelling, Identification and Control*, vol. 1, no. 1, pp. 30–37, 2006.

[7] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks, "Collective memory and spatial sorting in animal groups," *Journal of Theoretical Biology*, vol. 218, no. 1, pp. 1–11, 2002.

[8] D. S. Brown, M. A. Goodrich, S.-Y. Jung, and S. C. Kerman, "Two invariants of human swarm interaction," *Journal of Human-Robot Interaction*, vol. 5, no. 1, pp. 1–31, 2015.

[9] M. A. Goodrich, S. Kerman, and S.-Y. Jun, "On leadership and influence in human-swarm interaction." in *AAAI Fall Symposium: Human Control of Bioinspired Swarms*, 2012.

[10] A. L. Nevai, K. M. Passino, and P. Srinivasan, "Stability of choice in the honey bee nest-site selection process," *Journal of Theoretical Biology*, vol. 263, no. 1, pp. 93–107, 2010.

[11] D. M. Gordon, *Ant Encounters: Interaction Networks and Colony Behavior*. Princeton University Press, 2010.

[12] M. Haque, E. Baker, C. Ren, D. Kirkpatrick, and J. A. Adams, "Analysis of biologically inspired swarm communication models," in *Advances in Hybridization of Intelligent Methods*. Springer, 2018, pp. 17–38.

[13] D. Karaboga and B. Akay, "A survey: algorithms simulating bee swarm intelligence," *Artificial Intelligence Review*, vol. 31, no. 1-4, p. 61, 2009.

[14] D. Sumpter and S. Pratt, "A modelling framework for understanding social insect foraging," *Behavioral Ecology and Sociobiology*, vol. 53, no. 3, pp. 131–144, 2003.

[15] D. J. Sumpter, *Collective Animal Behavior*. Princeton University Press, 2010.

[16] S. Nagavalli, L. Luo, N. Chakraborty, and K. Sycara, "Neglect benevolence in human control of robotic swarms," in *IEEE Intl. Conf. on Robotics and Automation*. IEEE, 2014, pp. 6047–6053.

[17] J. W. Crandall and M. A. Goodrich, "Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 2. IEEE, 2002, pp. 1290–1295.

[18] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[19] D. S. Brown, S.-Y. Jung, and M. A. Goodrich, "Balancing human and inter-agent influences for shared control of bio-inspired collectives," in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4123–4128.

[20] N. Gilbert, *Agent-based models*. Sage, 2008, no. 153.

[21] *Swarm Intelligence: Proceedings of the 11th International ANTS Conference*. Springer, 2018.

[22] K. M. Passino and T. D. Seeley, "Modeling and analysis of nest-site selection by honeybee swarms: The speed and accuracy trade-off," *Behavioral Ecology & Sociobiology*, vol. 59, no. 3, pp. 427–442, 2006.

[23] T. D. Seeley and S. C. Buhrman, "Nest-site selection in honey bees: How well do swarms implement the "best-of-N" decision rule?" *Behavioral Ecology & Sociobiology*, vol. 49, no. 5, pp. 416–427, 2001.

[24] T. D. Seeley and R. A. Morse, "Nest site selection by the honey bee, apis mellifera," *Insectes Sociaux*, vol. 25, no. 4, pp. 323–337, 1978.