Abstraction and Persistence: Macro-Level Guarantees of Collective Bio-Inspired Teams under Human Supervision

Michael A. Goodrich* and Eric G. Mercer†

Brigham Young University, Provo, UT 84663, USA

It can be difficult for humans to control large numbers (100-200) of robots performing coordinated tasks. Organizational constraints may be imposed that allow a human to issue commands or plays that dictate collective behavior, but naive hierarchical approaches can suffer from robustness issues in cases where key robots or communication channels are compromised. Bio-inspired models can create robust and purposeful collective behavior that is implemented in a completely decentralized organization. Unfortunately, although robust, it may be difficult for a human to influence collectives in a way that the collective is responsive to the human input. We present results from an iterative modeling exercise that systematically relates how individual agents can be influenced by humans to ensure macro-level behaviors of the collective.

I. Introduction

There are advantages to having large numbers (100-200) of robots perform coordinated tasks, but such large teams of robots may be difficult for a human to control. Limitations on communication bandwidth and human attention make it difficult for a single human to gain and maintain situation awareness of each individual robot, and the same limitations also make it difficult for a human to influence individual robots in such a way that the collective behavior of the robots accomplishes a set of reasonable goals. In response to these limitations, organizational constraints may be imposed that allow a human to issue commands or plays that dictate collective behavior, but naive hierarchical approaches can suffer from robustness issues in cases where key robots or communication channels are compromised. In response to these issues, researchers have proposed using bio-inspired models to create robust and purposeful collective behavior that is implemented in a completely decentralized organization. Unfortunately, although robust, it may be difficult for a human to influence these collectives in such a way that the collective is responsive to human input. We present results from a modeling exercise that systematically relates how individual agents can be influenced by humans to ensure macro-level behaviors of the collective.

Motivation: As noted by Lewis,[9] current methods for controlling teams with large numbers or robots (N on the order of 20-200 robots) may grow faster than $O(N)$ because of the complexity of coordinating robot activities. Adding multiple operators can help with some problems, but if complexity grows as $O(> N)$ then the number of required operators will also grow faster than $O(N)$. This is counter to the common goal of maximizing human influence while simultaneously minimizing the number of humans involved

An important way to deal with complexity that arises from inter-agent interactions is to use decentralized control in which robots coordinate behavior in a decentralized fashion. There is an emerging body of research on the design of decentralized controllers and on distributed consensus algorithms that provide performance guarantees on the behavior of such systems (see, for example,[7, 8, 10]).

There has also been some interesting work on having a human control such decentralized teams.[3, 5, 6] This decentralized approach has an important benefit: it can help turn an $O(> N)$ problem into an $O(N)$ or $O(1)$ problem; decentralized control takes care of the required inter-agent coordination, leaving the human to

[*] Professor, Computer Science Department, 3361 TMCB, AIAA Member

[†] Associate Professor, Computer Science Department, 3361 TMCB,

manage the team rather than individuals. A limitation in much of the prior work is that the human takes on a role of centralized supervisor who controls distributed behavior through some sort of broadcast mechanism or who controls a key agent such as a leader with whom all other agents interact. A centralized supervisor can introduce brittleness into the system. If team behavior requires all agents to receive a consistent and centralized signal (from the human or from a leader), then delays or inaccuracies in this signal can cause instabilities or errors.

Proposed Solution: We propose to explore other roles that a human can adopt (a) to keep management complexity within bounds ($O(N)$ or better) and (b) to be robust. The main constraint on these other human roles is that the team must produce guaranteed performance. Two key ideas are required to manage complexity and maintain robustness in the system: abstraction and persistence.

Abstraction means that the behavior of the collective is more than the behavior of the individual members. This principle is well-known in biological systems where a colony or flock exhibits collective behavior that appears purposeful to a human observer in the large scale, even though the behavior of individual members may not appear predictable or sensible. The human must understand the collective at a macro level and focus influence on managing so-called collective states that appear purposeful. Because this collective phenomenon of purposeful behavior is so well-known in bio-inspired systems, we will begin our work by focusing exclusively on such systems.

Persistence means that, since we have replaced centralized control with decentralized human influence, the human must persist in influencing the agents in order to have high confidence that the collective state of the system behaves as needed. Simply put, since the human cannot reliably and quickly control every agent at once, he or she must sustain influence on some subset of agents in a way that behavior emerges as desired. Such influence must be understood at a level sufficient to ensure robustness and authority in the system.

In the next sections, we propose a simple model class that allows decentralized agents to exhibit multiple forms of collective behavior, we then abstract this model class to a Markov Decision Process that allows us to formalize the technical challenges associated with abstraction and persistence, and finally, we present a formal-methods approach for identifying performance guarantees.

II. Stochastic Model of Multi-agent System

We adopt a simplified version of Couzin's model.[2] Let $i \in \{1, \ldots, N\}$ be a set of N homogeneous agents with nonholonomic dynamics given by $\dot{x}_i = s\cos(\theta_i)$, $\dot{y}_i = s\sin(\theta_i)$, and $\dot{\theta}_i = \omega_i$, where $[x_i, y_i]^T \in \Re^2$ is the i^{th} agent's position, $\theta_i \in [-\pi, \pi]$ is the angular heading of the agent, s is the constant agent speed, and ω_i is the angular velocity control input. For simplicity we define: $v_i = [\cos(\theta_i), \sin(\theta_i)]^T$ and $c_i = [x_i, y_i]^T$. Further, let $A(t) = a_{ij}(t)$ denote the sensory adjacency matrix where $a_{ij}(t) = 1$ means that agent j is visible to agent i at time t . Each $a_{ij}(t)$ is determined at time t according to a Bernoulli random variable with parameter $p_{ij}(t) = \begin{cases} \frac{1}{d_{ij}(t)} & \text{if } d_{ij}(t) \geq 1 \\ 1 & \text{otherwise} \end{cases}$, where $d_{ij}(t)$ is the Euclidean distance between agents i and j at time t.

As with Couzin's model, agents react to neighbors within three different zones: repulsion, orientation, and attraction. The neighbors in these zones are determined by

$$n_i^r = \{j : (\| c_i - c_j \| \leq R_r) \wedge a_{ij} = 1\} \quad n_i^o = \{j : (\| c_i - c_j \| \leq R_o) \wedge a_{ij} = 1\} \quad n_i^a = \{j : a_{ij} = 1\}$$

where n_i^r, n_i^o, and n_i^a are the sets of agent i's neighbors in the regions of repulsion, orientation, and attraction, respectively. The parameters R_r and R_o are the associated radii of repulsion and orientation. Note that this model allows for overlapping regions of repulsion, orientation, and attraction. This overlap eliminates the hard switch between the repulsion, orientation, and attraction forces seen in[2] that may be sensitive to sensor transients in real robots.

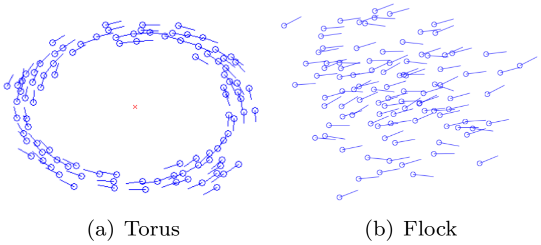
The control input ω_i is determined by first computing the repulsion, orientation, and attraction vectors

$$u_i^r = -\sum_{n_i^r} \frac{c_j - c_i}{\|c_j - c_i\|^2} \quad u_i^o = \frac{v_i + \sum_{n_i^o} v_j}{\|v_i + \sum_{n_i^o} v_j)\|} \quad u_i^a = \frac{\sum_{n_i^a} c_j - c_i}{\|\sum_{n_i^a} (c_j - c_i)\|}.$$

Next, the desired heading vector u_i is computed as $u_i = u_i^r + u_i^o + u_i^a$. Note that, because of the normalization in the equations above, orientation and attraction forces are always equally weighted in the model. This keeps one of these two forces from overpowering the other. It also allows the exponentially growing repulsion vector to overpower the orientation and attraction forces as agents move closer together, which aids in collision avoidance. Finally, angular velocity, ω_i, is computed as $\omega_i = k\alpha_i$ and $\alpha_i = \tan^{-1}(u_i) - \theta_i$, where we limit $\alpha_i \in [-\pi, \pi]$. Since $\pi = \max_{u_i} \tan^{-1}(u_i)$ it follows that $|\omega_i|$ is bounded by $k\pi$.

III. Abstraction to Collective States

In the simplified version of Couzin's model described in the previous section, individual agents are repelled by near agents, are attracted to far agents, and align with intermediate agents. Although the states of individual agents determine collective behavior, it is often useful to view the agents at a macro level where they display a suite of collective behaviors, illustrated in Figure 1, which we call abstract or collective states.


(a) Torus

(b) Flock

Figure 1. Snapshots of various phases from dynamic model: (a) Torus, (b) Flock.

The model in the previous section is capable of producing the flock and the torus behaviors and, importantly, can produce either collective behavior given the same set of model parameters (R_r and R_o, the radii of repulsion and orientation). Since either collective behavior can be produced by the agents, a human can influence the emergent behavior of the collective by manipulating initial states. Even more importantly, a human can cause the collective to switch from one abstract behavior state to another by influencing a subset of the agents.

From the perspective of the abstract collective states, the behavior of individual agents evolves as a stochastic difference equation, making it difficult to characterize or predict the behavior of any individual agent's behavior. Fortunately, both the abstract states of tori and flocks are attractors of the dynamic system that is created by the decentralized control laws from the previous section. As such, the abstract states represent attractors in the dynamic system. Characterizing the stochastic system with abstract states leads to the definition of a critical challenge in human interaction with bio-inspired robot teams: how does one allow the agents to maintain their decentralized implementation while allowing the human to influence individual agents and thereby influence which attractor or abstract state is exhibited by the collective. Before proceeding, it is useful to note that distinct abstract or collective states exist for many systems.[4, 11]

IV. Reduction to MDP Model

We propose that *efficient human interaction with a decentralized robot team must operate at the "macro-level"* encoded by abstract states that are attractors in the underlying dynamic system. Consistent with both bio-inspired and communication-limited systems, we assume that the decentralized implementation means that (a) the number of other agents that influence agent i's behavior is small, and (b) the number of agents influenced by the operator is also small. Furthermore, we assume that the inter-agent topology varies over time, as indeed it does in the model described above (such as when an agent travels in and out of another agent's neighborhood). By equating abstract states with attractors of the dynamic system, the human's task is to influence the collective in a way so that it travels from one abstract state to another using the tools available. A principal challenge in the macro-level abstraction is that the transitions between states behave non-deterministically over an undetermined period of time. Human influence is eventually implemented as commands to individual agents and these agents then influence other agents through the network topology. Thus, the human is *"injecting"* influence into the system, and that influence percolates over time to other agents. As such, transitions between abstract states take a non-deterministic amount of time depending on the percolation rate through the collective. To better understand how to influence a collective through individual agents, we build and analyze a probabilistic model that formally abstracts from individual behaviors to collective states.

In the simplified Couzin model presented earlier, we observe two distinct and controllable abstract states: flock and torus. Each state emerges depending on initial conditions of agents, and parameters can be selected so that each state is equally probable given a broad set of those initial conditions. From this observation, we build a Markov Decision Process (MDP) over these two abstract states, and we build the model in such a way that individual agent states are implicitly, rather than explicitly, modeled. In other words, we no longer directly track, as in the Couzin-based model, the underlying dynamic system governing individual agent behavior, and we instead view agents as either participating in the collective state or not.

Recall that a MDP is given by four components: state, actions, transition probability, and reward. We will discuss rewards in a subsequent section. The remainder of this section incrementally builds the rest of the MDP.

Abstract State: We represent the abstract state by a tuple $s = (\sigma, \delta)$ where $\sigma \in \{\text{flock, torus}\}$ is the collective state and $\delta \in [0, \frac{1}{N}, \ldots, \frac{N-1}{N}, 1]$ represents the fraction of the N total agents whose individual behavior is compatible with the state of the collective; δ represents the fraction of agents participating in the collective state. We choose parameters for the simplified version of Couzin's model such that the flock or torus states are equally likely for random initial agent locations and orientations.

Actions: It is possible to cause the agents to change from one state to another by controlling the number of agents in the collective who are informed about the location of some desired object. This changeover is done by allowing the human to select a subset of agents, we'll call them *stakeholders* to be consistent with prior work,[4] and inform them of some desired objective. A stakeholder is an agent that is influenced by the group and also influenced by its interaction with a human operator. In [3], Couzin showed how a limited number of stakeholders influence the behavior of a flock. In prior work, we have shown that not only can stakeholders allow a human to control the direction of a flock, but that stakeholders can also be used to control a torus and dynamically switch between attractors.

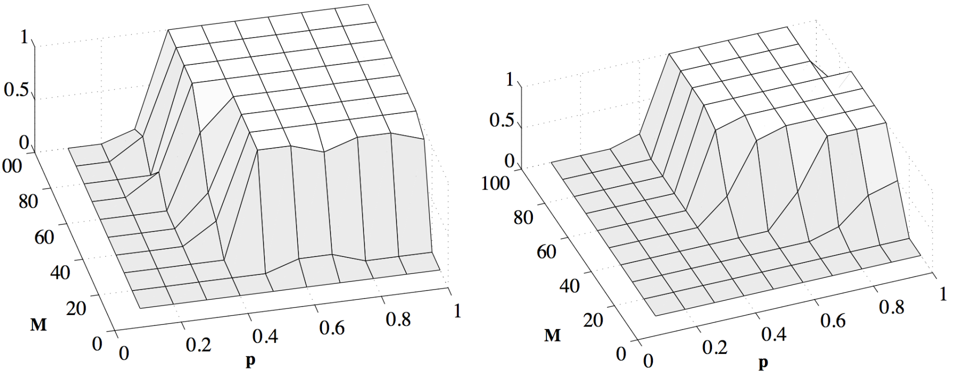
Recall that N represents the number of agents in the collective. We assume that the human operator can broadcast a desired location to $M < N$ stakeholder agents. By broadcasting a reference input to a limited number of agents a human may influence the movement of the collective and even cause the group to switch attractors. A stakeholder has dynamics and control inputs just like every other agent. The difference is that the control input u_i is modified to allow a human operator to influence the stakeholder behavior. This influence is added into the stakeholder attraction dynamics through a weighted sum as $u_i^{sa} = \frac{\rho\hat{q}_i + (1-\rho)u_i^a}{\|\rho\hat{q}_i + (1-\rho)u_i^a\|}$, where the sa superscript indicates the vector of attraction for a stakeholder, $\hat{q} \in \Re^2$ is a the desired location or reference, $\rho \in [0, 1]$ is the priority of the reference input (relative to the importance of the influence of other agents), and u_i^a is determined by the attraction dynamics described earlier. Orientation and repulsion are given as described previously and combine with the stakeholder attractor as $u_i = u_i^{sa} + u_i^o + u_i^r$. This control input enables human influence on the system without eliminating inter-agent dynamics. The input is limited to the attraction dynamics in order to avoid tampering with orientation dynamics and repulsion dynamics, which are necessary for collision avoidance.

Thus, the action available to the human through which collective state can be influenced is given by $a = (M, \rho)$, that is, the human can choose the number of agents who are turned into stakeholders and the relative priority of the desired location for those agents who are stakeholders.

Transition Probabilities: This model of influence allows the operator to manage either the number of agents influenced or the priority of reference input versus inter-agent influence. As a function of ρ and M, prior work has identified the probability, using Matlab simulation, of a swarm switching from torus to flock and from flock to torus; see Figure 2.

In the interest of simplicity, we consider only $\rho = 0.5$ so that the only action available to the human is $a = M$. The human can thus choose how many agents are influenced, and then sustain influence over those agents for a period of time while the influence percolates through the collective and a new desired abstract state emerges. In terms of the MDP, our goal is to model the transition from one abstract state s_t to another s_{t+1}, meaning that we must identify $p(s_{t+1}|s_t, M)$ where $a = M$.

Recall that the abstract state s has two components: the collective state of the agents, σ, and the number of agents whose behavior is consistent with the collective state, δ. For this paper, we assume a deterministic linear model for how δ changes as a function of M. Let RF denote the rate that agent influence propagates from agent to agent yielding $\delta_{t+1} = RF\delta_t + \delta_t$.



(a) Torus2Flock

(b) Flock2Torus

Figure 2. Probability of transitioning from (a) torus to flock and (b) flock to torus, as a function of ρ and M.

This rate of influence is a inaccurate assumption for two reasons. First, the rate of influence is not, in general, linear. It is described more accurately using a linear differential equation over some abstract state x as $\dot{x} = -Lx$ where L is the graph Laplacian associated with the inter-agent graph topology. For such a model, the rate of propagation is better described by a decaying exponential with parameter given by the Fiedler eigenvalue associated with L. The linear model can be thought of as describing the *"steep part"* of the exponential decay. Second, since agents are moving, the agent topology changes over time so influence *"spreads"* unevenly and stochastically among the agents.

Despite the limitation, the linear assumption is useful from a modeling perspective because it allows us to decompose the transition probability into two parts: a deterministic part and a simple stochastic part. To illustrate, consider the following example. Suppose that the collective is in state $\sigma = \text{flock}$ and we want to change the state to $\sigma = \text{torus}$. Since no agents are behaving consistent with torus-like behavior, $\delta = 0$. Thus, the initial state is given by $s = (\text{flock}, 0)$. Now suppose that the human chooses to only communicate with ten agents out of the possible 100 total agents. Then $a = 10$ and we want to know what state will occur after one time step. This is given abstractly by $p(s_{t+1}|s_t, 10) = p(s_{t+1}|(\text{flock}, 0), 10)$. There are two possible next states, $s_{t+1} \in \{(\text{torus}, 0.1 + 0.1RF), (\text{flock}, 0.1 + 0.1RF)\}$ corresponding to (a) a switch from flock to torus with only $0.1(1 + RF)$ of the agents behaving in a "torus-like" way but the rest to follow, and (b) a persistent flock but with $0.1(1 + RF)$ of the agents starting to behave in a "torus-like" way, respectively. We construct the transition probability by specifying values for $p((\text{torus}, 0.1 + 0.1RF)|(\text{flock}, 0), 10)$ and $p((\text{flock}, 0.1 + 0.1RF)|(\text{flock}, 0), 10)$.

More generally, we must specify the transition probability from any state s_t to any state s_{t+1} given any value of M. For the results below, we only consider $M \in \{10, 20, \ldots, 100\}$. We assume that when $M \geq 70$ the transition from flock to torus is certain and for $M < 70$ the probability equals $M/70$. This is loosely based on Figure 2. Similarly, we assume that when $M \geq 30$ the transition from torus to flock is certain and for $M < 30$ the transition probability equals $M/30$. Future work should refine these values using both simulations and by estimating the Fiedler eigenvalue for average graph Laplacians.

Analysis: Using the MDP, we show that the interaction between human and collective is sound and complete. The first step toward performing this analysis is defining the property of *persistence*. Persistence is a quantitative measure of authority in the MDP that represents the level of control exerted over a period of time by a human operator necessary to affect emergent behavior in a collective. Persistence encapsulates both the number of agents directly influenced by the human and a minimum time bound over which the human exerts input to the agents before the desired collective behavior emerges. Intuitively, persistence is a bound, $[\ell_t, u_t]$, on the amount of time a particular human input encoded as span-of-control M must be given to the robots to guarantee that an observed collective state, $\sigma \in \{\text{torus}, \text{flock}\}$, reflects a desired emergent collective behavior. The property is computed in our MDP model using the PRISM stochastic model checker – a tool for formal modeling and analysis of systems that exhibit random or probabilistic behaviors.[1] To measure time, the MDP is augmented with rewards. From the point where the action is resolved, each state is rewarded with the constant 1, representing a single time step in a discrete space, until the system adopts the desired collective behavior in the abstract state. It is now possible to measure persistence, or the expected time to transition to the abstract state, by asking PRISM to find computation paths in the MDP, starting from a given action, that maximize or minimize the reward.

From PRISM, expected time to transition to a flock from a torus in our MDP when for the action $M = 10$ is 2.43 units. The value, of course, changes with the probability distributions in the model. In general, the maximum time to transition from a flock to a torus in the worst case ($M = 10$) is 4.4 time units. The minimum time is 1 which occurs for $M \geq 70$. The maximum time to transition from flock to torus is 2.43 and the minimum time is 1 for $M \geq 30$. The running time to compute these bounds by PRISM is negligible.

V. Conclusion

Bio-inspired systems can be modeled at an abstract level as MDPs that use known collective states and that allow humans to inject influence through span of control. Formal methods can be used to relate the states and actions of the MDP to behaviors of the implementation-level state-space system, guaranteeing that the abstraction behaves correctly. Persistence (temporal locality and span of control) can be used to describe formal guarantees of behavior for the MDP.

Future work should use these formal guarantees as guidelines for designing user interfaces that (a) enforce sequencing that avoids incorrect behavior or (b) inform the operator when a choice is made that moves the system outside of guaranteed operational parameters. Additional future work should generalize these results to other decentralized systems.

Acknowledgments

This work was partially supported by grants from the Office of Naval Research and from the Army Research Collaborative Technology Alliance. Results and opinions are not necessarily those of the funding agencies. We appreciate the work of Sean Kerman and Daniel Brown on developing and analyzing the simplified form of Couzin's model and on generating the figures from the model.

References

[1] PRISM Model Checker. www.prismmodelchecker.org, May 2012.

[2] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and H. R. Franks. Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218(1), September 2002.

[3] X. C. Ding, M. Powers, M. Egerstedt, S. Young, and T. Balch. Executive decision support: Single agent control of multiple uavs. *IEEE Robotics and Automation Magazine*, 2009.

[4] M. A. Goodrich, P.B. Sujit, S. Kerman, B. Pendleton, and J. Pinto. Enabling human interaction with bio-inspired robot teams: Topologies, leaders, predators, and stakeholders. Technical Report BYU-HCMI Technical Report 2011-1, Brigham Young University, 2011.

[5] Z. Kira and M. A. Potter. Exerting human control over decentralized robot swarms. In *Proceedings of International Conference on Autonomous Robots and Agents*, pages 566–571, 2009.

[6] T. McLain and R. W. Beard. Cooperative rendezvous of multiple unmanned air vehicles. In *AIAA Guidance, Navigation and Control Conference*, Denver, Colorado, USA, 2000.

[7] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[8] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents part ii: Dynamic topology. In *Proceedings of the 42nd IEEE Conf. on Decision and Control*, Hawaii, December 2003.

[9] J. Wang and M. Lewis. Assessing coordination overhead in control of robot teams. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 2645–2649, Montreal, Canada, 2007.

[10] R. P. Wiegand, M. A. Potter, D. A. Sofge, and W. M. Spears. A generalized graph-based method for engineering swarm solutions to multiagent problems. In *Lecture Notes in Computer Science*, volume 4193. SpringerLink, 2006.

[11] L. Mier y Teran-Romero, E. Forgoston, and I. B. Schwartz. Noise, bifurcations, and modeling of interacting particle systems. Technical Report Rep US 2011, NIH, 2011.