Studying Direct and Indirect Human Influence on Consensus in Swarms

Saman Amirpour Amraii
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260
samirpour@acm.org

Nilanjan Chakraborty
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
nilanjan@cs.cmu.edu

Michael Lewis
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260
ml@sis.pitt.edu

Abstract

Many cooperative control problems ranging from formation following, to rendezvous to flocking can be expressed as consensus problems. The ability of an operator to influence the development of consensus within a swarm therefore provides a basic test of the quality of human-swarm interaction (HSI). Two plausible approaches are : Direct- dictate a desired value to swarm members or Indirect- control or influence one or more swarm members relying on existing control laws to propagate that influence. Both approaches have been followed by HSI researchers. The Indirect case uses standard consensus methods where the operator exerts influence over a few robots and then the swarm reaches a consensus based on its intrinsic rules. The Direct method corresponds to flooding in which the operator directly sends the intention to a subset of the swarm and the command then propagates through the remainder of the swarm as a privileged message. In this paper we compare these two methods regarding their convergence time and properties in noisy and noiseless conditions with static and dynamic graphs. We have found that average consensus method (indirect control) converges much slower than flooding (direct) method but it has more noise tolerance in comparison with simple flooding algorithms. Also, we have found that the convergence time of the consensus method behaves erratically when the graph's connectivity (Fiedler value) is high.

Introduction

Human control of swarms is a problem of particular difficulty because there is no ready correspondence between human goals, swarm behaviors, and actions an operator might take to influence a swarm. Robots coordinated as swarms rely on simple control laws replicated across platforms which interact with each other to give rise to emergent organized behavior. Flocking behavior, for example, can be generated from three simple rules: 1) move away from any sensed robot closer than d_1, 2) move toward any sensed robot further away than d_2, 3) adjust heading to average heading of sensed robots. The balancing of attractive and repulsive forces and consensus on heading leads to a swarm that sticks together and moves in common, perhaps changing, directions.

Flocking is an example of biomemetic control because the control laws were chosen to mimic the behavior of flocking animals with the presumption that the animals, themselves, use some similar mechanism to coordinate their behavior. Swarm behavior can also be generated from analogs to physical laws by treating robots as point masses and using attractive/repulsive forces and artificial potential fields to produce emergent coordinated behavior. In this case the swarm is referred to as physicomemetic. In either case swarm behavior can be influenced by altering the behavior of some members, altering the control laws, or altering the environment in which the swarm operates.

A basic distinction among approaches to Human-Swarm interaction lies in the ontological status of operator(s) communications in their influence on swarm behavior. One approach, Direct Influence (Kolling, Nunnally, and Lewis 2012), allows operators privileged communications that on receipt, directly alter a swarm members control laws and/or parameters to influence swarm behavior. This approach naturally distinguishes two basic roles of control laws: 1) maintaining swarm coherence and 2) producing behaviors that can be exploited to perform human desired tasks. Laws governing coherence specify the constraints on robot movement needed to maintain connectivity of the swarm. The severity of these constraints can range from a disc which requires that a robot never move to any position that takes it out of communication range of any robot to which it is currently connected to a minimum spanning tree that allows a robot to move to any position that maintains at least one link to the swarm. Human desired tasks are then performed by requesting behaviors subject to these and other constraints. Direct control has the advantage of unambiguously expressing the operators intent to any robot receiving the privileged message thereby accelerating convergence to the operators intent.

An alternate approach, Indirect Influence, limits operators to controlling or influencing a subset of swarm members and uses unaltered control laws to propagate this influence to the remainder of the swarm. In one version of this approach (Goodrich et al. 2011) the operator controls real or virtual swarm members called leaders or predators that stand in a special relationship with other members who (a) are attracted to and follow leaders and (b) are repelled by and flee predators. Operators influence the system by selecting agents to control and designating those agents as leaders/predators to the remainder of the swarm. In this case privileged communication of the operators intention is limited to special members controlled directly and those additional members in direct communication with the special members. In a weaker variant of this scheme operators communicate an influence to some swarm members that contributes but does not fully determine their behavior. These members referred to by (Goodrich 2012) as stakeholders and pacesetters are peers and their communications to other swarm members are not privileged. Goodrich (2012) argues that ceding full authority to a human operator can cause undesirable results such as accelerating members at rates that lead to loss of coherence and break-up of the swarm or highly inefficient state transitions that might be achieved much more smoothly if done in consonance with ongoing behaviors. By acknowledging that a swarm may have better knowledge of its situation than a remote human operator that operator may achieve her goals more effectively by working within the system by injecting control through a small number of agents and allowing the system to adjust to these inputs over time.

Analysis of Swarm Consensus

Our focus is on two distinct approaches to swarm control. In the Direct approach, the operator controls the swarm by sending a command to one or more robots. This command is assumed to be a privileged message and is given precedence over competing influences. For example, if the operator desires that flocking robots move in a new direction, that intention is expressed in a message to the swarm. Upon receipt each robot changes its heading to the desired value. In the Indirect approach the user expresses intent to one or more specialized leaders or predators that attract or repel other robots. This influence is then propagated through the remainder of the swarm through existing control laws. In both of these approaches, the swarm has an intrinsic set of rules that manages the low level functionality of the swarm. In current swarm robotics, this low level often is comprised of connectivity maintenance and obstacle avoidance rules. The operator commands function at a higher level, usually by giving a new direction to the swarm or assigning a new destination. Therefore the operator commands can only be performed in the set of performable operations by swarm (e.g. those operations that does not contradict with connectivity constraints of swarm or does not force robots to move into obstacles).

In either control approach, the operators' main desire is to observe a consensus on her intention in the swarm. For example if she sends a new orientation command to the swarm, she expects to see all the robots gradually change their orientation into her intended value. The reminder of this paper studies the effects of the aforementioned control methods on the consensus quality of the swarm. We analyse the "consensus quality" based on these aspects: How long does it take for the swarm to converge on the intended value/command of the operator? How robust is the convergence algorithm in the presence of noisy communication? What are the effects of graph size and graph connectivity on the convergence time? What happens when robots start to move and the connectivity graph of the swarm changes during time?

For doing our analyses we have used the simplest case in which the operator initiates influence through a single robot. In the Indirect case this influence is exercised in its weakest form through an influenced robot that lacks any special designation to attract or repel other robots that Goodrich (2011) refers to as a stakeholder. The only difference of this robot with others is that it keeps its designated value during time while other robots constantly change their information state based on the information state of their neighbors. On the other hand in the Direct control method, robots can distinguish between normal messages and privileged ones. A message from the operator is considered a privileged one and updates the internal information state of the recipient upon arrival. This privileged message is being adopted by other robots as soon as they get in contact with any of the previous holders of the message.

We can define a swarm as a set of robots occupying spatial positions in a plane during time: $S(t) = \{s_i(t) | s_i(t) \in \mathbb{R}^2\}$. Robots also have an internal state which represents their knowledge at any moment in time. For example their internal state may reflect their destination or an enumerator defining each robots set of internal rules (e.g. for switching between flocking and rendezvous behaviors). In our example this internal state is the robots orientation at each time step: $X(t) = \{x_i(t) | x_i(t) \in (-\pi, \pi)\}$. Our goal is to start from an arbitrary initial state $X(0)$, choose a random orientation value as our intention, x^*, send it to the swarm and wait until the swarm converges on our intention. Convergence is achieved if there exists a time τ, which after that the internal values of all robots remains in an error tolerance range δ^* of x^*:

$$\forall t \geq \tau, \forall i \quad |x_i(t) - x^*| \leq \delta^*$$

Robots communicate with each other through a limited disk connectivity graph. Meaning that robots that are in range κ can see each other and robots farther away would not be able to communicate directly. It results in a connectivity graph $G = (N, E)$ which has N nodes (the number of robots in our swarm) and there is an edge $e_{ij} \in E$ iff $\|s_i - s_j\| \leq \kappa$.

The operator chooses a random robot ϕ as her initial point of influence and updates the internal value of that robot with her desired random value:

$$x_\phi(0) = x^*$$

In the Direct control method, also known as flooding approach, a swarm has an additional internal state set $P(t)$ which indicates which robots have received the privileged information from the operator. Thus after the user sends x^* to ϕ, we would have $p_\phi(0) = 1$ while $\forall i \neq \phi \quad p_i(0) = 0$. The internal states X and P are updated by this rule: $\forall i, j$ if $e_{ij} \in E$ and $p_j(t) = 1$ and $p_i(t) = 0$ then $p_i(t+1) = 1$ and $x_i(t+1) = x_j(t)$. In case a robot has more than one neighbors with privileged information, it adopts the information from the first one.

In the Indirect control method, also known as averaging consensus, the internal state of each robot is updated by averaging over internal values of all of its neighbors (including itself). The only robot that doesn't change its internal value is ϕ (i.e. the robot which receives x^* from operator). The averaging algorithm works based on the method presented in (Xiao and Boyd 2003). Here, we have a weight matrix W which defines the averaging coefficients. Thus at each time t we have:

$$X(t) = W^t X(0)$$

The optimal solution can be expressed as a semidefinite programming problem, which often results in different edge weights (even negative ones) for the edges in the connectivity graph G. As the current swarm robots are usually very limited in computational capabilities, this optimal approach is not plausible for actual experiments with real robots. Instead, we can use a more common approach which assigns a fixed weight to all edges. Xiao (2003) demonstrates that if the swarm remains connected during time, any edge weight smaller than 1 will guarantee a convergence. He also proves that the optimal constant edge weight is $\alpha^* = \frac{2}{\lambda_1(L)+\lambda_{n-1}(L)}$ where $\lambda(.)$ denotes the ith largest eigenvalue of a symmetric matrix and L is the Laplacian matrix of the connectivity graph G. In our experiments, we use this edge weight in our Indirect approach.

We assume that the swarm is operating in an obstacle free environment and always maintains its connectivity (even when robots move). We start our analysis by assuming noiseless communication in a fixed connectivity graph. After comparing Direct and Indirect methods and finding some lower bounds on the worst case convergence time, we expand our analysis by assuming noisy communication and moving robots resulting in dynamic connectivity graphs. As (Wang and Liu 2009; Ren, Beard, and Atkins 2005) have demonstrated, even in the presence of noise, the averaging consensus method will converge as long as swarm's connectivity graph remains connected. Therefore our analysis would focus on its convergence time and its relation to the Direct method's convergence time.

Flooding in Noiseless and Static Communication Graph

Operator sends her intention to one robot and then it propagates from there with a breadth first search (BFS) graph traversing algorithm. As the graph is fixed, it takes at most the size of graph diameter steps (D) until the information from the first robot reaches all the other robots.

Averaging Consensus in Noiseless and Static Communication Graph

Here also the information propagates with a BFS. But instead of sending the complete information to the next layer of robots, only a fraction of information would be sent to them as the robots average the intended information with their own internal states. Therefore at least we need D steps until the information from ϕ can have any effect on the farthest robot. But as the consensus method works by averaging, the actual convergence needs more steps. As a result, in general it always takes more steps for the swarm to reach consensus in Indirect method in comparison with the flooding method.

Flooding in Noiseless and Dynamic Communication Graph

When we have a dynamic graph, the connectivities between robots changes during time, but based on our assumption, the total graph remains connected. In each time step there are two sets: privileged set and uninformed set. At the beginning, the privileged set has only 1 member: ϕ. As the graph is connected, there must be at least one member from the uninformed set that is adjacent to a member of the privileged set. Thus in the next step that robot will receive the information from its informed neighbor. It will then be removed from the uninformed set and be added to the privileged set at the next time step. Consequently, at each time step at least one more robot will receive x^*. Therefore the total convergence time will be at most N, the number of robots in the swarm.

Averaging Consensus in Noiseless and Dynamic Communication Graph

Like the flooding method, in the worst case scenario it takes N steps for the information from the leader to reach all robots, but again as the internal value of robots does not change suddenly to the desired level x^*, we need more steps until the actual convergence happens.

Flooding in Noisy and Static Communication Graph

We assume that our noise is a uniform distribution variable $\epsilon \in (-\delta, \delta)$. When the information $x_i(t)$ is transmitted from each robot to its neighbours, they will receive $x_i(t) + \epsilon$. As the maximum number of hops from ϕ is D, we would have at most $\pm D\delta$ noise added to the value of the farthest node. If we want to keep this value below our maximum error toleration δ^*, we should have $\delta \leq \delta^*/D$. If the noise value exceeds this threshold, convergence guarantee is lost.

The time of convergence is still D as it takes at most D steps for the information from the initial robot ϕ to reach the farthest robots in the swarm.

Averaging Consensus in Noisy and Static Communication Graph

Here, whenever a robot averages over its neighbors and its own value, at the end a uniform error $\epsilon \in (-\delta, \delta)$ is added to its value. As the graph is connected, the swarm will eventually converge on x^*. The only logical assumption that we have to make here is $\delta \leq \delta^*$. Otherwise the local perturbation of the internal state of the robot may exceed the error acceptance level. The important issue about the consensus method is that as robots average their values, most of the time the noise cancels out. This makes this method of controlling more robust in the presence of noise.

The time of convergence is still at least D in the worst case as it takes D steps for the information to have any effect on the farthest nodes in the swarm and then it takes more steps until the swarm converges on operators' intention.

Flooding in Noisy and Dynamic Communication Graph

When the graph is dynamic, the number of hops for a complete information propagation may increase. In the worst case scenario, there is only 1 robot that is added to the privileged set at each time step. In this case, we have N hops until the information is propagated and the swarm reaches consensus. Therefore in order to compensate for noise, we should have $\delta \leq \delta^*/N$. If the noise level gets above this upper bound, we lose our convergence guarantee.

Averaging Consensus in Noisy and Dynamic Communication Graph

As our graph remains connected at each time step, it will reach consensus (Wang and Liu 2009). Like the flooding method, we need at least N steps in the worst case scenario until information reaches all robots and then we have to perform more steps until the swarm converges.

Experiments

In order to test Direct and Indirect control approaches, we have created several random swarm configurations. Then we have analysed their convergence time in comparison to each other. The swarm is created by arbitrarily choosing from 1 to 50 robots and placing them randomly in a 200×200 board. Then we start with a connectivity range $\kappa = 1$ and create the connectivity graph. We will gradually increase κ until all robots form a single connected graph. We also assign some random values as internal state X to robots. These internal state values are selected from $(-\pi, \pi)$ in order to simulate an orientation value. Then we choose a random robot as the leader and communicate our random intention x^* to it. Both Direct and Indirect methods are performed by the swarm and their convergence time is measured. Our error acceptance value is $\delta^* = 0.1$. The results are averaged over 10,000 experiments.

The convergence times are compared based on the number of nodes in the graph and the Fiedler value. Fiedler value is the second smallest eigenvalue of the Laplacian of the connectivity graph (i.e. $\lambda_1(L(G))$) (Gross and Yellen 2003). Fiedler value is also called the algebraic connectivity. A larger Fiedler value means that the graph is more connected and one has to remove more edges in order to cut the graph into independent components (Jamakovic and Uhlig 2007). Therefore Fiedler value gives us a good estimate about the order of connectivity of the communication graph.

The graph diameter distribution of communication graphs used in our experiments is demonstrated in figure 1. As it can be seen, when $E \approx 2N$ graphs are less connected and the graph diameters are higher: $D \approx \frac{N}{2}$. When $E \approx 4N$, graphs are highly connected with $D \approx \frac{N}{10}$

The convergence time for the Direct and Indirect methods are presented in figure 2.

While the Direct method takes at most about 15-18 steps till it converges, Indirect method takes much more time,

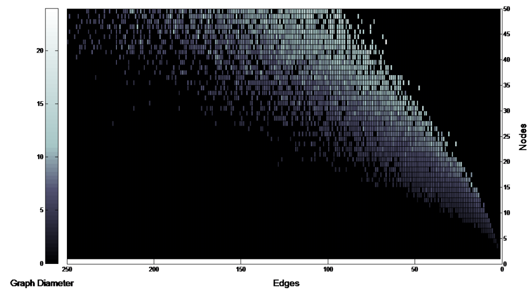
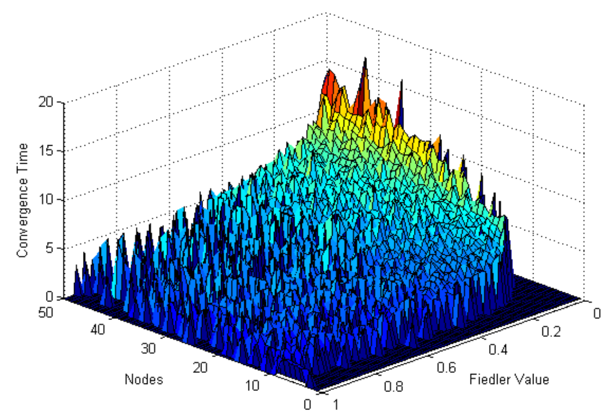
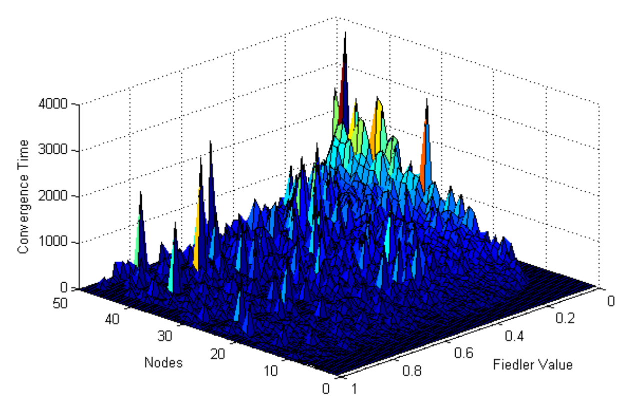


Figure 1: Average graph diameter based on the number of nodes and edges. This random graph distribution has been used in the experiments. This data is based on 10,000 randomly generated swarms. Swarm size is uniformly distributed between 1 to 50 robots. All graphs are connected.



(a) Flooding Convergence Time



(b) Averaging Consensus Convergence Time

Figure 2: Convergence time of (a) flooding (Direct control) vs (b) average consensus (Indirect control).

sometimes even more than 2000 steps. The Direct method is also much more robust. As the Fiedler value increases (the graph is more connected), the direct method converges faster regardless of the number of nodes. Also when the Fielder value is small, the convergence time of Direct method has a linear relationship with the number of nodes in the graph. On the other hand, the Indirect method behaves differently. When the Fiedler value is small, the convergence time increases exponentially with the number of nodes. Also, even when the Fiedler value is high and graph is well connected, sometimes the convergence time spikes. It seems that the Indirect method convergence time does not behave linearly versus the connectivity level of the graph.

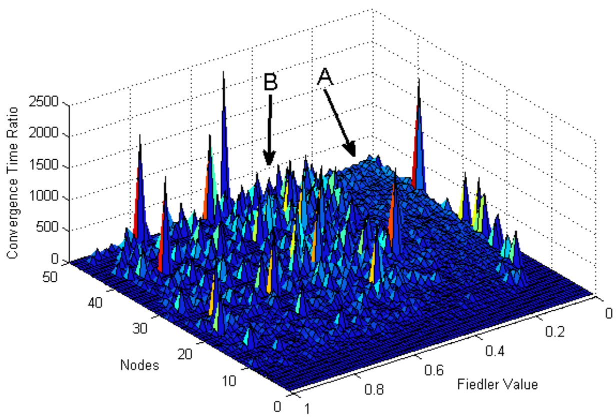


Figure 3: Convergence time of average consensus divided by convergence time of flooding. Region A shows a stable and linear relationship between two control methods while region B shows an erratic and unpredictable behavior in the convergence properties of the indirect control method of swarm.

Figure 3 shows the ratio of the convergence time of the Indirect method divided by the convergence time of the Direct method. There is a region A that is fairly stable: the convergence time of the Indirect method is usually around $2 * N$ times of the convergence time of the Direct method. But there are some unstable regions too. For example when Fiedler value is very small and the graph is loosely connected, the convergence time of the indirect method is considerably higher than the direct method. Also in region B in which the Fiedler value is high and the graph is highly connected, the Indirect method behaves erratically. One reason for this behavior is that in a large graph, the total connectivity of the graph may be high while some small regions of it may be loosely connected to the rest of the graph. If ϕ is selected from these regions, it takes much more time for the swarm to reach a consensus as the information must propagate from a few number of edges and then it has to overcome the initial values of the rest of the robots which are highly connected to each other.

The experiments had the same results for the case with noise. The only important issue about the introduction of noise to the system is regarding the swarms noise tolerance. The Direct method is very sensitive to the amount of noise. The noise must be less than 10% of the error tolerance level or the swarm may not converge. On the other hand, the indirect method is very robust when presented with noise. It can even tolerate noise levels up to the error tolerance threshold.

Discussion

We have seen that the flooding method has much faster convergence time (around 100 times faster than the averaging consensus method). It is also very stable in respect to graph size and as the graph size grows, it behaves linearly based on the connectivity level of the graph (Fiedler value). On the other hand the Indirect method takes more time to reach convergence and it behaves erratically when the graph gets larger, even if it is still well connected. It may be due to the fact that in larger graphs, the leader (ϕ) may be poorly connected while the other robots are well connected. Then it takes much more time for the leader to have influence on other robots.

The simple flooding algorithm performs poorly in the presence of noise while the consensus method has a higher tolerance. It may be better to combine the flooding method with the consensus method and create a hybrid one that also averages between the value of each robot's privileged neighbours in order to compensate for noise and increase the noise tolerance of the swarm algorithm. Hybrid methods of this sort have been used in a number of HSI studies (Kira and Potter 2009; Ding et al. 2009) and determining the most effective ways to combine the advantages of Direct and Indirect influence is a promising research direction.

Acknowledgements

This research has been sponsored in part by AFOSR FA955008-10356 and ONR Grant N0001409-10680.

References

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

Goodrich, M.; Pendleton, B.; Sujit, P.; and Pinto, J. 2011. Toward human interaction with bio-inspired robot teams. In *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2859 –2864.

Goodrich, M. 2012. Human factors issues for interaction with bio-inspired swarms. In *56th Annual Meeting Human Factors and Ergonomics Society (HFES12)*.

Gross, J. L., and Yellen, J., eds. 2003. *Handbook of Graph Theory*. CRC Press, 1 edition.

Jamakovic, A., and Uhlig, S. 2007. On the relationship between the algebraic connectivity and graph's robustness to node and link failures. In *3rd EuroNGI Conference on Next Generation Internet Networks*, 96 –102.

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

Kolling, A.; Nunnally, S.; and Lewis, M. 2012. Towards human control of robot swarms. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, HRI '12, 8996. New York, NY, USA: ACM.

Ren, W.; Beard, R.; and Atkins, E. 2005. A survey of consensus problems in multi-agent coordination. In *American Control Conference, 2005. Proceedings of the 2005*, 1859 – 1864 vol. 3.

Wang, L., and Liu, Z. 2009. Robust consensus of multi-agent systems with noise. *Science in China Series F: Information Sciences* 52(5):824–834.

Xiao, L., and Boyd, S. 2003. Fast linear iterations for distributed averaging. *Systems and Control Letters* 53:6578.