HIJACKING SWARMS

S. Kazadi
Jisan Research Institute
email: skazadi@jisan.org

J. Chang
Jisan Research Institute
email: jihyun@jisan.org

Abstract

Swarms are defined by both their local and global behaviors. Swarm Engineering concerns itself with discovering classes of local behaviors based on local sensing and actuation which guarantee global properties of swarms emerge from any of the class of behaviors. One design requirement of swarm systems is that the swarm be capable of withstanding perturbations, deliberate or accidental, that occur in the behaviors of the individual agents. In the natural world, some swarms exploit the behaviors of other swarms in order to hijack the victim swarms, generally as a predatory behavior. In this paper, examine the cases under which a swarm can be hijacked. We demonstrate that a general set of conditions exist under which a swarm and some subset of its global properties may be hijacked. Swarms not satisfying this set of conditions may be hijacked only by overpowering them.

KEY WORDS
swarm engineering, hijacking

1 Introduction

Swarms, both biological and artificial, are nonlinear complex adaptive systems. As a result, small changes in the design of the elements of the swarm can cause vast changes in the overall group dynamic. This makes the design of swarms difficult because the natural approach to the design of swarms, which involves trial and error, does not easily yield satisfactory progress when complex global behaviors are sought. It is generally impossible to determine what a swarm will do globally when only the local behaviors are known. As a result, the system must be run in order to truly know the global outcome. However, this has its own problems that very often cannot be foreseen.

Among these problems are those associated with designing a swarm made of agents built on a particular, pre-fabricated platform. It is quite possible that a particular desired swarm dynamic cannot be created due to many factors that are out of the engineers' control once the platform has been decided upon. The dynamic might require sensors in places different from those built into the platorm, behaviors that are impossible for the current agents, or processing not possible with the computational structure of the platform. This can make the development of robust swarm-based applications quite slow. Several studies examine methods of avoiding the problem during swarm design including Kazadi et al. [3] which introduces the Hamiltonian Method Swarm Design; Spears et al. [4] which introduced a graph-based method of swarm behavior design; and Winfield et. al. [8,9], which uses temporal logic to verify the emergent behaviours of a swarm system.

Swarms have been tacitly assumed to offer several advantages over individual, more complex platforms carrying out the same tasks. These advantages include robustness and emergence. Robustness allows swarms to be resistant to both expected and unexpected perturbations and gives them the capability to accomplish various tasks in a variety of enviornments. The malfunction of one agent in a swarm will not disturb the rest of the swarm's ability to accomplish a given task. Emergence allows the design of the swarm to create control of characteristics that are not explicitly part of the individual agents' design. This means that

Hijacking is nothing new in the field of science and in nature. Swarms constantly interfere with other swarms in the real world. For example, a swarm of bees perform what is called a "waggle dance" [10,11]. This is a figure-eight dance of the honeybee used to share information among its swarm about the direction to flowers. A swarm of honeybees can be hijacked by sending an agent that conducts the waggle dance, by which the agent can spread misleading information to the rest of the swarm.

Ants are also vulnerable to being hijacked [12,13,14]. They communicate through chemicals called pheromones, which mark trails to and from an ant colony [20]. An agent may interfere with the ant swarm by leaving off misleading trails of pheromone, causing disorder in the organization of an ant colony.

Swarms of barracudas prey on schools of fish by a form of hijacking [15]. These tropical fish can grow up to be 6-feet long and are capable of manipulating schools of fish in groups, controlling the density of a school of fish by surrounding and restricting the area of available space for movement.

This paper discusses the conditions under which swarms can or cannot be hijacked and provide proof of those conditions. This information is beneficial to engineers who create swarms because it offers ways to prevent hijackable swarms. It is critical for engineers to know what it takes to ensure the safety of a swarm. Swarms are constantly attacked, and the information this paper provides will stabilize and protect swarms against those attacks.

The remainder of the paper is organized as follows. Section 2 will provide a rigorous definition of hijackability. In section 3, we will define properties, including measurables, general properties, and feedback. Section 4 will employ these properties and give specific examples and simulation data. Lastly, section 5 will conclude the paper with a discussion of relevant issues.

2 Hijackability

Swarms are designed by first defining a global measurable known as a property (P_s) or properties $\left(\sum_{i=1}^{N_p} P_{s,i}\right)$ [4]. This(These) measurable(s), which form the defining characteristic(s) of the swarm are then manipulated by the behavior of the

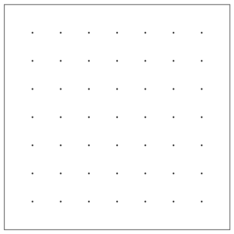

Figure 1. This figure illustrates the phase diagram for a swarm whose behaviors generate no pressure on the property(ies).

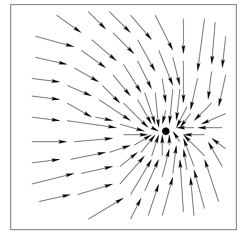

Figure 2. This figure illustrates the desired system design for a swarm system with one final state of the system.

agents. The goal of the swarm engineer is to provide a method for the behaviors of the agents to affect $\frac{dP_S}{dt}$ so that the integral

$$\int_0^{t_f} \frac{dP_s}{dt} dt = P_{f_d} - P_0 \tag{1}$$

for some desired final state P_{f_d} and any initial state P_0. The difficult part of the design is to make sure that the behavior of the agents are such that $sgn\left(P_{f_d} - P\left(t\right)\right) = sgn\left(\frac{dP}{dt}\right)$ and that the integral has the appropriate numerical value.

This section explores the question of robustness from a theoretical point of view. In terms of the global outcome of the system, there are three general classes of systems. The first class is one in which the global property in question is not affected directly or indirectly by the agents in the swarm. This system may be represented by a phase space diagram with zero attractors. The second class is one in which the global property in question always achieves the specific global goal under the action of the agents. This may be represented by a phase space diagram with one attractor. Finally, the third class is one in which the global property in question chooses from between multiple attractors. This may be represented by a phase space diagram with multiple attractors.

2.1 Zero attractors

It is interesting to delineate between the different types of swarms when considered in conjunction with a specific global property. The simplest case, of course, is the case in which the behaviors of the agents receive no information about, and therefore are not affected by, the global property. In such a case, the phase space is a particularly simple and uninteresting diagram, shown in Figure 1.

As the diagram illustrates, there is no pressure in any direction in phase space. As a result, this particular global property may be easily manipulated, as any pressure on the swarm cannot be counteracted by the agent behavior(s). We define this in the following definition.

A swarm whose global property P has a behavior-generated phase space diagram with no attractors is called ***hijackable***. This definition indicates that the swarm can be redirected without any resistance. This means that any perturbation from the outside or internal malfunctions (or even noise!) can affect the global property and that the swarm does nothing to correct the variation. This can happen in the case of properties that the agents ignore. For example, prior to the last century, humanity, viewed as a swarm, affected the global CO_2 levels, but were completely unaware of what they were doing. Were something happening to change the CO_2 level without humanity's knowledge, humanity would not have been able to tell and react.

Swarms of this type are not robust at all with respect to the property in question. This represents an example of a swarm that is not robust[1].

2.2 One attractor

Systems in which the global goal is assured to be achieved have the property that, under the behavior of the agents, the system has a single attractor at the desired state.

The system is in an attractor state when

$$\frac{d\overrightarrow{P_s}}{dt} \simeq \alpha \left(\overrightarrow{P_{s,f}} - \overrightarrow{P_s}\left(t\right)\right) \tag{2}$$

where $\overrightarrow{P_{s,f}}$ is the final state of the property, $\overrightarrow{P_s}\left(t\right)$ is the current state, and α is a proportionality constant.

A swarm is ***well defined*** if the behavior-generated phase diagram has one attractor which attracts the system state from any feasible initial state.

Such a system is generally the goal of swarm research - that is a system whose final state is known and reproducible and results no matter the initial state. There are many examples of such swarms including the artificial physics-based swarms and the puck clustering swarms.

For such swarms, interaction with malfunctioning agents or external agents with differing goals can interfere with the tendency of the system to reach the global attractor. We call such agents ***rogue agents***. We now examine the conditions under which such agents can derail the swarm.

First, suppose that there are N_r rogue agents. The rogue agents each have a behavior $b_{r,i} = \frac{dP_{S,i,r}}{dt}$ where i runs between 1 and N_r. This then means that

$$\frac{dP_S}{dt} = \sum_{i=1}^{N_a} b_i + \sum_{i=1}^{N_r} b_{r,i} = B_S + B_r \tag{3}$$

where B_S represents the system behavior and B_r represents the rogue behavior.

In general, the behavior B_S is associated with a vector in phase space which indicates the direction and magnitude of a change in the property. In the case that the property is a vector,

[1]Generally, it is assumed that the swarm works directly on the property in question, making this specific case extremely unlikely.

then B_S must be a vector as well, as must B_r. We delineate between the two types of convergence conditions.

Even in the case that the new system is well defined, it must be that the new system's attractor is not identical to the old system's attractor. This follows from the following proposition.

Proposition 1: In the presence of the behavior of rogue agents, the initial attractor is still a system attractor iff the behavior of the rogue agents has the property that $\frac{d\overrightarrow{P_r}}{dt}\Big|_{\overrightarrow{a}} = 0$.

Proof:Suppose that the opposite is true. Since an attractor has the property that $\frac{d\overrightarrow{P}}{dt}\Big|_{\overrightarrow{a}} = 0$. This is clearly no longer the case since the $P = P_S + P_r$ and $\frac{d\overrightarrow{P_S}}{dt}\Big|_{\overrightarrow{a}} = 0$.

This is extremely significant because it means that under the behavior of the rogue agents, excepting the case when they create a new attractor that is identical to the original one (as we'll see in the next theorem), the system behavior must change.

Theorem 2: (Strong Convergence Condition) Let $\overrightarrow{a}$ be the attractor in phase space under the original system behavior. If the new rogue agents have a new attractor under their behavior alone, say $\overrightarrow{x}$, then the overall system outcome will remain constant iff $\overrightarrow{x} = \overrightarrow{a}$.

Proof: This is the only condition under which $\frac{d\overrightarrow{P}}{dt}\Big|_{\overrightarrow{a}} = 0$ and a single convergence point exists. Of course, in the case that this condition is met, one can't really call the rogue agents rogue. They really just do the same thing, only a bit differently.

Under the behavior of the rogue agents, the system may form a new attractor, set of attractors, or a quasi-stable behavior in which a small subset of the overall phase space is visited continually by the system but does not converge to a single state.

Theorem 3: (Weaker Convergence Condition) Let $\overrightarrow{a}$ be the attractor in phase space under the original system behavior. If the new rogue agents have a new attractor under their behavior alone, say $\overrightarrow{x}$, then the new system will contain at least one attractor or attractor region.

Proof: In order for the two systems to have a single attractor, the behavior must be such that the system approaches these points from any other point. This means that there exists a neighborhood of the points such that all pathways enter the neighborhood and no pathways exit. Let A be the union of these two neighborhoods and let B be the minimal ε- ball containing both neighborhoods. Then, B is the region we are looking for. This region may have any number of attractors, but the region itself may be considered an attractor.

This means that if the rogue agents have their own specific goal, the mixture of the two sets of agents will produce a third swarm with at least one attractor or attractor region. This indicates that the system will still converge to a well defined or ensemble system that resembles one or the other or some mixture of the two.

2.3 Multiple attractors

Consider the case that the swarm's behavioral description in phase space has multiple attractors. Let the set of attractors be represented by $\{a_i\}_{i=1}^{N_a}$ and the regions which lead to these attractors be represented by $\{A_i\}_{i=1}^{N_a}$. In particular, a single attractor is a special case where $N_a = 1$. In this case, the swarm's outcome is known only when the initial state is known and restricted to one region A_i. In general, such a swarm would not be considered robust, as it's final outcome is not well defined.

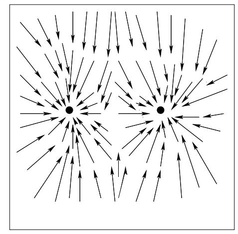

Figure 3. This figure illustrates a system design in which multiple final states occur.

All of the effects described above might happen in each of these regions. A complete examination of these effects is beyond the scope of this paper.

3 Examining the effects of rogue agents

In this section, we examine two different swarms and the effects that rogue agents have on these swarms. One of the swarms is hijackable while the other is not. We shall see that in the case of the hijackable swarm, usurping control over the swarm is quite easy. On the other hand, we shall also see that in a non-hijackable swarm, one must exert considerable pressure in the form of rogue agents in order to alter the global behavior.

3.1 A hijackable swarm

Our starting place is the description of a swarm of agents that is hijackable. Kazadi et. al. [1] describes a swarm whose behaviors are based on artificial physics or physicomemetics Spears et. al. 1991, 2006. These swarms are made up of agents which are aware of one another's positions via some kind of positional sensor. Each agent is pointlike in the sense that it has a position, but its size is somewhat unimportant. Each agent uses an analog of a physical system to determine how it will respond to the totality of the other agents. For instance, if the "swarm" consists of only two agents, each agent will be able to determine where the other agent is. From this data, each agent will determine what it's behavior will be according to a physical law. As an example, if that physical law is based on the law of gravitational attraction, then this means that the agents will each determine the acceleration vector accordingly:

$$\overrightarrow{a_2} = \frac{Gm_1}{r^2}\hat{r_{12}} \tag{4}$$

and

$$\overrightarrow{a_1} = \frac{Gm_2}{r^2}\hat{r_{21}}. \tag{5}$$

Each agent then attempts to follow this acceleration. In the event of multiple agents, each agent performs a similar calculation for all other agents and sums the result. Thus, the agent's microscopic behaviors create the overall system's behavior according to microscopic attention to physical laws.

Many studies examine the determination of methods of generating desired global behaviors using artificial physics [5,6,7,19,21]. In general, the correct application of local behaviors can generate a global lattice with specific properties. Among

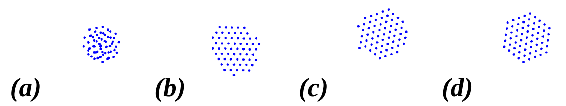

Figure 4. This set of images illustrates the basic behavior of the swarm of agents; the swarm anneals from a random initial dispersion to a global hexagonal dispersion. However, the swarm "wanders about" while it anneals due to random relative motion.

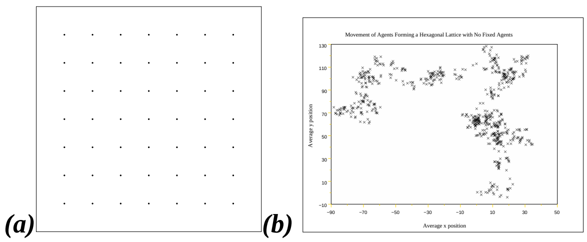

Figure 5. This figure illustrates the phase space of the hijackable system (a). Without an attractor, the average position of the system wanders randomly around phase space (b).

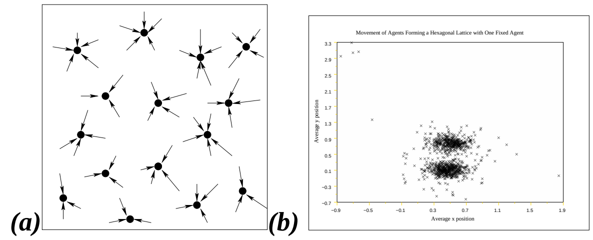

Figure 6. This figure illustrates the phase space of the hijackable system with a single rogue fixed agent (a). The average position of the system, when implemented, remains relatively static in phase space. (b)

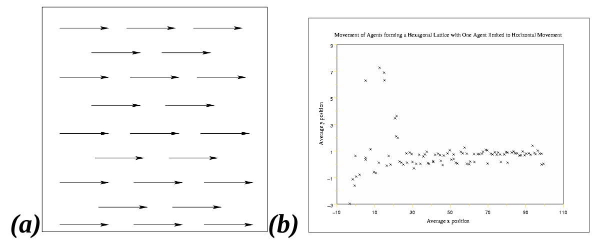

Figure 7. This figure illustrates the phase space of the hijackable system with a single rogue moving agent (a). The system, when implemented, moves in one direction in phase space (b).

these properties are the correct dispersion of agents and the correct arrangement of the global group. One such study, Kazadi et al. [1] describes a simple behavior that anneals the overall group structure into a globally desired hexagonal arrangement. The situation is depicted in the set of images given in Figure 4.

Note that the average position of the agents is

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N}\overrightarrow{x_i}. \tag{6}$$

This position is the global property of average position. It is a value which has a behavior given by

$$\frac{d\overline{X}}{dt} = \frac{1}{N}\sum_{i=1}^{N}\frac{d\overrightarrow{x}}{dt}. \tag{7}$$

In the original swarm, the quantity $\frac{d\overrightarrow{x}}{dt}$ exists, but no conditions are imposed on its behavior. As a result, if the quantity $\overline{X}$ changes, there is no restoring force to change it back to a desired quantity.

We can view using the phase diagrams discussed above. In this case, we can describe $\overline{X}$ as the average two-dimensional position $(\overline{x}, \overline{y})$. The phase diagram consists of the attractors and directional indicators at each point that the behaviors of the agents will make the system move in phase space. In this case, it is clear to see that the system is equally likely to go in any direction, and so the phase space has no attractor or directional component. As a result, we expect the system to have a random walk in phase and real space, which is, in practice, what occurs. The situation is depicted in Figure 5.

As we've seen above, the overall phase space diagram when including rogue agents is the sum of the phase space contributions of the agents and of the rogue agents. In our case, there is no contribution of to the overall phase space by the agents, and so the control of the measurable is in the hands of the swarm. For instance, suppose that the phase space consists of points in which the directional indicators are pointing towards the points in the phase space; such a space has an infinite number of attractors. Then the swarm will be anchored to the space and will tend not to wander (outside of a rearrangement of the hexagonal grid). The situation is depicted in Figure 6 along with a graph of the average position of the swarm. In this simulation, one agent is rogue, and it is immobile. As we can see, this is all it takes to immobilize the entire swarm; this swarm is hijackable.

A similar swarm is depicted in Figure 7. In this case, the rogue agent involved has a movement in one direction. The phase diagram of the combined state is one in which all directional arrows point in one direction. Such a system has no attractors, and therefore the overall system cannot converge. When this rogue agent is added to the system, the system continually moves in the indicated direction. No convergence occurs in the overall system state, as is indicated by the system's average position.

It is interesting to note that although the system is hijackable when considering the average position property, this does not affect the overall configuration properties. The overall configuration of the system remains consistent, forming a hexagonal global shape despite the rogue agents. This is not always so, but for these rogue agent additions, it is because the one property does not significantly affect the original swarm property. As positions involved in the initial property are relative, the average position can easily change without affecting the other properties as long as the agents are able to react to the change in the rogue agent's behavior more quickly than the rogue agent can move

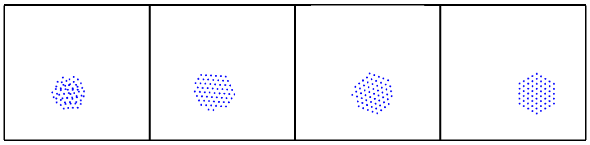

Figure 8. Despite being pulled to the right by the rogue agents, the swarm behavior hasn't changed; the swarm still forms a hexagon.

away from the swarm. This is demonstrated in Figure 8 for the case when the rogue agent moves monotonically to the right.

The main point with this swarm is that it is totally at the whim of rogue agents with respect to this property. As a result, the swarm can be made to do just about anything the rogue agent wants it to with minimal effort by the rogue agent. Moreover, using the phase diagrams, it is clearly understandable that the global outcome will be what it is.

3.2 A non-hijackable swarm

In [2], a swarm is described which consists of two different types of agents. One group of agents is known as the consumers while the second is known as the vendors. Each of the agents is modelled as a greedy agent, trying to earn maximal profits (in the case of the vendors) and maximal consumption given limited resources (in the case of the consumers). In the simulation, the agents iteratively interact with one another, with interactions initiated by the consumers. Each cycle, each consumer randomly chooses a vendor and decides whether or not to buy what the vendor is selling. If certain conditions are met, the consumers buy the vendor's product. At the end of the cycle, the vendors decide whether or not to raise their prices based on their cumulative interactions with the consumers and the comparison of that interaction with previous cycles.

The study focuses on generating global conditions under which prices of the commodities are held constant under the interactions of the agents and their greedy behaviors. The study concludes that the generation of stable non-inflating prices can occur if the demand for the commodities is tied to the prices of the commodities in such a way that as the profit level of the vendors increases, the demand decreases. The study generates a theoretical description of the system which indicates the conditions under which a swarm can stabilize prices. The condition under which a swarm can be expected to stabilize prices can be described by

$$-\sum_j \frac{dD_j}{dt} \geq$$

$$\sum_j \frac{1}{\frac{\partial f}{\partial D_j}} \left(\frac{\partial f}{\partial b} \left(D_j \left(t \right) \left(f \left(t \right) - c_c \left(t \right) \right) \right) + \frac{\partial f}{\partial c_c} \frac{dc_c}{dt} + \frac{\partial f}{\partial i} \frac{di}{dt} \right) \tag{8}$$

where D represents the demand for the commodity, f represents the price of the commodity, b represents the profit of the vendor, c_c represents the commodity cost, i represents the vendor's income, and the sum is taken over the set of consumers in the system. As long as the demand satisfies this equation, the price will be stabilized.

Rogue agents, then, can be defined as agents for which this inequality does not hold. Let us examine the situation.

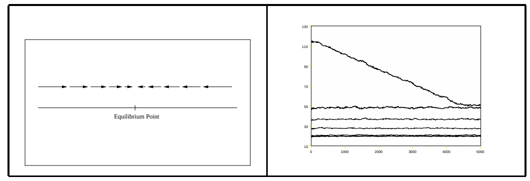

Figure 9. The baseline behavior of this swarm is to stabilize the prices (left). Each price has a single equilibrium point. When the price is out of equilibrium, the system returns it to its equilibrium value (right).

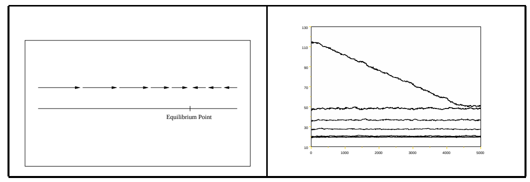

Figure 10. This figure illustrates the movement of the equilibrium point as a result of one rogue agent's influence (left). The equilibrium values of the prices are likewise changed with respect to the previous graph.

Firstly, the phase space diagram is a linear graph, since only one property is involved here. Our phase diagram shows, at each value for D the direction and magnitude that the change in D has. Clearly, it is only possible to have a single value for the price of the commodity settled upon by the system, with respect to the cost of the commodity, if there is only one attractor at the desired price. We also assume that each of the values of the various partial derivatives is bounded and that $\frac{dc_c}{dt} = 0$. Then in this case, the phase diagram is as given in Figure 9. Importantly, this has a driving force toward the equilibrium point from both directions. Moreover, the magnitude of the push is bounded, an important consideration, as we shall see.

Now, if we add a rogue agent to the system, say, one in which the change in demand is always zero, then the rogue agent adds nothing to the left hand side of the equation, but does add to the right hand side of the equation. This is equivalent to adding a right-pointing arrow everywhere in the phase diagram. Adding right pointing arrows everywhere to the phase diagram shifts the equilibrium point to the right - i.e. higher prices. If the amount that is added is sufficiently small, it will not overwhelm the swarm's convergence entirely. The situation is depicted in Figure 10.

We can determine the *capacity of the swarm* in two steps. First, we can ask what the sum on either side of the equation is. The difference between the left and the right hand side determines the amount by which the rogue agents must change the system behavior before the overall convergence behavior changes to non-convergence. As a result, the capacity of the swarm may be determined by the magnitude of this difference. In terms of the number of agents, once the effect of a single agent is determined, the capacity in terms of the number of agents can be determined. In graphical terms, as soon as the arrows in the phase diagram from the rogue agent(s) are longer than the greatest ones of the swarm, the overall system characteristic changes from bounded to unbounded. The effect of various values of the

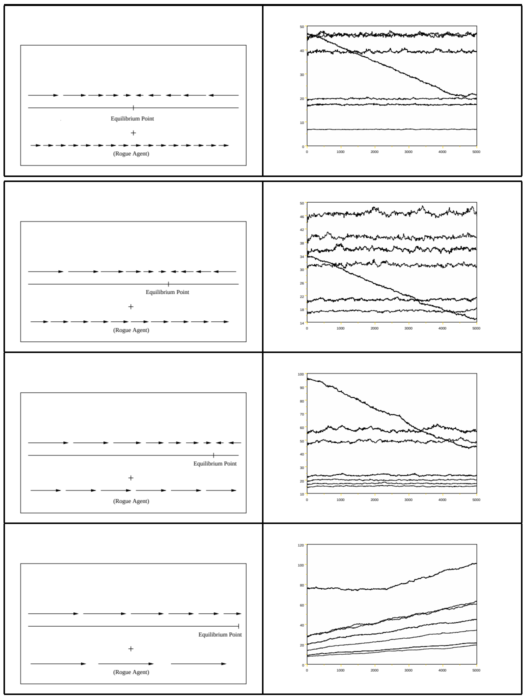

Figure 11. This figure illustrates the incremental changes in system behavior in both phase space (left) and actual price behaviors (right) as a result of an increasing number of rogue agents. The point at which the effect overwhelms the stabilization effect is the *capacity* of the swarm.

rogue agent effect is depicted in Figure 11.

We can also see easily that the effect of differing values for the effect of the rogue agent has an immediate effect on the convergence point of the system. The phase diagram makes the difference clear, as the combination of the two effects produces a convergence point at different points depending on the magnitude of the rogue agent's(s') behavior(s). Clearly, as the rogue agent's behavior becomes greater, the convergence point moves to the right. This is equivalent to the equilibrium price increasing. Both of these effects are clearly evident in Figure 3.8.

It is clear that such a swarm has a very different dynamic interaction with rogue agents than hijackable swarms. The non-hijackable swarms tend to have a gentle degradation of their behavior which may be followed by a complete loss of control of the property in question. This can be easily determined by the phase diagram by examining the magnitude of the attraction around the attractor. Attractors with relatively large slopes are very stable while those with relatively small slopes are somewhat unstable. Moreover, unlike the hijackable swarm above, they switch between two different behaviors. One behavior mimics the actual desired swarm property, and the other is completely outside of the desired range.

4 Discussion and Conclusions

In general, the idea of a robust swarm is an important one in any swarm engineering endeavor. Until now, there has been little in the way of a theoretical methodology to determine, in general, what a robust swarm is. In much of the swarm literature [8,9,16,17,18,22] the idea of robustness centers around the idea that when a single agent or multiple agents are inactivated, the swarm can still achieve its goal. This specific requirement is the most common defining concept in the idea of robustness of swarms. As we have seen, a swarm will achieve a global task from any initial set of conditions if the global task is the attractor in state space for the swarm system. The question of whether or not a swarm will achieve this goal when utilizing fewer agents equates to generating a phase space graph for swarms with progressively fewer agents. Once these phase space graphs have been created, one can easily determine whether or not the smaller swarm will succeed in achieving the task. More robust swarms will be able to do so with fewer agents, and their phase space diagrams will continue to exhibit a single attractor despite the small number of agents.

However, these considerations do not achieve a second kind of robustness measure which is also important when deploying swarms. In general, considerations about robustness in swarms to not consider the swarm's *robustness to attack*. Such a consideration is not a trivial matter, as it involves not only the swarm's ability to keep on completing its work despite a malfunctioning agent, but also its ability to respond to deliberate attacks which are meant to alter its behaviors. We have shown that not only can swarms have a capacity to absorb interference, but that swarms tend to have a finite *capacity* beyond which the attack on the swarm (or malfunction) will overwhelm the swarm in generating the overall behavior. On the other hand, the malfunction or attack might alter the system in such a way that it forms a stable set of orbits around the final goal or near the prior final goal. This could happen, for instance, in a puck clustering swarm if one agent *declustered* the pucks, thereby generating a stable state that approached but never achieved complete clustering.

Understanding how a swarm will behave under these types of malfunctions or attacks is critical to deploying a system that behaves as desired, even when faced with a set of unlikely circumstances. Since, in general, one cannot know how a system might behave in an unpredictable environment where agents can malfunction in unexpected ways, a careful examination of the phase space diagrams of the swarm under the behaviors of the various agents is crucial. Rogue behaviors which overwhelm the swarm in the magnitude of the movement in phase space will change or erase the attractor. Rogue behaviors that do not overwhelm the swarm in magnitude may cause quasi-steady states which do not converge to the desired global goal or simply change the final attractor so that the behavior is different from the desired behavior. As we have already shown that such a movement is inevitable unless the rogue behaviors have the same attractor, so understanding how much change is likely in the event of a malfunction or attack is important.

We have also demonstrated at least once condition under which a swarm is *never* robust - that is when the swarm is hijackable, or when its phase space diagram has no attractors. In such a case, even a minute change in the system by a malfunctioning or rogue agent can take over the swarm. Such a case must be strictly avoided as it means that something as unimportant as a light breeze pushing a flying swarm can control its group behavior.

Finally, it is clear that one must be careful to discern the difference between a hijackable swarm *with respect to one prop-*

erty and a wholly hijackable swarm. As we have seen, some swarms are hijackable with respect to a specific property and not with respect to others. This is an important distinction and criterion in designing a swarm. For instance, if your swarm was going to form a hexagonal matrix to allow better radio transmission and reception, it might be important to make sure that the swarm is not hijackable with respect to overall position. Consideration of what might initially seem to be peripheral properties is important when building a swarm of agents in order to make sure that interactions with such properties do not inactivate the overall swarm goal.

References

[1] S. Kazadi, J. R. Lee, J. Lee. Artificial Physics, Swarm Engineering, and the Hamiltonian Method World Congress on Engineering and Computer Science 2007 , San Francisco, California, USA, October 24-26, 2007.

[2] S. Kazadi, P. Kim, J. S. Lee, J. Lee. Swarm Economics World Congress on Engineering and Computer Science 2007 , San Francisco, California, USA, October 24-26, 2007.

[3] S. Kazadi. The Genesis of Swarm Engineering. Proceedings of the SCI2003 Conference, Special Session on Swarm Engineering, Orlando, Florida, USA, 2003.

[4] S. Kazadi. On the Development of a Swarm Engineering Methodology. Proceedings of IEEE Conference on Systems, Man, and Cybernetics, Waikoloa, Hawaii, USA, pp.1423-1428, October 2005.

[5] W. Spears, D. Gordon. 1999. Using artificial physics to control agents. In IEEE International Conference on Information, Intelli- gence, and Systems, Washington, DC, pp. 281■288.

[6] W. Spears, D. Spears, D. Zarzhitsky (2005, invited). Physicomimitics positioning methodology for distributed, autonomous swarms. GOMACTech-05 Intelligent Technologies

[7] P. Wiegand, M. Potter, D. Sofge, and W. Spears, A Generalized Graph-Based Method for Engineering Swarm Solutions to Multiagent Problems. Proceedings of PPSN 2006

[8] A. Winfield, J. Sa, M.C. Gago, C. Dixon, M. Fisher. On Formal Specification of Emergent Behaviours in Swarm Robotics Systems. Int. J. Advanced Robotic Systems, 2(4),363-370, 2005.

[9] A. Winfield, F.T. Alan, J. Christopher Harper, and Julien Nembrini. Towards Dependable Swarms and a New Discipline of Swarm Engineering. UK: Springer Berlin / Heidelberg, 2005. 3 May 2008.

[10] D.W. Stokes. 1983. A Guide to Observing Insect Lives. Little, Brown and Co., Boston.

[11] J.L. Gould, C.G. Gould. 1988. The Honey Bee. Scientific American Library, W.H. Freeman, New York.

[12] M. Dorigo, V. Maniezzo, A. Colorni. Ant system: optimization by a colony of cooperating agents. Bruxelles: IEEE Systems, Man, and Cybernetics Society, 1996. IEEE Xplore. Feb. 1996. 3 May 2008.

[13] J. Klotz, D. Williams, B. Reid, K. Vail, P. Koehler. Ant Trails: A Key to Management with Baits. University of Florida.

[14] R.W. Matthews, J.R. Matthews. 1978. Insect Behavior. John Wiley and Sons, New York.

[15] Barracuda- Predatory Behavior, The Great Barracuda, The Pacific Barracuda, Human Fear of Barracudas. Science Encyclopedia Volume I. Net Industries. 2008.

[16] P.C. Butler, R.P. Kelley, B.E. Bishop. Control of Heterogeneous Combat Swarms. Dept. of Weapons & Syst. Eng., United States Naval Acad., Annapolis, MD;

[17] S.M. Chen, H.J Fang. Modelling and Stability Analysis of Emergent Behavior of Scalable Swarm System. Jan. 16, 2006.

[18] S.M Chen, H.J. Fang. 2005. Modeling and stability analysis of large-scale swarm. Control and Decision, 20:490-494.

[19] L. Barnes, W. Alvis, M.A. Fields, K. Valavanis, W. Moreno, "Swarm Formation Control with Potential Fields Formed by Bivariate Normal Functions," med, pp. 1-7, 2006 14th Mediterranean Conference on Control and Automation, 2006.

[20] W. Shen, Pe. Will, A. Galstyan, and C. Chuong. Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms. Autonomous Robots, 17(1):93-105, July 2004.

[21] Chu, T., Wang, L., Chen, T., 2003. Self-organized motion in anisotropic swarms. Journal of Control Theory and Applications, 1:77-81

[22] Gazi, V., Passino, K.M., 2003. Stability analysis of swarms. IEEE Transactions on Automatic Control, 48(4):692-697.