

On the impact of variation on self-organizing systems

Adam Campbell, Cortney Riggs, and Annie S. Wu
Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816
{acampbel,aswu}@cs.ucf.edu, cortr@hotmail.com

Abstract—The goal of this work is to gain a better understanding of the role that inter-agent variation plays in self-organizing systems. We develop both continuous and discrete models of a multi-agent coordination procedure based on response thresholds and use these models to analyze the average behavior of a system as well as examine the dynamics of single instances of the problem. Results indicate that variation in the behaviors of agents can lead to increased stability and more effective self-organization in cooperative multi-agent systems.

Keywords—self-organization; variation; multi-agent systems; emergent coordination

I. INTRODUCTION

When working with computational and robotic agents, we often assume by default that error is bad and that variation and noise in anticipated behaviors and outcomes should be minimized or eliminated. Yet there are related disciplines, namely the study of complex adaptive systems, in which it has long been established that error or variation is not only beneficial but essential to how some systems work [1], [2]. As the applications of self-organizing multi-agent systems become more complex, we believe that variation will play an increasingly important role in these systems. The goal of this work is to gain a better understanding of the role that variation plays in self-organizing multi-agent systems.

Specifically, we investigate the effects of inter-agent variation on the ability of a multi-agent team to solve a distributed task allocation problem. The problem we investigate involves a single task, and the team of agents solving this problem must self-organize such that an appropriate number of them are working and not working on the task at any given time. Both the biological [3], [4], [5] and multi-agent [6], [7], [8], [9], [10] literatures provide examples of systems where behavioral differences among the individuals of a team help the team as a whole reach its objective. We hypothesize that inter-agent variation is not only beneficial but also essential to a distributed team's ability to self-organize.

©2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

The importance of variation in self-organizing systems may be better explained through the use of an example. Honeybees maintain their nest temperatures within a desired range to optimize brood development. If external forces such as ambient temperature cause the nest temperature to fall below the desired range, honeybees will shiver to generate heat to warm up the nest [4]. The number of bees allocated to the task of shivering determines the rate of heating within the nest. In order to maintain a desirable nest temperature, a colony must be able to maintain a “correct” number of bees acting (shivering) at any given time as dictated by the difference between the current and desired nest temperatures at that time. Jones *et al.* [4] observe that there are differences in the temperatures at which individual bees begin to shiver and that these differences help to prevent all of the bees from acting simultaneously, thus avoiding large fluctuations in nest temperature. As a result, the nest as a whole is able to make more gradual changes in the number of bees that are and are not shivering, resulting in smoother adjustments in nest temperature. Variation in individual bees leads to a more effective and stable division of labor for the colony as a whole.

Biology provides multiple examples of natural systems in which variation in agent behaviors help distributed teams self organize and complete their objectives. The sources of variation in these systems may be genetic [4], [11], experiential [5], morphological [12], [13], [14], [15], social [16], [17], [18], or age-related [15]. Computational simulations of natural systems support the hypothesis that variation among individuals leads to more stable and effective teams [19], [20]. Engineered multi-agent systems have also effectively used variation to generate appropriate divisions of labor in distributed teams. Variation may be manually inserted and static throughout a run [7], [8] or dynamically evolved over the course of a run [9], [10], [15]. In all of these examples, variation in the behaviors of individual agents increases the number of possible sets of agent actions that a team as a whole can offer, thus increasing a team's ability to adapt to varying stimuli [1].

This work focuses on investigating the effects of inter-agent variation on a multi-agent task allocation problem. In particular, we examine how differences in the thresholds

that determine when each agent starts to work help to promote a self-organized division of labor throughout the system. We first analyze the behavior of systems without any inter-agent variation, and then use a dynamical systems approach to predict the behavior of systems where inter-agent variation is present. This analysis is then used as a way to program agents so that they work together to solve the given multi-agent task allocation problem. Finally, we discuss the importance of inter-agent variation and self-organization based on a definition of self-organization given by cyberneticist William Ross Ashby in 1947 [21]. When taken together, these experiments and analysis support the hypothesis that inter-agent variation can help cooperative multi-agent systems to self-organize more effectively.

II. THE PROBLEM

The problem we investigate models situations where a team of agents must work together to maintain the level of a constantly changing resource or system variable, herein referred to as “the task.” We assume that there is a single task with a defined target value and an observable current value. External forces can act on and change the current value of the task. If the task’s current value drops below the target value, agents can act on the task to raise its current value. The number of agents that act on a task determines the rate at which the current value increases. The goal of the team, then, is to maintain an appropriate number of agents acting on the task at any given time to counteract the changes that move the current value of the task away from the target value. We assume that agents are not able to communicate with each other, and thus, must determine when to work based only on their perceptions of the task’s current value. This presents a challenge to the agents as they must decide independently whether or not to act on the task at any given time. If too many agents act at once, they can overshoot the target; if too few act, then they may never reach the target.

Variation is added into this problem in the form of a limited amount of error in the each agent’s ability to correctly sense the current value of the task. This error affects when an agent perceives that the current value has fallen below the target value which determines when an agent will work on the task. Without variation, all agents behave in the exact same way and only one of two events can occur: either all of the agents act on the task, or none of the agents act. With variation, agents react over a wider range of current values, thus increasing the number of possible system-level responses to the task’s current value. Variation among the individual agents is expected to allow the team as a whole to temper their responses to the current task value, and thus, allow the system to behave in a more stable manner.

Algorithm 1 shows the simulation used here to investigate the effects of inter-agent variation on the aforementioned problem. The simulation consists of a task t and a set of

Algorithm 1 *Simulation(A, t, maxTime)*

```

1: for  $i := 1..n$  do
2:    $a_i.error := U(t.maxE)$ 
3: end for
4:  $timestep := 1$ 
5:  $totalDiff := 0$ 
6: for  $timestep := 1..maxTime$  do
7:    $t.current := t.current - t.\delta$ 
8:   for  $i := 1..n$  do
9:     if  $t.current + a_i.error < t.target$  then
10:       $a_i.needsToWork := true$ 
11:     else
12:       $a_i.needsToWork := false$ 
13:     end if
14:   end for
15:   for all  $i := 1..n$  do
16:     if  $a_i.needsToWork$  then
17:        $t.current := t.current + t.contribution$ 
18:     end if
19:   end for
20:    $totalDiff := totalDiff + abs(t.current - t.target)$ 
21: end for
22: return  $totalDiff$ 

```

n agents, $A = \{a_1, a_2, \dots, a_n\}$. Associated with the task are a target value $-\infty < t.target < \infty$, a current value $-\infty < t.current < \infty$, a rate of change $t.\delta > 0$, a contribution value $t.contribution > 0$, and a maximum error value $t.maxE \geq 0$. Associated with each agent a_i is an error value $-t.maxE \leq a_i.error < t.maxE$. We examine two distributions of error values between $\pm t.maxE$: uniform and Gaussian. The Gaussian distribution assumes that $t.maxE$ is equivalent to three standard deviations of error denoted, $t.stdE$.

The simulation begins by setting the error value of each agent (Lines 1-3). The function $U(t.maxE)$ returns a random number in $[-t.maxE, t.maxE]$ generated based on the given distribution. By definition, $U(0) = 0$. The value $totalDiff$ records the sum of differences between the current and target value of the task after each time step has completed (Line 20). The goal of the agents is to minimize $totalDiff$.

During each time step of a run, the following three events occur. First, the current value of the task is decreased by the rate-of-change value $t.\delta$ (Line 7). Next, each agent makes a decision about whether or not to work on the task during the current time step (Line 8-14). An agent remains idle if it believes that the task does not require any work. Finally, each agent that has decided to work increases the current value of the task by $t.contribution$ (Lines 15-19). These three events repeat $maxTime$ iterations.

III. DEFINITIONS

Definition 1: The value $tc = n * t.contribution$ is the **total contribution** of the team, i.e., the amount of work that would go into the task if every agent were to act at the same time.

Definition 2: A system is deemed **feasible** if $tc \geq t.\delta$ and **infeasible** if $tc < t.\delta$. In an infeasible system, the agents cannot contribute enough work to counteract the rate-of-change value, $t.\delta$, and thus, the $t.current$ value will tend towards $-\infty$. Throughout this text, it is assumed that all systems are feasible.

Definition 3: The **state**, s_k , at timestep k is equal to the $t.current$ value before $t.\delta$ has been subtracted from $t.current$, i.e., the $t.current$ value before Line 7 of Algorithm 1 is executed. The set $s^* = \{s_1, s_2, \dots, s_l\}$ is the ordered set of all unique states for some system. By definition, s_{k+1} refers to the state succeeding s_k , even in the situation where $k = l$.

Definition 4: The **pre-state**, p_k , is the $t.current$ value after $t.\delta$ has been subtracted from $t.current$ and before the agents act, i.e., the $t.current$ value between Lines 8 and 14 in Algorithm 1. The pre-state represents the actual state of the system when the agents make their decisions about whether or not to work and is used when determining the expected behavior of the agents each time step. The ordered set of all pre-states is defined as $p^* = \{p_1, p_2, \dots, p_l\}$.

Definition 5: The **post-state**, d_k , is the $t.current$ value after the contribution of each agent has been added to it, i.e., the $t.current$ value at Line 20 in Algorithm 1. The post-state represents the state of the system when computing *totalDiff*. The set $d^* = \{d_1, d_2, \dots, d_l\}$ is the ordered set of all post-states. Note, the value of post-state d_k is equal to the value of the state s_{k+1} .

Definition 6: The function $actors(p_k)$ is defined as the set of agents that act when the system is in pre-state p_k .

Definition 7: The value e_{p_k} is defined as the expected proportion of agents to act on the task when the system is in pre-state p_k . This value is used when modeling the behavior of Algorithm 1.

IV. NO INTER-AGENT VARIATION

Suppose we have a system with no error and a task with an initial current value and a target value of zero.¹ Assuming the system is feasible, what is the the expected value returned by the simulation? To compute this value, we determine all values in d^* , compute their average, and multiply the average by *maxTime*.

The total number of states, l , for a system without error is equal to the number of iterations needed for the system

¹For simplicity, and without loss of generality, we assume that the systems throughout the remainder of this text have an initial current value and a target value of zero. These assumptions allow for a simpler and more straightforward analysis of how variation can affect self-organization in a distributed, decentralized system.

to reenter the initial state, i.e., a state where $t.current = 0$. This amount of time is found by determining the minimal l value that satisfies $x * tc - l * t.\delta = 0$ for some positive integers l and x . The smallest positive value of l that satisfies the previous equation is

$$l = \frac{LCM(tc, t.\delta)}{t.\delta} = \frac{tc}{GCD(tc, t.\delta)}, \quad (1)$$

where $LCM(a, b)$ returns the least common multiple of a and b , and $GCD(a, b)$ returns the greatest common divisor of a and b . The average difference between the post-state and the target value during any given time step is

$$\bar{d}_{noVar} = \frac{tc - GCD(tc, t.\delta)}{2}. \quad (2)$$

To obtain Equation 2, we first determine the values in d^* , sum those values, and then divide by l . If tc and $t.\delta$ are relatively prime, then $l = tc$ and $d^* = \{0, 1, \dots, tc - 1\}$. In general, d^* is the set of values equal to $GCD(tc, t.\delta)$ multiplied by each value in $\{0, 1, \dots, l - 1\}$:

$$d^* = \{0, GCD(tc, t.\delta), \dots, (l - 1) * GCD(tc, t.\delta)\}. \quad (3)$$

By noting that $l = \frac{tc}{GCD(tc, t.\delta)}$ and $\sum_{i=0}^{n-1} i = \frac{n*(n-1)}{2}$, the average value in d^* becomes

$$\begin{aligned} \bar{d}_{noVar} &= \frac{\sum_{k=1}^l d_k}{l} = \frac{GCD(tc, t.\delta) \sum_{z=0}^{l-1} z}{l} \\ &= \frac{GCD(tc, t.\delta) * l(l - 1)}{2l} \\ &= \frac{GCD(tc, t.\delta) * \left(\frac{tc}{GCD(tc, t.\delta)} - 1\right)}{2} \\ &= \frac{tc - GCD(tc, t.\delta)}{2}, \end{aligned}$$

and thus, the expected return value of a run with no variation and *maxTime* steps is $\bar{d}_{noVar} * maxTime$. In the following sections, we use both continuous and discrete models of the system to show how variation in the agents' response to the task can result in a lower return value than the same system without inter-agent variation.

V. UNIFORM VARIATION

We begin by examining how the system behaves when variation is uniformly distributed within a radius of the target value. Our model shows that systems with inter-agent variation are able to reach a stable point where $s_{k+1} = s_k$. Systems with little-to-no variation are not able to reach such a point.

The expected proportion of agents that act when a system is in pre-state p_k is

$$e_{p_k} = \begin{cases} 0 & \text{if } p_k \geq t.maxE \\ 1 & \text{if } p_k < -t.maxE \\ 1 - \frac{p_k + t.maxE}{2 * t.maxE} & \text{otherwise.} \end{cases} \quad (4)$$

The expected contribution of the agents is $e_{p_k} * tc$, and thus, the expected post-state, d_k , of a system in state s_k is

$$d_k = \begin{cases} p_k & \text{if } p_k \geq t.maxE \\ p_k + tc & \text{if } p_k < -t.maxE \\ p_k + tc * \left(1 - \frac{p_k + t.maxE}{2 * t.maxE}\right) & \text{otherwise.} \end{cases} \quad (5)$$

Recall from Definition 5, that the post-state d_k of state s_k is equal to the value of the successive state, s_{k+1} . Thus, Equation 5 is the recurrence relation mapping s_k to s_{k+1} .

Figure 1 uses Equation 5 to trace the trajectory of systems with different values of tc and $t.\delta$ and an initial current value of 0.0. In each plot, $t.\delta = 0.2$, and for the first three columns, $t.maxE = 1$, and for the right-most column, $t.maxE = 0$. The tc values vary between the plots. By tracing the trajectory of the system, we see that systems with error eventually settle in to a state where $s_{k+1} = s_k$. This stable point can be found by setting s_{k+1} equal to s_k in the third part of Equation 5 and by solving for s_k :

$$s_k = t.maxE + t.\delta - \frac{2 * t.maxE * t.\delta}{tc}. \quad (6)$$

When this value is plugged in to the third part of Equation 4, we see that the system reaches a point where $tc * e_{p_k} = t.\delta$, i.e., the system is stabilizing at a point where the contribution of the agents each time step equals the rate-of-change value.

Without error, Equation 5 becomes discontinuous as there exists no point where $s_{k+1} = s_k$. Instead of settling to a stable point, the system cycles between the states described by Equation 3. Similarly, systems with small amounts of inter-agent variation are not able to converge to a stable point in the state space. Figure 2 plots the bifurcation diagram [22] showing all of the states reached between time steps 1000 and 5000 for systems with $tc = 1$, $t.\delta = 0.2$, and $0 \leq t.maxE \leq 0.3$. States before time step 1000 are not plotted in order to focus on if, and where, the system eventually settles. With small amounts of inter-agent variation, the systems shown in Figure 2 are not able to reach a point where s_{k+1} equals s_k because as $t.maxE$ decreases, so too does the likelihood of satisfying the third condition in Equation 5.

Three types of behaviors can be seen in Figure 2: periodic behavior where the system cycles through more than one state, chaotic behavior, and stable behavior where the system settles into a single state. When $t.maxE = 0$, the system cycles between the states given by Equation 3, i.e., the states 0.8, 0.6, 0.4, 0.2, and 0. This behavior is illustrated in the top-right plot in Figure 1. When $0 < t.maxE \leq 0.1$, the system cycles between the states 0.9, 0.7, 0.5, 0.3, and 0.1. The behavior of these particular systems are given as follows. First, the task's current value is reduced to -0.2 and then increased to 0.8 because all of the agents find that the task's current value is below their perceived target values. The state then proceeds to drop to 0.6, 0.4, and 0.2.

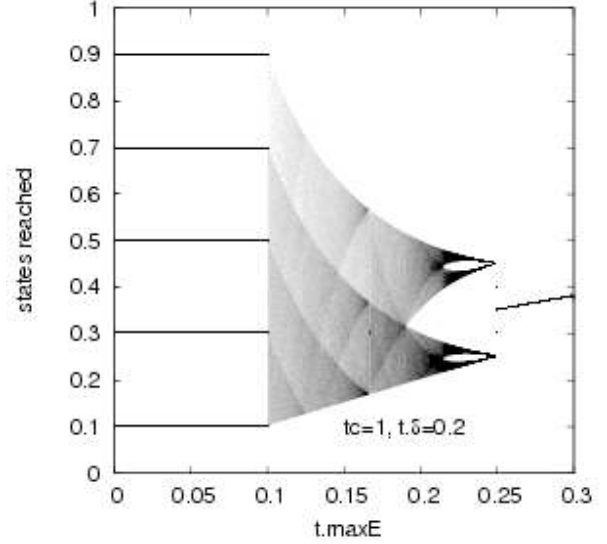


Figure 2. This plot shows how, under uniform distribution, systems with small amounts of inter-agent variation are not able to reach a point where s_{k+1} equals s_k . Each system is run for 5000 time steps and only states after time step 1000 are plotted.

When the task reaches a state of 0.2, and thus, a pre-state of 0.0, an average of half of the agents work on the task, thus increasing its value to 0.5. From here, the state decreases to 0.3, and at state 0.1 (i.e., pre-state -0.1), all of the agents act on the task and its value is increased to 0.9. Again, none of the agents act and the task's state reaches values of 0.7, 0.5, 0.3, and 0.1. At this point, the cycle repeats.

Chaotic-like behavior is seen when $t.maxE$ is between 0.1 and approximately 0.25. After $t.maxE$ has passed 0.25, the system is able to reach a stable point where $s_k = s_{k+1}$. Future work will look at exploring the chaotic behavior of systems with low error in order to determine analytically how much error is needed for the system to reach stability. For now, the goal is to show how to take advantage of systems with adequate amounts of inter-agent variation in order to produce stability within such systems.

VI. GAUSSIAN VARIATION

Often, in real systems, variation occurs with a Gaussian distribution. This distribution models imperfections and physical limitations. Using a Gaussian distribution of error may be more realistic. This model also shows stability where $s_{k+1} = s_k$ with enough inter-agent variation.

In modeling Gaussian distribution, the third case of Equation 4, which calculates the expected proportion of agents to act in pre-state p_{s_k} , becomes

$$e_{p_k} = \frac{1}{\sqrt{2\pi}} \int_z^{\infty} e^{-\frac{x^2}{2}} dx, \quad (7)$$

where $z = \frac{p_k - t.target}{t.stdE}$. This is equivalent the statistical Q-function evaluated at $Q\left(\frac{p_k - t.target}{t.stdE}\right)$. Thus, the third case in

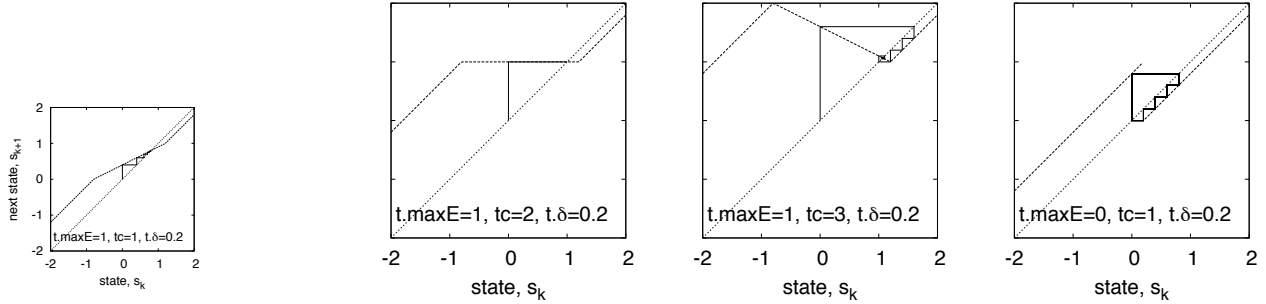


Figure 1. These figures plot the recurrence relation between s_k and s_{k+1} for a uniform distribution from Equation 5 and show how the trajectory of systems with $t.\delta = 0.2$ and an initial current value of 0.0 evolve under different conditions. For the first three columns $t.maxE = 1.0$, and in the fourth column $t.maxE = 0.0$. From left to right, the columns show plots for tc values of 1.0, 2.0, 3.0, and 1.0, respectively. The diagonal line shows where $s_k = s_{k+1}$.

Equation 5 for the expected post-state, d_k , simply becomes

$$d_k = p_k + \frac{tc}{\sqrt{2\pi}} \int_z^{\infty} e^{-\frac{x^2}{2}} dx, \quad (8)$$

where again $z = \frac{p_k - t.target}{t.stdE}$. Figure 3 uses Equation 8 to trace the trajectory of systems with different values of tc and $t.\delta$ and an initial current value of 0.0.

Following the trajectories, we find that systems with error are again able to find a stable state, such that $s_{k+1} = s_k$. This point can be approximated using the bounds for the Q-function. A simple and relatively tight approximation is the Chernoff bound

$$Q(x) \leq \frac{e^{-(x)^2}}{2}.$$

This approximation gives a simple equation for the finding $s_{k+1} = s_k$,

$$s_k \leq t.stdE \sqrt{-2 \ln \frac{2 * t.\delta}{tc}} + t.\delta + t.target. \quad (9)$$

This equation gives a complex number for $\frac{t.\delta}{tc} \geq 0.5$. However, ignoring the imaginary part still gives an approximation if the result from the square root is negated. Using this value finds a point in the system where, $tc * e_{p_k} \approx t.\delta$.

State behavior with Gaussian error distribution shows the same general behaviors with increasing error as seen with uniform error distribution. Figure 4 shows the Gaussian bifurcation diagram for time steps 1000 to 5000 in a system defined with $tc = 0, t.\delta = 0.2$, and $0 \leq t.maxE \leq 0.9$. Again small amounts of variation lead to periodic behavior and a stable state is not reached. Until enough error is introduced to produce a stable system, chaotic and periodic behaviors exist. Also, in each cycle found between chaotic phases as $t.maxE$ increases, the number of states that the system cycles through decreases by one. In Figure 4, at approximately $t.maxE = 0.3$, the system abandons chaotic behavior all together before finding a stable point near

$t.maxE = 0.42$.

Comparing Figures 2 and 4, we find that the Gaussian distribution appears to need a larger error range than the uniform distribution in order to stabilize. With the Gaussian distribution of error, a majority of the agents sense the target as relatively close to the actual value. Since most of the agents respond at values closer to the target, e_{p_k} has greater fluctuations with small differences in the pre-state. Larger spread of error gives less differences in e_{p_k} and less drastic changes in states.

VII. EXAMINING INDIVIDUAL RUNS

We provide further evidence to show that variation in agent behavior can improve a system's ability to self-organize by examining the behavior of individual runs. Additionally, we describe how the analysis done so far can be used to optimize a team's behavior on the given multi-agent task allocation problem.

The plots in Figure 5 show the average value returned by Algorithm 1 for $n = 100, t.contribution = 1, t.\delta \in \{10, 30, 60, 90\}$, and $0 \leq t.maxE \leq 150$. Except for when the error values are low (this case is described in the previous section and illustrated in Figure 2), Equation 6 accurately predicts the values returned by Algorithm 1. Specifically, the top two plots in Figure 5 show that when $t.\delta$ equals 10 or 30, the average post-state of the system increases as $t.maxE$ increases. The bottom two plots show that higher $t.\delta$ values result in a decreasing value of the average post-state as $t.maxE$ increases. Each point in Figure 5 is obtained by running Algorithm 1 with the same set of parameters fifty times. The only difference between the runs used for a single data point is the assignment of the agents' error values. For each run, the post-state values are summed over five-hundred time steps, and these fifty values are averaged to obtain a single point on the plot. Error bars show the standard deviation for these fifty values, and the line in each plot shows the expectation given by taking the absolute value

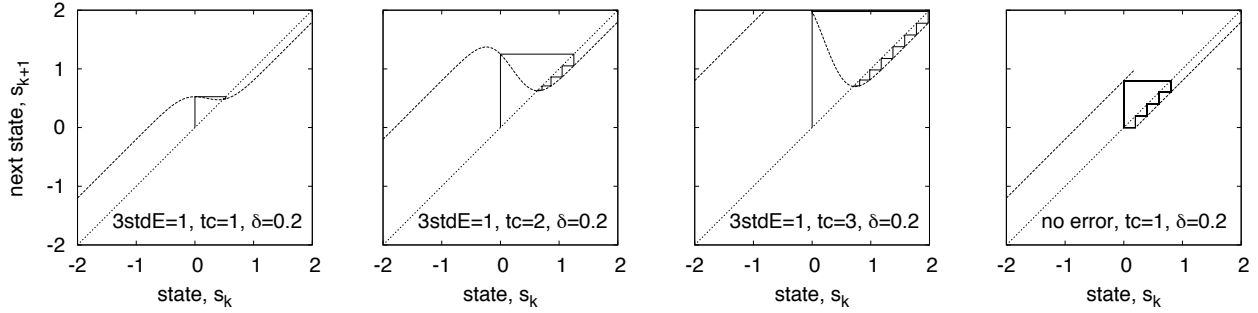


Figure 3. These figures plot the recurrence relation between s_k and s_{k+1} for a Gaussian distribution from Equation 8 and show how the trajectory of systems with $t.\delta = 0.2$ and an initial current value of 0.0 evolve under different conditions. For the first three columns $t.maxE = 1.0$, and in the fourth column $t.maxE = 0.0$. From left to right, the columns show plots for $t.c$ values of 1.0, 2.0, 3.0, and 1.0, respectively. The diagonal line shows where $s_k = s_{k+1}$.

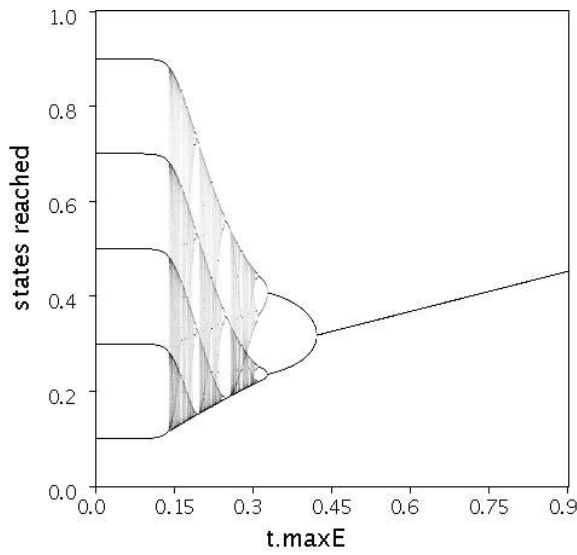


Figure 4. This plot shows how, under Gaussian distribution, systems with small amounts of inter-agent variation are not able to reach a point where s_{k+1} equals s_k . Each system is run for 5000 time steps and only states after time step 1000 are plotted.

of Equation 6 and multiplying it by $maxTime = 500$. For Gaussian variation, the data is similar but with gentler slopes due to the slower dispersion of agents away from the target as error increases.

Recall that the goal of the agents is to minimize the difference between the post-state and the target value of the task. On the surface then, Figure 5 would indicate that certain ranges of error values should be avoided because they cause the system to perform worse than a system without error. However, this is not necessarily the case, as we show that a system with error and a high $totalDiff$ value can be desirable as long as two conditions are met: first, the average post-state value must be predictable, and second, the variance of the post-state values throughout each individual

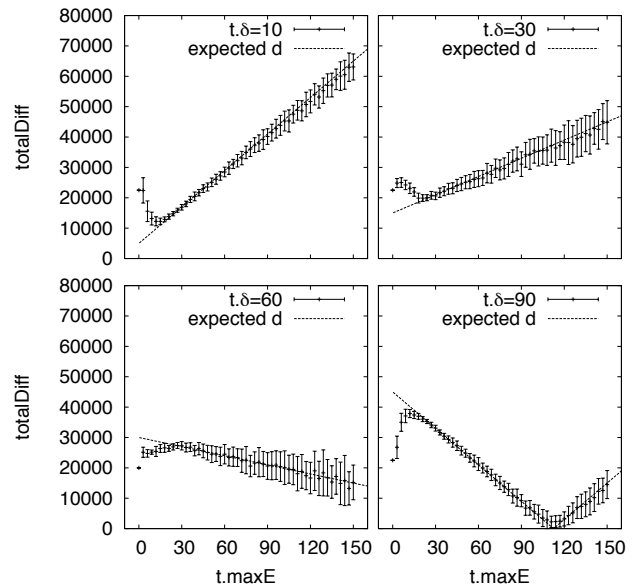


Figure 5. These plots show the average value returned by Algorithm 1 for systems with $maxTime = 500$, $n = 100$, $t.contribution = 1$, $t.\delta \in \{10, 30, 60, 90\}$, and $0 \leq t.maxE \leq 150$. Values closer to zero indicate better performance. The line in each plot shows the expectation given by taking the absolute value of Equation 6 and multiplying it by 500.

run must be low. When these two conditions are met, we can adjust the perceived target value of the agents so that their actions cause the post-states to be at or near the actual target value.

It can be seen from Figure 5 that the first condition is satisfied because Equation 6 is a good predictor of the average post-states in all cases except for where $t.maxE$ is low. Regarding the second requirement, Figure 5 says nothing about the variance *within a single run*, since only the average $totalDiff$ values are used when plotting its points. Figures 6 and 7 show the average standard deviation of the post-states *within* each of the fifty runs for uniform and Gaussian distribution, respectively. Lower values indicate

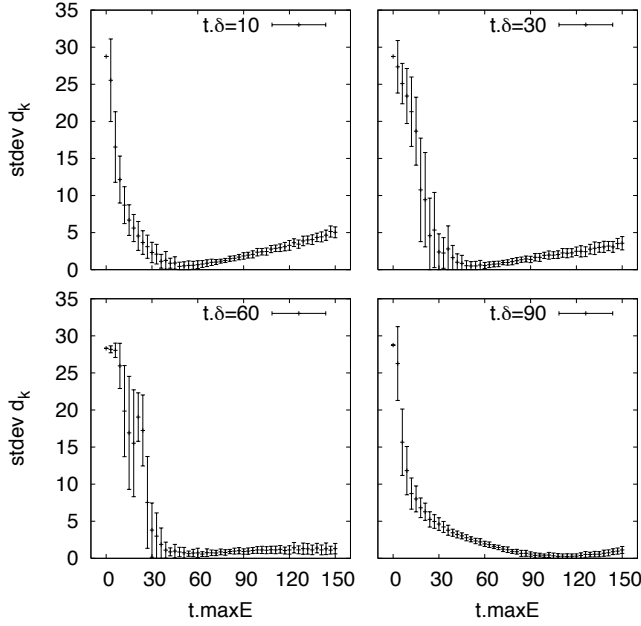


Figure 6. The points in each figure show the average standard deviation of the post-states for the uniform distribution runs given in Figure 5. Lower values indicate that the system is able to settle down into a stable region of the state space.

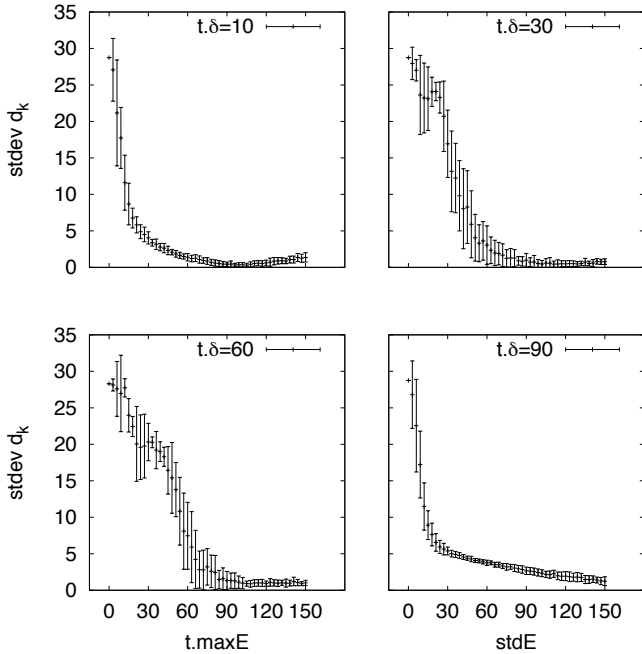


Figure 7. The points in each figure show the average standard deviation of the post-states for the Gaussian distribution runs. Lower values indicate that the system is able to settle down into a stable region of the state space.

that the system is able to settle down in to a stable region of the state space. Except for when the error values are low, the post-states have low variance, thus indicating that the agents are able to reach an area of the state space where

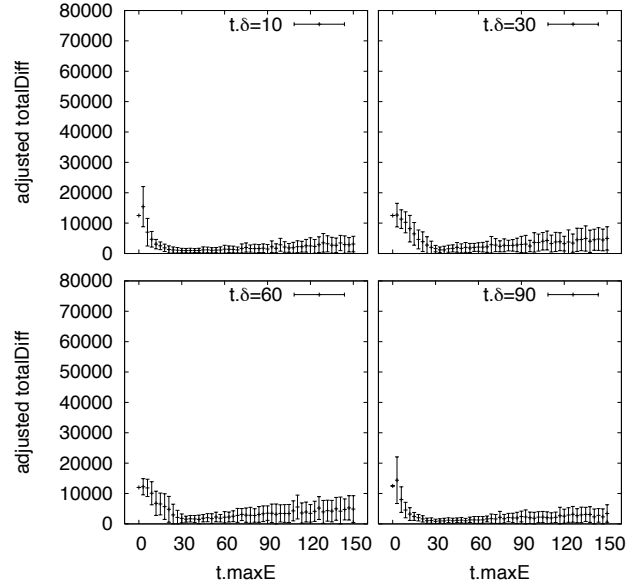


Figure 8. These plots show the average value returned by Algorithm 1 when the agents' perceived target values are adjusted to equal the negative of Equation 6. Values closer to zero indicate better performance. Compare these plots to those of Figure 5 where the agents' perceived target values are equal to the actual target value of the task.

an appropriate number of them act each time step in order to counteract the $t.\delta$ value. In comparing the uniform and Gaussian models, it is again apparent that more error is needed for the Gaussian distribution model to have the same performance as the uniform distribution model.

Given $t.maxE$, $t.\delta$, and tc , we can use Equation 6 or 9 to compute the expected post-state of the system, and thus, determine how to adjust the target value of each agent so that their actions cause the post-states of the task to be at or near the actual target value. This can be achieved by having the agents perceive the target value as being equal to the negative of Equation 6 or 9. Figure 8 shows the average value returned by Algorithm 1 with a uniform distribution after the agents' perceived target values are adjusted. Adjusting the point at which the agents react to the stimulus allows for the system to settle at a point closer to the desired target value. This result can be seen by comparing the plots in Figure 8 to the corresponding ones in Figure 5.

VIII. A LINK TO ASHBY'S WORK

In 1947, cyberneticist William Ross Ashby used the phrase *self-organization* to describe machines that undergo a "self-induced change of organisation" [21]. In Ashby's description of self-organization it is assumed that the system in question is deterministic, can be measured objectively, and its variables can be described numerically. Ashby claims that if at least one of the variables in the system is a step-function of time, then a "spontaneous change of organisation will appear to occur" [21]. To illustrate this point, he gives an example of a system containing n binary variables, $\{x_1, x_2, \dots, x_n\}$, where variable x_n can take one

k	1	2	3
s_k	0	5	10
p_k	-10	-5	0
$actors(p_k)$	A	A	$\{a_1\}$
d_k	5	10	5
$f(s_k)$	s_2	s_3	s_2

Table I

STATE TABLE FOR SYSTEM WITH THREE AGENTS, AN INITIAL $t.current$ VALUE EQUAL TO ZERO, $t.target = 0$, $t.contribution = 5$, $t.\delta = 10$, $a_1.error = 2.5$, $a_2.error = 0.0$, AND $a_3.error = -2.5$.

of two possible values $\{x_n^1, x_n^2\}$. Because the behavior of the system only depends on the values of its variables, i.e., no external influences, then it can be described by the following set of equations:

$$\frac{dx_i}{dt} = f_i\{x_1, x_2, \dots, x_n\}, \quad 1 \leq i \leq n. \quad (10)$$

Equation 10 can be broken into the following set of equations based on the current value of x_n :

$$\frac{dx_i}{dt} = \begin{cases} g_i\{x_1, x_2, \dots, x_{n-1}\}, & 1 \leq i \leq n & \text{if } x_n = x_n^1 \\ h_i\{x_1, x_2, \dots, x_{n-1}\}, & 1 \leq i \leq n & \text{if } x_n = x_n^2. \end{cases}$$

The variable x_n is considered to be *absorbed* by the functions g_i and h_i since it is constant in both cases. These two systems of $n - 1$ variables are considered to be an *organization* in themselves and switch between each other depending on x_n 's value. In this sense, Ashby claims the system has self-organized. Using Ashby's definition of self-organization to describe a task allocation problem, our analysis supports the idea that inter-agent variation in a cooperative multi-agent system creates more organizations, thus allowing for the possibility of a wider range of possible system-level responses to changing stimuli than are possible without variation [1].

Except for when the error values are being set, Algorithm 1 is a deterministic process, thus allowing us to construct a deterministic finite state automaton (DFA) that captures the behavior of any feasible system. Function $f : s^* \rightarrow s^*$ maps each state to its successive state, and because s^* is finite for feasible systems, the DFA eventually enters in to a cycle. The set s^* and the function f can be constructed by running the simulation and keeping a history of the states and their transitions. The run can be stopped whenever a state is reached that is already in s^* .

As an example, consider a system with three agents and the following parameters: $t.target = 0$, $t.contribution = 5$, $t.\delta = 10$, $a_1.error = 2.5$, $a_2.error = 0.0$, and $a_3.error = -2.5$. Table I shows the state transition table for this system. Using Ashby's definition of self-organization, the DFA described in Table I can be seen to possess four different organizations with respect to p_k and the function

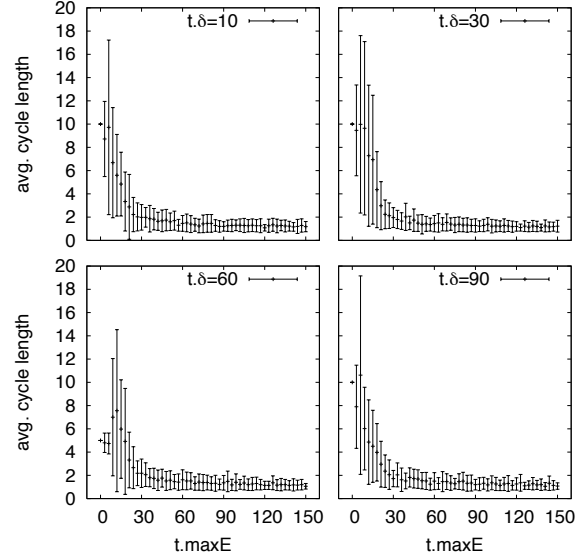


Figure 9. These plots correspond to those in Figures 8 and 10 and show the average length of the cycles that systems eventually settle in to.

$actors(p_k)$:

$$actors(p_k) = \begin{cases} \emptyset & \text{if } p_k \geq 2.5 \\ \{a_1\} & \text{if } 0 \leq p_k < 2.5 \\ \{a_1, a_2\} & \text{if } -2.5 \leq p_k < 0 \\ \{a_1, a_2, a_3\} & \text{if } p_k < -2.5. \end{cases}$$

In general, the number of different organizations that a system has is equal to one plus the number of unique error values that the agents have. A system without any variation has only two organizations. A system containing agents all with different error values has $n + 1$ organizations. Variation in agent behavior allows for a wider range of possible subsets of agents that can act at each time step than does systems without variation. Stability is achieved when the system enters an organization that results in the contribution of the agents equaling the rate-of-change value, i.e., an organization where $|actors(p_k)| * t.contribution = t.\delta$. Because the error values are randomly assigned in Algorithm 1, the conditions for entering the organization where $t.contribution * |actors(p_k)| = t.\delta$ may be difficult or impossible to meet; however, as $t.maxE$ increases, so too does the range of states defining each organization. Figure 9 shows the average lengths of the cycles that the fifty uniform distribution runs from Figure 8 eventually settle in to, and Figure 10 shows the proportion of those runs where the cycle lengths are one. As $t.maxE$ increases the average cycle length decreases and the proportion of runs that reach a stable point where $s_k = s_{k+1}$ increases. By creating more organizations, inter-agent variation allows agents to temper their reactions to the task-stimulus, thus stabilizing the task's state. By understanding the effects of inter-agent variation, we are able to predict and adjust the behavior of the agents

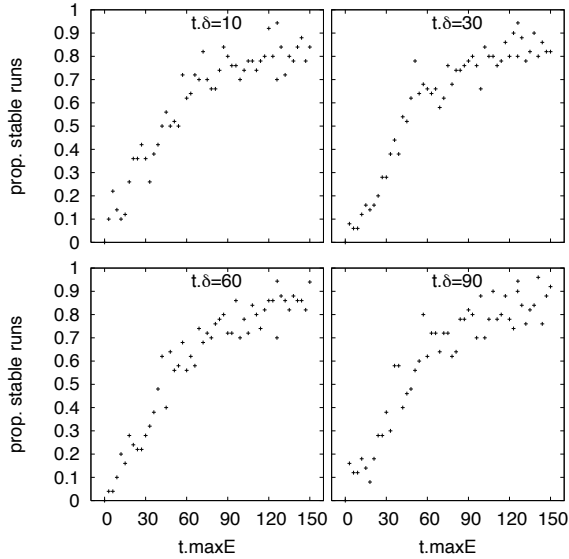


Figure 10. These plots correspond to those in Figures 9 and 8 and show the proportion of runs where the system settles into an organization where $s_k = s_{k+1}$

in a way that takes advantage of the stability caused by inter-agent variation, and thus, provide a solution to the problem.

IX. DISCUSSION

The goal of this work is to gain a better understanding of the impact of variation on distributed multi-agent systems. We hypothesize that some levels of inter-agent variation may be beneficial to the ability of the system as a whole to self-organize. We investigate this hypothesis by studying the behavior of a team of agents working together to solve a multi-agent task allocation problem in which the agents must maintain the level of some constantly decreasing resource or system variable. When the value of the task drops below some pre-defined target value, the agents react and increase the task's value in order to bring it closer to the target. Inter-agent variation is introduced by giving the agents erred views of the current value of the task. To explore the effects of inter-agent variation on this problem, we begin by analyzing the behavior of systems without any variation. This analysis provides a way to determine the value returned by the simulation for systems without inter-agent variation and is used in comparisons with systems where inter-agent variation is present. A dynamical systems approach is used to predict the behavior of systems containing inter-agent variation, and we find that systems with an adequate amount of inter-agent variation reach a stable point where the state of the system at time t equals the state of the system at time $t + 1$. Furthermore, we are able to compute this point, and then use these results to program the agents so that they work together effectively to solve the given multi-agent task allocation problem. Finally, we discuss the importance of inter-agent variation and self-organization based on a definition of self-

organization given by cyberneticist William Ross Ashby in 1947 [21]. Specifically, we show how variation helps to create more “organizations” of the state-space, thus allowing for more combinations of agent-actions than are possible without inter-agent variation. When taken together, these experiments and analysis support the hypothesis that inter-agent variation can help cooperative multi-agent systems self-organize and complete their objectives.

In future work, we will look to determine the exact amount of inter-agent variation needed for a system to stabilize. Above, we state that systems need an “adequate” amount of inter-agent variation to stabilize; however, the amount inter-agent variation necessary for stability is not currently known. Factors such as the task's rate of change, the agents' contribution, and the team size will most likely contribute to how much inter-agent variation is needed for a system to stabilize. Additional directions for future work include investigating problems containing more than one type of task, problems where the rate-of-change value and contribution values change over time, and problems where agents must adjust their behaviors when teammates begin to break down and can no longer work. This future work looks to provide further evidence of the significant effects of inter-agent variation on cooperative multi-agent systems. By understanding how inter-agent variation affects the dynamics of such systems, we can begin to harness the power of variation and use it to our advantage.

X. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful feedback. This work was supported by ONR grant N000140911043.

REFERENCES

- [1] W. R. Ashby, “Requisite variety and its implications for the control of complex systems,” *Cybernetica*, vol. 1, no. 2, pp. 83–99, 1958.
- [2] R. Axelrod and M. D. Cohen, *Harnessing Complexity*. New York: Free Press, 1999.
- [3] D. E. Jackson, S. J. Martin, F. L. W. Ratnieks, and M. Holcombe, “Spatial and temporal variation in pheromone composition of ant foraging trails,” *Behavioral Ecology*, vol. 18, no. 2, pp. 444–450, 2007.
- [4] J. C. Jones, M. R. Myerscough, S. Graham, and B. P. Oldroyd, “Honey bee nest thermoregulation: Diversity promotes stability,” *Science*, vol. 305, pp. 402–404, 2004.
- [5] F. Ravary, E. Lecoutey, G. Kaminski, N. Châline, and P. Jaisson, “Individual experience alone can generate lasting division of labor in ants,” *Current Biology*, vol. 17, pp. 1308–1312, 2007.
- [6] A. Campbell and A. S. Wu, “On the significance of synchronicity in emergent systems,” in *Proceedings of the Eighth International Conference on Autonomous Agents and Multi-agent Systems*, 2009, pp. 449–456.

- [7] A. Campbell, A. S. Wu, K. Garfield, R. Shumaker, S. Luke, and K. A. D. Jong, "Empirical study on the effects of synthetic social structures on teams of autonomous vehicles," in *Proceedings of the IEEE International Conference on Networking, Sensing, and Control*, 2006, pp. 440–445.
- [8] M. J. B. Krieger and J.-B. Billeter, "The call of duty: Self-organised task allocation in a population of up to twelve mobile robots," *Robotics and Autonomous Systems*, vol. 30, pp. 65–84, 2000.
- [9] O. Kittithreerapronchai and C. Anderson, "Do ants paint trucks better than chickens? Markets versus response thresholds for distributed dynamic scheduling," *IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1431–1439, December 2003.
- [10] T. H. Labella, M. Dorigo, and J.-L. Deneubourg, "Division of labor in a group of robots inspired by ants' foraging behavior," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 1, no. 1, pp. 4–25, 2006.
- [11] B. P. Oldroyd and J. H. Fewell, "Genetic diversity promotes homeostasis in insect colonies," *Trends in Ecology and Evolution*, vol. 22, no. 8, pp. 408–413, 2007.
- [12] S. N. Beshers and J. H. Fewell, "Models of division of labor in social insects," *Annual Review of Entomology*, vol. 46, pp. 413–440, 2001.
- [13] G. F. Oster and E. O. Wilson, *Caste and Ecology in the Social Insects*. Princeton, New Jersey: Princeton University Press, 1978.
- [14] G. E. Robinson, "Regulation of division-of-labor in insect societies," *Annual Review of Entomology*, vol. 37, pp. 637–665, 1992.
- [15] T. D. Seeley, "Adaptive significance of the age polyethism schedule in honeybee colonies," *Journal of Behavioral Ecology and Sociobiology*, vol. 11, no. 4, pp. 287–293, 1982.
- [16] E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg, "Phase diagram of a model of self-organizing hierarchies," *Physica A*, vol. 217, no. 3, pp. 373–392, 1995.
- [17] D. A. Markiewicz and S. O'Donnell, "Social dominance, task performance and nutrition: implications for reproduction in eusocial wasps," *Journal of Comparative Physiology A*, vol. 187, no. 5, pp. 327–333, 2001.
- [18] S. Picault and A. Collinot, "Designing social cognition models for multi-agent systems through simulating primate societies," in *Proceedings of the Third International Conference on Multiagent Systems*, 1998, pp. 238–245.
- [19] S. Graham, M. R. Myerscough, J. C. Jones, and B. P. Oldroyd, "Modelling the role of intracolony genetic diversity on regulation of brood temperature in honey bee (*Apis Mellifera* L.) colonies," *Insectes Sociaux*, vol. 53, pp. 226–232, 2006.
- [20] M. R. Myerscough and B. P. Oldroyd, "Simulation models of the role of genetic variability in social insect task allocation," *Insectes Sociaux*, vol. 51, pp. 146–152, 2004.
- [21] W. R. Ashby, "Principles of the self-organizing dynamic system," *The Journal of General Psychology*, vol. 37, pp. 125–128, 1947.
- [22] R. C. Hilborn, *Chaos and Nonlinear Dynamics*. New York: Oxford University Press, Inc., 1994.