

University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

CSE Journal Articles

Computer Science and Engineering, Department
of

2019

Co-Regulated Consensus of Cyber-Physical Resources in Multi-Agent Unmanned Aircraft Systems

Chandima Fernando

University of Nebraska - Lincoln, wfernando@huskers.unl.edu

Carrick Detweiler

University of Nebraska - Lincoln, cdetweiler2@unl.edu

Justin Bradley

University of Nebraska - Lincoln, jbradley@cse.unl.edu

Follow this and additional works at: <https://digitalcommons.unl.edu/csearticles>

Fernando, Chandima; Detweiler, Carrick; and Bradley, Justin, "Co-Regulated Consensus of Cyber-Physical Resources in Multi-Agent Unmanned Aircraft Systems" (2019). *CSE Journal Articles*. 208.

<https://digitalcommons.unl.edu/csearticles/208>

This Article is brought to you for free and open access by the Computer Science and Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in CSE Journal Articles by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Article

Co-Regulated Consensus of Cyber-Physical Resources in Multi-Agent Unmanned Aircraft Systems

Chandima Fernando *, Carrick Detweiler and Justin Bradley

Department of Computer Science and Engineering, University of Nebraska–Lincoln, 256 Avery Hall, Lincoln, NE 68588, USA; carrick@cse.unl.edu (C.D.); jbradley@cse.unl.edu (J.B.)

* Correspondence: wfernando@huskers.unl.edu; Tel.: +1-701-412-4641

Received: 11 April 2019; Accepted: 17 May 2019; Published: 23 May 2019



Abstract: Intelligent utilization of resources and improved mission performance in an autonomous agent require consideration of cyber and physical resources. The allocation of these resources becomes more complex when the system expands from one agent to multiple agents, and the control shifts from centralized to decentralized. Consensus is a distributed algorithm that lets multiple agents agree on a shared value, but typically does not leverage mobility. We propose a coupled consensus control strategy that co-regulates computation, communication frequency, and connectivity of the agents to achieve faster convergence times at lower communication rates and computational costs. In this strategy, agents move towards a common location to increase connectivity. Simultaneously, the communication frequency is increased when the shared state error between an agent and its connected neighbors is high. When the shared state converges (i.e., consensus is reached), the agents withdraw to the initial positions and the communication frequency is decreased. Convergence properties of our algorithm are demonstrated under the proposed co-regulated control algorithm. We evaluated the proposed approach through a new set of cyber-physical, multi-agent metrics and demonstrated our approach in a simulation of unmanned aircraft systems measuring temperatures at multiple sites. The results demonstrate that, compared with fixed-rate and event-triggered consensus algorithms, our co-regulation scheme can achieve improved performance with fewer resources, while maintaining high reactivity to changes in the environment and system.

Keywords: consensus; co-regulation; cyber-physical system; distributed control; unmanned aircraft system

1. Introduction

The success of an autonomous, robotic mission can be measured by the effectiveness and efficiency in completing the mission [1]. While physical resources (e.g., time, energy, power, and space) have historically dominated the metrics of success, more capable, complex cyber-physical agents require the judicious allocation of cyber resources (e.g., communication and computation) as well. Without a strongly coupled co-design strategy, cyber and physical resources are typically optimized either independent of one another, or perhaps in an iterative design loop until requirements are satisfied [2]. For example, controllers are typically designed to handle worst-case conditions or maneuvers while providing good margins of stability and robustness [3,4]. Selection of sampling rate, or period of execution of the control law often becomes a software or real-time system implementation issue [5]. However, even if co-designed, cyber and physical resources should be co-regulated at run-time, in conjunction with holistic (i.e., both physical and cyber)

system performance. The situation is further complicated in the case of multi-agent, or distributed systems. Concisely, current design methods do not allow for dynamic adjustment of physical, communication, and cyber resources in response to changing conditions, objectives, or system faults.

Motivating Example

During a prescribed burn, building a temperature profile of the area enables detection of areas that are too hot (e.g., fires that may damage the crown of the trees) and helps monitor which areas need a different ignition profile. Sometimes, thermocouple loggers equipped with probes are buried underground in the area of the burn to log the fire temperatures [6]. Alternatively, small wearable temperature sensors can be used by firefighters to build a shallow, but serviceable temperature profile. However, this offline data gathering method does not help in monitoring the live fire and requires physical access to the area selected for the burn.

We are interested in utilizing a team of Unmanned Aircraft Systems (UAS) to build a temperature profile of a fire. Teams of UASs are already being tasked with igniting controlled burns in areas where access is challenging or unsafe [7–9]. A natural additional capability is to leverage the distributed, multi-agent system to simultaneously build the temperature profile by moving and sensing multiple locations. Information consensus [10] provides an excellent strategy for estimating the temperature profile with multiple, moving agents in a decentralized fashion. In this strategy, improved measurements can be obtained by being closer to the point of interest, which also may restrict communication with other agents given the topography. As a result, there are times when it is prudent for each agent to fly close to the area in question, change its frequency of communication, or fly higher to communicate its value with others. In this scenario, an online co-regulated and co-designed information consensus algorithm can dynamically adjust cyber, physical, and communication resources in response to holistic, multi-agent system performance.

Summary of Approach

To solve this challenge, we propose co-regulated consensus controllers which allocate cyber, physical, and communication resources with the goal of obtaining fast convergence and high accuracy, but that require fewer resources than traditional approaches. The resources are adjusted in accordance with the difference between the agents' shared, sensed values. In consensus algorithms, convergence time depends upon the rate of communication between agents as well as the connectivity of the entire group of agents. We co-regulate both the communication frequency and the positions of the agents to change connectivity.

Connectivity and communication rate of the agents are traded off to improve both time to convergence and convergence value against the need to be at a particular altitude for sensing. Specifically, when connectivity becomes sparse, the shared value may only be propagated through a small number of neighboring agents—thereby converging more slowly. Connectivity of the network can be improved either by using long range communicating devices or by changing the physical location of the agent. In the case of UASs, increasing the altitude can improve connectivity; however, this can reduce the accuracy or reliability of the sensing.

The communication rate can be increased to improve convergence time as well, but will impact convergence value if the shared value does not change at a similar frequency. This has given way to the idea of event-triggered consensus wherein the shared value is only communicated between agents when current and sensed value differ more than some threshold [11]. While potentially minimally allocating communication resources, event-triggered and self-triggered consensus algorithms may not react quickly enough to changes in the shared state and typically are more difficult to analyze.

In this paper, we propose an information consensus control scheme in which position of the agents (i.e., connectivity), communication frequency, and the shared information state are simultaneously co-regulated in a single framework. Motion of the agents, and hence connectivity, is co-regulated based on shared state error. When shared state error is high agents are moved to positions that increase network connectivity, and as error diminishes the agents are moved back to a more effective position that increases their ability to closely sense the shared state. Simultaneously, for each agent, propagation of the shared value occurs at a time-varying communication rate. This rate is increased when shared state error is high, and lowered as error diminishes, thus preserving computation and communication when not needed. The result is a dynamic algorithm that provides a densely connected network with high communication rates during transient periods of the shared state, and a more loosely connected, slower communicating network during quiescent periods of the shared state.

Previously, we introduced an initial co-regulated consensus controller strictly for communication frequency [12]. This paper uses the same controller to co-regulate the communication frequency when the information state changes. While the controller achieved improved convergence times and lower communication costs, it does not dynamically adjust connectivity. The work presented here builds on this result in a significant way by adding a co-regulated position controller to maneuver agents in their physical position to achieve higher connectivity. This is then combined with the co-regulated communication frequency controller to improve both convergence rate and accuracy of convergence value. Our algorithm provides a new capability to design controllers that can optimize new performance criteria—an issue we explore in depth in this paper. This paper makes the following contributions:

- Introduction of a position controller to move agents in physical space to adjust connectivity
- Proofs for the convergence of the proposed algorithm
- Introduction of new cost metrics to assess distributed, cyber-physical co-regulation performance
- Comparison of the effectiveness of the proposed co-regulation algorithm against non co-regulated and event-triggered consensus strategies

2. Related Work

Consensus algorithms rely on exchanging state information among the agents. Exchanged states are used to update the state of each agent. The update model used in the paper is similar to the models proposed in [13,14]. Both are predecessors of the distributed coordination algorithms presented in [15,16]. Vicsek's model in [16] studies a leader follower problem where every agent is initialized with a similar speed and a different heading. The state update model adjusts the headings of the agents to the heading of the leader agent. Both Jadbabaie et al. [15] and Ren and Beard [13] improved this model to accommodate leader-less coordination. The consensus model in [13] introduced the information consensus that considered the shared state value as an observation made by the agents of a random variable. The convergence properties of the model are proved for both continuous and discrete time domains.

Most of the theoretical work in consensus assumes *continuous* communication between the agents is possible. Continuous communication assumes the agents broadcast and receive the state values and update their individual states continuously. Guaranteeing convergence properties of such systems requires continuous mathematics and is well established [11]. In recent work, systems with higher-order, or nonlinear dynamics are studied for their convergence properties [17]. If communication and computation rates are sufficiently fast the assumption of continuous communication holds and the aforementioned theoretical analysis is sound. However, in real systems, communication and computation are not continuous but digital, and are inherently discrete (e.g., WiFi and Xbee) [18]. Particularly at slow communication rates, this reality may invalidate convergence guarantees based on continuous mathematics.

Consensus algorithms with discrete-time communication traditionally use a fixed communication period to simplify the analysis of the system [19]. Agents communicate, receive, and update states at these periodic intervals. This period can either be synchronous between agents [20] or asynchronous [21]. Discrete time algorithms are also simple to schedule as periodic tasks in a real-time computing system [22], and convergence properties are easily analyzed using conditions of the Laplacian matrix [23]. Asynchronous communication among the agents is more realistic in real world implementations with the presence of packet losses, communication delays, and clock synchronization errors. When analyzing the convergence of asynchronous systems, the collectively connected topologies are used. In this case “frequent enough” communication by all agents is sufficient to make the network collectively connected [13,24].

Achieving fast convergence times are essential once a multi-agent system is deployed in the real world. There have been extensive studies carried out both on the rate of convergence and the time to converge. The definitions for both the convergence rate and convergence time are stated in [25]. The authors of [26] quantified the convergence rate of an averaging consensus algorithm to be the second eigenvalue (sorted in descending order) of the Laplacian matrix drawn from the connectivity graph. This second eigenvalue, also known as the *Fiedler eigenvalue*, increases with the connectivity of the graph. The more densely the agents are connected, the bigger the second eigenvalue will be. Hence, increasing the number of connections within the multi-agent system yields faster convergence. Increasing the connectivity of agents has been popular in power-grid-related research. For example, consensus algorithms with added innovative methods are used in power management of a micro-grid in [27]. The connectivity of the network is increased by creating additional links between nodes which are at the opposite end of the communication network. This effort has led to faster convergence rates in these types of systems.

Research into improving connectivity through repositioning can be done in the mobile network/robotics domain. In [28], mobile ad-hoc networks move autonomous and mobile agents to positions where the network will maintain maximum connectivity using a flocking-based heuristic algorithm. Underwater multi-agent communication is challenging, and, as a result, in [29], a mobile surface vehicle provides a linkage between underwater agents to improve connectivity of the network. Optimal waypoints are generated and followed by the surface vehicle to ensure high connectivity [29]. Algorithmically, a semi-definite programming approach to move agents’ location to improve the algebraic connectivity of the network has been proposed [30]. The pairwise distances between the agents decide the movement. In similar work, potential fields have been used to position agents to gain higher connectivity in a consensus example [31]. The method is resilient even when the network is dynamic. These works are related to our proposed methodology by dynamically adjusting position to modify connectivity. However, our strategy offers a low-level reactive coupled feedback mechanism that adjusts position based on consensus error. This does not require computationally complex optimization or trajectory generation algorithms since low-level feedback is used.

In the work presented here, we build on ideas in [27] and our previous work [12] to produce a novel, holistic model for co-design and analysis of co-regulated shared information consensus controllers. The motivating example for our work is a team of UASs monitoring temperatures of a fire at multiple locations.

3. Background

Here, we summarize the key concepts for consensus control, namely: graph theory and matrix theory for consensus, discrete information consensus, and linear quadratic regulators (LQR).

3.1. Graph Theory in Consensus

Graph theory has been used to abstract the communication network among the group of agents by representing agents as nodes and communication as edges of a graph. Analyzing the matrix representation of the same graph helps prove certain consensus properties such as convergence.

Let N -node graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent a N -agent communication system where the set of nodes represents the set of agents $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$. Each edge in \mathcal{G} represents a communication link between two agents and is denoted by $e_{ij} \in \mathcal{E} = (v_i, v_j)$. The edge is directed to represent one-way communication [32]. The adjacency matrix $\mathbf{A} = [a_{ij}]$ of the graph \mathcal{G} is defined as

$$a_{ij} = \begin{cases} 1, & e_{ij} \in \mathcal{E} \\ 0, & \text{otherwise.} \end{cases}$$

Define $\bar{\mathcal{G}} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_W\}$ as the finite set of all possible communication subgraphs between N agents. Let $\hat{\mathcal{G}}_i = \{\mathcal{G}_{i_1}, \mathcal{G}_{i_2}, \dots, \mathcal{G}_{i_w}\} \subset \bar{\mathcal{G}}$ be the union of simple graphs for $v_i \in \mathcal{V}$. We define a simple graph to be a single communication instance between two agents. The edge set of $\hat{\mathcal{G}}_i$ is given by the union of the edge set of $\{\mathcal{G}_{i_1}, \mathcal{G}_{i_2}, \dots, \mathcal{G}_{i_w}\}$ where $j = 1, \dots, w$.

In a *strongly connected* graph, there exist bi-directional paths between any pair of nodes in the graph. A graph wherein all nodes except one have at least one parent is known as a *tree* and the orphan node is named the *root*. In a *spanning tree*, there exists at least one path through all the nodes in the graph.

3.2. Matrix Theory for Consensus

A nonnegative matrix, denoted as $\mathbf{L} \geq 0$, only contains entries greater than or equal to zero. A *stochastic* matrix is a nonnegative matrix with all its row sums equal to one [33]. Let matrix $\mathbf{L} \geq 0$ be a $n \times n$ stochastic matrix. \mathbf{L} is said to be *stochastic indecomposable aperiodic (SIA)* if $\lim_{m \rightarrow \infty} \mathbf{L}^m = \mathbf{1}\mathbf{y}^T$ for some column vector \mathbf{y} where $\mathbf{1}$ is a $n \times 1$ column vector with all the entries equal to one [34].

3.3. Discrete Information Consensus

The graph \mathcal{G} is assigned with N agents, each node representing an agent. Let \mathcal{M}_i denote the set of neighbors of agent i . An agent j is considered a neighbor of agent i if and only if $e_{ij} \in \mathcal{E}$. We assume each agent is connected to itself and hence every agent i is a member of \mathcal{M}_i . Let $\mathbf{x}^I = (x_1^I, x_2^I, \dots, x_N^I)^T$ denote the information state of the agents. \mathbf{x}^I will be shared among the agents and eventually be agreed upon via consensus. Assuming the communication to be either synchronous or asynchronous, we use the discrete time consensus algorithm presented in [13],

$$x_i^I[k + 1] = \frac{1}{\sum_{j \in \mathcal{M}_i} 1} \sum_{j \in \mathcal{M}_i} x_j^I[k] \quad i = 1, \dots, N. \tag{1}$$

$k \in \{1, 2, 3, \dots\}$ is the discrete time step. Given a set of initial conditions $\mathbf{x}^I[0]$, the system is said to achieve consensus asymptotically [10] if,

$$\lim_{k \rightarrow \infty} \left\| x_i^I[k] - x_j^I[k] \right\| \rightarrow 0, \quad i, j = 1, \dots, N. \tag{2}$$

Further, if the convergence value is equal to the average of the initial state values, it is known as the average consensus problem and formally denoted as

$$\text{Average} \left(\mathbf{x}^I[0] \right) = \frac{1}{n} \sum_{i=1}^N x_i^I [0]. \quad (3)$$

3.4. Linear Quadratic Regulator Problem

A Linear Quadratic Regulator (LQR) solves an optimal control problem for a linear, time-invariant system. We use LQR to construct the controller which co-regulates the agents' physical positions. For a linear time invariant system, the infinite-horizon LQR control input, u , is given by,

$$u(t) = -\mathbf{K}\mathbf{x}(t)$$

where $\mathbf{x}(t)$ is the state to be regulated and \mathbf{K} is the constant feedback gain matrix [35]. The control input $u(t)$ minimizes the cost function

$$J = \int_0^{\infty} (\mathbf{x}(t)' \mathbf{Q} \mathbf{x}(t) + u(t)' \mathbf{R} u(t)) dt. \quad (4)$$

\mathbf{Q} and \mathbf{R} are two weighting matrixes and chosen to represent the desired performance—a tradeoff between control effort and state error. The feedback gain matrix is calculated as

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}' \mathbf{P} \quad (5)$$

where \mathbf{A} is the system matrix, \mathbf{B} is the input matrix, and the \mathbf{P} is a solution to the Algebraic Riccati equation (ARE),

$$\mathbf{A}' \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}' \mathbf{P} + \mathbf{Q} = 0. \quad (6)$$

4. Co-Regulation for Information Consensus

This section develops the control strategies to vary the agents' position and communication frequency to achieve faster convergence on the information state. We layout the control strategy on regulating the communication frequency followed by the strategy on regulating the position. Then, the convergence guarantees of the proposed control strategies are analyzed.

4.1. Co-Regulated Communication Consensus

A digital communication task requires computational resources to form a message, encode it, and transmit it via a channel. This means the scheduled communication tasks onboard the computer are what dictate the allocation of communication resources. This creates a powerful mechanism for dynamically allocating communication resources by adjusting the scheduling of the communication task. We leverage this capability to design our co-regulated communication strategy. We assume that the task schedule onboard the computer can be dynamically adjusted to increase or decrease how often an agent communicates.

Quicker convergence times are realized through higher communication rates as it facilitates rapid exchange of information. However, upon reaching consensus, exchanging the same information adds no value and will simply occupy computational cycles and bandwidth that might be otherwise allocated to other tasks. On the other hand, higher rates of communication are required once a change in state variable is detected. This suggests a strategy that varies communication rate in accordance with consensus value

could improve resource allocation. Hence, we make use of the following controller on the communication rate to satisfy the co-regulated communication objective [12].

Let $T_i[k]$ be the time period between two successive communications of agent i , which evolves as a discrete time system

$$T_i[k + 1] = T_i[k] + x_i^F[k]u_i^F[k], \tag{7}$$

where the frequency of communication is denoted by $x_i^F[k] = 1/T_i[k]$, and a control input, which modifies the rate for each agent is denoted by u_i^F . The control input is calculated by the following controller,

$$u_i^F[k] = \underbrace{-\alpha_1^F \left| \sum_{j \in \mathcal{M}_i} (x_i^I[k] - x_j^I[k]) \right|}_{\text{Pushes comm rate toward } x_{i,\max}^F} + \underbrace{\alpha_2^F \left| x_i^F[k] - x_{i,\min}^F \right|}_{\text{Pushes comm rate toward } x_{i,\min}^F}. \tag{8}$$

The convergence guarantees for the above controller are proved in [12].

4.2. Co-Regulated Position Consensus

Connectivity of the multi-agent system can be modified either by changing the existing communication mechanisms (e.g., more powerful antenna), or by moving toward or away from neighboring agents. Given our agents have fixed communication resources onboard during a mission, in this strategy we leverage the mobility of each agent to change proximity to neighboring agents and subsequently adjust the connectivity of the communication network.

In co-regulated position consensus, we manipulate the agents’ connectivity by changing their physical position, based on the local value of the shared state. In this scenario, sparsely distributed agents are moved to a prior known location with good connectivity so that each agent would have a larger set of neighbors to connect to. For a particular mission, it is possible the agents are required to maintain a formation with certain distances, be tied to a fixed geographical position, or perhaps need to improve proximity to sensed phenomena. Such external reasons may prohibit the agents from having full connectivity. As a result, once the states converge, the agents are moved back to their initial positions.

Let x_{1i}^P and x_{2i}^P denote the position and the velocity of agent i . Let u_i^P denotes the physical control input for agent i . We model each agent’s dynamics as a double integrator:

$$\begin{aligned} \dot{x}_{1i}^P &= x_{2i}^P \\ \dot{x}_{2i}^P &= u_i^P. \end{aligned} \tag{9}$$

We now introduce a controller to regulate each agent’s position depending on the difference in the shared state of each agent. This state difference is caused either when a neighbor updates its state upon communicating with another agent or sensing a new state value as a result of an external event. The need to increase connectivity is governed by error in the shared state. Once the state error is reduced, the increased connectivity has no real benefits. As a result, the connectivity can be decreased without affecting the shared state value.

We introduce the following controller for each agent’s physical position,

$$u_i^P = -\mathbf{K} \left(x_{1i}^P - x_{1i,ref}^P \right) \quad i \in 1, 2, \dots, N \tag{10}$$

where \mathbf{K} is the state feedback gain produced from an LQR control design strategy (see Section 3.4). We define $x_{1i,ref}^P$ to be a reference to the position where an agent should move to. $x_{1i,ref}^P$ is bounded above by the maximum position an agent can move to, x_{max}^P , and bounded below by $x_{1i}^P[0]$, which is the initial

position of an agent i . We define x_{common}^P to be the common position for increased connectivity. We select $x_{1i}^P[0]$ and x_{max}^P such that $x_{1i}^P[0] \leq x_{common}^P \leq x_{max}^P$. $x_{1i,ref}^P$ is changed according to:

$$\dot{x}_{1i,ref}^P = \overbrace{-\beta_1^P (x_{1i,ref}^P - x_{max}^P) \sum_{j \in \mathcal{M}_i} |x_i^I - x_j^I|}^{\text{Pushes } x_{1i,ref}^P \text{ toward } x_{max}^P} - \overbrace{\beta_2^P (x_{1i,ref}^P - x_{1i}^P[0])}^{\text{Pushes } x_{1i,ref}^P \text{ toward } x_{1i}^P[0]} \quad (11)$$

Upon choosing proper gains for β_1^P and β_2^P , the first term of Equation (11) increases the reference when a state difference is present. The increase is capped by the upper bound x_{max}^P . Hence, the agents are moved towards the maximum position by Equation (10) and the agents move beyond x_{common}^P . This results the agents to increase the connectivity. Once the agents' states converge, the error term $\sum_{j \in \mathcal{M}_i} |x_i^I - x_j^I| \rightarrow 0$, and the second term in Equation (11) pushes $x_{1i,ref}^P$ back to its lower bound $x_{1i}^P[0]$, moving the agent back to its initial position. Once the agents retrieve from x_{common}^P , connectivity falls back to the initial level of connectivity.

4.3. Convergence Guarantees

Lemma 1. *If the union of a set of simple graphs $\{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_w\} \subset \bar{\mathcal{G}}$ has a spanning tree, then the matrix product $\mathbf{D}_w \cdots \mathbf{D}_2 \mathbf{D}_1$ is SIA, where \mathbf{D}_j is a matrix corresponding to each simple graph \mathcal{G}_j , $j = 1, \dots, w$. This proof is provided in [13].*

Lemma 2. *Let $\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_k$ be a finite set of SIA matrices with the property that for each sequence $\mathbf{S}_{i1}, \mathbf{S}_{i2}, \dots, \mathbf{S}_{ij}$ of positive length, the matrix product $\mathbf{S}_{ij} \mathbf{S}_{ij-1} \cdots \mathbf{S}_{i1}$ is SIA. Then, for each infinite sequence $\mathbf{S}_{i1}, \mathbf{S}_{i2}, \dots$ there exists a column vector \mathbf{y} such that*

$$\lim_{j \rightarrow \infty} \mathbf{S}_{ij} \mathbf{S}_{ij-1} \cdots \mathbf{S}_{i1} = \mathbf{1} \mathbf{y}^T.$$

The proof can be read in full in [34]. Lemmas 1, and 2 are, respectively, stated as lemmas 3.5, and 3.2 in [13].

Theorem 1. *Let \mathcal{G} be the communication graph of N agents. Assume each agent is connected to at least one other agent, making \mathcal{G} a spanning tree. Assume each agent i 's communication radius is r and the straight line distance between two agents i, j is d_{ij} ($d_{ij} \leq r$). Let each agent's communication frequency, $x_i^F[k] = \frac{1}{T_i[k]}$, evolve as in Equation (7). Let each agent change position according to Equations (10) and (11). Define the maximum straight line distance an agent can travel, $|x_{1i}^P[0] - x_{max}^P|$, to be within a circle of radius $\frac{r-d_{ij}}{2}$ centered around agent i . The discrete consensus algorithm in Equation (1) achieves global asymptotic consensus if there exists a non-overlapping infinite sequence of hyperperiods, denoted as $T_H[k]$, where $T_H[k] = \max\{T_i[k]\}$, and the union of communication subgraphs in $\bar{\mathcal{G}}$ has a spanning tree within each $T_H[k]$.*

Proof. Three cases are produced to analyze the convergence of Equation (1) under the variable communication frequency, which is regulated by Equation (7) [12]:

1. Synchronized and fixed T_H ;
2. Synchronous and time-varying $T_H[k]$; and
3. Asynchronous and time-varying $T_H[k]$.

The controller for agent position in Equation (11) is not directly coupled with Equation (7), however, both are indirectly coupled through the state error $|\sum_{j \in \mathcal{M}_i} (x_i^I[k] - x_j^I[k])|$. Under Equations (10) and (11), varying agent positions would add or remove connections from the connectivity matrix. Hence, added or removed connections would make $T_H[k]$ time-varying. Let \mathbf{S}_w be the matrix representation of a subgraph

in \mathcal{G} where $w = 1, \dots, W$. Define \mathbf{H}_l to be the product of any permutations of subgraph matrices at a hyperperiod l ,

$$\mathbf{H}_l = \mathbf{S}_1 \mathbf{S}_2 \mathbf{S}_3 \cdots \mathbf{S}_w.$$

From Lemma 1, \mathbf{H}_l is a SIA matrix. As all agents have communicated once at each hyperperiod $T_H[k]$, the subgraph represented by the matrix product \mathbf{H} at each hyperperiod $T_H[k]$ will have a minimum spanning tree. Let the information consensus protocol be applied for k steps in Equation (1). As $0 < T_H[k] < \infty$, when $k \rightarrow \infty, l \rightarrow \infty$. The propagation of the initial information throughout the network for k steps can be written,

$$\mathbf{x}^l[k] = \mathbf{S}_k \mathbf{S}_{k-1} \cdots \mathbf{S}_1 \mathbf{x}^l[0].$$

By substituting the matrix product at each hyperperiod with \mathbf{H} ,

$$\lim_{k \rightarrow \infty} \mathbf{x}^l[k] = \lim_{l \rightarrow \infty} \mathbf{H}_l \mathbf{H}_{l-1} \cdots \mathbf{H}_1 \mathbf{x}^l[0].$$

From Lemma 2, the product of \mathbf{H} matrixes can be substituted with,

$$\lim_{l \rightarrow \infty} \mathbf{H}_l \mathbf{H}_{l-1} \cdots \mathbf{H}_1 = \mathbf{1y}^T$$

where \mathbf{y} is some constant column vector of positive entries. Therefore,

$$\lim_{k \rightarrow \infty} \mathbf{x}^l[k] = \mathbf{1y}^T \mathbf{x}^l[0].$$

The convergence is guaranteed as long as the necessary condition is not violated by Equations (10) and (11). The necessary condition to achieve consensus is to maintain a spanning tree in the agent connectivity graph [36]. Once the distance between any two agents becomes greater than r , the agent is unreachable and the agent connectivity graph no longer maintains the spanning tree property. Equations (10) and (11) vary the agent positions between x_{max}^P and $x_{i}^P[0]$. By defining an upper bound at x_{max}^P and a lower bound at $x_{i}^P[0]$ for $x_{1i,ref}^P$ to be within a circle of radius $\frac{r - d_{ij}}{2}$, $x_{1i,ref}^P$ is restricted to move agents to unreachable locations. This preserves the spanning tree property of the connectivity graph of the agents. Therefore, by Equation (7) together with Equations (10) and (11), the system defined in Equation (1) achieves consensus. \square

5. Performance Metrics

Because co-regulation actively trades off resources and performance traditional metrics may not capture the dynamical nature of the holistic system. Here, we propose the set of performance metrics that was used to evaluate the proposed co-regulated consensus. The change in connectivity and the communication frequency among the agents change the time required for convergence and impact the converged value. In digital communication, a computational task must execute and send packets through the communication device. As a result, communication frequency is directly linked with computation and, hence, increased communication causes increased computational load. To capture holistic system performance, we present the following three cost metrics in our performance evaluation: (1) convergence time; (2) average cost of communication; and (3) error in convergence.

5.1. Convergence Time

Time to convergence depends on the rate of convergence. The fastest rate is achieved once all the agents are connected and the slowest when the agents are sparsely connected. We provide the following

definition for the convergence time, denoted as t_c , which slightly differs from the one given in [25]. t_c is defined as the first time instance after which the disagreement between the information states between agents become smaller than some ϵ

$$t_c = \left| x_i^I - x_j^I \right| \leq \epsilon, \quad i, j = 1, \dots, N.$$

Further, let T_{sim} be the length of a global clock tick used in simulation. This represents a global, synchronized time between the agents. We then define k_c to be the discrete time index at which convergence occurs:

$$k_c = \left\lfloor \frac{t_c}{T_{sim}} \right\rfloor$$

5.2. Average Cost of Communications

The number of times each agent exchanges information has a direct impact on the energy expenditure of the agent. Let C_{ij} be the impact on energy and computation from single communication between agent i and j . We calculate the cost for communications between an agent and its neighboring agents until convergence time, t_c as

$$\Omega_i = \sum_{k=1}^{k_c} \sum_{j \in \mathcal{M}_i} f(C_{ij}, k), \tag{12}$$

$$f(C_{ij}, k) = \begin{cases} C_{ij}, & \text{if } k = nT_{sim}, \quad n = 1, \dots, k_c \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

We then calculate the average cost of communication, $\bar{\Omega}$, by averaging the total cost of communication in the network over N total agents.

$$\bar{\Omega} = \frac{1}{N} \sum_{i=1}^N \Omega_i. \tag{14}$$

5.3. Average Error in Converged Value

Let $x_i^I[0]$ be the initial information state and $x_i^I[k_c]$ be the information state value of agent i at convergence time t_c . The error in converged value of agent i is calculated as

$$Z_i = \left| x_i^I[0] - x_i^I[k_c] \right|.$$

The average error in converged value is the simple average of the errors of all N agents,

$$\bar{Z} = \frac{1}{N} \sum_{i=1}^N Z_i. \tag{15}$$

6. Simulation Setup and Experimental Design

We now describe our simulation setup, design of the LQR controllers, experimental design, and provide a motivating scenario for our experiments.

6.1. Simulation Setup

MATLAB was used as the simulation environment to carry out the experiments. Agents were defined as independent objects, each with their own set of properties. Each agent had fields for position and velocity in x , y , and z coordinates, information state value, communication frequency, and initial values. We mapped the real world in a three-coordinate system in the simulation where x and y coordinates mapped the horizontal plane and z mapped the vertical plane. Forward and right movements of an agent resulted in a positive increase in x and y coordinates, respectively. Likewise, agent's upwards movement was an increase in z coordinate. The agents' motion was modeled as a second-order dynamical systems, defined in Equation (9). MATLAB's non-stiff differential equation solver, `ode45()`, was used to solve Equation (9) for x_1^P and x_2^P . Section 6.2 describes the design of the gain matrix for the LQR controller.

Each agent was assumed to be able to communicate at a variable rate bounded between $x_{i,\min}^F$ and $x_{i,\max}^F$. Zeno behavior is a phenomenon in switched system wherein a finite number of switches may occur in finite time. We imposed the bounds on communication rate, in part, to avoid this destructive behavior [37]. Table 1 lists the parameters used in the experiment and their initialized values. The discrete-time consensus algorithm in Equation (1) was independently implemented in each agent and the frequency of communication and physical position were co-regulated with the shared state by applying Equations (7) and (10). Each simulation allowed sufficient time for the shared information state to converge and the return of each agent to the initial states of frequency and position.

Table 1. Definitions and initialized value of parameters.

| Parameter | Value | Description |
|-----------------------|----------|--|
| T_{sim} | 0.01 s | Length of a global clock tick |
| ϵ | 0.001 °C | Threshold of state error for convergence |
| $x_{i,\min}^F$ | 1 Hz | Minimum communication frequency |
| $x_{i,\max}^F$ | 10 Hz | Maximum communication frequency |
| $x_i^F[0]$ | 1 Hz | Initial communication frequency |
| $x_i^P[0]$ | 15 m | Initial altitude of an agent |
| x_{\max}^P | 30 m | Max altitude allowed |
| x_{common}^P | 25 m | Common altitude for increased connectivity |

6.2. LQR Design

MATLAB's `lqr()` command was used to calculate the gain matrix \mathbf{K} for the controller in Equation (10) for each agent. `lqr()` takes the system matrix, \mathbf{A} , input matrix, \mathbf{B} , and two weighting matrices, \mathbf{Q} and \mathbf{R} , as the input parameters. For the experiments, the \mathbf{Q} and \mathbf{R} matrices were chosen such that agent position x_{i1}^P was prioritized an order of magnitude over velocity x_{i2}^P . We used the following system matrix \mathbf{A} , input matrix \mathbf{B} , and two weighting matrices \mathbf{Q} and \mathbf{R} for the experiment:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 1 \end{bmatrix}.$$

`lqr()` solved the Algebraic Riccati equation in Equation (6) with respect to the performance in Equation (4), and returned the gain matrix \mathbf{K} as in Equation (5).

6.3. Experiment Design

Figure 1 provides an overview of the experiment design. Agents sensed the temperature at predefined areas of interest (green locations in Figure 1) while in the lower orbit. The objective was to estimate,

as accurately as possible, the temperature at each location. This was accomplished via information consensus as the agents orbited around the areas of interest. Agents’ positions were co-regulated to the upper orbit (x_{common}^P) to improve connectivity, thereby sharing more information, and occurred in response to shared state error, as described in Section 4.2.

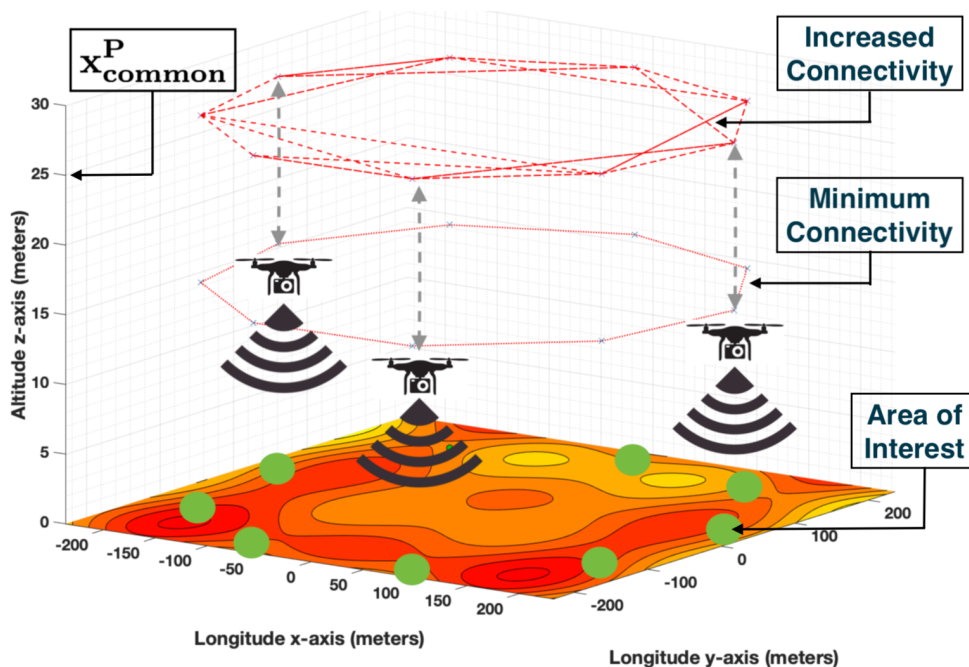


Figure 1. Overview of the experimental design.

The homogeneous agents were assumed to be capable of translational motion in three directions: x , y , and z . They were equipped with thermal imaging cameras to measure temperature on the ground at prescribed locations. Agents orbited between the D number of areas of interests and measured the temperature at each area. The environment was simulated as a coordinate system with x , y , and z dimensions and each dimension was measured in meters. Table 2 contains the x and y coordinates for $D = 9$ number of areas of interest. The temperature reading of all agents was synchronized, which required all agents to be at their respective areas of interest to take the readings at the same time instance.

The agents were initially at a low altitude, which was needed for good sensing of each area of interest. While at this low altitude, we assumed that communication was limited due to environmental obstructions and that the communication had a ring-like topology. The obstructions were avoided and connectivity could only be improved once the agents increased their altitude beyond a height x_{common}^P . Once the altitude of x_{common}^P or above was reached, the communication graph was assumed to be fully connected since an agent could communicate with all other agents. Agents initial, maximum and common altitudes for increased connectivity are listed in Table 1. Upon completing each orbit, the recorded temperature measurements of each agent were averaged with the neighboring agents per Equation (1). Each agent separately applied a Gaussian process on the shared variable to estimate the temperature field around D points.

We assumed that each of the temperature measurements through the thermal imaging cameras was subject to an error. The error was selected to be normally distributed with zero mean and a standard deviation of 5°C . The errors created a difference in the observed temperature state from agent to agent, which we addressed using our consensus algorithm.

Table 2. Predefined points of interest.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---------|----------|----------|----------|---------|---------|---------|----------|----------|----------|
| x_x^P | −187.3 m | −216.8 m | −144.7 m | −5.4 m | 136.9 m | 214.7 m | 192.6 m | 80.2 m | −70.2 m |
| x_y^P | −114.5 m | 32.8 m | 164.6 m | 219.5 m | 171.6 m | 43.6 m | −104.9 m | −204.5 m | −208.2 m |

6.4. Co-Regulation of Multiple Shared States

The co-regulating communication and the position algorithms in Equations (8) and (10) regulate the respective states based on a scaler value, x_i^I . However, in our experiment, we conducted consensus on temperature values from multiple locations. This means we must modify the co-regulation controllers in Equations (8) and (10) to work with multiple sensed values. We define \mathbf{x}_i^I to be the vector of all shared states representing temperature readings from each location. Then, Equation (8) becomes:

$$\begin{aligned}
 u_i^F[k] &= -\alpha_1^F \left| \sum_{j \in \mathcal{M}_i} \left(x_i^I[k] - x_j^I[k] \right) \right| + \alpha_2^F \left| x_i^F[k] - x_{i,\min}^F \right| \\
 &\quad \downarrow \\
 u_i^F[k] &= -\alpha_1^F \left\| \sum_{j \in \mathcal{M}_i} \left(\mathbf{x}_i^I[k] - \mathbf{x}_j^I[k] \right) \right\|_\infty + \alpha_2^F \left| x_i^F[k] - x_{i,\min}^F \right|
 \end{aligned}$$

The infinity norm selects the maximum state difference in \mathbf{x}_i^I for an agent i . The control input applied to Equation (7) and the increase in frequency are the highest when the maximum state difference is selected over any other state difference. This guarantees that an agent communicates frequently enough and transmits all the states in \mathbf{x}_i^I to its neighbors before deciding to decrease the frequency of communication.

7. Results

We now present the experiments and results depicting the effectiveness of the proposed method. We applied the proposed co-regulated consensus methods in a simulated prescribed fire. We discuss the cost savings of the proposed methods against event-triggered and traditional fixed-rate consensus strategies.

The overall behavior of the co-regulated consensus algorithm is plotted in Figure 2 for two orbits. The temperature states are plotted only for Area 5 in Table 2 (136.9 m, 171.6 m). The agents took 43 s to orbit the areas of interest once and to start communicating the individual temperature states. Agents repeated the process for each orbit. Throughout the orbit phase, the communication frequency and the vertical position were kept at the initial values. The initial values for the temperature states were the agents’ individual temperature measurements at each area. The co-regulation of communication frequency, agent connectivity, and the application of the consensus algorithm started after completing one round of orbit.

We selected $\alpha_1^F = 0.1 \text{ s}^2 \text{ }^\circ\text{C}^{-1}$ and $\alpha_2^F = 0.1 \text{ s}^3$ parameters for Equation (8) and $\beta_1^P = 1500 \text{ s}^{-1} \text{ }^\circ\text{C}^{-1}$ and $\beta_2^P = 1 \text{ s}^{-1}$ parameters for Equation (11) to simulate the system shown in the Figure 2. The difference between the individual temperature measurements and the shared temperature caused the frequency controller in Equation (8) to apply a negative control input to Equation (7). As a result, the time period to communicate, $T_i[k + 1]$, decreased, which in turns increased the frequency of the communication, as shown in the “Frequency (Hz)” plot in Figure 2. At the same time, the state error caused Equation (11) to shift the reference position towards the maximum altitude causing the controller in Equation (10) to move the agents, as shown in the “Height (m)” plot in Figure 2. Once the agents moved towards the maximum altitude, they moved past the threshold altitude for increased connectivity x_{common}^P and could communicate with a larger subset of agents. The improved connectivity allowed rapid convergence of the shared temperature state values.

As the states converged, term $\left\| \sum_{j \in M_i} (x_i^j[k] - x_j^j[k]) \right\|_{\infty} \rightarrow 0$ and the second term in Equation (7) dominated. It applied a positive control input to lengthen the time interval of successive communications and the communication frequency was lowered to its minimum. Similarly, Equation (11) pushed the reference position back to the initial altitudes. The agents lowered their altitude below x_{common}^P and reduced the connectivity to the initial ring-like topology.

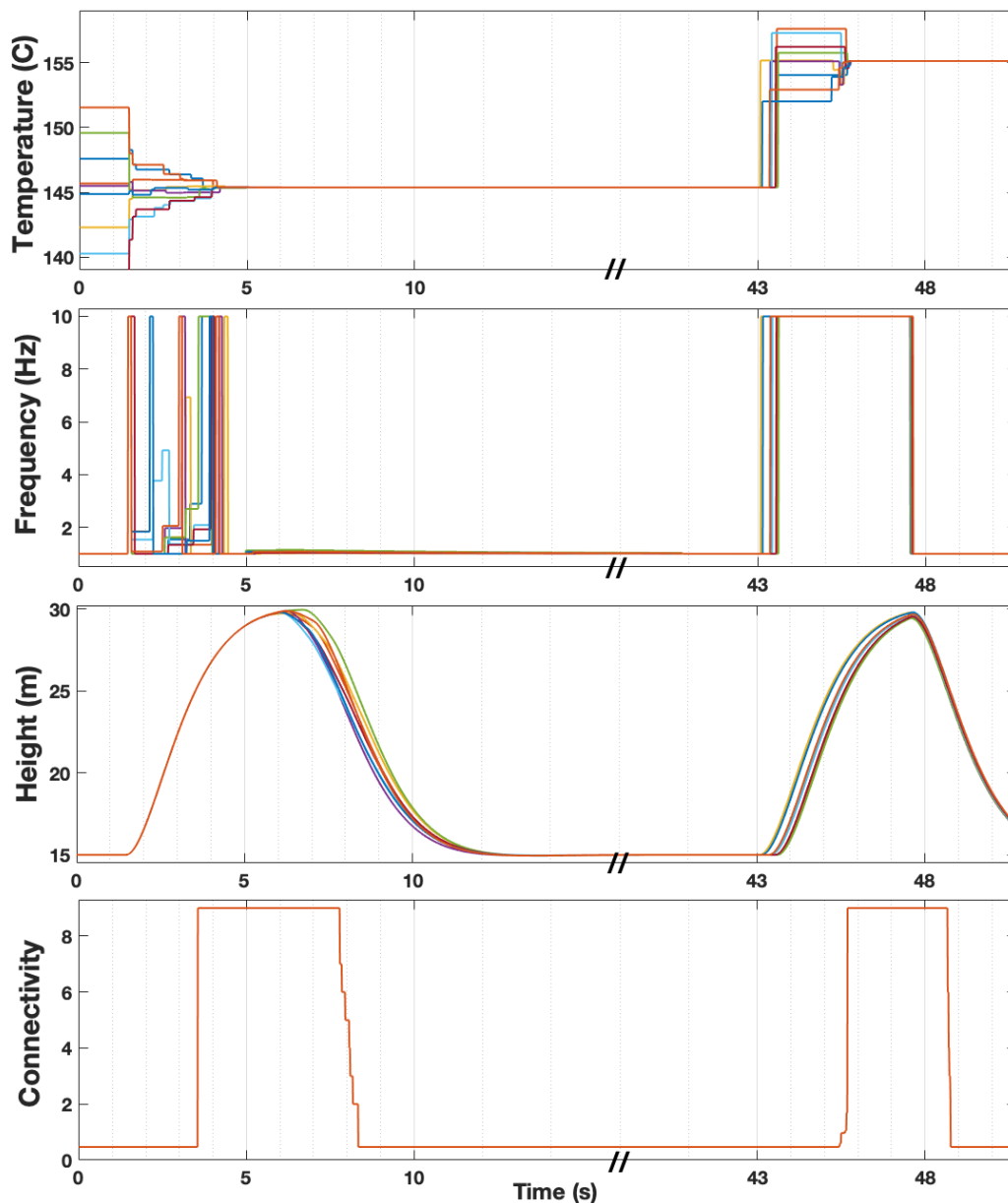


Figure 2. Behavior of the co-regulated algorithm.

Estimation Using Gaussian Process

We assumed the area of the prescribed burn was in different stages of the burn process. We assigned nine UASs to orbit around nine areas of interest, which were positioned according to Table 2. The true temperature of the fire at each area was approximated from the temperature readings provided in [6].

The approximated values of the true temperature and erroneous temperature recording through thermal imagery by each UAS are recorded in Table 3. We assumed the mean temperature of the fire over the interested area did not vary significantly during the time the UASs were in motion [6]. Upon completing the orbit, UASs calculated the average temperature across all areas of interest using consensus algorithm and applied Gaussian process regression on converged temperature values. MATLAB’s `fitrgp()` and `resubPredict()` commands were used to fit a Gaussian regression model and predict the values from the trained model. The converged and estimated temperatures readings are recorded in Table 3.

Table 3. Temperature measurements and estimations by agents across all areas of interest.

| UAS ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|------------------------|-------|--------|--------|--------|--------|--------|-------|-------|-------|
| True Temperature °C | 28.00 | 192.00 | 242.00 | 191.00 | 146.00 | 99.00 | 78.00 | 70.00 | 55.00 |
| Area 1 °C | 30.67 | 201.17 | 230.71 | 195.31 | 147.59 | 92.46 | 75.83 | 71.71 | 72.89 |
| Area 2 °C | 41.85 | 185.25 | 257.17 | 194.63 | 145.68 | 102.57 | 76.98 | 69.38 | 62.45 |
| Area 3 °C | 35.05 | 199.09 | 245.36 | 184.96 | 149.59 | 107.15 | 80.44 | 75.17 | 58.63 |
| Area 4 °C | 26.48 | 193.47 | 238.06 | 195.44 | 140.26 | 93.66 | 73.95 | 55.28 | 62.19 |
| Area 5 °C | 29.63 | 188.23 | 248.85 | 182.44 | 145.49 | 97.79 | 79.60 | 71.56 | 50.68 |
| Area 6 °C | 27.85 | 191.18 | 245.14 | 196.47 | 151.55 | 94.68 | 78.39 | 63.93 | 49.43 |
| Area 7 °C | 27.97 | 199.66 | 238.15 | 192.86 | 144.87 | 104.59 | 72.55 | 70.16 | 57.76 |
| Area 8 °C | 33.50 | 199.72 | 242.43 | 183.54 | 142.29 | 93.69 | 89.75 | 66.92 | 58.74 |
| Area 9 °C | 27.04 | 196.44 | 238.18 | 183.99 | 138.89 | 101.44 | 77.11 | 69.02 | 62.10 |
| Converged Value °C | 31.12 | 194.91 | 242.67 | 189.96 | 145.13 | 98.67 | 78.29 | 68.13 | 59.43 |
| Gaussian Estimation °C | 31.17 | 194.87 | 242.67 | 189.96 | 145.13 | 98.68 | 78.28 | 68.15 | 59.40 |

Comparison to Related Methods

It is important to find what choices for the parameters α_1^F and α_2^F yield the least cost. The cost metrics to be reduced are time to converge, average communication cost, and error in converged value (see Section 5). After multiple simulations over a range of choices for α_1^F and α_2^F parameters, $\alpha_1^F = 1 \text{ s}^2 \text{ } ^\circ\text{C}^{-1}$ and $\alpha_2^F = 10 \text{ s}^3$ resulted in the lowest cost.

These optimal choices were used to compare the proposed co-regulated method against four other methods. Table 4 shows the results of the comparisons. Our full co-regulation consensus is in the bottom row, and compared against fixed-rate consensus, two variants of our co-regulation algorithm, and event-triggered consensus. In the first row, we show event-triggered control representing the most relevant comparison due to recent advances in that area [38]. The event-triggered algorithm communicates when the error in shared state between an agent and its neighboring agents becomes greater than a threshold $5.0 \times 10^{-5} \text{ } ^\circ\text{C}$.

In the second row in Table 4, we compare against a traditional fixed-position and fixed-rate consensus algorithm. Rows 3 and 4 in Table 4 are variations of our co-regulation algorithm where we hold either position or communication rate fixed while varying the other. This provides some intuition for the tradeoffs between changing connectivity and communication rate, and the subsequent impact on our metrics.

It is clear from the results in Table 4 that the fastest convergence time was achieved through full co-regulation of both the communication frequency and the position (connectivity). The increased communication frequencies were aided by the increased connectivity to achieve faster convergence times where the time could have been $10\times$ worse had the connectivity not been increased (see Row 4 in Table 4). However, this benefit was realized at the cost of moving agents further. In the event-triggered approach, the smaller convergence time was achieved through very frequent communications (41.90 average number of communications in Row 1). In fact, event-triggered consensus communicated the most out of all methods we tried (There are many tuning parameters in event-triggered control and our implementation represents a basic event-triggered consensus strategy.). This disadvantage was avoided in

the proposed co-regulation strategies as the co-regulation allowed the communication frequency to be lowered once the state errors diminished. Interestingly, fixed-rate and fixed-position consensus scored the worst in all metrics, wasting communication resources, and taking $>10\times$ longer to converge. Row 3 of Table 4 shows a strategy with co-regulated position (connectivity) and fixed-rate communication. This strategy was only mildly worse than full co-regulation, demonstrating the power of improving connectivity to achieve fast convergence. In that case, the algorithm compensated for the fixed-rate communication by moving the agents further to improve the connectivity, and still did not manage to converge more quickly than full co-regulation. This suggests that it is likely better to increase connectivity and less aggressively increase communication frequency.

Table 4. Comparison of co-regulation against other strategies.

| | Convergence Time | Avg # of Comms | Distance Travelled |
|---|------------------|----------------|--------------------|
| Event-triggered consensus | 4.33 s | 41.90 | 0 m |
| Traditional fixed position ($u^P = 0$) and fixed-rate communication ($u^F = 0$) consensus | 40.31 s | 41.23 | 0 m |
| Co-regulated position and fixed-rate communication ($u^F = 0$) | 3.02 s | 4.00 | 311.34 m |
| Fixed position ($u^P = 0$) and co-regulated communication | 27.44 s | 33.17 | 0 m |
| Full co-regulation of x^P and x^F | 2.23 s | 5.00 | 290.10 m |

Comparison against Position-Only Co-Regulation

The results in Table 4 suggest that co-regulating agent position only to improve the connectivity provides most of the benefit of co-regulation without having to co-regulate communication rate. However, often the agents may be unable to achieve full connectivity even though they travel towards the common location. This could be due to radio interference, topographical interference, or other issues. Table 5 shows the cost of convergence and average cost of communication for agents connected with varying connectivity levels. When the agents are able to connect to only a small subset of agents at their common reference, co-regulating both the communication and position can achieve significantly shorter convergence times with smaller cost in communication compared with a fully connected network.

Table 5. Comparison of position-only co-regulation compared against full co-regulation of both communication and position with varying connectivity. CR, full co-regulation; PO, position-only co-regulation.

| | 2 Neighbors | | 3 Neighbors | | 4 Neighbors | | 5 Neighbors | | Full Connectivity | |
|--------------------|-------------|---------|-------------|---------|-------------|---------|-------------|--------|-------------------|--------|
| | PO | CR | PO | CR | PO | CR | PO | CR | PO | CR |
| Convergence Time | 39.97 s | 27.12 s | 29.31 s | 19.54 s | 22.18 s | 14.27 s | 13.03 s | 9.72 s | 3.02 s | 2.23 s |
| Communication Cost | 40.94 | 32.59 | 30.28 | 24.15 | 23.15 | 18.43 | 14.01 | 13.15 | 4.00 | 5.00 |

In short, if full connectivity cannot be achieved, improved convergence performance can be realized by additionally co-regulating the communication rate.

Error in Converged Value

The error between the converged value and the true mean of the initial temperature values were near zero in each of the methods, suggesting that consensus is indeed a good estimator in this multi-agent system. Our previous work using co-regulated communication rate [12] showed that asynchronous

communication can lead to converging to the wrong value. This occurs when an agent communicates more often than another—effectively over weighting its measurement in the consensus calculation. Here, we explore the impact of this phenomenon and how it is mitigated in our results above.

Figure 3 plots the co-regulated communication frequency using optimal alpha parameters (α_1^F and α_2^F). Synchronous and switching-like behavior of the communication frequency could be observed in all agents using this algorithm. This means once an error in shared state was observed, the frequency controller in Equation (8) pushed the communication rate to its maximum. Once the error started to decrease, it subsequently pushed it to its minimum resulting in the switching-like behavior. This occurred multiple times before convergence was obtained. This synchronous behavior resulted in convergence with minimal error.

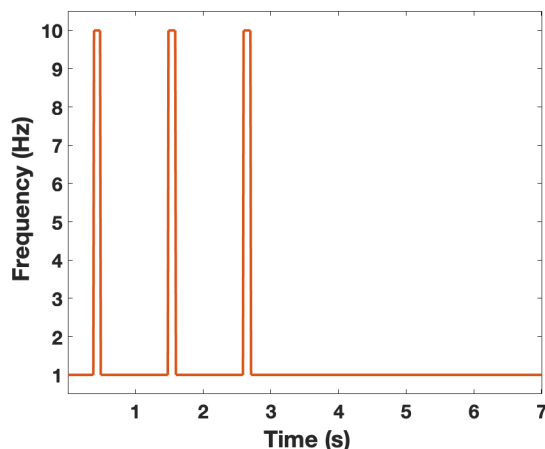


Figure 3. Communication frequency with optimal alpha parameters, $\alpha_1^F = 1$ and $\alpha_2^F = 10$.

High α_1^F and α_2^F values may not be ideal for different applications, particularly if resources are scarce, cannot be allocated as quickly, or there are communication limitations. Under lower gain values, a switching behavior may not occur, and the asynchronous behavior would lead to a mismatch between the converged value and true mean of the initial values. We show an example co-regulated communication rate response in Figure 4, which plots the communication frequency of nine agents under smaller ($\alpha_1^F = 0.05 \text{ s}^2 \text{ } ^\circ\text{C}^{-1}$ and $\alpha_2^F = 0.05 \text{ s}^3$) gains.

To provide insight into how this impacts the converged shared state, in Table 6, we show the error in converged value (calculated using Equation (15)) with varying α_1^F and α_2^F gains. As shown in the table, the error increased with decreasing α_1^F and α_2^F up to a certain point, after which the error was again reduced. This was because the system behaved more synchronously at the extreme values for the gains. This demonstrated the importance of finding optimal gain values in co-regulated multi-agent systems.

The power of our co-regulation framework is that this behavior can be adjusted using the gains α_1^F and α_2^F . These gains can be tuned to dampen, or sharpen, the response to achieve different objectives.

Table 6. Error between the converged value and the true mean of the initial conditions with varying α_1 and α_2 .

| | $\alpha_1^F (s^2 \text{ } ^\circ\text{C}^{-1})$ | 1 | 0.5 | 0.1 | 0.05 | 0.01 | 0.005 | 0.001 |
|---|---|------|-----------|-----------|-----------|-----------|-----------|-----------|
| | $\alpha_2^F (s^3)$ | 1 | 0.5 | 0.1 | 0.05 | 0.01 | 0.005 | 0.001 |
| Fixed position ($u^P = 0$) and co-regulated communication (u^F) | | 0 °C | 0.0027 °C | 0.0095 °C | 0.0218 °C | 0.0448 °C | 0.0373 °C | 0.0280 °C |
| Full co-regulation of u^P and u^F | | 0 °C | 0.0124 °C | 0.0686 °C | 0.1194 °C | 0.0861 °C | 0.1124 °C | 0.0831 °C |

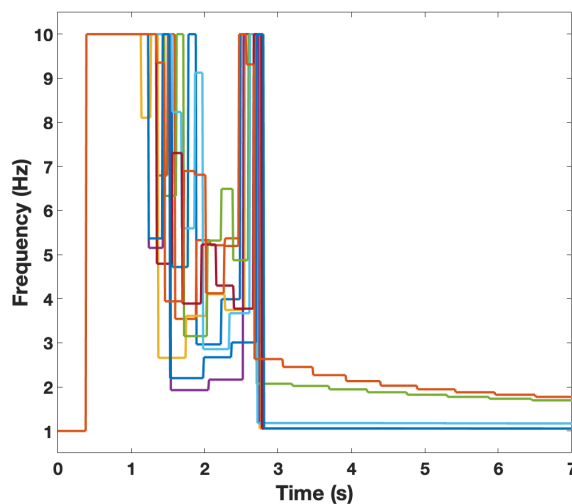


Figure 4. Communication frequency at small alpha parameters, $\alpha_1 = 0.05$ and $\alpha_2 = 0.05$.

8. Conclusions

We have developed a novel consensus scheme in which communication rate and connectivity are co-regulated alongside error in shared information state. This allows for the dynamic reallocation of computation, communication, and agent position to improve estimation of a shared state value. To do this, we introduced a new controller to move the agents to a common location to achieve higher connectivity and combined this with our previous co-regulated communication rate consensus in [12] to build a holistic consensus algorithm. New cost metrics were introduced to assess performance and comparisons against traditional fixed-rate and event-triggered consensus were shown. Our co-regulation consensus strategy is shown to have shorter convergence time with fewer numbers of communication while improving communication connectivity. Applicability of the proposed algorithm was demonstrated in a simulated environment where the agents implemented averaging co-regulated consensus to calculate the mean temperature of a prescribed fire.

Future directions of this work include analyzing the performance of a larger number of co-regulated agents, exploring optimization strategies to better tune the controller gains and meet new performance objectives, and implementing the co-regulated algorithms in real prescribed fire UAS-Rx [8].

Author Contributions: Conceptualization, C.F., C.D. and J.B.; Funding acquisition, C.D. and J.B.; Resources, C.D. and J.B.; Software, C.F.; Writing—original draft, C.F.; and Writing—review and editing, C.D. and J.B.

Funding: This research was funded by USDA-NIFA under grant number 2017-67021-25924 and NSF under grant number IIS-1638099.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, B.; Moridian, B.; Kamal, A.; Patankar, S.; Mahmoudian, N. Multi-robot mission planning with static energy replenishment. *J. Intell. Robot. Syst.* **2018**, 1–15. [[CrossRef](#)]
2. Lee, E.A. The past, present and future of cyber-physical systems: A focus on models. *Sensors* **2015**, *15*, 4837–4869. [[CrossRef](#)]
3. Åström, K.J.; Wittenmark, B. *Computer-Controlled Systems: Theory and Design*; Prentice-Hall: New York, NY, USA, 1984.
4. Franklin, G.F.; Powell, J.D.; Emami-Naeini, A. *Feedback Control of Dynamics Systems*; Pearson Higher Education: London, UK, 2010.
5. Bradley, J.; Atkins, E. Coupled Cyber-Physical System Modeling and Coregulation of a CubeSat. *IEEE Trans. Robot.* **2015**, *31*, 443–456. [[CrossRef](#)]
6. Kennard, D.K.; Outcalt, K.W.; Jones, D.; O'Brien, J.J. Comparing techniques for estimating flame temperature of prescribed fires. *Fire Ecol.* **2005**, *1*, 75–84. [[CrossRef](#)]
7. Twidwell, D.; Allen, C.R.; Detweiler, C.; Higgins, J.; Laney, C.; Elbaum, S. Smokey comes of age: Unmanned aerial systems for fire management. *Front. Ecol. Environ.* **2016**, *14*, 333–339. [[CrossRef](#)]
8. Beachly, E.; Detweiler, C.; Elbaum, S.; Duncan, B.; Hildebrandt, C.; Twidwell, D.; Allen, C. Fire-Aware Planning of Aerial Trajectories and Ignitions. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 685–692.
9. Beachly, E.; Detweiler, C.; Elbaum, S.; Twidwell, D.; Duncan, B. UAS-Rx Interface for Mission Planning, Fire Tracking, Fire Ignition, and Real-Time Updating. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; pp. 67–74. [[CrossRef](#)]
10. Ren, W.; Beard, R.W.; Atkins, E.M. Information consensus in multivehicle cooperative control. *IEEE Control Syst.* **2007**, *27*, 71–82.
11. Garcia, E.; Cao, Y.; Casbeer, D.W. Decentralized Event-Triggered Consensus with General Linear Dynamics. *Automatica* **2014**, *50*, 2633–2640. [[CrossRef](#)]
12. Fernando, C.; Detweiler, C.; Bradley, J. Co-Regulating Communication for Asynchronous Information Consensus. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018; pp. 6994–7001.
13. Ren, W.; Beard, R.W. Consensus of Information under Dynamically Changing Interaction Topologies. In Proceedings of the IEEE American Control Conference, Boston, MA, USA, 30 June–2 July 2004; Volume 6, pp. 4939–4944.
14. Moreau, L. Stability of multiagent systems with time-dependent communication links. *IEEE Trans. Autom. Control* **2005**, *50*, 169–182. [[CrossRef](#)]
15. Jadbabaie, A.; Lin, J.; Morse, A.S. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control* **2003**, *48*, 988–1001. [[CrossRef](#)]
16. Vicsek, T.; Czirók, A.; Ben-Jacob, E.; Cohen, I.; Shochet, O. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.* **1995**, *75*, 1226. [[CrossRef](#)]
17. Cao, Y.; Yu, W.; Ren, W.; Chen, G. An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination. *IEEE Trans. Ind. Inform.* **2013**, *9*, 427–438. [[CrossRef](#)]
18. Dorfler, F.; Pasqualetti, F.; Bullo, F. Continuous-Time Distributed Observers with Discrete Communication. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 296–304. [[CrossRef](#)]
19. Ren, W.; Cao, Y. Convergence of Sampled-data Consensus Algorithms for Double-integrator Dynamics. In Proceedings of the 2008 47th IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; pp. 3965–3970.
20. Lin, P.; Jia, Y. Consensus of Second-Order Discrete-Time Multi-Agent Systems with Nonuniform Time-Delays and Dynamically Changing Topologies. *Automatica* **2009**, *45*, 2154–2158. [[CrossRef](#)]
21. Xiao, F.; Wang, L. Asynchronous Consensus in Continuous-Time Multi-Agent Systems with Switching Topology and Time-Varying Delays. *IEEE Trans. Autom. Control* **2008**, *53*, 1804–1816. [[CrossRef](#)]

22. Krishna, C.M.; Shin, K.G. *Real-Time Systems*; Tata McGraw-Hill Education: New York, NY, USA, 1997.
23. Saber, R.O.; Murray, R.M. Consensus protocols for networks of dynamic agents. In Proceedings of the 2003 American Control Conference, Denver, CO, USA, 4–6 June 2003; pp. 951–956.
24. Fang, L.; Antsaklis, P.J. Information consensus of asynchronous discrete-time multi-agent systems. In Proceedings of the American Control Conference, Portland, OR, USA, 8–10 June 2005; pp. 1883–1888.
25. Olshevsky, A.; Tsitsiklis, J.N. Convergence speed in distributed consensus and averaging. *SIAM J. Control Optim.* **2009**, *48*, 33–55. [[CrossRef](#)]
26. Olfati-Saber, R.; Murray, R.M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* **2004**, *49*, 1520–1533. [[CrossRef](#)]
27. Hug, G.; Kar, S.; Wu, C. Consensus+ Innovations Approach for Distributed Multiagent Coordination in a Microgrid. *IEEE Trans. Smart Grid* **2015**, *6*, 1893–1903. [[CrossRef](#)]
28. Konak, A.; Buchert, G.E.; Juro, J. A flocking-based approach to maintain connectivity in mobile wireless ad hoc networks. *Appl. Soft Comput.* **2013**, *13*, 1284–1291. [[CrossRef](#)]
29. Francolin, C.; Rao, A.; Duarte, C.; Martel, G. Optimal control of an autonomous surface vehicle to improve connectivity in an underwater vehicle network. *J. Aerosp. Comput. Inf. Commun.* **2012**, *9*, 1–13. [[CrossRef](#)]
30. Simonetto, A.; Keviczky, T.; Babuška, R. On distributed maximization of algebraic connectivity in robotic networks. In Proceedings of the American Control Conference, San Francisco, CA, USA, 29 June–1 July 2011; pp. 2180–2185.
31. Ajorlou, A.; Momeni, A.; Aghdam, A.G. A class of bounded distributed control strategies for connectivity preservation in multi-agent systems. *IEEE Trans. Autom. Control* **2010**, *55*, 2828–2833. [[CrossRef](#)]
32. Ren, W.; Beard, R.W.; Atkins, E.M. A Survey of Consensus Problems in Multi-Agent Coordination. In Proceedings of the 2005 American Control Conference, Portland, OR, USA, 8–10 June 2005; Volume 3, pp. 1859–1864. [[CrossRef](#)]
33. Horn, R.A.; Johnson, C.R. *Matrix Analysis*; Cambridge University Press: Cambridge, UK, 1990.
34. Wolfowitz, J. Products of indecomposable, aperiodic, stochastic matrices. *Proc. Am. Math. Soc.* **1963**, *14*, 733–737. [[CrossRef](#)]
35. Kirk, D.E. *Optimal Control Theory: An Introduction*; Dover Publications: New York, NY, USA, 2004.
36. Ren, W.; Beard, R.W. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Trans. Autom. Control* **2005**, *50*, 655–661. [[CrossRef](#)]
37. Fan, Y.; Liu, L.; Feng, G.; Wang, Y. Self-Triggered Consensus for Multi-Agent Systems With Zeno-Free Triggers. *IEEE Trans. Autom. Control* **2015**, *60*, 2779–2784. doi:10.1109/TAC.2015.2405294. [[CrossRef](#)]
38. Li, X.; Hirche, S. Event-Triggered Consensus of Multi-Agent Systems on Strongly Connected Graphs. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018; pp. 1311–1316.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).