Co-Regulated Information Consensus with Delays for Multi-Agent UAS*

Chandima Fernando[1], Carrick Detweiler[1], and Justin Bradley[1]

***Abstract*— Consensus algorithms provide a framework for the distributed coordination of a multi-agent system. However, widespread application and deployment of consensus algorithms may be limited in real-world multi-agent coordination problems due to implementation on size, power, and weight constrained vehicles. In this case, limited resources may contribute to delay and packet loss causing algorithm deterioration and violation of performance guarantees. This calls for novel strategies for intelligent resource utilization and computationally simple implementation. Towards this goal, we propose co-regulation strategies for discrete time average consensus under delays allowing dynamic resource utilization while coping with communication limitations. This is done by dynamically adjusting communication frequency to facilitate higher state exchange rates while simultaneously adjusting agents' locations to increase inter-agent connectivity for rapid convergence. We prove that convergence is still guaranteed for co-regulation strategies for discrete time average consensus under bounded delays. In addition, we propose a pause for agents' locations to mitigate adverse behavior caused by delay. To simplify implementation we devise a consensus strategy that decouples the co-regulated consensus from low-level vehicle feedback control. The usability of our proposed system is evaluated through a series of simulations, and we show our proposed co-regulation strategies in fact result in faster convergence time. We evaluate the approach with an outdoor experiment using 4 customized unmanned aircraft systems (UASs).**

I. Introduction

Consensus algorithms are decentralized algorithms that are used to compute weighted averages of values between multiple agents. This weighted average has been used as an estimation strategy [1], collective, multi-agent controller [2], and clock synchronization scheme [3] amongst other things. Much of the research in this area focuses on guarantees about convergence, time to convergence, and the differences associated with time-triggered and event-triggered formulations [4]. While consensus algorithms have been deployed on indoor robots [5] and on unmanned aircraft systems (UASs) [6], [7], additional research is needed into the practical aspects needed for deployment. This includes handling delay, integrating motion with communication, and careful coordination between the higher- and lower-level control systems.

In this paper, we propose consensus-based control algorithms that dynamically adjust each agent's physical position and communication frequency, which we consider resources, according to collective consensus performance. Co-regulating resources also changes the inherent inter-agent communication delay that arises in real-world communication. While performance guarantees that ignore this delay under ideal conditions can be computed, overlooking delay can have significant consequences in real-world systems [8]. Communication delay in a consensus algorithm causes an agent to update its state based on obsolete states of its neighbors. Our co-regulation strategy dynamically adjusts the frequency proportional to the difference in information state. Convergence of states is further assisted by dynamically changing the physical position of agents to improve the connectivity of the system. This has the benefit of achieving reduced times in state convergence. Co-regulated consensus has many of the same advantages of event-triggered consensus [9], including minimizing communication resources, but, in contrast, allows for easier analysis by maintaining a time-triggered architecture, albeit a time-varying one. These advantages are primarily realized when applied and implemented in real-world applications because the time-triggered architecture allows the use of existing real-time operating system strategies, guarantees, and robustness in implementation [10].

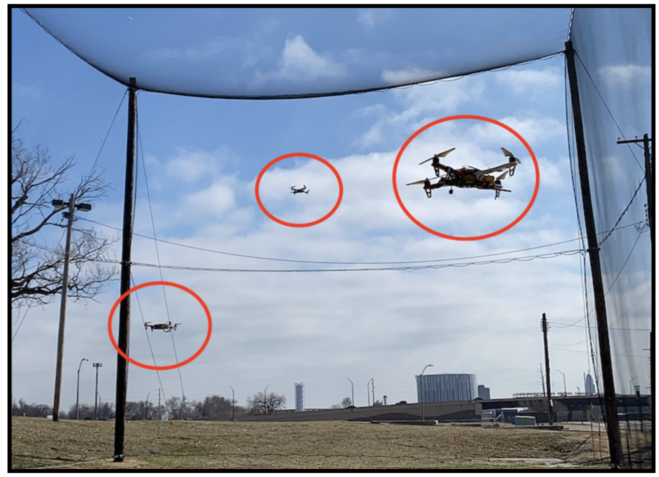

Fig. 1: Distributed agents apply co-regulated consensus strategies in an outdoor experiment.

To ease the design of co-regulated consensus algorithms and their corresponding implementation across non-homogeneous vehicles, we develop a tiered architecture that decouples consensus control from low-level control. Figure 2 provides an overview of the proposed architecture. We prove both convergence and stability of this tiered architecture.

In our prior work, we provided proofs of convergence and rates-of-convergence under dynamic communication frequency and agent movements which change connectivity [11].

*This work was supported in part by USDA-NIFA 2017-67021-25924, NSF IIS-1638099, NSF IIS-1925052, and NSF IIS-1925368.

[1]Department of Computer Science & Engineering, University of Nebraska–Lincoln, Lincoln, NE 68588, USA {wfernando2, carrick, jbradley}@cse.unl.edu

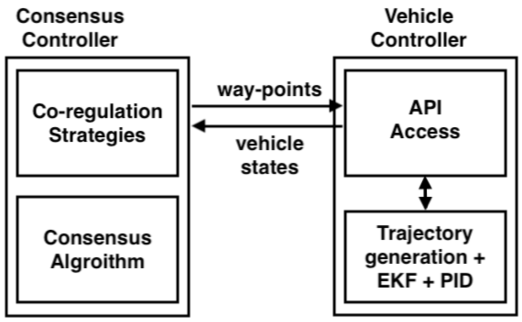

Fig. 2: Two-tier architecture for consensus implementation alongside traditional vehicle controller

In this work, we extend our approach and performance guarantees to incorporate communication delay and implement it on a team of small UASs (see Figure 1) for potential use in surveillance, scientific, monitoring, and other applications. The key contributions of this work are:

- Theoretical guarantees on the convergence of co-regulation consensus algorithms under co-regulation of communication frequency and physical position with delays;
- A tiered consensus architecture to decouple co-regulated consensus control from a vehicle's low-level feedback control to simplify implementation; and
- Evaluation and demonstration of our approach on a real UAS testbed with outdoor flights results.

II. Related Work

Consensus for leaderless coordination among information-exchanging agents provide early examples of distributed averaging [12], [13]. Each agent's update in state and subsequent information exchange between agents results in further reduction of state value differences to converge on single state value - consensus. We use a model similar to the "state update model" in [13], [14]. Update models that exchange observations on a random variable are known as information consensus problems [13]. We propose co-regulating controllers that utilize state information to form an information consensus problem and achieve an adjustable level of performance in state convergence while minimizing resource utilization of a delay-affected, discrete-time system.

Digital communication devices used to implement such algorithms are inherently discrete [15], suited more to discrete consensus algorithms to develop regularization strategies [16]. Our work changes the controller rate and communication rate which makes the distributed agents asynchronous [17] and variable in communication topology. We consider a connected communication topology between agents with sufficiently frequent communications to keep the network connected [13], [18].

Communication delay is a significant roadblock consensus algorithms should account for. Work on convergence and stability properties in continuous time consensus affected by delay was done using Lyapunov-Krasovskii techniques through LMI [19], [20], and furthered for second order systems under delay [21]. The discrete-time system under delay is can be analyzed using a matrix representation of the communication topology [22], [17]. However, the discrete-time algorithm we use dynamically adjusts the duration of the discrete time step in accordance with state difference [11]. While we have shown this to be effective under ideal conditions we have not considered communication time delay - a necessary component for real-world deployment and one of the objectives of this paper.

Multi copters are a compelling platform for deploying consensus algorithms due to their rapid maneuverability, payload capabilities, and application possibilities. Consensus algorithms proposed in [23], [24] use three multi copters as a testbed. Research-grade multi copters often communicate through XBee modules[1] and use location consensus waypoints for planning for the agents while a distributed, low-level controller computes trajectories for each multi copter to achieve the consensus waypoint. Such strategies have also been augmented with a decentralized model predictive controller for flocking of a group of multi copters in outdoor experimentation [7]. To cleanly implement such algorithms, research in [7] proposes a two-layer architecture: a coordination layer to compute consensus values and a flight control layer for the multi copter navigation. Our work follows a two-layer architecture to separate consensus computations from low-level flight control similar to that in [7].

III. Background

Graphs are both suitable and highly useful to map a network of inter-communicating agents. A node in the graph represents an agent, and an edge between two nodes represents a communication instance between two agents. Matrix representation of a such graph can be analyzed to prove the convergence properties of the consensus algorithm.

A. Graph Theory in Consensus

Let a set of N agents be represented in a N node graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a mapping of the set of nodes $\mathcal{V} = \{v_1, v_2, .., v_N\}$ to N agents and the set of edges $\mathcal{E}$ to agent communications. Single communication instance between two agents in $\mathcal{E}$ is denoted by $\varepsilon_{ij} = (v_i, v_j)$. A directed edge in $\mathcal{E}$ represents one way communication [8].

The equivalent adjacency matrix representation of the graph $\mathcal{G}$ is denoted by $\mathcal{A} = [a_{ij}]$, where

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

A weighting factor w_{ij} can be assigned to each edge to denote communication strength or the trustworthiness of states between two agents i and j.

A communication subgraph is a graph with N nodes and a set of edges that maps the communications between N agents at a given time instance. For a N agents system, let $\bar{\mathcal{G}} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_M\}$ be the finite set of all possible communication subgraphs.

[1] https://www.digi.com/xbee

B. Matrix Theory for Consensus

A matrix is known as a non-negative matrix if all the entries are greater than or equal to zero, $\mathbf{L} \geq 0$. A nonnegative matrix becomes a *stochastic* matrix if raw sum of the non-negative matrix equals to one [25]. A stochastic matrix $\mathbf{L}$ is known as *stochastic indecomposable aperiodic (SIA)* if $\lim_{m\to\infty} \mathbf{L}^m = \mathbf{1y}^T$ [26]. $\mathbf{1}$ represents a column vector of size $n \times 1$ with all the entries equal to one.

C. Delayed Discrete Information Consensus

Agents i and j are neighbors at a given time instant if $\varepsilon_{ij} \in \mathcal{E}$. For any given agent i, we use M_i to denote the set of agent i's neighbors. We assume each agent shares its state with itself, hence, the set M_i includes agent i.

We define the information state of the agents as $x^I = (x_1^I, x_2^I, \cdots, x_N^I)^T$. Agents share x^I among themselves and calculate the common consensus value individually. Let K be the global discrete clock indexed as $K \in \{1, 2, 3, \ldots\}$. Let τ_{ij} be the communication time delay, which is discrete, on a state transmitted from agent i to agent j, and $\tau_{ii} = 0$ (i.e., an agent has instant access to its own state). Assuming digital communication, we consider communication to be a discrete event and use the discrete-time consensus algorithm [13] extended to represent communication delay as in [22]:

$$x_i^I[k+1] = \frac{1}{\sum_{j\in M_i} w_{ij}} \sum_{j\in M_i} w_{ij} x_j^I[k-\tau_{ij}] \quad i = 1, \cdots, N \quad (1)$$

where $k \in K$, and $k - \tau_{ij}$ represents a discrete time index prior to the k^{th} time index. Therefore, $x_j^I[k - \tau_{ij}]$ is the information state value of agent j at $[k - \tau_{ij}]^{th}$ discrete time index. Let τ_{max} be the maximum value of all such communication delays such that for any $\tau_{ij} \leq \tau_{max} < \infty$. If the communication delay in each agent varies with time, τ_{max} varies respectively. Large communication delays increase time to converge thereby reduce the system performance. Therefore, knowing τ_{ij} helps with algorithm design but it is not a necessity.

For a given set of initial conditions $x^I[0]$ [1], the system (1) achieves consensus once

$$\lim_{k\to\infty} \left\| x_i^I[k] - x_j^I[k] \right\| \to 0, \quad i, j = 1, \cdots, N. \quad (2)$$

When edge weights are equal, and time delays are equal or non-existent, Equation (1) averages the initial state values - known as the average consensus problem, $\lim_{k\to\infty} \left(x^I[k]\right) = \frac{1}{n} \sum_{i=1}^N x_i^I[0]$.

IV. Co-Regulated Consensus with Delay

Here we discuss the derivation of our co-regulation control strategies for both information, and position consensus. We prove convergence properties of the controllers under delay and discuss the impacts of our strategy in improving performance in state convergence.

A. Information Consensus

We implement the delay-adjusted information consensus algorithm in Equation (1) with $w_{ij} = 1 \quad \forall i, j = 1, \cdots, N$. Each agent exchanges its information state in a communication step and subsequently calculates the average of received information states. We assume that delay, τ_{ij}, is primarily dictated by communication delay ($\tau_{ij,\text{comm}}$), and that nonzero computation time ($\tau_{i,\text{comp}}$) is significantly smaller (e.g., $\tau_{ij,\text{comm}} \gg \tau_{i,\text{comp}} \ \forall i, j$).

B. Communication Frequency Co-Regulation under Delay

Depending on the implementation, computation of the average information state value and exchange of states between agents could account for a large portion of resources available in size, weight, and power (SWaP) constrained system. This could be due to inefficient polling for new information, transmission power settings, or overly conservative sampling rates. Hence, while computationally simple, a consensus controller executing at a fixed, high rate may severely limit available resources that can be allocated to other critical processes (e.g., machine learning, perception). In contrast, a low computation and communication rate drastically hinders holistic multi-agent system performance. In our proposed co-regulation strategy increased resources are allocated to computation and communication when there exists a large difference in information state value between the agents and reduced resources are allocated when the difference in information state value diminishes. We make direct use of our communication frequency controller, x_i^F, provided in our previous work [27] but augment our performance guarantees to account for communication delay. From [27],

$$u_i^F[k] = \overbrace{-\alpha_1^F \left|\Sigma_{j\in\mathcal{M}_i} \left(x_i^I[k] - x_j^I[k-\tau_{ij}]\right)\right|}^{\text{Pushes comm rate toward } x_{i,\max}^F} + \underbrace{\alpha_2^F \left|x_i^F[k] - x_{i,\min}^F\right|}_{\text{Pushes comm rate towards } x_{i,\min}^F}. \quad (3)$$

Communication frequency is modeled as a discrete time variable with $x_i^F[k] = 1/T_i[k]$, Where $T_i[k]$ is the time period between two successive communications of agent i. We allow $T_i[k]$ to evolve according to the discrete-time equation,

$$T_i[k+1] = T_i[k] + x_i^F[k] u_i^F[k] \quad (4)$$

controller by a designed forcing function, u_i^F. Now we provide convergence guarantees of the delay affected system in Equation (1) and Equation (3) by extending the Theorem 4.4 which proves convergence of Equation (1) regulated by Equation (3) in [27]. Theorem 4.4 does not account for the delay in the system. We re-write Theorem 4.4 as Lemma 4.1.

Lemma 4.1: Let $\mathcal{G}$ be a communication graph with N nodes, each representing an agent. Let $\bar{\mathcal{G}} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_M\}$ be the finite set, cardinality M, of all possible communication subgraphs between N agents. Let each agent's communication frequency, $x_i^F[k] = 1/T_i[k]$, evolve as in Equation (4). That is, an agent, i, sends a communication of its shared state variable every $T_i[k]$ time units. The discrete consensus algorithm in

Equation (1) achieves global asymptotic consensus if there exists a non-overlapping infinite sequence of hyper periods, 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 $T_H[k]$. Such a condition is met as long as every agent's communication rate is above 0 Hz.

Theorem 4.2: Let $\mathcal{G}$ be an N-agent communication graph. Let $\bar{\mathcal{G}} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_M\}$ represents the finite set of all possible communication subgraphs of the N-agent system. Each agent, i, shares its state variable at every $T_i[k]$ time units while $T_i[k]$ evolves according to Equation (4). Let τ_{ij} be the communication delay between agent i and j. Then the delayed discrete consensus algorithm in Equation (1) achieves global asymptotic consensus if there exists a non-overlapping infinite sequence of hyper periods where each hyper period, $T_H[k] = \max\{T_i[k] + \tau_{ij}\}$, and the union of corresponding communication subgraphs from $\bar{\mathcal{G}}$ at each hyper period contains a spanning tree. Zeno behavior is avoided as long as every agent's communication frequency is bounded between $x^F_{i,max}$ and $x^F_{i,min}$.

proof: See Appendix I.

Theorem 4.2 proves that bounded communication delay does not affect state convergence even under our time-varying controller and communication frequency. However, average consensus is not guaranteed except when communication and controller frequency is synchronized and communication delay is identical for all agents. It is important to note this exception likely does not hold true for real-world systems, and hence we cannot guarantee the average consensus value will be the convergence value. This is a limitation of real-world-deployed consensus analogous to how noise prohibits true state estimation in other methods. However, in our experience, under normal operation, communication frequency and delay are typically close to ideal conditions and the consensus value is close to the average.

C. Co-regulated Position Consensus under Delay

Higher connectivity in a multi-agent network typically results in faster convergence speeds. Here, we move agents to a common location to improve agent connectivity due to limitations in the range of their wireless communication devices. In [11], we introduced a consensus controller to change the altitude of a set of agents to improve connectivity and consensus performance. In that work the common altitude that improved connectivity was pre-determined. Here, we replace x^I_i, the information state in Equation (1), with agents' physical positions, $x^P_{i,cons}$, to dynamically derive the common location. Communication delay has the unfortunate effect of causing agents to update their state with outdated position information leading to leaving to possible error in the consensus position value. We define variable $x^P_{i,cons} = [x^P_{ix,cons}, x^P_{iy,cons}]$ to be the consensus position state along agents latitude and longitude directions. The position consensus algorithm can be defined formally as

$$x^P_{i,cons}[k+1] = \frac{1}{\sum_{j\in M_i} w_{ij}} \sum_{j\in M_i} w_{ij} x^P_{j,cons}[k-\tau_{ij}], \quad (5)$$

where $i = 1, \cdots, N$ and $k \in \{1, 2, 3, \ldots\}$ is the discrete time index. We define the initial value, $x^P_{i,cons}[0]$, to be the initial physical location of agent i.

D. Tiered Architecture

To support our tiered architecture, we introduce a reference point to decouple the consensus controller from the lower-level control of the agent. Before discussing the details of the reference point, we start by defining the dynamics of the agents. Agents are modeled with second order dynamics,

$$\dot{x}^P_{1i} = x^P_{2i}, \qquad \dot{x}^P_{2i} = u^P_i. \quad (6)$$

x^P_{1i}, x^P_{2i}, and u^P_i are position, velocity, and control input for agent i respectively. Furthermore,

$$x^P_{1i} = \begin{bmatrix} x^P_{1ix} \\ x^P_{1iy} \end{bmatrix} \qquad x^P_{2i} = \begin{bmatrix} x^P_{2ix} \\ x^P_{2iy} \end{bmatrix}$$

accounting for position and velocity in the x and y directions (e.g., longitude and latitude).

Connectivity is improved only if the agents move to the consensus position calculated in Equation (5). Once the agents achieve information state consensus, improved connectivity has little advantage and hence agents move back to the lower-connectivity positioning they were in to resume individual tasks. To do this, we introduce a reference point for each agent, which is used as a way-point for an agent's low-level controller. We model the reference point as a discrete-time first order system. A reference point is defined, $x^P_{1i,ref} = [x^P_{1ix,ref}, x^P_{1iy,ref}]^T$. The change in reference point is given by the discrete-time equation:

$$\begin{aligned} x^P_{1i,ref}[k+1] = & \overbrace{-\beta^P_1\left(x^P_{1i,ref}[k] - x^P_{cons}[k]\right)}^{\text{Pushes } x^P_{1i,ref} \text{ toward } x^P_{i,cons}} \\ & * \Sigma_{j\in\mathcal{M}_i}\left|x^I_i[k] - x^I_j[k-\tau_{ij}]\right| \\ & - \overbrace{\beta^P_2\left(x^P_{1i,ref}[k] - x^P_{1i}[0]\right)}^{\text{Pushes } x^P_{1i,ref} \text{ toward } x^P_{1i}[0]}. \end{aligned} \quad (7)$$

The first term of Equation 7 pushes the reference point of each agent towards the consensus position $x^P_{i,cons}$. The term calculating the difference in information state acts both as a gain to the control input and a switch to the movement towards $x^P_{i,cons}$(An agent is moved when there exists a difference in information states). Once agents converge on information state it is no longer desirable to move towards the consensus position and the first term decays while the second term pushes agents toward their initial position. We now provide a guarantee of convergence of our co-regulated position consensus.

Theorem 4.3: Let $\mathcal{G}$ be the communication graph for a multi-agent system with N agents. Assume the agents in $\mathcal{G}$ can at least communicate with one agent in the system and maintain the connectivity at all the time. This ensures $\mathcal{G}$ contains a spanning tree. Communication time period between agents and communication frequency are related by $x^F_i[k] = \frac{1}{T_i[k]}$, and the communication frequency is changed as in Equation (4). Let agents' positions be regulated by

Equation (7). Then the multi-agent system under position consensus algorithm in Equation (5) achieves a consensus position if there exists an infinite sequence of hyper periods $T_H[k]$ with $T_H[k] = \max\{T_i[k] + \tau_{ij}\}$, and the connectivity subgraph at each hyper period contains a spanning tree.

proof: See Appendix II.

V. Simulations

In this section we analyze and discuss the behavior and effectiveness of the position consensus algorithm under various conditions in simulation. But first we start by giving an overview of our simulation environment.

A. Simulation Setup

We model agents as independent objects in a MATLAB simulation environment. Agents are free to move horizontally with a mapping of the world in longitude to the x-axis and latitude to the y-axis. On a multi-agent system of 6 agents, we fix the altitude of all agents to $5\,\mathrm{m}$ above ground level. We model agent motion as in Equation (6). We use `ode45()` in MATLAB to solve Equation (6). Agents are capable of communicating their states to other agents at a variable frequency from $0.1\,\mathrm{Hz}$ to $1\,\mathrm{Hz}$. Equation (4) implements frequency co-regulation with $\alpha_1 = 1$ and $\alpha_2 = 5$. We initialize each agent with a random information state value, $[-5, 5]$, and set the sum of states to zero. Initially, agents are positioned 5.1 meters from each other in a ring configuration so that they can only communicate with their immediate neighbor. We set w_{ij} to be 1 for every agent. Delay in state communication in each agent is assumed to be time varying, and modeled by the function $y = 3 + 3sin(ik)$ where i is the agent index in simulation and k is the discrete time index. We implement Equation (7) with gains of $\beta_1 = 10$ and $\beta_2 = 15$.

We let each simulation run until the agents converge on the information state and finish their movements by reaching the initial positions. We use 0.1 to be the desired convergence value in the information state. We simulate the global clock at a tick length of $0.1\,\mathrm{s}$.

B. Behavior of the Position Consensus

Figure 3 plots information state convergence, communication frequency, connectivity of the network, convergence in consensus position on the x-axis, the reference point for agents' movement on the x-axis, and actual positions of agents on the x-axis. When agents receive information from their neighbors, they increase communication frequency depending on the difference in information state error. Similarly, agents distributively calculate x^P_{cons} and x^P_{Ref} that tracks x^P_{cons}. This causes the agents to move towards x^P_{cons}. Being closer to each other increases system connectivity. Information state difference vanishes rapidly because of improved connectivity and increased communication frequency. Note that the delay in agents prevents them from achieving a single step convergence even when connectivity is at its highest. When information state difference vanishes, Equation (4) and Equation (7) reduce communication frequency and move the reference and physical states towards initial starting positions.

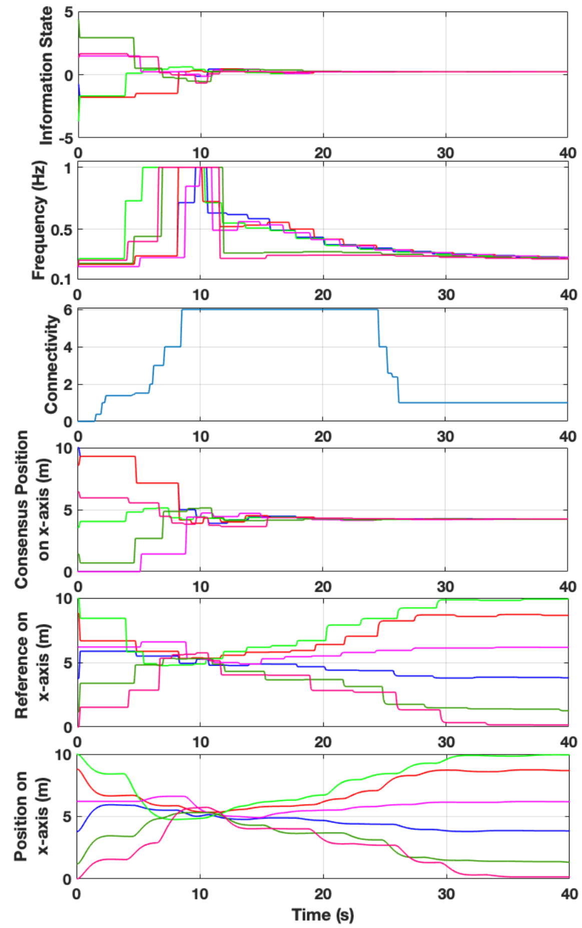

Fig. 3: Top to bottom : Agents' information state; All agent communication frequency change; All agent connectivity change; All agent x^P_{cons} on x-axis; All agent x^P_{ref} on x-axis; All agent x^P_1 on x-axis.

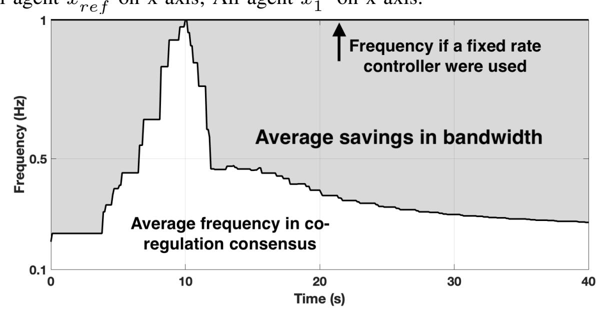

Fig. 4: Low utilization in bandwidth as a result of variable frequency by co-regulation technique

Figure 4 plots the savings in communication bandwidth in using co-regulation method as opposed to using a fixed rate controller.

Figure 5 plots convergence of reference point x^P_{Ref} and physical location x^P_1 towards the consensus position x^P_{cons}. Initially, asynchronous agents connect at minimum connectivity and minimum frequency of communication. Hence, the calculation of the consensus position x^P_{cons} evolves and achieves consensus only after subsequent communications. Asynchronous clocks and nonidentical time delays of the agents may trigger some agents to act prior to the others. The result is a zigzag path for the consensus position. The reference point that tracks the consensus position takes a

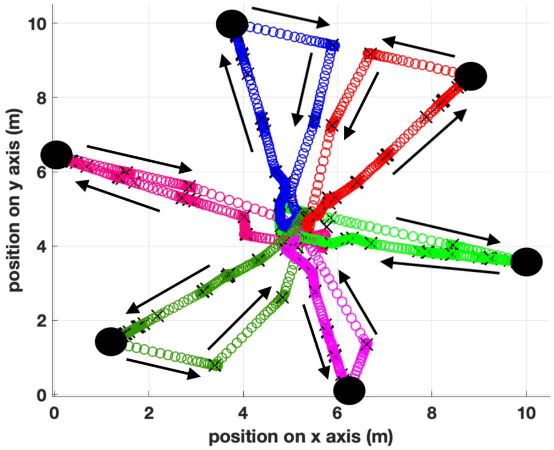

Fig. 5: Paths of x^P_{ref} and x^P on x and y axis over the complete simulation time period are denoted by $\times$ and $\circ$ respectively. Black circles mark the initial agent positions and arrows point the directions that the agents moved.

non-straight line path towards the converged x^P_{cons}.

C. Improved Following of the Reference Point

Asynchronous agents with distinct time delays result in agents that update states on outdated values of their neighbors. Hence, x^P_{cons} takes a non-ideal zigzag like path towards its consensus value and clearly seen in Figures 5 and 3. We propose to mitigate this zigzag behavior by adding a short pause before allowing agent movement. We pause the physical movement of agent i for t^p_i seconds, and let $x^P_{i,cons}$ converge beyond a threshold ϵ such that when $k \to \infty$, for every $x^P_{i,cons}[k] - x^P_{i,cons}[k-1] \leq \epsilon$ holds true afterwards. However, an agent can only calculate t^p_i after a significant number of time steps. As a solution, we estimate the regression of $x^P_{i,cons}$ locally by recursively fitting a polynomial to $x^P_{i,cons}[k]$ at every time step k. This allows each agent to have an estimate of time t^p_i that satisfies $x^P_{i,cons}[k] - x^P_{i,cons}[k-1] \leq \epsilon$. When the real world discrete time step k reaches t^p_i, an agent starts to move towards $x^P_{i,cons}$.

Note that the pause of t_p seconds does not affect information state convergence as agents continue to maintain the initial connectivity and exchange states. However, longer pauses adversely impact the time to information state convergence. Figure 6 shows that choosing a large convergence value on ϵ makes the agents move immediately; however, the path may be suboptimal despite fast information state convergence time is achieved in $29.3\,$s. In contrast, a small ϵ value delays the motion excessively and the system fails to realize the benefits of improved connectivity as the information state converged in $4.27\,$s. Intelligent selection of ϵ could optimize the trade-off between delayed convergence and reduced agent motions - an issue we leave for future research.

We analyze the advantage of regulating both the position and the communication frequency for multi-agent systems with 3, 6, and 9 agents. We analyze time to convergence and distance traveled by an agent in five variants of co-regulation. We create these variants by regulating agents' position only, regulating agents' communication frequency only, not regulating either agents' position or agents' frequency, and adding delay to agents' movement as suggested in Subsection V-C.

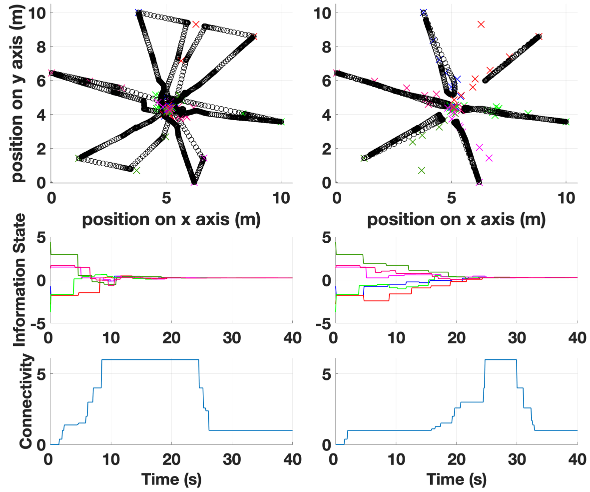

Fig. 6: Left: x^P of 6 agents, information state convergence at $29.3\,$s, and agent connectivity plot when $\epsilon = \infty$. Right: x^P of 6 agents, information state convergence at $42.7\,$s, and agent connectivity plot when $\epsilon = 0.1$

N	No Regulation		Frequency Reg. Only		Position Reg. Only		Position and Frequency Reg		Initial Pause	
	Dist(m)	Time(s)	Dist(m)	Time(s)	Dist(m)	Time(s)	Dist(m)	Time(s)	Dist(m)	Time(s)
3	0	10.1	0	10.1	7.96	10.1	8.54	10.1	5.61	10.1
6	0	52.00	0	42.00	6.48	27.50	7.40	23.07	7.26	25.79
9	0	117.6	0	78.4	6.59	35.1	7.81	24.19	7.52	32.05

TABLE I: Average time to converge and average distance traveled by each agent in different network sizes.

The simulation set up is identical to the one described in Subsection V-A. Table I summarizes the simulation results. It is clear that the fastest convergence times are achieved when we co-regulate both the communication frequency, agent position, and when we do not impose any pause to agent initial movement. However, the cost in this is visible in high traveling distance compared to position non-pause imposing method. Similarly, the initial pause does result in reduced agent travel distances, however, the increase in convergence time is inevitable.

VI. Experiments

To validate our approach, we implemented the proposed co-regulation algorithms on 4 UASs in an outdoor setting.

A. UAV System Overview

We used three customized quadcopter UASs shown in Figure 7. The frame of the quadcopter is a DJI450[2] type frame with a 450mm arms span. We selected open-source flight controller board Pixhawk 4[3] and connected an external Global Positioning System (GPS) which aids low-level navigation of the UAS. Attached to each UAS is an Odroid XU4[4], a single board computer that powers on-board processing and runs Ubuntu Mate operating system. The onboard computer uses MAVROS installed on top of the Robotic Operating System (ROS) to handle message passing between the onboard computer and the Pixhawk flight controller. An XBee S3B 900MHz radio communication module is attached to each

[2] https://www.dji.com/flame-wheel-arf/feature

[3] https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk4.html

[4] https://wiki.odroid.com/odroid-xu4/odroid-xu4

onboard computer for communication. All radio modules are set to broadcast to all UASs in range.

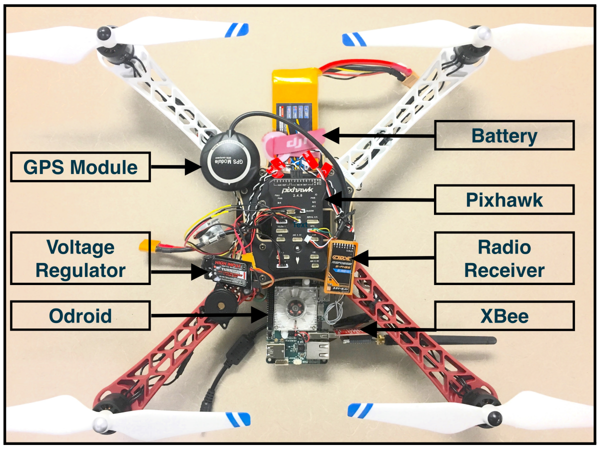

Fig. 7: Top view of experimental UAS platform

B. Two-tier Architecture and Consensus Implementation.

The consensus control portion of the two tire architecture resides in the onboard computer. We use the onboard computer to connect to the XBee device to receive and transmit the information state and consensus position state. In addition, the onboard computer implements algorithms in Equation (1), Equation (5), and Equation (7). It results in a target way-point in latitude, longitude, altitude, and yaw. Vehicle control is handled through the Pixhawk flight controller. The target way-point is sent to the Pixhawk flight controller through a serial communication link from the onboard computer. We use the ROS topic "mavros/setpoint_raw/global" to publish the target way-point to the Pixhawk flight controller. In turn, Pixhawk flight controller calculates necessary actuator input commands to the motors such that the UAS is navigated to the target waypoint.

We set up initial states of the agents to be a random valued 3x3 matrix between 0 and 1. Agents' x^F_{max} and x^F_{min} are set to 0.3Hz and 0.1Hz which controls both the computation and communication frequency. We select gains of 1 and 10 for α^F_1 and α^F_2 in Equation (3) and 0.01 and 0.05 for β^P_1 and β^P_2 in Equation (7). The 4 agents are initially dispersed in an area such that each neighbor is guaranteed to stay within the communication radius of its immediate neighbor.

Figure 8 plots the information state change and the agents' actual paths x^P_1 towards the common position x^P_{cons}. When there exists a difference between an agent's information state and its neighbors' information states, Equation (4) increases the communication frequency to its maximum. Dispersed agents exchange their positions, and Equation (5) updates the consensus position x^P_{cons} recursively at each execution cycle defined by their respective x^F. The differences in x^P_{cons} and x^I cause agents to update the reference position x^P_{ref} and move towards the consensus location. Subsequent communications and increased connectivity, from being in close proximity to each other, resulting in fast convergence of the information state value. This causes x^P_{ref} to change its position towards the agents' initial location, and the agents move back to the starting locations gradually.

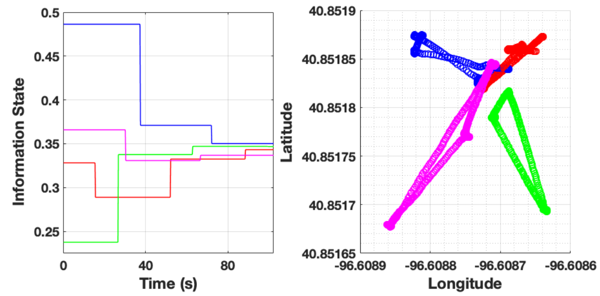

Fig. 8: Positions of 4 UASs in GPS coordinates (left) and Information State convergence (right)

VII. Conclusion

This paper analyzes co-regulation consensus strategies under delay and provides a framework for decoupling consensus from low-level feedback control. Co-regulation strategies are implemented on a system of 4 UASs and experimental results presented. This paper provides both theoretical and implementation advancements towards consensus deployment in real-world applications. Toward this goal, while theoretical guarantees abound in consensus research there is relatively little empirical evidence supporting real-world deployment. Our primary future objective is to construct a highly reliable platform and deployment mechanism for repeatable multi-agent demonstrations of co-regulated consensus. We expect to carry out an in-detailed study on adjusting controller gains and evaluating the proposed co-regulation methods with event triggered techniques as future works.

Appendix I
Proof of Theorem 4.2

$T_H[k] = \max\{T_i[k] + \tau_{ij}\}$ denotes the longest time for any agent i's state to reach an any agent j. Selecting $T_H[k]$ to be the longest time period provides a guarantee that all agents have broadcasted and received states at least once within $T_H[k]$. As x^F_i is time varying, multiple exchanges between agents can occur during a hyper period, $T_H[k]$, indicating possible Zeno behavior. However, binding x^F_i to an upper bound of $x^F_{i,max}$ guarantees $0 < T_H[k]$, and, therefore, avoids and infinite number of agent communication within a $T_H[k]$, or Zeno behavior.

Define $\mathbf{S}_m$ to be the equivalent matrix representation of a communication subgraph in $\bar{\mathcal{G}}$ where $m = 1, \ldots, M$. Let there be q communication instances at the l^{th} hyper period, the product of such subgraph matrices, $\mathbf{H}_l$, will be, $\mathbf{H}_l = \mathbf{S}_1\mathbf{S}_2\mathbf{S}_3 \cdots \mathbf{S}_q$. From Lemmas 3.1 and 3.2 in [13], the product of communication subgraph matrices results in a SIA matrix, and therefore, $\mathbf{H}_l$ is a SIA matrix.

Now consider we apply Equation (1) for k discrete time steps and represent it in terms of communication subgraphs at each discrete step k: $x^I[k] = \mathbf{S}_k\mathbf{S}_{k-1}\cdots\mathbf{S}_1 x^I[0]$. As we bind the communication frequency of each agent between $x^F_{i,min}$ and $x^F_{i,max}$, hyper period $T_H[k]$ agrees to the following inequality: $0 < T_H[k] < \infty$. Therefor, when $k \to \infty$, $l \to \infty$.

The state propagation can be expressed at each hyper period $\mathbf{H}_l$,

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

From [27], the product of SIA matrices converge to a vector as $k \to \infty$ as $l \to \infty$, $\lim_{l\to\infty} \mathbf{H}_l \mathbf{H}_{l-1} \cdots \mathbf{H}_1 = \mathbf{1} y^T$ where y denotes a constant column vector of positive entries representing the consensus values. Therefore,

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

APPENDIX II
PROOF OF THEOREM 4.3

This theorem is similar to the theorem on information state convergence under variable frequency in Theorem 4.2. At each hyper period, the communication subgraph contains a spanning tree as we select $T_H[k] = \max\{T_i[k] + \tau_{ij}\}$. Let $\mathbf{Q_m}$ be the equivalent matrix representation of a communication subgraph that represents x^P_{cons}, and $m = 1, \ldots, M$. Weighted average calculation of x^P_{cons} in Equation (5) ensures $\mathbf{Q_m}$ is a SIA matrix. Therefore, at each hyper period l, for q communicating instances, we write the union of communication subgraphs $\mathbf{H}_l$ as, $\mathbf{H}_l = \mathbf{Q}_1 \mathbf{Q}_2 \mathbf{Q}_3 \cdots \mathbf{Q}_q$.

From Lemmas 3.1 and 3.2 in [13], $\mathbf{H}_l$ is a SIA matrix as it is a product of SIA matrices. When considering the system for k discrete time steps, $x^P_{cons}[k] = \mathbf{Q}_k \mathbf{Q}_{k-1} \cdots \mathbf{Q}_1 x^P_{cons}[0]$.

Re-writing it in terms of each hyper period l, we get the following equation, $x^I[k] = \mathbf{H}_l \mathbf{H}_{l-1} \cdots \mathbf{H}_1 x^I[0]$. As we bind the communication frequency by a minimum and maximum, $0 < T_H[k] < \infty$, when $k \to \infty$, $l \to \infty$. Therefore,

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

The product of SIA matrices converges to a matrix with identical rows, $\mathbf{1} y^T$ when $k \to \infty$, $\lim_{l\to\infty} \mathbf{H}_l \mathbf{H}_{l-1} \cdots \mathbf{H}_1 = \mathbf{1} y^T$. Hence,

$$\lim_{k\to\infty} \mathbf{x}^P_{cons}[k] = \mathbf{1} \mathbf{y}^T \mathbf{x}^P_{cons}[0].$$

The necessary condition for convergence, that is to maintain a spanning tree in connectivity at each hyper period, is satisfied as long as the agents maintain the connectivity throughout. The bounded time delay ensures that each agent will receive a state update within a finite amount of time and spanning tree property is maintained.

REFERENCES

[1] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems*, vol. 27, no. 2, pp. 71–82, 2007.

[2] W. Qiao and R. Sipahi, "Consensus control under communication delay in a three-robot system: Design and experiments," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 2, pp. 687–694, 2016.

[3] L. Schenato and F. Fiorentin, "Average timesynch: A consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, 2011.

[4] Y. Cao, W. Yu, W. Ren, and G. Chen, "An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 427–438, Feb. 2013.

[5] R. Aragues, J. Cortes, and C. Sagues, "Distributed consensus on robot networks for dynamically merging feature-based maps," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 840–854, 2012.

[6] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature uavs," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306–1324, 2006.

[7] Q. Yuan, J. Zhan, and X. Li, "Outdoor flocking of quadcopter drones with decentralized model predictive control," *ISA transactions*, vol. 71, pp. 84–92, 2017.

[8] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proceedings of the American Control Conference, 2005.*, June 2005, pp. 1859–1864 vol. 3.

[9] X. Li and S. Hirche, "Event-triggered consensus of multi-agent systems on strongly connected graphs," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1311–1316.

[10] C. M. Krishna and K. G. Shin, *Real-time Systems*. Tata McGraw-Hill Education, 1997.

[11] C. Fernando, C. Detweiler, and J. Bradley, "Co-regulated consensus of cyber-physical resources in multi-agent unmanned aircraft systems," *Electronics*, vol. 8, no. 5, p. 569, 2019.

[12] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on automatic control*, vol. 48, no. 6, pp. 988–1001, 2003.

[13] W. Ren and R. W. Beard, "Consensus of information under dynamically changing interaction topologies," in *Proceedings of the IEEE American Control Conference 2004*, vol. 6. IEEE, 2004, pp. 4939–4944.

[14] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on automatic control*, vol. 50, no. 2, pp. 169–182, 2005.

[15] F. Dorfler, F. Pasqualetti, and F. Bullo, "Continuous-time distributed observers with discrete communication," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 296–304, 2013.

[16] W. Ren and Y. Cao, "Convergence of sampled-data consensus algorithms for double-integrator dynamics," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 3965–3970.

[17] F. Xiao and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, 2008.

[18] L. Fang and P. J. Antsaklis, "Information consensus of asynchronous discrete-time multi-agent systems," in *Proceedings of the American Control Conference, 2005*. IEEE, 2005, pp. 1883–1888.

[19] P. Lin and Y. Jia, "Average consensus in networks of multi-agents with both switching topology and coupling time-delay," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 1, pp. 303–313, 2008.

[20] A. Seuret, D. V. Dimarogonas, and K. H. Johansson, "Consensus under communication delays," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 4922–4927.

[21] Y. Zhang and Y.-P. Tian, "Consensus of data-sampled multi-agent systems with random communication delay and packet loss," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 939–943, 2010.

[22] F. Xiao and L. Wang, "State consensus for multi-agent systems with switching topologies and time-varying delays," *International Journal of Control*, vol. 79, no. 10, pp. 1277–1284, 2006.

[23] A. Joshi, A. Wala, M. Ludhiyani, S. Singh, M. Gagrani, S. Hazra, D. Chakraborty, D. Manjunath, and H. Chung, "Implementation of distributed consensus with guaranteed real-time communication on an outdoor quadrotor testbed," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 2158–2163.

[24] S. Reyes, E. B. Ramirez, M. Rivas, E. S. Espinoza, F. Munoz, and S. Salazar, "Testbed for implementation of consensus strategies on a team of multi-agent aircraft systems at outdoors environments," in *2018 15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*. IEEE, 2018, pp. 1–6.

[25] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 1990.

[26] J. Wolfowitz, "Products of indecomposable, aperiodic, stochastic matrices," *Proceedings of the American Mathematical Society*, vol. 14, no. 5, pp. 733–737, 1963.

[27] C. Fernando, C. Detweiler, and J. Bradley, "Co-regulating communication for asynchronous information consensus," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 6994–7001.