

Distributed Multi-agent Coordination over Cellular Sheaves

Tyler Hanks, Hans Riess, Samuel Cohen, Trevor Gross, Matthew Hale, James Fairbanks*

Abstract—Techniques for coordination of multi-agent systems are vast and varied, often utilizing purpose-built solvers or controllers with tight coupling to the types of systems involved or the coordination goal. In this paper, we introduce a general unified framework for heterogeneous multi-agent coordination using the language of cellular sheaves and nonlinear sheaf Laplacians, which are generalizations of graphs and graph Laplacians. Specifically, we introduce the concept of a nonlinear homological program encompassing a choice of cellular sheaf on an undirected graph, nonlinear edge potential functions, and constrained convex node objectives. We use the alternating direction method of multipliers to derive a distributed optimization algorithm for solving these nonlinear homological programs. To demonstrate the wide applicability of this framework, we show how hybrid coordination goals including combinations of consensus, formation, and flocking can be formulated as nonlinear homological programs and provide numerical simulations showing the efficacy of our distributed solution algorithm.

I. INTRODUCTION

Optimal control and coordination of multi-agent systems is of paramount importance in modern applications ranging from multi-robot systems [1] to cyber-physical systems [2]. In anticipation of next-generation systems composed of highly autonomous heterogeneous agents with multiple and evolving individual and group objectives, our goal in this paper is to present a unified framework for modeling a wide array of multi-agent coordination problems, along with a general-purpose solver for these problems. In this framework, system designers specify requirements about the behavior of a multi-agent system, compile these requirements into a *homological program*, and synthesize a decentralized control policy with a universal solver.

The main abstractions we utilize in constructing this framework are cellular sheaves and nonlinear sheaf Laplacians. Our proposed framework uses cellular sheaves with nonlinear edge potentials to model various types of multi-agent interactions and coordination goals, including but not

limited to, consensus, translation invariant formation, and flocking. The generality of this framework enables heterogeneity not only in the systems being controlled, but also in the communication patterns and coordination goals between different agents. To design distributed optimal controllers for such multi-agent systems, we use the alternating direction method of multipliers (ADMM) to derive a distributed solution algorithm for optimization problems defined over cellular sheaves. ADMM is a widely used technique for distributed optimization due to its attractive convergence properties provided only mild assumptions on the problem. The resultant algorithm allows agents to choose control inputs that minimize their own control objectives while driving the global system towards the coordination goal, all while utilizing only local computations and communication with neighboring agents.

Related Work: Several efforts have emerged to systematically study controller design patterns, including layered control architectures [3], co-design, [4], and information structures [5]. Through our reformulation of well-known multi-agent coordination problems as homological programs, we similarly illuminate fundamental design principles of networked multi-agent systems.

The graph-theoretic approach to modeling multi-agent systems has lead to a corpus of algorithms addressing a diversity of coordination problems, including consensus [6], [7], dissensus [8], formation control [9], [10], flocking [11], [12], and many other tasks [13]. In this literature, distributed optimization [14] is commonly integrated with cooperative control strategies in order to simultaneously pursue private goals as well as shared common objectives [15].

Cellular sheaves were introduced in stratification theory [16] and rediscovered in applied topology [17]. The proposed marriage of methods from spectral graph theory and cellular sheaves [18] led to the observation that sheaf Laplacians induce a dynamical system generalizing (multidimensional) distributed mean consensus protocols where contemporaneous generalizations of consensus such as matrix-weighted [19] and matrix-scaled consensus [20] are straightforward instantiations of the cellular sheaf construction. Subsequent sheaf-theoretic developments [21] yield a similar generalization of max consensus [20].

Control of cellular sheaves was initiated for non-physical systems in which linear and nonlinear sheaf Laplacians are deployed to model opinion dynamics via discourse sheaves [22]. Another sheaf-theoretic approach to optimal network control [23] bears similarities to ours, without the computational advantages of a distributed implementation. Sheaves have appeared elsewhere in the control systems lit-

*Tyler Hanks, Samuel Cohen, and Trevor Gross are with the Department of Computer and Information Science and Engineering and James Fairbanks is with the Department of Mechanical and Aerospace Engineering, University of Florida, emails: {t.hanks, samuel.cohen, trevorgross, fairbanksj}@ufl.edu. Hans Riess and Matthew Hale are with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, emails: {riess, mhale30}@gatech.edu

Hanks was supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1842473. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF. Riess and Hale were supported by AFOSR under grants FA9550-23-1-0120 and FA9550-19-1-0169, ONR under grants N00014-22-1-2435 and N00014-21-1-2495. All authors were supported by DARPA under grant HR00112220038.

erature, including modeling event-based systems [24], information diffusion [25], game theory [26], as well as network theory [27]. Distributed optimization [28], LQR [29], and model-predictive control [30] have also been studied through the lens of category theory, an area closely related to sheaf theory.

Homological programming was developed for mobile sensor networks [31], and extended to a generalized notion of distributed optimization [32], with applications in distributed model-predictive control and graph signal processing. Our work expands the scope of homological programming introduced in that work with the nonlinear sheaf Laplacian and constrained node objectives.

Outline: In Section II, we review cellular sheaves and sheaf Laplacians. Then, in Section III, we establish a general framework for representing multi-agent coordination problems. In Section IV, we propose a solution method. To encourage the adoption of this framework, we present an in-depth case study in Section V. In Section VI, we present several numerical simulations. Finally, in Section VII we discuss future directions.

II. PRELIMINARIES

In this section we introduce cellular sheaves and sheaf Laplacians. Together with optimization techniques, these constructions will serve as a general framework for us to analyze multi-agent coordination problems.

Suppose $G = (V, E)$ is a undirected graph with node set $V = [N]$ and edge set $E \subseteq V \times V$, where $[N]$ denotes the set $\{1, 2, \dots, N\}$. An edge between given nodes i and j is given by an unordered pair of concatenated indices $ij = ji \in E$. When particular endpoints of an edge are not specified, an edge can be denoted by $e \in E$. The set $N_i = \{j : ij \in E\}$ are the neighbors of i .

Interpretation. In a multi-agent system (MAS), an undirected graph models bidirectional communication channels between agents. We assume these communication networks have unchanging topologies.

Given a fixed graph G , we define a data structure for organizing assignments of vectors.

Definition 1 (Cellular Sheaf). Given $G = (V, E)$, a cellular sheaf \mathcal{F} (valued in Euclidean spaces over a graph) is a data structure that assigns:

- A Euclidean space $\mathcal{F}(i)$ with the standard inner-product to every node $i \in V$ called a **stalk**.
- A Euclidean space $\mathcal{F}(ij)$ with the standard inner-product to every edge $ij \in E$ called an **edge stalk**.
- A linear transformation $\mathcal{F}(i) \xrightarrow{\mathcal{F}_i \lhd ij} \mathcal{F}(ij)$ for every $i \in V, j \in N_i$ called a **restriction map**.

Example 1 (Constant Sheaf). Given a graph G and a particular vector space \mathbb{R}^k , the constant sheaf, denoted \mathbb{R}^k , assigns the vector space $\mathbb{R}^k(i) = \mathbb{R}^k$ to every $i \in V$, the vector space $\mathbb{R}^k(ij) = \mathbb{R}^k$ to every edge $ij \in E$, and the identity map $\mathbb{R}^k_i \lhd ij = \text{id}_{\mathbb{R}^k}$ for every $i \in V, j \in N_i$.

We refer to a particular assignment $x_i \in \mathcal{F}(i)$ as a **local section**. This is the datum assigned to an individual node. Collecting all the data over nodes yields the space of 0-cochains, $C^0(G; \mathcal{F}) = \bigoplus_{i \in V} \mathcal{F}(i)$, and data over edges is similarly gathered in the space of 1-cochains, $C^1(G; \mathcal{F}) = \bigoplus_{e \in E} \mathcal{F}(e)$. $C^0(G; \mathcal{F})$ and $C^1(G; \mathcal{F})$ are endowed with the inner products $\langle \mathbf{x}, \mathbf{x}' \rangle_{C^0} = \sum_{i \in V} \langle x_i, x'_i \rangle_{\mathcal{F}(i)}$ and $\langle \mathbf{y}, \mathbf{y}' \rangle_{C^1} = \sum_{e \in E} \langle y_e, y'_e \rangle_{\mathcal{F}(e)}$.

Interpretation. A stalk is interpreted as an agent's state space, and a local section is interpreted as a particular state (e.g. position and velocity measurements). In a multi-agent systems, a 0-cochain is, then, a global state, and $C^0(G; \mathcal{F})$ is the global state space. Note that dimensions of stalks are not required to be the same for every agent, hence, the state spaces of agents can vary from agent to agent (e.g. a team of both ground and ariel vehicles). In certain cases, 1-cochains can be seen as observed outputs when internal states are transformed by restriction maps.

A ubiquitous problem in sheaf theory is to decide whether 0-cochains are coherent with respect to restriction maps. This consistency is expressed by the notion of a global section.

Definition 2 (Global Sections). Suppose \mathcal{F} is a cellular sheaf over G . A **global section** is a 0-cochain $\mathbf{x} \in C^0(G; \mathcal{F})$ such that

$$\mathcal{F}_i \lhd ij(x_i) = \mathcal{F}_j \lhd ij(x_j) \quad \forall i \in V, \forall j \in N_i. \quad (1)$$

The set of global sections is $\Gamma(G; \mathcal{F}) \subseteq C^0(G; \mathcal{F})$.

Interpretation. In a multi-agent coordination problem, agents are often required to satisfy locally-defined constraints that propagate globally, the simplest example being consensus. The global section condition (Eq. 1) allows these local constraints to be encoded as restriction maps. For instance, the constant sheaf encodes a consensus constraint when G is connected. This follows from $\Gamma(G; \mathbb{R}^k) \cong \mathbb{R}^k$. If G has c connected components, it follows that $\Gamma(G; \mathbb{R}^k) \cong (\mathbb{R}^k)^c$. A global section, then, corresponds to a particular consensus vector assigned to each connected component.

Deciding whether a 0-cochain is a global section is not enough. Often, we want a flow to a global section from an arbitrary 0-cochain. To construct such flows, we first introduce a few notions from homological algebra. Define the **coboundary operator** $C^0(G; \mathcal{F}) \xrightarrow{\delta_{\mathcal{F}}} C^1(G; \mathcal{F})$ given by $(\delta_{\mathcal{F}} \mathbf{x})_{ij} = \mathcal{F}_i \lhd ij(x_i) - \mathcal{F}_j \lhd ij(x_j)$. The **degree-0** and **degree-1 cohomology** of a cellular sheaf is given by

$$\begin{aligned} H^0(G; \mathcal{F}) &= \ker \delta_{\mathcal{F}} \\ H^1(G; \mathcal{F}) &= C^1(G; \mathcal{F}) / \text{im } \delta_{\mathcal{F}} \end{aligned} \quad (2)$$

It follows immediately that $H^0(G; \mathcal{F}) = \Gamma(G; \mathcal{F})$ because $x \in \ker \delta_{\mathcal{F}}$ precisely when $\mathcal{F}_i \lhd ij(x_i) - \mathcal{F}_j \lhd ij(x_j) = 0$ for all $ij \in E$. Global sections, then, can be computed as the kernel of the coboundary operator with a linear solver, where the relevant matrix grows in the number of nodes of the graph and the dimensions of the stalks.

TABLE I: Sheaf Theory Concepts and Potential Functions in Multi-Agent Systems

Sheaf Theory Concepts			Potential Functions	
Concept	Notation	Interpretation	Potential Function	Coordination Goal
graph	$G = (V, E)$	bidirectional communication network	$(1/2)\ y\ _2^2$	Consensus
vertex stalk	$\mathcal{F}(i)$	state space of agent i	$y^\top A y$	Matrix-weighted consensus [19]
local section	x_i	local state	$-(1/2)\ y\ _2^2$	Dissensus [8]
edge stalk	$\mathcal{F}(ij)$	communication space between agent i and j	$(1/2)\ y - b\ _2^2$	Reach displacement of b
restriction map	$\mathcal{F}_i \rightharpoonup_{ij}$	how agent i sends messages to agent j	$(\ y\ _2^2 - r^2)^2$	Reach distance of r
sheaf	\mathcal{F}	multi-agent communication structure		
0-cochains	$C^0(G; \mathcal{F})$	global state space		
0-cochain	\mathbf{x}	global state		
global sections	$\Gamma(G; \mathcal{F})$	feasible global states		

A decentralized approach to computing global sections is to construct an appropriate local operator on $C^0(G; \mathcal{F})$ whose dynamics converge to a global section. A linear operator $L_{\mathcal{F}}$ – a **linear sheaf Laplacian** – whose construction relies on the Hodge Laplacian [33] leads to a heat equation $\dot{\mathbf{x}} = -L_{\mathcal{F}}\mathbf{x}$ converging to a global section for an arbitrary initial condition [18, Proposition 8.1]. To analyze a wider class of coordination problems, we utilize a generalization of the linear sheaf Laplacian [22, §10]. In addition to the specification of a graph and a sheaf, the nonlinear sheaf Laplacian is determined by a choice of edge potentials.

Definition 3 (Nonlinear Sheaf Laplacian). Suppose \mathcal{F} is a cellular sheaf over G , and suppose $\{U_e : \mathcal{F}(e) \rightarrow \mathbb{R}\}_{e \in E}$ are (possibly nonlinear) potential functions. Then, the **nonlinear sheaf Laplacian** is the map $C^0(G; \mathcal{F}) \xrightarrow{L_{\mathcal{F}}^{\nabla U}} C^0(G; \mathcal{F})$ defined $L_{\mathcal{F}}^{\nabla U} = \delta^\top \circ \nabla U \circ \delta$ where $\nabla U : C^1(G; \mathcal{F}) \rightarrow C^1(G; \mathcal{F})$ is defined as the gradient of $U(\mathbf{y}) = \sum_{e \in E} U_e(y_e)$.

Examples 2. Given G and \mathcal{F} , suppose $U_e(y_e) = (1/2)\|y_e\|_2^2$ for all $e \in E$. Then, $L_{\mathcal{F}}^{\nabla U} = \delta^\top \circ \delta$ and is denoted $L_{\mathcal{F}}$. In particular, the linear sheaf Laplacian $L_{\mathbb{R}^k}$ of the constant sheaf on G is the Kronecker product of the unnormalized graph Laplacian L_G with an identity matrix or simply L_G when $k = 1$.

The nonlinear sheaf Laplacian acts on local sections as

$$(L_{\mathcal{F}}^{\nabla U} \mathbf{x})_i = \sum_{j \in N_i} \mathcal{F}_i^\top \left(\nabla U_{ij} (\mathcal{F}_{i \rightharpoonup ij}(x_i) - \mathcal{F}_{j \rightharpoonup ij}(x_j)) \right).$$

Consequently, the nonlinear sheaf Laplacian can be computed locally in the network because $(L_{\mathcal{F}}^{\nabla U} \mathbf{x})_i$ only requires information about the local sections and restriction maps of neighbors N_i .

While linear sheaf Laplacians converge to global sections specifying local linear constraints, nonlinear Laplacian dynamics are more subtle. For example, nonconvex potentials can lead to diverging trajectories. For instance, given G and \mathcal{F} , suppose $U_e(y) = -(1/2)\|y\|_2^2$ for all $e \in E$. Then, $L_{\mathcal{F}}^{\nabla U} = -L_{\mathcal{F}}$, and trajectories with $\mathbf{x}(0) \notin H^0(G; \mathcal{F})$ diverge from their the projection onto $H^0(G; \mathcal{F})$.

III. NONLINEAR HOMOLOGICAL PROGRAMMING FOR MULTI-AGENT COORDINATION

This section presents the primary modeling tool used in this framework – nonlinear homological programming – and formulates a general multi-agent coordination problem within this framework.

A. Nonlinear Homological Programming

The following definition extends homological programming [32] with nonlinear constraints and objective functions valued in the extended real numbers $\mathbb{R} = \mathbb{R} \cup \{\infty\}$.

Definition 4 (Nonlinear Homological Program). A **nonlinear homological program** consists of the following data:

- 1) Undirected graph $G = (V, E)$;
- 2) Cellular sheaf \mathcal{F} on G ;
- 3) Objective functions $\{f_i : \mathcal{F}(v) \rightarrow \mathbb{R}\}_{i \in V}$;
- 4) Potential functions $\{U_e : \mathcal{F}(e) \rightarrow \mathbb{R}\}_{e \in E}$.

The homological program defined by this data is the optimization problem

$$\begin{aligned} & \underset{\mathbf{x} \in C^0(G; \mathcal{F})}{\text{minimize}} && \sum_{i \in V} f_i(x_i) \\ & \text{subject to} && L_{\mathcal{F}}^{\nabla U} \mathbf{x} = 0 \end{aligned} \quad (\text{P})$$

Example 3 (Distributed Optimization). In a system of N agents where each agent has a private convex objective function $f_i(x)$ with decision variable $x \in \mathbb{R}^k$, a ubiquitous distributed optimization problem [34] is to minimize the sum of each agent’s cost function: $\text{minimize}_{x \in \mathbb{R}^k} \sum_{i=1}^N f_i(x)$. A typical distributed solution strategy is to introduce local copies of x for each agent to decouple the objective and use a coupling consensus constraint. This can be defined as a homological program for the constant sheaf \mathbb{R}^k on a connected agent communication topology, as shown in [32].

The following theorem describes when nonlinear homological programs are convex optimization problems, enabling us to solve them efficiently.

Theorem 1. Let $\text{P} = (V, E, \mathcal{F}, \{f_i\}, \{U_e\})$ be a nonlinear homological program. If U_e is differentiable and convex for each $e \in E$, and f_i is convex for every $i \in V$, then P is a convex optimization problem.

For illustrative purposes, we introduce single-agent optimal control as a simple example of a homological program

which will be further utilized when posing multi-agent optimal control problems.

Example 4 (Single-Agent Optimal Control). Single-agent optimal control problems for a fixed time horizon T form homological programs on cellular sheaves whose underlying graphs are $T+1$ length paths. To see why, consider a control system with states in \mathbb{R}^n , control inputs in \mathbb{R}^m , and discrete dynamics $x(t+1) = Ax(t) + Bu(t)$. We model this agent's dynamics for T time-steps from an initial condition $x(1) = c$ as a cellular sheaf \mathcal{D} on the $(T+1)$ -vertex path graph P_{T+1} (see Fig. 1). The overall potential function for \mathcal{D} is $\Phi(\mathbf{y}) = (1/2)\|y_0 - c\|_2^2 + \sum_{e \in E} (1/2)\|y_e\|_2^2$ for all $\mathbf{y} \in C^1(P_{T+1}; \mathcal{D})$. Zeros of the associated nonlinear Laplacian $L_{\mathcal{D}}^{\nabla\Phi} = \delta_{\mathcal{D}}^{\top} \circ \nabla\Phi \circ \delta_{\mathcal{D}}$ then correspond precisely to admissible trajectories of the system across T time-steps starting from $x(1) = c$.

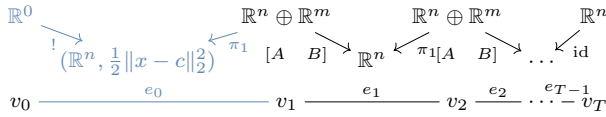


Fig. 1: The cellular sheaf encoding the evolution of a discrete LTI system from an initial condition $x(1) = c$ over T time-steps. Note that $!: \mathbb{R}^0 \rightarrow \mathbb{R}^n$ is the unique map from \mathbb{R}^0 , and π_1 is the first projection. The graph which the sheaf is defined on is given on the bottom with the stalks and restriction maps lying above. The blue section uses an edge potential function to fix the initial condition to $x(1) = c$. All other edge potential functions are the standard consensus potentials so we omit them from the diagram for clarity. The black section then encodes the evolution of the dynamics from c for T time-steps.

Now, suppose we wish to drive the system to a desired state using stage cost functions $f^t: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ for $t \in [T-1]$. A typical multistage optimal control problem starting from initial state $x(1) = c$ is

$$\begin{aligned} & \text{minimize} && \sum_{t=1}^{T-1} f^t(x(t), u(t)) \\ & \text{subject to} && x(t+1) = Ax(t) + Bu(t) \quad \forall t \in [T-1] \\ & && x(1) = c. \end{aligned} \quad (3)$$

Note that our framework can easily incorporate other convex constraints on the states and controls; however, we leave these out for simplicity of presentation. For notational convenience, let $\mathbf{x} = \text{vec}(x(1), \dots, x(T))$ and $\mathbf{u} = \text{vec}(u(1), \dots, u(T-1))$, where vec denotes the concatenation of column vectors into a new column vector. We have already seen that the constraints in Eq. 3 correspond to the zeros of the nonlinear Laplacian for \mathcal{D} . Thus the problem in Eq. 3 is equivalent to the homological program

$$\begin{aligned} & \text{minimize} && \sum_{t=1}^{T-1} f^t(x(t), u(t)) \\ & \text{subject to} && L_{\mathcal{D}}^{\nabla\Phi}(\mathbf{x}, \mathbf{u}) = 0. \end{aligned} \quad (4)$$

B. General Multi-Agent Coordination Homological Program

Now suppose we have a multi-agent control system with N agents on a connected communication topology $G = (V, E)$. Each agent $i \in [N]$ has state space \mathbb{R}^{n_i} , control space \mathbb{R}^{m_i} , and discrete dynamics $x_i(t+1) = A_i x_i(t) + B_i u_i(t)$. The total system dynamics are then given by the direct

sum of the dynamics for each agent. Let $Q = \sum_{i \in [N]} n_i$ and $R = \sum_{i \in [N]} m_i$ denote the global state and control input dimensions respectively. For notational convenience, we introduce block matrix variables $\mathbf{X} \in \mathbb{R}^{Q \times T}$ and $\mathbf{U} \in \mathbb{R}^{R \times T}$ to represent T -length trajectories of the global state space (resp. control space), where block i, t is in $\mathbb{R}^{n_i \times 1}$ (resp. $\mathbb{R}^{m_i \times 1}$). Then, $\mathbf{X}[:, t]$ denotes the global state at time t , $\mathbf{X}[i, :]$ denotes the state trajectory of agent i across all time-steps, and $\mathbf{X}[i, t]$ denotes the state of agent i at time-step t . The same goes for \mathbf{U} .

A general distributed multi-agent coordination problem is that of choosing control inputs to drive this global system to a desired goal while utilizing only local computations and communication between agents. We formulate this as a multi-stage optimal control problem over a fixed time horizon T . We allow each agent to have its own local optimal control problem of the same form as Eq. 3. Specifically, each agent $i \in [N]$ has stage costs f_i^t and dynamic evolution sheaf \mathcal{D}_i with potential function Φ_i from a given initial condition $x_i(1) = c_i$, as in Example 4. Let \mathcal{C}_i denote the set $\{(\mathbf{x}, \mathbf{u}) \in C^0(P_{T+1}; \mathcal{D}_i) \mid L_{\mathcal{D}_i}^{\nabla\Phi_i}(\mathbf{x}, \mathbf{u}) = 0\}$. Then we define the functional form of Eq. 4 as $J_i: C^0(P_{T+1}, \mathcal{D}_i) \rightarrow \mathbb{R}$ given by

$$J_i(\mathbf{x}, \mathbf{u}) = \sum_{t=1}^{T-1} f_i^t(x(t), u(t)) + \chi_{\mathcal{C}_i}(\mathbf{x}, \mathbf{u}), \quad (5)$$

where $\chi_{\mathcal{C}_i}$ denotes the convex indicator function of the set \mathcal{C}_i . The global objective function $\sum_{i=1}^N J_i(\mathbf{X}[i, :], \mathbf{U}[i, :])$ is then a decoupled multi-agent optimal control problem where each agent wants to fulfill its individual control goal disregarding other agents.

To coordinate our multi-agent system, we introduce a cellular sheaf \mathcal{F} on G with the interpretations given in Table I (left). We can then formulate the coordination goal between each pair of communicating agents $e = i \sim j \in E$ by choosing an appropriate edge potential function U_e from Table I (right). Such a choice of potential functions determines a global potential function $U(\mathbf{y}) = \sum_{e \in E} U_e(\mathbf{y}_e)$ defined over 1-cochains and yields the coordination objective

$$\text{minimize}_{\mathbf{X}[:, t] \in C^0(G; \mathcal{F})} U(\delta_{\mathcal{F}} \mathbf{X}[:, t])$$

for a given $t \in [T]$. Taken together, this lets us define a general multi-agent coordination problem as

$$\text{minimize}_{i=1}^N J_i(\mathbf{X}[i, :], \mathbf{U}[i, :]) + \gamma U(\delta_{\mathcal{F}} \mathbf{X}[:, T]), \quad (6)$$

In other words, this objective asks each agent in the MAS to achieve its own control objective as much as possible subject to its dynamic constraints and initial condition while also cooperating with the other agents to achieve the coordination goal by the end of the time horizon. The free parameter γ can be increased to emphasize the importance of the group objective over the individual objectives. When U is convex and bounded below, minimizing $U \circ \delta_{\mathcal{F}}$ is equivalent to finding a point where $\nabla(U \circ \delta_{\mathcal{F}}) = \delta_{\mathcal{F}}^{\top} \circ \nabla U \circ \delta_{\mathcal{F}} = 0$. This

gives us a strengthened multi-agent homological program:

$$\begin{aligned} & \underset{\mathbf{X}, \mathbf{U}}{\text{minimize}} && \sum_{i=1}^N J_i(\mathbf{X}[i, :], \mathbf{U}[i, :]) \\ & \text{subject to} && L_{\mathcal{F}}^{\nabla U} \mathbf{X}[:, T] = 0, \end{aligned} \quad (7)$$

where the coordination goal is imposed as a hard constraint rather than an objective term.

Remark 1. It should be noted that there are multiple sheaf homological constraints present in Eq. 7. Specifically, the sheaves \mathcal{D}_i encode the dynamic evolution of each agent $i \in [N]$ from an initial condition to the end of the time horizon. Meanwhile, the sheaf \mathcal{F} and its associated nonlinear Laplacian specifies the coordination constraint which must be satisfied by the end of the time horizon. We can think of this as a sort of *nested* or *hierarchical* homological program, where the objectives on each node of the coordination sheaf are themselves homological programs.

IV. SOLVING HOMOLOGICAL PROGRAMS WITH ADMM

We examine how we can use ADMM to derive a distributed solver for homological programs, thus allowing the design of distributed controllers for achieving multi-agent coordination. Algorithm 1 presents a distributed solution algorithm for a general nonlinear homological program $P = (V, E, \mathcal{F}, \{f_i\}, \{U_e\})$ subject to the following assumptions.

Assumption 1. The homological program P satisfies the convexity conditions of Theorem 1, and each node objective f_i is closed and proper.

Assumption 2. Each edge potential U_e is strongly convex with unique minimizer of b_e such that $U_e(b_e) = 0$.

To use ADMM to solve P , we first introduce extra variables $z_i \in \mathcal{F}(i)$ for each vertex stalk and rewrite the problem in the equivalent consensus form:

$$\begin{aligned} & \underset{\mathbf{z}}{\text{minimize}} && \sum_i f_i(x_i) + \chi_C(\mathbf{z}) \\ & \text{subject to} && \mathbf{x} - \mathbf{z} = 0 \end{aligned} \quad (8)$$

where $C = \{\mathbf{z} \in C^0(G; \mathcal{F}) \mid L_{\mathcal{F}}^{\nabla U} \mathbf{z} = 0\}$. This problem has the following augmented Lagrangian with scaled dual variable as in [35]:

$$\mathcal{L}_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = \sum_i f_i(x_i) + \chi_C(\mathbf{z}) + (\rho/2) \|\mathbf{x} - \mathbf{z} + \mathbf{y}\|_2^2.$$

Applying ADMM gives the iterative update rule:

$$\begin{aligned} x_i^{k+1} &:= \operatorname{argmin}_{x_i} f_i(x_i) + (\rho/2) \|x_i - z_i^k + y_i^k\|_2^2 \\ \mathbf{z}^{k+1} &:= \Pi_C(\mathbf{x}^{k+1} + \mathbf{y}^k) \\ y_i^{k+1} &:= y_i^k + x_i^{k+1} - z_i^{k+1} \end{aligned} \quad (9)$$

where Π_C denotes Euclidean projection onto C . The dynamics of the x_i and y_i updates can be computed in parallel. The core difficulty of the algorithm then is computing the \mathbf{z} update in a distributed fashion. It turns out that running the nonlinear heat equation, which is a local operation, converges precisely to the desired projection. We prove this in the following theorem.

Theorem 2. Let \mathcal{F} be a cellular sheaf on a graph $G = (V, E)$. For each edge $e \in E$, let $U_e: \mathcal{F}(e) \rightarrow \mathbb{R}$ be a

differentiable, strongly convex function with unique global minimum point at b_e with minimum value $U_e(b_e) = 0$. Define $U = \sum_{e \in E} U_e$ and $b = \operatorname{vec}(b_1, \dots, b_{|E|})$. If $b \in \operatorname{im} \delta_{\mathcal{F}}$, then trajectories of the nonlinear Laplacian dynamics

$$\dot{x} = -\alpha L_{\mathcal{F}}^{\nabla U} x \quad (10)$$

converge to the orthogonal projection of the initial condition onto $\delta_{\mathcal{F}}^+ b + H^0(G; \mathcal{F}) = \ker L_{\mathcal{F}}^{\nabla U}$ for a given diffusivity $\alpha > 0$, where $\delta_{\mathcal{F}}^+$ denotes the Moore-Penrose pseudoinverse.

This gives us Algorithm 1 for solving nonlinear homological programs. In this algorithm, each iteration of the

Algorithm 1: Distributed Solve

Input: Sheaf \mathcal{F} , potential functions U_e , and initial state $(\mathbf{z}^0, \mathbf{y}^0)$
Input: Step size ρ , diffusivity α , tolerances ϵ_1, ϵ_2 , and $K \in \mathbb{N}$
Output: $\mathbf{z}^*, \mathbf{y}^*$
 $\mathbf{z} \leftarrow \mathbf{z}^0$;
 $\mathbf{y} \leftarrow \mathbf{y}^0$;
while $j \leq K$ **do**
 for $i \leftarrow 1$ **to** N **do**
 $x_i \leftarrow \operatorname{argmin}_{x_i} f_i(x_i) + (\rho/2) \|x_i - z_i + y_i\|_2^2$;
 end
 $\mathbf{x} \leftarrow \operatorname{vec}(\{x_i\}_{i \in [N]})$;
 $\mathbf{z} \leftarrow \text{sheafDiffusion}(\mathcal{F}, U, \alpha, \epsilon_2, \mathbf{x} + \mathbf{y})$;
 if $\|\mathbf{x} - \mathbf{z}\| < \epsilon_1$ **then**
 break;
 end
 for $i \leftarrow 1$ **to** N **do**
 $y_i \leftarrow y_i + x_i - z_i$;
 end
end
return \mathbf{z}, \mathbf{y}

two for-loops can be computed in parallel by each agent. The operation $\text{sheafDiffusion}(\mathcal{F}, U, \alpha, \epsilon, \mathbf{z}^0)$ denotes running the nonlinear heat equations for \mathcal{F} with diffusivity α to ϵ convergence from initial condition \mathbf{z}^0 .

Theorem 3. Suppose a homological program P satisfies assumptions 1 and 2. Assume also that the Lagrangian of Eq. 8 has a saddle point. Then Algorithm 1 applied to P has the following properties:

- *Residual convergence:* $\mathbf{x}^k - \mathbf{z}^k \rightarrow 0$ as $k \rightarrow \infty$
- *Objective convergence:* $\sum_i f_i(x_i) + \chi_C(\mathbf{z}) \rightarrow p^*$ as $k \rightarrow \infty$.
- *Dual variable convergence:* $\mathbf{y}^k \rightarrow \mathbf{y}^*$ as $k \rightarrow \infty$ where \mathbf{y}^* is a dual optimal point.

Remark 2. When potential functions are chosen that violate Assumption 2, the same ADMM algorithm can still be used to solve the relaxed problem Eq. 6 with slight modification to the \mathbf{z} update.

V. CASE STUDY: MULTI-DOMAIN TEAM VEHICLE COORDINATION

Multi-vehicle systems (MVS) provide a rich class of examples of multi-agent coordination problems that can be formulated and solved as homological programs. In this section, we describe some of these coordination problems in more detail through a case study and explain how they are instances of homological programs.

We consider a team of heterogeneous autonomous vehicles that includes unmanned aerial vehicles (UAV), unmanned surface vehicles (USV), and unmanned underwater vehicles (UUV). The vehicles are indexed by $V = \{1, 2, \dots, N\}$. In this setting, communication modalities (e.g. radio frequency, sonar, optical, or tethered links) are domain-dependent, varying with the operational environment of each vehicle type.

We assume vehicles have a fixed communication network. UAVs have a dense communication network between other UAVs, and UAVs also communicate with USVs. USVs serve as relays for communication between USVs and UAVs. Consequently, USVs have links with other UAVs as well as other USVs and links with designated UUVs. Finally, UUVs communicate with other UUVs, and UUVs communicate with designated USVs. We model the entire communication network according to an undirected graph $G = (V, E)$ where an edge $e \in E$ denotes the presence of a bidirectional communication channel between the vehicles at its endpoints. Let G_{UAV} , G_{USV} , and G_{UUV} denote the respective subgraphs corresponding to each respective inter-team communication network, and let $G_{\text{UAV-USV}}$ and $G_{\text{UUV-USV}}$ denote the subgraphs corresponding to each intra-team network.

Vehicles navigate in a rectangular domain $D \subseteq \mathbb{R}^3$, which includes airspace, a surface region, and an undersea region. We assume that for each $i \in V$ the dynamics of vehicle i are modeled by double integrator dynamics

$$\begin{aligned} \dot{p}_i &= v_i \\ \dot{v}_i &= u_i, \end{aligned} \quad (11)$$

where $p_i \in \mathbb{R}^d$ is agent i 's position, $v_i \in \mathbb{R}^d$ is agent i 's velocity, and $u_i \in \mathbb{R}^d$ is agent i 's control input. Then, the state of agent i is $x_i = \text{vec}(p_i, v_i)$. The dimensions of the agents' state spaces depend on the type of vehicle: UUVs and UAVs move in \mathbb{R}^3 and hence have $d = 3$, while USVs are modeled as moving in \mathbb{R}^2 and hence have $d = 2$. We assume that agents fully observe their own position and velocity. Consider the following scenario which we describe as an optimal control problem:

- The USV team is tasked with making a translationally-invariant grid formation that is specified by displacement vectors $\hat{p}_{i,j}$. A subset of edges $E_{\text{USV}} \subseteq E_{\text{surface}}$ indicates which pairs of agents must maintain a specified relative position in order to assemble the given formation.
 - The UAV team flocks together, forming consensus on their speeds and bearings while maintaining a fixed altitude h_{UAV} . The UAVs also need to remain within a given distance $r_{\text{UAV-USV}}$ from the USVs that they communicate with in order to remain within communication range.
 - The UUV team flocks together, forming consensus on their speeds and bearings, but are free to vary their depth from the surface. In order to maintain communication, UUVs also need to remain within a given distance $r_{\text{UUV-USV}}$ to USVs, and they also need to remain with a given distance of other UUVs for the same reason.
 - All vehicles minimize the actuation required for their tasks.
- With the interpretation (see Section II) of stalks as local state-spaces, 0-cochains as global state-spaces, global sections as

global coordination constraints, and edge potentials as agent-to-agent coordination objectives, we synthesize a homological program $P = (V, E, \mathcal{F}, \{U_e\}, \{f_i\})$ describing the above complex coordination problem.

a) *Stalks*: The cellular sheaf \mathcal{F} is defined over the communication graph $G = (V, E)$. For each node $i \in V$, the vertex stalk is assigned based on the vehicle type:

$$\mathcal{F}(i) = \begin{cases} \mathbb{R}^3 \oplus \mathbb{R}^3 & \text{if } i \in V_{\text{UAV}} \cup V_{\text{UUV}} \\ \mathbb{R}^2 \oplus \mathbb{R}^2 & \text{if } i \in V_{\text{USV}} \end{cases}.$$

For each edge $e = ij \in E$, the edge stalk is defined based on the type of connection. The edge stalks are:

$$\mathcal{F}(e) = \begin{cases} \mathbb{R}^3 \oplus \mathbb{R}^3 & \text{if } e \in E_{\text{UAV}} \cup E_{\text{UUV}} \\ \mathbb{R}^2 & \text{if } e \in E_{\text{USV}} \\ \mathbb{R}^3 & \text{if } e \in E_{\text{UAV-USV}} \\ \mathbb{R}^3 & \text{if } e \in E_{\text{UUV-USV}} \end{cases}$$

Restriction maps: In this case study, restriction maps serve as an interface between agent types as well as communication modalities. Thus, restriction maps $\mathcal{F}_{i \triangleleft e} : \mathcal{F}(i) \rightarrow \mathcal{F}(e)$ project the state of each agent to the shared edge state-space:

- For $e = ij \in E_{\text{UAV}}$:

$$\mathcal{F}_{i \triangleleft e}(x_i) = \begin{bmatrix} p_i \\ v_i \end{bmatrix}, \quad \mathcal{F}_{j \triangleleft e}(x_j) = \begin{bmatrix} p_j \\ v_j \end{bmatrix}$$

$$\text{so } y_e = (\delta \mathbf{x})_e = \begin{bmatrix} p_i - p_j \\ v_i - v_j \end{bmatrix} \in \mathbb{R}^3 \oplus \mathbb{R}^3.$$

- For $e = ij \in E_{\text{UUV}}$:

$$\mathcal{F}_{i \triangleleft e}(x_i) = \begin{bmatrix} p_i \\ v_i \end{bmatrix}, \quad \mathcal{F}_{j \triangleleft e}(x_j) = \begin{bmatrix} p_j \\ v_j \end{bmatrix}$$

$$\text{so } y_e = (\delta \mathbf{x})_e = \begin{bmatrix} p_i - p_j \\ v_i - v_j \end{bmatrix} \in \mathbb{R}^3 \oplus \mathbb{R}^3.$$

- For $e = ij \in E_{\text{USV}}$:

$$\mathcal{F}_{i \triangleleft e}(x_i) = \begin{bmatrix} p_{i,x} \\ p_{i,y} \end{bmatrix}, \quad \mathcal{F}_{j \triangleleft e}(x_j) = \begin{bmatrix} p_{j,x} \\ p_{j,y} \end{bmatrix}$$

$$\text{so } y_e = (\delta \mathbf{x})_e = \begin{bmatrix} p_{i,x} - p_{j,x} \\ p_{i,y} - p_{j,y} \end{bmatrix} \in \mathbb{R}^2.$$

- For $e = ij \in E_{\text{UAV-USV}}$:

$$\mathcal{F}_{i \triangleleft e}(x_i) = p_i, \quad \mathcal{F}_{j \triangleleft e}(x_j) = \begin{bmatrix} p_{j,x} \\ p_{j,y} \\ 0 \end{bmatrix}$$

$$\text{so } y_e = (\delta \mathbf{x})_e = p_i - \begin{bmatrix} p_{j,x} \\ p_{j,y} \\ 0 \end{bmatrix} \in \mathbb{R}^3.$$

- For $e = ij \in E_{\text{UUV-USV}}$:

$$\mathcal{F}_{i \triangleleft e}(x_i) = p_i, \quad \mathcal{F}_{j \triangleleft e}(x_j) = \begin{bmatrix} p_{j,x} \\ p_{j,y} \\ 0 \end{bmatrix}$$

$$\text{so } y_e = (\delta \mathbf{x})_e = p_i - \begin{bmatrix} p_{j,x} \\ p_{j,y} \\ 0 \end{bmatrix} \in \mathbb{R}^3.$$

Edge Potentials: Edge potentials encode heterogeneous coordination objectives. Edge potentials $\{U_e\}$ are defined as follows:

- For $e = ij \in E_{\text{form}} \subseteq E_{\text{USV}}$:

$$U_e(y_e) = \frac{1}{2} \|y_e - \hat{p}_{i,j}\|_2^2$$

where $\hat{p}_{i,j} \in \mathbb{R}^2$ is the desired displacement. For edges $e \in E_{\text{USV}} \setminus E_{\text{form}}$ not involved in formation, we set $U_e(y_e) = 0$.

- For $e = ij \in E_{\text{UAV}}$:

$$U_e(y_e) = \frac{1}{2} \|v_i - v_j\|_2^2.$$

- For $e = ij \in E_{\text{UUV}}$:

$$U_e(y_e) = \frac{1}{2} \|v_i - v_j\|_2^2 + \frac{1}{2} (\|p_i - p_j\|_2^2 - r_{\text{UUV}}^2)^2$$

where $r_{\text{UUV}} > 0$ is the desired UUV-UUV communication distance.

- For $e = ij \in E_{\text{UAV-USV}}$:

$$U_e(y_e) = \frac{1}{2} (\|y_e\|_2 - r_{\text{UAV-USV}})^2$$

where $r_{\text{UAV-USV}}$ is the desired UAV-USV communication distance.

- For $e = ij \in E_{\text{UUV-USV}}$:

$$U_e(y_e) = \frac{1}{2} (\|y_e\|_2^2 - r_{\text{UUV-USV}^2})^2$$

where $r_{\text{UUV-USV}}$ is the desired UUV-USV communication distance.

Objective Functions: For each agent $i \in V$, the objective is to minimize the control effort over a time horizon T , subject to dynamics and coordination constraints:

$$f_i(x_i, u_i) = \sum_{t=1}^T \|u_i^t\|_2^2$$

where $u_i^t \in \mathbb{R}^3$ (for UAVs and UUVs) or \mathbb{R}^2 (for USVs) is the control input at time t , and the optimization is subject to the dynamics (Eq. 11), and the coordination constraints encoded by the sheaf \mathcal{F} and potentials $\{U_e\}$.

VI. NUMERICAL EXAMPLES

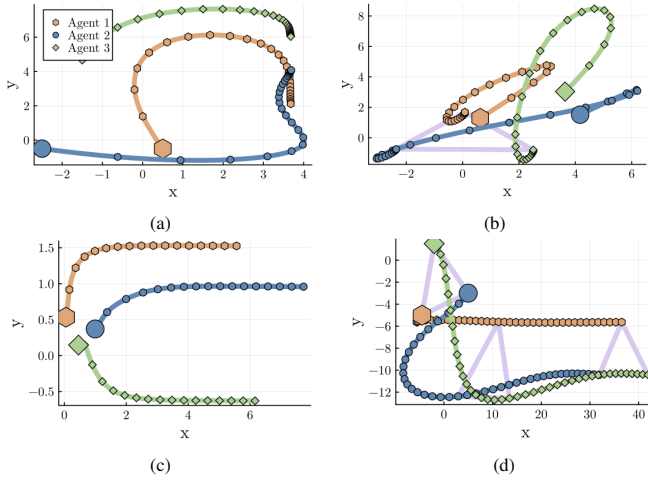


Fig. 2: (a) Agents perform a hybrid goal of consensus in x and tracking in y . (b) Agents perform the goal of reaching a triangular formation centered at the origin. (c) Agents perform the goal of flocking. (d) Agents perform a moving formation goal. In (c) and (d), the leader agent (1) tracks a constant rightward velocity vector.

To demonstrate the broad applicability of our framework, we implemented representative multi-agent optimal control examples of consensus, formation, and flocking. We implemented our ADMM algorithm for nonlinear homological programs in the Julia programming language, using JuMP.jl with IPOPT to solve the individual agent subproblems and the conjugate gradient method from Krylov.jl to compute the projection operations [36]–[38]. We implemented each example in a model predictive control setup, where at each timestep, the controllers for each agent solve the relevant optimal control homological program using Algorithm 1, implement the first control input, and repeat the process until convergence to the coordination goal.

The multi-agent system used for each example involved three vehicles obeying discrete double integrator dynamics in the plane ($d = 2$), i.e. the Euler integration of Eq. 11. Thus, each agent's state space is \mathbb{R}^4 and their control space is \mathbb{R}^2 . Each dimension of each agent's control input was constrained to the range $[-2, 2]$. Each agent's optimal control problem spanned a time horizon of $T = 10$. We ran only 10 iterations of Algorithm 1 for each optimization step in MPC. This was sufficient for our MPC controllers to achieve convergence to the coordination goals while keeping the computation time low. In all experiments, each agent's initial positions and velocities were chosen randomly from the range $[-5, 5]$. The code to reproduce the experiments can be found in our package AlgebraicOptimization.jl

a) Consensus: For this example, the agents are to reach consensus in the x -axis while reaching individual tracking goals in the y -axis. Thus each agent's subproblem is a linear quadratic tracking problem in y -space with a quadratic cost on the control activation. The coordination sheaf is defined over a fully connected communication topology, where each restriction map is simply the projection onto the x coordinate. Edge potentials are the standard norm squared potential function encoding the consensus goal. The results of this controller run for 100 iterations are shown in Fig. 2.a.

b) Stationary Formation: For this example, the agents reach an isosceles triangle formation around the origin. As such, each agent's subproblem is a standard linear quadratic regulation problem to drive the state to 0 with a quadratic penalty on control activation. The coordination sheaf is over a fully connected communication topology with the restriction maps projecting onto the position components of the state vector. The formation goal is encoded using edge potential functions of the form $\phi_e(y) = (1/2)\|y - b_e\|_2^2$ for desired b_e . The results of this controller run for 100 iterations are shown in Fig. 2.b.

c) Flocking: For this example, agents implement the standard flocking goal of reaching consensus in velocities while staying a fixed distance away from all other agents. The constant sheaf \mathbb{R}^4 on a fully connected communication topology along with potential functions summing the standard consensus potential function on the velocity components and the fixed distance potential function with $r^2 = 5$ on the position components. Each agents' objective is to minimize total control activation. Additionally, a designated leader agent tracks a constant rightward velocity vector. The results of this controller run for 65 iterations are shown in Fig. 2.c. Computing the distance between each agent confirms that they reached the desired pairwise distance of $\sqrt{5}$.

d) Moving Formation: This example combines a formation goal in positions with a consensus goal in velocities. As such, the coordination sheaf is the constant sheaf \mathbb{R}^4 on the three vertex path graph. This encodes a leader-follower topology with the middle agent in the path acting as the leader. The leader's objective is to track the velocity vector $[3, 0]$ (rightward with constant velocity) and minimize its control actuation while the followers' objectives are to simply minimize control actuation. The edge potential

functions are of the form $\phi_e(y) = (1/2)\|y - b_e\|_2^2$ where the velocity coordinates of each b_e are 0 encoding consensus in velocity while the position coordinates are chosen to achieve a fixed displacement between the leader and its followers. The results of this controller run for 160 iterations are shown in Fig. 2.d.

VII. FUTURE WORK

The results presented here can be extended in many directions. Having proposed a pipeline to translate coordination problems into homological programs and solve them with ADMM, a next step would be correct-by-construction software synthesis given user-specified coordination specifications. Other promising areas for further developments include incorporating sheaf models of coordination into graph neural network architectures (see [39]–[41]) in order to solve agent-level or system-level tasks. While we attended primarily to convex potential functions, there is a rich theory to be developed for non-differentiable or non-convex potentials and the pursuit of local minima.

APPENDIX

Proof of Theorem 1. Because each f_i is convex and the sum of convex functions is convex, the objective is clearly convex. It remains to show that the constraint set $\mathcal{C} = \{x \mid L_{\mathcal{F}}^{\nabla U} x = 0\}$ is convex. For this, we utilize the fact that $L_{\mathcal{F}}^{\nabla U}$ is the gradient of the function $g(x) = U(\delta_{\mathcal{F}} x)$. This function is clearly differentiable and is convex because the precomposition of a convex function with a linear map is convex ([42], Theorem 5.7). Thus we can apply Theorem 23.5 of [42] to obtain the following

$$x' = \nabla g(x) \iff x \in \partial g^*(x'),$$

where ∂ denotes the subdifferential and g^* is the Legendre transform of g . Instantiating the above with $x' = 0$, we see that the set $\{x \mid \nabla g(x) = 0\}$ is equivalent to the set $\{x \mid x \in \partial g^*(0)\}$. The set of subgradients of a convex function at any point is closed and convex, therefore, the set \mathcal{C} is closed and convex. So P asks to minimize a convex function restricted to a closed convex subset of its domain and is therefore a convex optimization problem. \square

Proof of Theorem 2. For notational convenience, we let $L = L_{\mathcal{F}}^{\nabla U}$ and $\delta = \delta_{\mathcal{F}}$ for the remainder of the proof. First note that plainly b is the unique global minimizer of U , and $U(b) = 0$. Thus $\nabla U(y) = 0$ only when $y = b$. We know that $L = \nabla(U \circ \delta) = \delta^{\top} \circ \nabla U \circ \delta$. This entails that \dot{x} is always in $\text{im } \delta^{\top}$ and therefore orthogonal to $\ker \delta$. We thus decompose x as $x^{\parallel} + x^{\perp}$ where x^{\parallel} is the orthogonal projection onto $\ker \delta = H^0(G; \mathcal{F})$ and focus on the evolution of x^{\perp} under the dynamics restricted to $\text{im } \delta^{\top}$.

Let $\Psi(x^{\perp}) = U(\delta x^{\perp})$. Because b is the unique minimizer of U with $U(b) = 0$, Ψ vanishes only at $\delta x^{\perp} = b$. Moreover, $\delta x^{\perp} = b$ holds if and only if $x^{\perp} = \delta^+ b$. To show the first direction of this implication, we have

$$\delta x^{\perp} = b \implies \delta^+ \delta x^{\perp} = \delta^+ b \implies x^{\perp} = \delta^+ b,$$

where the second implication holds because $\delta^+ \delta$ is projection onto $\text{im } \delta^{\top}$ and is therefore the identity on x^{\perp} . For the reverse direction, we have

$$x^{\perp} = \delta^+ b \implies \delta x^{\perp} = \delta \delta^+ b \implies \delta x^{\perp} = b,$$

where the second implication holds because $\delta \delta^+$ is projection onto $\text{im } \delta$ and is therefore the identity on b .

So, $\Psi(x^{\perp})$ vanishes only when $x^{\perp} = \delta^+ b$ implying that $\nabla \Psi(x^{\perp}) = L(x^{\perp}) = 0$ if and only if $x^{\perp} = \delta^+ b$ and that $\Psi(x^{\perp})$ is globally positive definite about $\delta^+ b$. Furthermore, radial unboundedness of U implies radial unboundedness of $\Psi(x^{\perp})$. Finally, we see that

$$\dot{\Psi} = \langle \nabla \Psi(x^{\perp}), \dot{x}^{\perp} \rangle = \langle L(x^{\perp}), -\alpha L(x^{\perp}) \rangle \leq 0$$

with equality only when $x^{\perp} = \delta^+ b$. This shows the $\dot{\Psi}$ is globally negative definite around $\delta^+ b$, meaning Ψ is a global Lyapunov function about $\delta^+ b$ for the dynamics in Eq. 10 restricted to $\text{im } \delta^{\top}$. Therefore $\delta^+ b$ is globally asymptotically stable meaning $x^{\perp} \rightarrow \delta^+ b$. For the unrestricted dynamics in Eq. 10, this means

$$\lim_{t \rightarrow \infty} x(t) = x^{\parallel}(0) + \delta^+ b,$$

which is orthogonal projection of $x(0)$ onto $\delta^+ b + \ker \delta = \delta^+ b + H^0(G; \mathcal{F})$.

To finish the proof, it remains to show that $\delta^+ b + H^0(G; \mathcal{F}) = \ker L$. For the first direction, let $z \in \delta^+ b + \ker \delta$. Then we know z is of the form $\delta^+ b + x$ for some x such that $\delta x = 0$. We have

$$\begin{aligned} L(z) &= \delta^{\top} (\nabla U(\delta(\delta^+ b + x))) \\ &= \delta^{\top} (\nabla U(\delta \delta^+ b + \delta x)) = \delta^{\top} (\nabla U(b)) = 0, \end{aligned}$$

where the third equality holds because $\delta \delta^+$ is projection onto $\text{im } \delta$ and $x \in \ker \delta$. For the reverse direction, let $L(x) = 0$ for some x . We know that this only holds when $\delta x = b$ which implies that $\delta^+ \delta x = \delta^+ b$. Because $\delta^+ \delta$ is projection onto $\text{im } \delta^{\top} = (\ker \delta)^{\perp}$, we know that x must be of the form $\delta^+ b + x'$ for some $x' \in \ker \delta$. This completes the proof. \square

Proof of Theorem 3. This setup satisfies the criterion for convergence in [35, §3.2.1]. \square

REFERENCES

- [1] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The Robotarium: A remotely accessible swarm robotics research testbed," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1699–1706, May 2017.
- [2] E. A. Lee, "Cyber Physical Systems: Design Challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pp. 363–369, May 2008. ISSN: 2375-5261.
- [3] N. Matni, A. D. Ames, and J. C. Doyle, "A Quantitative Framework for Layered Multirate Control: Toward a Theory of Control Architecture," *IEEE Control Systems Magazine*, vol. 44, pp. 52–94, June 2024. Conference Name: IEEE Control Systems Magazine.
- [4] G. Zardini, A. Censi, and E. Frazzoli, "Co-Design of Autonomous Systems: From Hardware Selection to Control Synthesis," in *2021 European Control Conference (ECC)*, pp. 682–689, June 2021.
- [5] A. Mahajan, N. C. Martins, M. C. Rotkowitz, and S. Yüksel, "Information structures in optimal decentralized control," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 1291–1306, 2012.

- [6] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [7] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [8] A. Franci, M. Golubitsky, I. Stewart, A. Bizyaeva, and N. E. Leonard, "Breaking Indecision in Multiagent, Multioption Dynamics," *SIAM Journal on Applied Dynamical Systems*, vol. 22, no. 3, pp. 1780–1817, 2023.
- [9] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, 2001.
- [10] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.
- [11] H. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents, part I: fixed topology," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, vol. 2, pp. 2010–2015 Vol.2, 2003.
- [12] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
- [13] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.
- [14] Y. Zheng and Q. Liu, "A review of distributed optimization: Problems, models and algorithms," *Neurocomputing*, vol. 483, pp. 446–459, Apr. 2022.
- [15] R. Raffard, C. Tomlin, and S. Boyd, "Distributed optimization for cooperative agents: application to formation flight," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 3, pp. 2453–2459 Vol.3, 2004.
- [16] A. D. Shepard, *A Cellular Description of the Derived Category of a Stratified Space*. PhD thesis, Brown University, 1985.
- [17] J. M. Curry, *Sheaves, cosheaves and applications*. Ph.D., University of Pennsylvania, 2014.
- [18] J. Hansen and R. Ghrist, "Toward a spectral theory of cellular sheaves," *Journal of Applied and Computational Topology*, vol. 3, no. 4, pp. 315–358, 2019.
- [19] M. H. Trinh, C. Van Nguyen, Y.-H. Lim, and H.-S. Ahn, "Matrix-weighted consensus and its applications," *Automatica*, vol. 89, pp. 415–419, Mar. 2018.
- [20] M. H. Trinh, D. Van Vu, Q. V. Tran, and H.-S. Ahn, "Matrix-Scaled Consensus," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 346–351, Dec. 2022. ISSN: 2576-2370.
- [21] R. Ghrist and H. Riess, "Cellular sheaves of lattices and the Tarski Laplacian," *Homology, Homotopy and Applications*, vol. 24, pp. 325–345, Apr. 2022. Publisher: International Press of Boston.
- [22] J. Hansen and R. Ghrist, "Opinion Dynamics on Discourse Sheaves," *SIAM Journal on Applied Mathematics*, vol. 81, no. 5, pp. 2033–2060, 2021.
- [23] G. M. Kearney, K. F. Palmowski, and M. Robinson, "Sheaf-theoretic framework for optimal network control," Nov. 2020. arXiv:2012.00120 [math].
- [24] G. Zardini, D. I. Spivak, A. Censi, and E. Frazzoli, "A Compositional Sheaf-Theoretic Framework for Event-Based Systems (Extended Version)," *Electronic Proceedings in Theoretical Computer Science*, vol. 333, pp. 139–153, 2021.
- [25] H. Riess and R. Ghrist, "Diffusion of Information on Networked Lattices by Gossip," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 5946–5952, 2022.
- [26] H. Riess, M. Munger, and M. M. Zavlanos, "Max-Plus Synchronization in Decentralized Trading Systems," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 221–227, Dec. 2023.
- [27] M. Robinson, "Understanding networks and their behaviors using sheaf theory," in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 911–914, Dec. 2013.
- [28] T. Hanks, M. Klawonn, E. Patterson, M. Hale, and J. Fairbanks, "A compositional framework for first-order optimization," *arXiv preprint arXiv:2403.05711*, 2024.
- [29] B. She, T. Hanks, J. Fairbanks, and M. Hale, "Characterizing compositionality of lqr from the categorical perspective," 2023.
- [30] T. Hanks, B. She, M. Hale, E. Patterson, M. Klawonn, and J. Fairbanks, "Modeling model predictive control: A category theoretic framework for multistage control problems," *arXiv preprint arXiv:2305.03820*, 2024.
- [31] R. Ghrist and S. Krishnan, "Positive Alexander Duality for Pursuit and Evasion," *SIAM Journal on Applied Algebra and Geometry*, vol. 1, no. 1, pp. 308–327, 2017.
- [32] J. Hansen and R. Ghrist, "Distributed Optimization with Sheaf Homological Constraints," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019.
- [33] B. Eckmann, "Harmonische Funktionen und Randwertaufgaben in einem Komplex," *Commentarii Mathematici Helvetici*, vol. 17, no. 1, pp. 240–255, 1944.
- [34] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.
- [35] S. Boyd, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.
- [36] I. Dunning, J. Huchette, and M. Lubin, "Jump: A modeling language for mathematical optimization," *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.
- [37] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Prog.*, vol. 106, pp. 25–57, 2006.
- [38] A. Montoisson and D. Orban, "Krylov.jl: A julia basket of hand-picked krylov methods," *Journal of Open Source Software*, vol. 8, no. 89, p. 5187, 2023.
- [39] O. Zaghen, A. Longa, S. Azzolin, L. Telyatnikov, A. Passerini, and P. Lio, "Sheaf Diffusion Goes Nonlinear: Enhancing GNNs with Adaptive Sheaf Laplacians," in *ICML 2024 Workshop GRaM*, 2024.
- [40] C. Bodnar, F. Di Giovanni, B. Chamberlain, P. Lió, and M. Bronstein, "Neural Sheaf Diffusion: A Topological Perspective on Heterophily and Oversmoothing in GNNs," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 18527–18541, Dec. 2022.
- [41] J. Hansen and T. Gebhart, "Sheaf Neural Networks," in *NeuralIPS 202 Workshop TDA and Beyond*, 2020.
- [42] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.