Model-Free Practical Cooperative Control for Diffusively Coupled Systems

Miel Sharf, *Member, IEEE*, Anne Koch, *Graduate Student Member, IEEE*, Daniel Zelazo, *Senior Member, IEEE*, and Frank Allgöwer, *Member, IEEE*

***Abstract*—In this article, we develop a data-based controller design framework for diffusively coupled systems with guaranteed convergence to an ϵ-neighborhood of the desired formation. The controller is composed of a fixed controller with an adjustable gain on each edge. Via passivity theory and network optimization, we not only prove that there exists a gain attaining the desired formation control goal, but we present a data-based method to find an upper bound on this gain. Furthermore, by allowing for additional experiments, the conservatism of the upper bound can be reduced via iterative sampling schemes. The introduced scheme is based on the assumption of passive systems, which we relax by discussing different methods for estimating the systems' passivity shortage, as well as applying transformations passivizing them. Finally, we illustrate the developed model-free cooperative control scheme with a case study.**

***Index Terms*—Asymptotic stability, convex functions, data models, iterative learning control, iterative methods, multiagent systems, networked control systems, nonlinear dynamical systems, optimization, sampling methods.**

I. Introduction

Multiagent systems have received extensive attention in the past years, due to their appearance in many fields of engineering, exact sciences, and social sciences. Examples include robotics [1], traffic engineering [2], and ecology [3]. The state-of-the-art approach to model-based control for multiagent systems offers rigorous stability analysis, performance guarantees, and systematic insights into the considered problem. However, with the growing complexity of systems, the modeling process is reaching its limits. Obtaining a reliable mathematical model of the agents becomes a time-intensive and arduous task.

At the same time, modern technology allows for gathering and storing more and more data from systems and processes, inciting an increasing interest in *data-driven control*. There are two main approaches for data-driven control. The first is model-based data-driven control, which uses data to identify a model from the problem, which is in turn used to solve the synthesis problem [4], [5]. In this case, the model estimation errors must be taken into account when solving the synthesis problem. The second is model-free control, which does not try to estimate a model for the system. The latter can be further bisected into approximate dynamic programming methods and direct policy search. The former evaluates a score for each state-action pair, and then obtains an optimal control policy using dynamic programming [6], and the latter tries to find the optimal policy directly, e.g., by gradient descent or via a nonparametric description of the possible trajectories [7]. These methods have all been applied to multiagent systems as well, with varying degrees of success [6], [8]–[10].

In this article, we develop a data-driven controller synthesis approach for multiagent systems, which comes with rigorous theoretical analysis and stability guarantees for the closed loop, with almost no assumptions on the agents and few measurements needed. Our approach is based on high-gain control as well as passivity. Some ideas on high-gain approaches to cooperative control can be found in [11] and references therein. Zheng *et al.* [12] provide a high-gain condition in the design of distributed H_∞ controllers for platoons with undirected topologies, while there are also many approaches to (adaptively) tune the coupling weights (e.g. [13]). Our approach provides an upper bound on a high-gain controller using passivity measures. Passivity properties of the components can provide sufficient abstractions of their detailed dynamical models for guaranteed control. Such passivity properties can be obtained from data as ongoing work shows (e.g., [14]–[16]).

Passivity is a well-known tool for controller synthesis [17], which is useful, beyond convergence analysis, for its powerful properties such as compositionality. It was first introduced in the field of multiagent systems in the seminal works of Arcak [18], [19], and was since explored in many variants in the context of multiagent systems in many other works [1], [20]–[26]. We focus on the variant known as maximal equilibrium independent passivity (MEIP), presented in [22]. The notion of MEIP

Manuscript received February 18, 2020; revised September 25, 2020; accepted January 29, 2021. Date of publication February 2, 2021; date of current version January 28, 2022. This work was supported by the German-Israeli Foundation for Scientific Research and Development. Recommended by Associate Editor M. K. Camlibel. *(Corresponding author: Miel Sharf.)*

Miel Sharf is with the Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm 2633513, Sweden (e-mail: sharf@kth.se).

Anne Koch and Frank Allgöwer are with the Institute for Systems Theory and Automatic Control, University of Stuttgart, Stuttgart 70569, Germany (e-mail: anne.koch@ist.uni-stuttgart.de; frank.allgower@ist.uni-stuttgart.de).

Daniel Zelazo is with the Faculty of Aerospace Engineering, Israel Institute of Technology, Haifa 32000, Israel (e-mail: dzelazo@technion.ac.il).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2021.3056582.

Digital Object Identifier 10.1109/TAC.2021.3056582

establishes a connection between multiagent systems and network optimization (see [22]–[26]). Different synthesis problems have been solved using this network optimization framework assuming an exact model for each of the agents exists [23], [24], [27]. More precisely, one needs a perfect description of the steady-state input–output behavior of the agents. Thus, these methods cannot be applied in our case.

As we said, this article generally studies the problem of controller synthesis for diffusively coupled systems. The control objective is to converge to an ϵ-neighborhood of a constant prescribed relative output vector. That is, for some tolerance $\epsilon > 0$, we aim to design controllers so that the steady-state limit of the relative output is ϵ-close to the prescribed values. The related problem of practical synchronization of multiagent systems has been considered in [28], in which the agents were assumed to be known up to some bounded additive disturbance. However, a nominal model was needed to get practical synchronization. It was also pursued in [29], where strong coupling was used to drive agents close to a common trajectory, but again, a model for the agents was needed.

Our contributions are as follows. We present a model-free data-driven method for solving the practical formation control problem. This is done by cascading a fixed controller and an adjustable gain on each edge. We show that this gain can be chosen to guarantee a solution to the practical formation control problem. We then provide schemes to compute this gain offline only from input–output data without any model of the agents. In fact, this gain can be computed from only three experiments (per agent), and it can become less conservative with further data samples. If iterative experiments can be performed, we also provide an approach for applying different gains over different edges to further reduce any conservatism. We survey the different advantages for each of the methods and discuss their applicability in terms of the number of required measurements, or trade-offs in terms of energy. We also provide simulations presenting the effectiveness of the presented model-free control methods. To the best of the authors' knowledge, no prior works consider data-driven control of multiagent systems using passivity. Furthermore, this is the first application of the network optimization framework of [22], [24] where the agents do not have an exact model.

Notations: We employ notions from algebraic graph theory [30]. An undirected graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ consists of finite sets of vertices $\mathbb{V}$ and edges $\mathbb{E} \subset \mathbb{V} \times \mathbb{V}$. We denote the edge having ends i and j in $\mathbb{V}$ by $k = \{i, j\} \in \mathbb{E}$. For each edge k, we pick an arbitrary orientation and denote $k = (i, j)$. The incidence matrix $\mathcal{E} \in \mathbb{R}^{|\mathbb{E}|\times|\mathbb{V}|}$ of $\mathcal{G}$ is defined such that for an edge $k = (i, j) \in \mathbb{E}$, we have $[\mathcal{E}]_{ik} = +1$, $[\mathcal{E}]_{jk} = -1$, and $[\mathcal{E}]_{\ell k} = 0$ for $\ell \neq i, j$. $\mathrm{diam}(\mathcal{G})$ denotes the diameter of $\mathcal{G}$.

We also use notions from linear algebra. For every vector $v \in \mathbb{R}^n$, $\mathrm{diag}(v)$ denotes the $n \times n$ diagonal matrix with the entries of v on its diagonal. The image of any linear map T between vector spaces will be denoted by $\mathrm{Im}(T)$. Also, for two sets $A, B \subseteq \mathbb{R}^d$, we define $A + B = \{a + b : \ a \in A, \ b \in B\}$. Furthermore, $\|\cdot\|$ is the Euclidean norm.

Lastly, if Σ is a dynamical system, and M is a linear map of appropriate dimension, we can consider the cascaded system of Σ and M. The cascade of Σ after M is denoted ΣM, and the cascade of Σ before M is denoted $M\Sigma$.

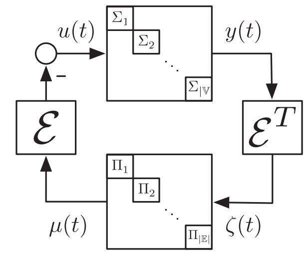

Fig. 1. Block-diagram of the diffusively coupled network $(\Sigma, \Pi, \mathcal{G})$.

II. Background: Network Optimization and Passivity in Cooperative Control

Our goal in this subsection is to describe the diffusive coupling networks studied in [22], and to present the passivity-based network optimization framework achieved for multiagent systems. See also [23], [24].

A. Diffusively Coupled Systems and Steady-State Relations

Diffusively coupled networks are composed of agents $\{\Sigma_i\}_{i\in\mathbb{V}}$ interacting over a graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ using edge controllers $\{\Pi_e\}_{e\in\mathbb{E}}$. Each vertex $i \in \mathbb{V}$ represents an agent and each edge $e \in \mathbb{E}$ represents a controller. We model them as SISO dynamical systems

$$\Sigma_i : \begin{cases} \dot{x}_i = f_i(x_i, u_i) \\ y_i = h_i(x_i, u_i) \end{cases}, \Pi_e : \begin{cases} \dot{\eta}_e = \phi_e(\eta_e, \zeta_e) \\ \mu_e = \psi_e(\eta_e, \zeta_e) \end{cases} \quad (1)$$

where the agents' state, input, and output are x_i, u_i, and y_i, respectively, and the controllers' state, input, and output are η_e, ζ_e, and μ_e, respectively. To understand the coupling of these systems, we consider the stacked inputs and outputs of the agents and controllers as $y = [y_1, \ldots, y_{|\mathbb{V}|}]^T$, and similarly for u, ζ, μ. The system is connected via the relations $\zeta = \mathcal{E}^T y$ and $u = -\mathcal{E}\mu$, where $\mathcal{E}$ is the incidence matrix of the graph $\mathcal{G}$. In other words, if we stack all agents to a dynamical system Σ, and stack all controllers to a dynamical system Π, the closed-loop is the feedback connection of Σ and $\mathcal{E}\Pi\mathcal{E}^T$. See Fig. 1 for an illustration of the network, which we will denote by $(\mathcal{G}, \Sigma, \Pi)$.

We will be interested in steady states of the closed-loop system. It is clear that if the stacked vectors $(\mathrm{u}, \mathrm{y}, \zeta, \mu)$ are a steady state for $(\mathcal{G}, \Sigma, \Pi)$, then for every vertex $i \in \mathbb{V}$, $(\mathrm{u}_i, \mathrm{y}_i)$ is a steady-state input–output pair for the system Σ_i, and for every edge, $e \in \mathbb{E}$, (ζ_e, μ_e) is a steady-state input–output pair for the system Π_e. This motivates the exploration of *steady-state input–output relations*, first defined in [22].

Definition 1: The *steady-state relation* of a system is a set containing all the steady-state input–output pairs of the system.

We will denote the steady-states relations of Σ_i, Π_e, Σ, and Π as k_i, γ_e, k, and γ, accordingly.

Remark 1: We will sometimes abuse the notation and consider this relation as a set-valued map. Indeed, for any input u, we can define the set $k(\mathrm{u})$ by $k(\mathrm{u}) = \{\mathrm{y} :\ (\mathrm{u},\mathrm{y}) \in k\}$, and similarly for k_i, γ_e, and γ. We also consider the inverse relation k^{-1} as the set-valued map assigning to a steady-state output y the set $k^{-1}(\mathrm{y}) = \{\mathrm{u} :\ \mathrm{y} \in k(\mathrm{u})\}$, i.e., the set of all steady-state inputs corresponding to the steady-state output y. We define this similarly for k_i, γ_e, and γ.

Thus, $(\mathrm{u},\mathrm{y},\zeta,\mu)$ is a steady state of $(\mathcal{G},\Sigma,\Pi)$ if and only if $\mathrm{y} \in k(\mathrm{u}), \mu \in \gamma(\zeta), \zeta = \mathcal{E}^T\mathrm{y}$, and $\mathrm{u} = -\mathcal{E}\mu$. Equivalently, y is a steady-state output of $(\mathcal{G},\Sigma,\Pi)$ if and only if the zero vector 0 lies in the set $k^{-1}(\mathrm{y}) + \mathcal{E}\gamma(\mathcal{E}^T\mathrm{y})$ [22], [23].

B. Maximum Equilibrium-Independent Passivity and the Network Optimization Framework

The main tool allowing us to connect multiagent systems to the network optimization world is monotone relations.

Definition 2: A steady-state relation is *monotone* if for any two points $(\mathrm{u}_1,\mathrm{y}_1)$ and $(\mathrm{u}_2,\mathrm{y}_2)$ in the relation, $u_1 < u_2$ implies $y_1 \le y_2$. We say that a monotone relation is *maximally monotone* if it is not contained in a larger monotone relation.

In order to connect this definition to the system-theoretic world, we define the following variant of passivity:

Definition 3. ([22]): A SISO system is said to be (output-strictly) *maximum equilibrium-independent passive* (MEIP) if the following two conditions hold.

i) The system is (output-strictly) passive with respect to any steady-state input–output pair it has.

ii) Its steady-state relation is maximally monotone.

One important property of maximally monotone relations is that they are subgradients of convex functions [31]. In this direction, we assume that the agents and controllers of the diffusively coupled network $(\mathcal{G},\Sigma,\Pi)$ are MEIP. Let K_i and Γ_e be the corresponding convex integral functions for the steady-state relations k_i, γ_e. In other words, $\partial K_i = k_i$ and $\partial\Gamma_e = \gamma_e$, where ∂ denotes the subdifferential of convex functions [31]. We shall denote $K = \sum_{i\in\mathbb{V}} K_i$ and $\Gamma = \sum_{e\in\mathbb{E}} \Gamma_e$, so that $\partial K = k$ and $\partial\Gamma = \gamma$. The dual functions of K_i, Γ_e, K, Γ are defined using the Legendre transform, $K^\star(\mathrm{y}) = \sup_{\mathrm{u}}\{\mathrm{u}^T\mathrm{y} - K(\mathrm{u})\} = -\inf_{\mathrm{u}}\{K(\mathrm{u}) - \mathrm{u}^T\mathrm{y}\}$, and similarly for $K_i^\star, \Gamma_e^\star$, and $\Gamma^\star$. We note that $\partial K^\star = k^{-1}, \partial\Gamma^\star = \gamma^{-1}, \partial K_i^\star = k_i^{-1}$, and $\partial\Gamma_e^\star = \gamma_e^{-1}$ [31].

We now resume our interest in steady states for the diffusively coupled network $(\mathcal{G},\Sigma,\Pi)$. We recall that y was the steady-state output of the diffusively coupled network if and only if $0 \in k^{-1}(\mathrm{y}) + \mathcal{E}\gamma(\mathcal{E}^T\mathrm{y})$. Restating this result in the language of convex functions gives the following theorem.

Theorem 1 ([22]): Consider the diffusively coupled network $(\mathcal{G},\Sigma,\Pi)$. Assume all agents Σ_i are MEIP, and all controllers Π_e are output-strictly MEIP (or vice versa). Let k_i, γ_e, k, and γ be the steady-state relations of Σ_i, Π_e, Σ, and Π accordingly, and let K_i, Γ_e, K, and Γ be the corresponding convex integral functions. Then, the closed-loop system converges to a steady state $(\mathrm{u},\mathrm{y},\zeta,\mu)$, such that (y,ζ) and (u,μ) are dual optimal solutions to the following convex optimization problems:

Optimal Potential Problem		***Optimal Flow Problem***	
$\min\limits_{y,\zeta}$	$K^\star(y) + \Gamma(\zeta)$	$\min\limits_{u,\mu}$	$K(u) + \Gamma^\star(\mu)$
s.t.	$\mathcal{E}^T y = \zeta$	s.t.	$\mu = -\mathcal{E}u.$

The two network optimization problems above will be denoted often by (OPP) and (OFP), respectively. These problems are fundamental in the field of network optimization, dealing with optimization problems defined on graphs [31]. The names "optimal potential problem" and "optimal flow problem" are inspired from standard nomenclature in this field.

III. Problem Formulation

We focus on relative-output-based formation control. In this problem, the agents know the relative output $\zeta_e = y_i - y_j$ with respect to their neighbors, and the control goal is to converge to a steady state with prescribed relative outputs $\zeta_e = \mathrm{y}_i - \mathrm{y}_j$. Examples include the consensus problem, in which all outputs must agree, as well as relative-position-based formation control of robots, in which the robots are required to organize themselves in a desired spatial structure [32].

More specifically, we are given a graph $\mathcal{G}$ and agents Σ, and our goal is to design controllers Π so that the signal $\zeta(t)$ of the diffusively coupled network $(\mathcal{G},\Sigma,\Pi)$ will converge to a desired, given steady-state vector $\zeta^\star$. One evident solution is to apply a (shifted) integrator as a controller. However, this solution will not always work even when the agents are MEIP.

Example 1: Consider agents Σ_i with integrator dynamics, together with (shifted) integrator controllers Π_e, where we desire consensus (i.e., $\zeta^\star = 0$) over a connected graph $\mathcal{G}$

$$\Sigma_i : \begin{cases} \dot{x}_i = u_i \\ y_i = x_i \end{cases} \qquad \Pi_e : \begin{cases} \dot{\eta}_e = \zeta_e \\ \mu_e = \eta_e. \end{cases}$$

The trajectories of the diffusively coupled system can be understood by noting that the closed-loop system yields the second-order dynamics $\ddot{x} = -\mathcal{E}\mathcal{E}^T x$. Decomposing x using a basis of eigenvectors of the graph Laplacian $\mathcal{E}\mathcal{E}^T$, which is a positive semi-definite matrix, we see that the trajectory of $x(t)$ oscillates around the consensus manifold $\{\mathrm{x} :\ \exists\lambda \in \mathbb{R}\ \mathrm{x} = \lambda\mathbb{1}_n\}$. Specifically, $x(t) - 1/n\mathbb{1}_n^T x(t)\mathbb{1}_n = \sum_{i=2}^n c_i\cos(\sqrt{\lambda_i}t + \varphi_i)v_i$, where $\lambda_2,\ldots,\lambda_n > 0$ are the nontrivial eigenvalues of the graph Laplacian, $v_2,\ldots,v_n$ are corresponding unit-length eigenvectors, and c_i, φ_i are constants depending on the initial conditions $x(0), \eta(0)$. Thus, $x(t) = y(t)$ does not converge anywhere, let alone to consensus. Moreover, $\zeta(t)$ does not converge as $t \to \infty$, as $\mathcal{E}\zeta(t) = \mathcal{E}\mathcal{E}^T y(t) = \sum_{i=2}^n \lambda_i c_i\cos(\sqrt{\lambda_i}t + \varphi_i)v_i$. Thus, the integrator controller does not solve the formation control problem in this case.

Even if the integrator would solve this problem in general, we would like more freedom in choosing the controller. In practice, one might want to design the controller to satisfy extra requirements (like $\mathcal{H}_2$- or $\mathcal{H}_\infty$-norm minimization, or making sure that certain restrictions on the behavior of the system are not broken). We do not try and satisfy these more complex requirements, but instead show that a large class of controllers can be used to solve the practical formation control problem. In turn, this

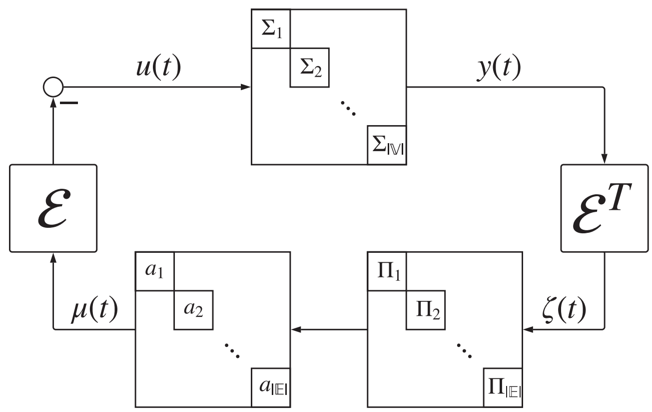

Fig. 2. Block-diagram of the diffusively coupled network $(\Sigma, \Pi, \mathcal{G}, A)$.

allows one to choose from a wide range of controllers, and try and satisfy additional desired properties. Sharf and Zelazo [23] offer an algorithm solving the problem, assuming the agents are MEIP and a perfect model of them is known. This algorithm allows a lot of freedom in the choice of controllers. However, in practice, we oftentimes have no exact model of the agents, or any closed-form model. To formalize the goals we aim at, we define the notion of *practical* formation control.

Problem 1: Given a graph $\mathcal{G}$, agents Σ, a desired formation $\zeta^\star \in \mathrm{Im}(\mathcal{E}^T)$, and an error margin ε, find a controller Π so that the relative output vector $\zeta(t)$ of the network $(\mathcal{G}, \Sigma, \Pi)$ converges to some ζ_0 such that $\|\zeta^\star - \zeta_0\| \leq \varepsilon$.

By choosing suitable error margins ε, practical formation control (compared to formation control) comprises no restriction or real drawback in any application case. Therefore, solving the practical formation control problem constitutes an interesting problem, especially for unknown dynamics of the agents. Thus, we strive to develop an algorithm solving this practical formation control problem without a model of the agents while still providing rigorous guarantees.

The underlying idea of our approach is amplifying the controller output. Consider the scenario depicted in Fig. 2, where the graph $\mathcal{G}$, the agents Σ, and the nominal controller Π are fixed, and the gain matrix A is a diagonal matrix $A = \mathrm{diag}(\{a_e\}_{e\in\mathbb{E}})$ with positive entries. We will show in the following that when the gains a_e become large enough, the controller dynamics Π become much more emphasized than the agent dynamics Σ. By correctly choosing the nominal controller Π according to $\zeta^\star$, we can hence achieve arbitrarily close formations to $\zeta^\star$, as the effect of the agents on the closed-loop dynamics will be dampened. We denote the diffusively coupled system in Fig. 2 as the 4-tuple $(\mathcal{G}, \Sigma, \Pi, A)$, or as $(\mathcal{G}, \Sigma, \Pi, a)$ where a is the vector of diagonal entries of A. In case A has uniform gains, i.e., $A = \alpha I$, we denote the system as $(\mathcal{G}, \Sigma, \Pi, \alpha \mathbb{1}_n)$. We make an assumption in order to apply the network optimization framework of Theorem 1.

Assumption 1: The agents $\{\Sigma_i\}_{i\in\mathbb{V}}$ are all MEIP, and the chosen controllers $\{\Pi_e\}_{e\in\mathbb{E}}$ are all output-strictly MEIP.

Before expanding on the suggested controller design, we discuss Assumption 1. In practice, we might not know if an agent is MEIP. Hence, we discuss how to either verify MEIP for the agents, or otherwise determine their shortage of passivity. We

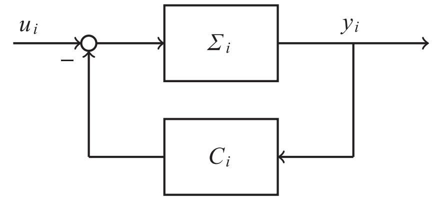

Fig. 3. Passivation of a passivity-short agent using feedback.

also discuss how to passivize the agents in the latter case. First, in some occasions, we might not know a model for the agents, but some known general structure properties. For example, one might know that an agent can be modeled by a gradient system, or a Euler–Lagrange system, but the exact model is unknown due to uncertainty on the characterizing parameters. In that case, we can use analytical results to verify MEIP. To exemplify this idea, we show how a very rough model can be used to prove that a system is MEIP.

Proposition 1: Consider a control-affine SISO system

$$\dot{x} = -f(x) + g(x)u;\ y = h(x). \tag{2}$$

Assume that g is positive, that $f/g, h : \mathbb{R} \to \mathbb{R}$ are continuous ascending functions, and that either $\lim_{|x|\to\infty} |f(x)/g(x)| = \infty$ or $\lim_{|x|\to\infty} |h(x)| = \infty$. Then, (2) is MEIP.

The proof is available in the Appendix. See also [24] for a treatment on gradient systems with oscillatory terms. Similarly, one can use a highly uncertain model to give an estimate about equilibrium-independent passivity indices.

Another approach for verifying Assumption 1 is learning input–output passivity properties from trajectories. For LTI systems, the shortage of passivity can be asymptotically revealed by iteratively probing the agents and measuring the output signal [15]. In [16], the authors showed that even one input–output trajectory (with persistently exciting input) is sufficient to find the shortage of passivity of an LTI system. For nonlinear agents, one can apply approaches presented in [14], [33], under an assumption on Lipschitz continuity of the steady-state relation. However, for general nonlinear systems, this is still a work in progress. We note that for LTI systems, output-strict passivity directly implies output strict MEIP [22].

Using either approach, we can either find that an agent is MEIP, or that it has some shortage of passivity, and we need to render the agent passive in order to apply the model-free control approaches presented in this article. We can use passivizing transformations in order to get a passive augmented agent. For example, if the agent has output-shortage of passivity $s_i > 0$, we can apply a controller $C_i : y_i \mapsto \nu_i y_i$ to the agent as in [25], with $\nu_i > s_i$, as shown in Fig. 3. It can be shown that the augmented agent is output-strictly MEIP in this case. More generally, one could deal with more complex shortages of passivity, namely simultaneous input- and output-shortage of passivity, using more complex transformations [26].

With this discussion and relaxation of Assumption 1, we return to our solution of the practical formation control problem. We considered closed-loop systems of the form $(\mathcal{G}, \Sigma, \Pi, a)$, where a is a vector of edge gains. From here, the article diverges into two sections. The next section deals with theory and analysis for uniform edge gains. The following section deals with the case of heterogeneous edge gains.

IV. Practical Formation Control With Uniform Gains

The chapter is split into two parts. The first part deals with the theory, and the second part deals with the corresponding implementation of practical formation control synthesis using uniform gains on the edges.

A. Theory

We wish to understand the effect of amplification on the steady state of the closed-loop system. For the remainder of the section, we fix a graph $\mathcal{G}$, agents Σ, and controllers Π such that Assumption 1 holds. We consider the diffusively coupled system $(\mathcal{G}, \Sigma, \Pi, \alpha\mathbb{1}_n)$ in Fig. 2, where the gains over all edges are identical and equal to $\alpha > 0$, and wish to understand the effect of α. We let K and Γ denote the sum of the integral functions of the agents and of the controllers, respectively. We first study the steady states of this diffusively coupled system.

Lemma 1: Under the assumptions above, the closed loop converges to a steady state, and the steady states y, ζ of the closed-loop system are minimizers of the following problem:

$$\begin{aligned} \min_{y,\zeta} \quad & K^\star(y) + \alpha\Gamma(\zeta) \\ \text{s.t.} \quad & \mathcal{E}^T y = \zeta. \end{aligned} \tag{3}$$

Proof: We define a new stacked controller, $\bar{\Pi} = \alpha\Pi$, by cascading the previous controller Π with the gain α. The resulting controller $\bar{\Pi}$ is again output-strictly MEIP, and we let $\bar{\gamma}, \bar{\Gamma}$ denote the corresponding steady-state input–output relation and integral function. Theorem 1 implies that the closed-loop system (with $\bar{\Pi}$) converges to minimizers of (OPP) for the system $(\mathcal{G}, \Sigma, \bar{\Pi})$. Hence, we have $\bar{\gamma}(\zeta) = \alpha\gamma(\zeta)$ for any $\zeta \in \mathbb{R}^{|\mathbb{E}|}$. Therefore, (OPP) for the system $(\mathcal{G}, \Sigma, \bar{\Pi})$ reads

$$\begin{aligned} \min_{y,\zeta} \quad & K^\star(y) + \alpha\Gamma(\zeta) \\ \text{s.t.} \quad & \mathcal{E}^T y = \zeta \end{aligned}$$

as $\bar{\Gamma} = \alpha\Gamma$. ■

Our goal is to show that when $\alpha \gg 1$, the relative output vector ζ of the diffusively coupled system $(\mathcal{G}, \Sigma, \Pi, \alpha\mathbb{1}_n)$ globally asymptotically converges to an $\varepsilon = \varepsilon(\alpha)$-ball around the minimizer of Γ, and $\lim_{\alpha\to\infty} \varepsilon(\alpha) = 0$. Thus, if we design the controllers so that Γ is minimized at $\zeta^\star$, then $\alpha \gg 1$ provides a solution to the ε-practical formation control problem. Indeed, we can prove the following theorem.

Theorem 2: Consider the closed-loop system $(\mathcal{G}, \Sigma, \Pi, \alpha\mathbb{1}_n)$, where the agents are MEIP and the controllers are output-strictly MEIP. Assume Γ has a unique minimizer in $\mathrm{Im}(\mathcal{E}^T)$, denoted ζ_1. For any $\varepsilon > 0$, there exists some $\alpha_0 > 0$, such that for all $\alpha > \alpha_0$ and for all initial conditions, the closed-loop converges to a vector y satisfying $\|\mathcal{E}^T\mathrm{y} - \zeta_1\| < \varepsilon$. In particular, if $\zeta_1 = \zeta^\star$, we solve the practical control problem.

In order to prove the theorem, we study (OPP) for the diffusively coupled system $(\mathcal{G}, \Sigma, \Pi, \alpha\mathbb{1}_n)$, as described in Lemma 1. In order to do so, we need to prove a couple of lemmas. The first deals with lower bounds on the values of convex functions away from their minimizers.

Lemma 2: Let U be a finite-dimensional vector space. Let $f : U \to \mathbb{R}$ be convex, and suppose $x_0 \in U$ is the unique minimum of f. Then, for any $\delta > 0$, there is some $M > f(x_0)$ such that for any point $x \in U$, if $f(x) < M$, then $\|x - x_0\| < \delta$.

Proof: We assume without loss of generality that $f(x_0) = 0$. Let μ be the minimum of f on the set $\{x \in U : \ ||x - x_0|| = \delta\}$, which is positive since x_0 is f's unique minimum and the set $\{x \in U : \|x - x_0\| = \delta\}$ is compact. We know that, for any $y \in U$, the difference quotient $\frac{f(x_0+\lambda y) - f(x_0)}{\lambda}$ is an increasing function of $\lambda > 0$ (see Theorem 23.1 of [31]). Manipulating this inequality shows that for any $x \in U$, $||x|| \geq \delta$ implies $f(x) \geq \frac{||x||}{\delta}\mu$, and in particular $f(x) \geq \mu$ whenever $||x|| \geq \delta$. Thus, if $f(x) < \mu$, then we must have $||x - x_0|| < \delta$, so we can choose $M = \mu$ and complete the proof. ■

The second lemma deals with minimizers of perturbed versions of convex functions on graphs.

Lemma 3: Fix a graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ and let $\mathcal{E}$ be its incidence matrix. Let $K : \mathbb{R}^{|\mathbb{V}|} \to \mathbb{R}$ be a convex function, and let $\Gamma : \mathbb{R}^{|\mathbb{E}|} \to \mathbb{R}$ be a convex function with a unique minimum ζ_1 when restricted to the set $\mathrm{Im}(\mathcal{E}^T)$. For any $\alpha > 0$, consider the function $F_\alpha(\mathrm{y}) = K^\star(\mathrm{y}) + \alpha\Gamma(\mathcal{E}^T\mathrm{y})$. Then, for any $\varepsilon > 0$, there exists some $\alpha_0 > 0$ such that if $\alpha > \alpha_0$, then all of F_α's minima, y, satisfy $\|\mathcal{E}^T\mathrm{y} - \zeta_1\| < \varepsilon$.

Proof: By subtracting constants from $K^\star$ and Γ, we may assume without loss of generality that $\min(K^\star) = \min(\Gamma) = 0$. Choose some $\mathrm{y}_0 \in \mathbb{R}^{|\mathbb{V}|}$ such that $\mathcal{E}^T\mathrm{y}_0 = \zeta_1$ and let $m = K^\star(\mathrm{y}_0)$. Note that $F_\alpha(y_0) = m$, meaning that if y is any minimum of F_α, it must satisfy $F_\alpha(\mathrm{y}) \leq m$, and in particular $\Gamma(\mathcal{E}^T\mathrm{y}) \leq \frac{m}{\alpha}$. Now, from Lemma 2, we know that there is some $M > 0$ such that if $\Gamma(\mathcal{E}^T\mathrm{y}) < M$ then $\|\mathcal{E}^T\mathrm{y} - \zeta_1\| < \varepsilon$. If we choose $\alpha_0 = \frac{m}{M}$, then whenever $\alpha > \alpha_0$, we have $\Gamma(\mathcal{E}^T\mathrm{y}) < M$, implying $\|\mathcal{E}^T\mathrm{y} - \zeta_1\| < \varepsilon$. ■

We now connect the pieces and prove Theorem 2.

Proof: Lemma 1 implies that the closed-loop system always converges to a minimizer of (3). Lemma 3 proves that there exists $\alpha_0 > 0$ such that if $\alpha > \alpha_0$, then all minimizers of (OPP) satisfy $\|\mathcal{E}^T\mathrm{y} - \zeta_1\| < \varepsilon$. This proves the theorem. ■

Remark 2: The parameters ε and $\zeta^\star$ can be used to estimate the minimal required gain α_0 by following the proofs of Lemmas 2 and 3. Namely, $\alpha_0 \leq \frac{m}{M}$, where M is the minimum of Γ on the set $\{\zeta \in \mathrm{Im}(\mathcal{E}^T) : \ \|\zeta - \zeta^\star\| = \varepsilon\}$, and $m = K^\star(\mathrm{y}_0) - \min_y K^\star(\mathrm{y}) = K^\star(\mathrm{y}^\star) + K(0)$, where $\mathrm{y}^\star \in \mathbb{R}^{|\mathbb{V}|}$ is any vector satisfying $\mathcal{E}^T\mathrm{y}^\star = \zeta^\star$.

Corollary 1: Let $(\mathcal{G}, \Sigma, \Pi, \alpha\mathbb{1}_n)$ satisfy Assumption 1 and let $(\mathcal{G}, \Sigma_{\mathrm{Int}}, \Pi)$ be a network composed of integrator dynamics for each agent. Denote the relative outputs of each system as $\zeta(t)$ and $\zeta_{\mathrm{Int}}(t)$ respectively. Then, for any $\varepsilon > 0$, there exists an $\alpha_0 > 0$ such that if $\alpha \geq \alpha_0$, then the relative outputs $\zeta(t)$ and $\zeta_{\mathrm{Int}}(t)$ both converge to constant vectors $\zeta^\star$ and $\zeta^\star_{\mathrm{Int}}$, respectively, and satisfy $\|\zeta^\star - \zeta^\star_{\mathrm{Int}}\| \leq \varepsilon$.

Proof: The agents Σ_{Int} are MEIP. Thus, by Theorem 1, we know that the diffusively coupled system $(\mathcal{G}, \Sigma_{\mathrm{Int}}, \Pi)$ converges to a steady state, and its steady-state output is a minimizer of the associated problem (OPP). Note that the input–output relation of Σ_{Int} is given via $k^{-1}(\mathrm{y}) = 0$, meaning the integral function $K^\star$ is the zero function. Thus, the associated problem (OPP) is the unconstrained minimization of $\Gamma(\mathcal{E}^T \mathrm{y})$, meaning that the system $(\mathcal{G}, \Sigma_{\text{Int}}, \Pi)$ converges, and its output converges to a minimizer of $\Gamma(\mathcal{E}^T \mathrm{y})$, i.e., its relative output $\zeta(t)$ converges to the minimizer of Γ on $\mathrm{Im}(\mathcal{E}^T)$. Applying Theorem 2 now completes the proof. ■

Remark 3 (Almost Data-Free Control): Corollary 1 can be thought of as a synthesis procedure. Indeed, we can solve the synthesis problem as if the agents were single integrators, and then amplify the controller output by a factor α. The corollary shows that for any $\varepsilon > 0$, there is a threshold $\alpha_0 > 0$ such that if $\alpha > \alpha_0$, the closed-loop system converges to an ε-neighborhood of $\zeta^\star$. We note that we only know that α_0 exists as long as the agents are MEIP. Computing an estimate on α_0, however, requires one to conduct a few experiments.

There are a few possible approaches to overcome this requirement. One can try an iterative scheme, in which the edge gains are updated between iterations. Gradient-descent and extremum-seeking approaches are discussed in the next section (see Algorithm 3), but both require to measure the system between iterations. Another approach is to update the edge gains on a much slower time-scale than the dynamics of the system. This results in a two time-scale dynamical system, where the gains a_e of the system $(\mathcal{G}, \Sigma, \Pi, a)$ are updated slowly enough to allow the system to converge. Taking a_e as uniform gains of size α, and slowly increasing α, assures that eventually, $\alpha > \alpha_0$, so the system will converge ε-close to $\zeta^\star$. The only data we need is whether or not the system has already converged to an ε-neighborhood of $\zeta^\star$[1], to know whether α should be updated or not. This requires no data on the trajectories themselves, nor information on the specific steady-state limit, but only knowing whether the control goal has been achieved, which is the coarsest form of data. This results in an *almost data-free* solution of the practical formation control problem, which is valid as long as the agents are MEIP.

B. Data-Driven Determination of Gains

In the previous subsection, we introduced a formula for a uniform gain α described by the ratio of m and M, that solves the practical formation problem, where m and M are as defined in Remark 2. The parameter M depends on the integral function Γ of the controllers, evaluated on well-defined points, namely $\{\zeta \in \mathrm{Im}(\mathcal{E}^T) : \|\zeta - \zeta^\star\| = \varepsilon\}$. Thus, we can compute M exactly with no prior knowledge on the agents. This is not the case for the parameter m, which depends on the integral function of the agents. Without knowledge of any model of the agents, we need to obtain an estimate of m solely on the basis of input–output data from the agents.

From Remark 2, we know that $m = \sum_{i=1}^n (K_i^\star(\mathrm{y}_i^\star) + K_i(0)) = \sum_{i=1}^n m_i$ for some $\mathrm{y}^\star \in \mathbb{R}^n$ such that $\mathcal{E}^T \mathrm{y}^\star = \zeta^\star$. Without a model of the agents, m cannot be computed directly,

[1]Such data can be obtained by different methods, e.g., checking the size of the derivatives, using physical intuition in some cases, or using passivity to determine convergence rates as in [27].

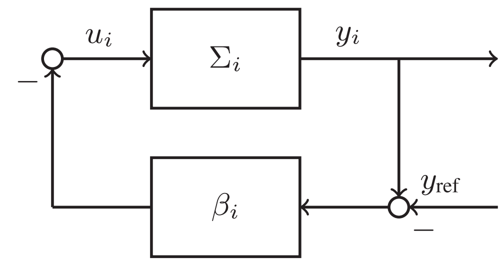

Fig. 4. Experimental setup of the closed-loop experiment for estimating m_i as used in Algorithm 1.

but we can find an upper bound on m from measured input–output trajectories via the inverse relations k_i^{-1}, $i = 1, \ldots, n$.

Proposition 2: Let $(\mathrm{u}_i^\star, \mathrm{y}_i^\star)$, $(\mathrm{u}_{i,1}, \mathrm{y}_{i,1})$, $(\mathrm{u}_{i,2}, \mathrm{y}_{i,2})$, …, $(\mathrm{u}_{i,r}, \mathrm{y}_{i,r})$ and $(0, \mathrm{y}_{i,0})$ be steady-state input–output pairs for agent i, for some $r \geq 0$. Then

$$m_i \leq \mathrm{u}_{i,1}(\mathrm{y}_{i,1} - \mathrm{y}_{i,0}) + \cdots + \mathrm{u}_{i,r}(\mathrm{y}_{i,r} - \mathrm{y}_{i,r-1}) + \mathrm{u}_i^\star(\mathrm{y}_i^\star - \mathrm{y}_{i,r}).$$

Proof: We prove the claim by induction on the number of steady-state pairs, $r + 2$. First, consider the case $r = 0$ of two steady-state pairs. Because $(0, \mathrm{y}_{i,0})$ is a steady-state pair, we know that $K_i(0) = -K_i^\star(\mathrm{y}_{i,0})$ by Fenchel duality. Similarly, $K_i(\mathrm{u}_i^\star) = \mathrm{u}_i^\star \mathrm{y}_i^\star - K_i^\star(\mathrm{y}_i^\star)$. Thus

$$m_i = K_i^\star(\mathrm{y}_i^\star) + K_i(0) = K_i^\star(\mathrm{y}_i^\star) - K_i^\star(\mathrm{y}_{i,0}) \leq \mathrm{u}_i^\star(\mathrm{y}_i^\star - \mathrm{y}_{i,0})$$

where we use the inequality $K_i^\star(b) - K_i^\star(c) \geq k_i^{-1}(c)(b - c)$ for $b = \mathrm{y}_{i,0}$ and $c = \mathrm{y}_i^\star$. Now, we move to the case $r \geq 1$. We write m_i as $(K_i^\star(\mathrm{y}_i^\star) - K_i^\star(\mathrm{y}_{i,r})) + (K_i^\star(\mathrm{y}_{i,r}) - K_i(0))$. The first element can be shown to be bounded by $\mathrm{u}_i^\star(\mathrm{y}_i^\star - \mathrm{y}_{i,r})$ by the case $r = 0$. The second element is bounded by $\mathrm{u}_{i,1}(\mathrm{y}_{i,1} - \mathrm{y}_{i,0}) + \cdots + \mathrm{u}_{i,r}(\mathrm{y}_{i,r} - \mathrm{y}_{i,r-1})$ by induction hypothesis, as we use a total of $r + 1$ steady-state pairs. Thus, m_i is no greater than the sum of the two bounds, $\mathrm{u}_{i,1}(\mathrm{y}_{i,1} - \mathrm{y}_{i,0}) + \cdots + \mathrm{u}_{i,r}(\mathrm{y}_{i,r} - \mathrm{y}_{i,r-1}) + \mathrm{u}_i^\star(\mathrm{y}_i^\star - \mathrm{y}_{i,r})$. ■

Remark 4: If we only have two steady-state pairs, $(\mathrm{u}_i^\star, \mathrm{y}_i^\star)$ and $(0, \mathrm{y}_{i,0})$, the estimate on m_i becomes $m_i \leq \mathrm{u}_i^\star(\mathrm{y}_i^\star - \mathrm{y}_{i,0})$. Thus, two steady-state pairs, corresponding to two measurements/experiments, are enough to yield a meaningful bound on m_i. We do note that more experiments yield better estimates of m_i, i.e., if $r \geq 1$, the estimate in Proposition 2 is better as long as $(\mathrm{y}_{i,0}, \mathrm{y}_{i,1}, \ldots, \mathrm{y}_{i,r}, \mathrm{y}_i^\star)$ is a monotone series.

We can use Remark 4 to compute an upper bound on m from the two steady-state pairs $(\mathrm{u}_i^\star, \mathrm{y}_i^\star)$ and $(0, \mathrm{y}_{i,0})$ per agent. Designing experiments to measure these quantities is possible, but can require additional information on the plant, e.g., output-strict passivity. Instead, we take another path and estimate $\mathrm{y}_{i,0}$ and $\mathrm{u}_i^\star$ instead of computing them directly. Indeed, we use the monotonicity of the steady-state input–output relation to bound $\mathrm{u}_i^\star$ and $\mathrm{y}_{i,0}$ from above and below. The approach is described in Algorithm 1. It is important to note that the closed-loop experiments are done with a output-strictly MEIP controller, which assures that the closed-loop system indeed converges.

We prove the following.

Proposition 3: The output $\mathbb{m}_i$ of Algorithm 1 is an upper bound on m_i.

Algorithm 1: Estimating m_i for an MEIP Agent.

```
Run the system in Fig. 4 with β_i small and y_ref = 1/β_i;
Wait for convergence, and measure the steady-state output
   y_{i,+} and the steady-state input u_{i,+};
Run the system in Fig. 4 with β_i small and y_ref = −1/β_i;
Wait for convergence, and measure the steady-state output
   y_{i,−} and the steady-state input u_{i,−};
if y_{i,+} < y_i^⋆ then
 |  Run the system in Fig. 4 with β_i = 1 and y_ref ≫ y_i^⋆;
 |  Wait for convergence, and measure the steady-state input
 |    u_{i,2} and output y_{i,2};
else
 |  Run the system in Fig. 4 with β_i = 1, y_ref ≪ y_i^⋆;
 |  Wait for convergence, and measure the steady-state input
 |    u_{i,2} and output y_{i,2};
end
Sort {u_{i,−}, u_{i,+}, u_{i,2}}. Denote the result by {U_1, U_2, U_3};
Sort {y_{i,−}, y_{i,+}, y_{i,2}}. Denote the result by {Y_1, Y_2, Y_3};
if U_2 > 0 then
 |  Define underline(y_{i,0}) = Y_1 and overline(y_{i,0}) = Y_2;
else
 |  Define underline(y_{i,0}) = Y_2 and overline(y_{i,0}) = Y_3;
end
if Y_2 > y_i^⋆ then
 |  Define underline(u_i^⋆) = U_1 and overline(u_i^⋆) = U_2;
else
 |  Define underline(u_i^⋆) = U_2 and overline(u_i^⋆) = U_3;
end
return m_i as the maximum over ω(y_i^⋆ − υ), where
   ω ∈ {underline(u_i^⋆), overline(u_i^⋆)} and υ ∈ {underline(y_{i,0}), overline(y_{i,0})};
```

Proof: First, we show that the closed-loop experiments conducted by the algorithm indeed converge. The plant Σ_i is assumed to be passive with respect to any steady-state input–output pair it possesses. Moreover, the static controller $u = \beta_i(y - y_{\mathrm{ref}})$ is output-strictly passive with respect to any steady-state input–output pair it possesses. Thus, it is enough to show that the closed-loop system has a steady state, which will prove convergence as this is a feedback connection of a passive system with an output-strictly passive system. Indeed, a steady-state input–output pair $(\mathrm{u_i}, \mathrm{y_i})$ of the system must satisfy $\mathrm{u_i} \in k_i^{-1}(\mathrm{y_i})$ and $\mathrm{u}_i = -\beta_i(\mathrm{y}_i - \mathrm{y_{ref}})$, or $-\beta_i(\mathrm{y_i} - \mathrm{y_{ref}}) \in k^{-1}(\mathrm{y}_i)$. This is equivalent to

$$0 \in k^{-1}(\mathrm{y}_i) + \beta_i(\mathrm{y}_i - \mathrm{y_{ref}}) = \nabla\left(K_i^\star(\mathrm{y}_i) + \frac{\beta_i}{2}(\mathrm{y}_i - \mathrm{y_{ref}})^2\right)$$

so y_i exists and is equal to the minimizer of $K_i^\star(\mathrm{y}_i) + \frac{\beta_i}{2}(\mathrm{y}_i - \mathrm{y_{ref}})^2$. This shows that the closed-loop experiments converge. It remains to show that it outputs an upper bound on m_i.

Using Remark 4, it is enough to show that $\mathrm{y}_{i,0} \in [\underline{\mathrm{y}_{i,0}}, \overline{\mathrm{y}_{i,0}}]$ and $\mathrm{u}_i^\star \in [\underline{\mathrm{u}_i^\star}, \overline{\mathrm{u}_i^\star}]$. To do so, we first claim that $U_1 \le \mathrm{u}_i^\star \le U_3$ and $Y_1 \le \mathrm{y}_{i,0} \le Y_3$. We first show that $Y_1 \le \mathrm{y}_{i,0}$, by showing that $\mathrm{y}_{i,-} \le \mathrm{y}_{i,0}$. Indeed, because k_i is a monotone map, this is equivalent to saying that $\mathrm{u}_{i,-} \le 0$. By the structure of the second experiment, the steady-state input is close to -1, and in particular smaller than 0. The inequality $\mathrm{y}_{i,0} \le \mathrm{y}_{i,+}$ is proved similarly. We note that because $\mathrm{u}_{i,-} \approx -1$ and $\mathrm{u}_{i,+} \approx 1$, we have $\mathrm{u}_{i,-} \le \mathrm{u}_{i,+}$ and thus $\mathrm{y}_{i,-} \le \mathrm{y}_{i,+}$. as k_i is monotone.

Next, we prove that $U_1 \le \mathrm{u}_i^\star$. By monotonicity of k_i, this is equivalent to $Y_1 \le \mathrm{y}_i^\star$. Because $\mathrm{y}_{i,-} \le \mathrm{y}_{i,+}$, it is enough to show that either $\mathrm{y}_{i,-} \le \mathrm{y}_i^\star$ or $\mathrm{y}_{i,2} \le \mathrm{y}_i^\star$. If the first case is true, then the proof is complete. Otherwise, $\mathrm{y}_{i,-} > \mathrm{y}_i^\star$, so the algorithm finds $\mathrm{y}_{i,2}$ by running the closed-loop system in Fig. 4 with $\beta_i = 1$ and $\mathrm{y_{ref}} \ll y_i^\star$. The increased coupling strength implies that the steady-state output $\mathrm{y}_{i,2}$ should be close to $\mathrm{y_{ref}}$, which is much smaller than $\mathrm{y}_i^\star$. Thus, $\mathrm{y}_{i,2} < \mathrm{y}_i^\star$, which shows that $Y_1 \le \mathrm{y}_1^\star$, or equivalently $U_1 \le \mathrm{u}_1^\star$. The proof that $\mathrm{u}_1^\star \le U_3$ is similar. This completes the proof. ■

Remark 5: Algorithm 1 demands us to run a certain system with parameter β_i small and wait until convergence. In practice, determining when the system has converged can be done by measuring the output or its derivative. Alternatively, one can use a Lypaunov-based approach [27]. One could also use engineering intuition and simulations to conclude an estimate on the termination time of the experiments. The parameter β_i can be chosen in a similar manner.

Remark 6: One can run more experiments to give tighter estimates of m_i. Indeed, by construction, take a collection $\{(U_{i,k}, Y_{i,k})\}_{k=0}^{r+1}$ of steady-state input–output pairs, and use the monotonicity of the steady-state relation to sort them in such a way that $Y_{i,0} \le \cdots \le Y_{i,r} \le y_i^\star \le Y_{i,r+1}$, $U_{i,0} \le 0 \le U_{i,1} \le \cdots \le U_{i,r+1}$. We would like to use Proposition 2, but we do not know the exact values of $y_{i,r}, u_{i,0}$. Instead, we again use the monotonicity of the steady-state relation and bound $U_{i,r} \le u_i^\star \le U_{i,r+1}$ and $Y_{i,0} \le y_{i,0} \le Y_{i,1}$. Thus,

$$m_i \le M_i \triangleq \sum_{k=1}^{r} U_{i,k}(Y_{i,k} - Y_{i,k-1}) + U_{i,r+1}(\mathrm{y}_i^\star - Y_{i,r}). \tag{4}$$

We claim that M_i is a relatively tight estimate of m_i.

Proposition 4: Let $\Delta_k = Y_{i,k} - Y_{i,k-1} \ge 0$, and assume that k_i^{-1} is an L-Lipschitz function. Then, $|M_i - m_i| \le C\max\{L, 1\}\max_k \Delta_k$, for a constant $C = C(\mathrm{y}_{i,0}, \mathrm{u}_i^\star, \mathrm{y}_i^\star)$.

Proof: First, $m_i = K_i^\star(\mathrm{y}_i^\star) - K_i^\star(\mathrm{y}_{i,0}) = \int_{\mathrm{y}_{i,0}}^{\mathrm{y}_i^\star} k_i^{-1}(s)ds$. Thus, it is enough to bound each of the following terms:

i) $|\int_{Y_{i,k-1}}^{Y_{i,k}} k_i^{-1}(s)ds - U_{i,k}(Y_{i,k} - Y_{i,k-1})|, k = 2, \ldots, r$;
ii) $|\int_{Y_{i,r}}^{\mathrm{y}_i^\star} k_i^{-1}(s)ds - u_i^\star(\mathrm{y}^\star - Y_{i,r})|$;
iii) $|\int_{\mathrm{y}_{i,0}}^{Y_{i,1}} k_i^{-1}(s)ds - U_{i,1}(Y_{i,1} - \mathrm{y}_{i,0})|$;
iv) $|(U_{i,r+1} - u_i^\star)(\mathrm{y}_i^\star - Y_{i,r})|$;
v) $|U_{i,r+1}(Y_{i,r+1} - \mathrm{y}_i^\star)|$;
vi) $|U_{i,1}(\mathrm{y}_{i,0} - Y_{i,0})|$.

The first term can be bounded by

$$\int_{Y_{i,k-1}}^{Y_{i,k}} |k^{-1}(s) - k^{-1}(Y_{i,k})|ds \le L\int_{Y_{i,k-1}}^{Y_{i,k}} |s - Y_{i,k}|ds = L\Delta_k^2.$$

Similarly, the second term is bounded by $L(y_i^\star - Y_{i,r})^2$ and the third term is bounded by $L(Y_{i,1} - \mathrm{y}_{i,0})^2$. The sum of the three terms is thus bounded by

$$L\left[\sum_{k=2}^{r}(Y_{i,k} - Y_{i,k-1})^2 + (y_i^\star - Y_{i,r})^2 + (Y_{i,1} - \mathrm{y}_{i,0})^2\right]$$
$$\le L\max(\Delta_k)\left[\sum_{k=2}^{r}(Y_{i,k} - Y_{i,k-1}) + (y_i^\star - Y_{i,r}) + (Y_{i,1} - \mathrm{y}_{i,0})\right]$$
$$\le L\max(\Delta_k)(\mathrm{y}_i^\star - \mathrm{y}_{i,0})$$

Algorithm 2: Synthesis Procedure for Practical Formation Control.

1 Choose some output-strictly MEIP controllers Π_e such that the integral function Γ has a single minimizer $\zeta^\star$ when restricted to the set $\mathrm{Im}(\mathcal{E}^T)$;
2 Choose some $\mathrm{y}^\star \in \mathbb{R}^n$ such that $\mathcal{E}^T \mathrm{y}^\star = \zeta^\star$;
3 **for** $i = 1, ..., n$ **do**
4 | Run Algorithm 1. Let $\mathbb{m}_i$ be the output;
5 **end**
6 Let $\mathbb{m} = \sum_{i=1}^n \mathbb{m}_i$;
7 Compute $M = \min\{\zeta \in \mathrm{Im}(\mathcal{E}^T) : \ \|\zeta - \zeta^\star\| = \varepsilon\}$;
8 Compute $\alpha = \mathbb{m}/M$;
9 **return** the controllers $\{\alpha \Pi_e\}_{e \in \mathbb{E}}$;

where we use $y_i^\star - Y_{i,r} \leq \Delta_{r+1}$ and $Y_{i,1} - \mathrm{y}_{i,0} \leq \Delta_1$. Similarly, the fourth term is bounded by $L\Delta_{r+1}^2$, the fifth term is bounded by $U_{i,r+1}\Delta_{r+1}$, and the last term is bounded by $U_{i,1}\Delta_0$. This completes the proof, as $0 \leq U_{i,1} \leq u_i^\star = k_i^{-1}(\mathrm{y}_i^\star)$, and $U_{i,r+1} = k^{-1}(Y_{i,r+1}) \leq k^{-1}(\mathrm{y}_i^\star) + L\Delta_{r+1}$. ■

Remark 7: A natural question that arises is how to conduct experiments assuring that $\max_k \Delta_k$ is small. If we run the system in Fig. 4 with $\beta_i \gg 1$, then the steady-state output would be very close to y_{ref}. Thus, if we run r additional experiments with $\beta_i \gg 1$ and references $y_{\mathrm{ref,1}} \leq y_{\mathrm{ref,2}} \leq \cdots \leq y_{\mathrm{ref,r}}$ (i.e., a total of $r+3$ experiments), then $\max \Delta_k = O(\max_k |y_{\mathrm{ref,k+1}} - y_{\mathrm{ref,k}}|)$. Thus, choosing $y_{\mathrm{ref,k}}$ as r equally spaced points in an appropriate interval would give $\max_k \Delta_k = O(1/r)$, and a uniformly random choice gives $\max_k \Delta_k = O(\log r/r)$ with high probability [34].

We saw that m_i can be bounded using three experiments for general MEIP agents, and that additional measurements can be used to provide more accurate estimates of m_i. Other recent works about data-driven control focus on the case of LTI systems, using them as a base to build toward a solution for nonlinear systems [4], [7], [35]. If we restrict ourselves to this case, we can exactly compute m_i from a single experiment.

Proposition 5: Suppose that the agent Σ_i is known to be both MEIP and LTI. Let $(\tilde{\mathrm{u}}, \tilde{\mathrm{y}})$ be any steady-state input–output pair for which either $u_1 \neq 0$ or $y_1 \neq 0$.[2] Then, $m_i = \frac{(y_i^\star)^2 \tilde{\mathrm{u}}}{2\tilde{\mathrm{y}}}$. Thus, m_i can be exactly calculated using a single experiment.

Proof: We know that k is a linear function, and the system state matrix is Hurwitz [21], [24]. Moreover, unless the transfer function of the agent is 0, k^{-1} is a linear function $k^{-1}(\mathrm{y}) = s\mathrm{y}$ for some $s > 0$ [36]. Thus, $K^\star(\mathrm{y}) = \frac{s}{2}\mathrm{y}^2$. Now, $k^{-1}(0) = s \cdot 0 = 0$, so (0,0) is a steady-state input–output pair, meaning that $\mathrm{y}_{i,0} = 0$. Moreover, we know that $\tilde{\mathrm{u}} = s\tilde{\mathrm{y}}$, and not both are zero, so we conclude that $\tilde{\mathrm{y}} \neq 0$, and that $s = \frac{\tilde{\mathrm{u}}}{\tilde{\mathrm{y}}}$. Thus, $K_i^\star(\mathrm{y}_{i,0}) = K_i^\star(0) = 0$ and $K_i^\star(\mathrm{y}_i^\star) = \frac{s}{2}(\mathrm{y}_i^\star)^2 = \frac{s}{2}\mathrm{y}^2$. This completes the proof, as $m_i = K_i^\star(\mathrm{y}_i^\star) - K_i^\star(\mathrm{y}_{i,0})$. ■

The chapter concludes with Algorithm 2 for solving the practical formation control problem using the single-gain amplification scheme, which is applied in Section VI.

Remark 8: Step 1 of the algorithm allows almost complete freedom of choice for the controllers. One possible choice is the static controllers $\mu_e = \zeta_e - \zeta_e^\star$. Moreover, if Π_e is any MEIP controller for each $e \in \mathbb{E}$, and $\gamma_e(\zeta_e) = 0$ has a unique solution for each $e \in \mathbb{E}$, then the "formation reconfiguration" scheme from [23] suggests a way to find the required controllers using mild augmentation.

Remark 9: The algorithm allows one to choose any vector $\mathrm{y}^\star$ such that $\mathcal{E}^T\mathrm{y}^\star = \zeta^\star$. All possible choices lead to some gain α which assures a solution of the practical formation control problem, but some choices yield better results (i.e., smaller gains) than others. The optimal $\mathrm{y}^\star$, minimizing the estimate α, can be found as the minimizer of the problem $\min\{K^\star(\mathrm{y}) : \mathcal{E}^T\mathrm{y} = \zeta^\star\}$, which we cannot compute using data alone. One can use physical intuition to choose a vector $\mathrm{y}^\star$ which is relatively close to the actual minimizer, but the algorithm is still valid, no matter which $\mathrm{y}^\star$ is chosen.

V. Iterative Practical Formation Control: Applying Different Gains on Different Edges

Let us revisit Fig. 2 and let $A = \mathrm{diag}(\{a_e\}_{e\in\mathbb{E}})$ with positive, but distinct entries a_e. These additional degrees of freedom can be used, for example, to reduce the conservatism and retrieve a smaller norm of the adjustable gain vector a while still solving the practical formation control problem. It follows directly from Theorem 2 that there always exists a bounded vector a solving the practical formation control problem. However, the question remains how a can be chosen based on sampled input–output data and passivity properties.

Our idea here is to probe our diffusively coupled system for given gains a_e and adjust the gains according to the resulting steady-state output. By iteratively performing experiments in this way, we strive to find controller gains that solve the practical formation control problem. This approach is tightly connected to iterative learning control, where one iteratively applies and adjusts a controller to improve the performance of the closed-loop for a repetitive task [37]. Our approach here is based on passivity and network optimization with only requiring the possibility to perform iterative experiments.

One natural idea in this direction is to define a cost function that penalizes the distance of the resulting steady state to the desired formation control goal and then apply a gradient descent approach, adjusting the gain a for each experiment. However, to obtain the gradient of $\|\mathcal{E}^T\mathrm{y}(a) - \zeta^\star\|^2$ with respect to the vector a, where $\mathrm{y}(a)$ is the steady-state output of $(\mathcal{G}, \Sigma, \Pi, a)$, one requires knowledge of the inverse relations k_i^{-1} for all $i = 1, \ldots, n$. With no model of the agents available, a direct gradient descent approach is hence infeasible. We thus look for a simple iterative multigain control scheme without knowledge on the exact steepest descent direction.

We start off with an arbitrarily chosen gain vector a_0 with only positive entries. Due to Assumption 1, the closed-loop converges to a steady state. According to the measured state, the idea is then to iteratively perform experiments and update the gain vector until we reach our control goal, i.e., practical formation control. The update formula can be summarized by

$$a_e^{(j+1)} = a_e^{(j)} + hv_e, \quad e \in \mathbb{E} \tag{5}$$

[2] For example, by running the system in Fig. 4 with some $\beta > 0$ and $\mathrm{y_{ref}} \neq 0$.

Algorithm 3: Practical Formation Control with Derivative-free Optimization.

```
Initialize a^(0), e.g. with 𝟙_|𝔼| ;
Choose step size h and set j = 0;
while F(a) = ||𝓔^T y(a) − ζ*||^2 > ε do
    Apply a^(j) to the closed loop ;
    Compute v_e = { (f_e − ζ*_e)/γ_e(f_e)   γ(f_e) ≠ 0
                  { 0                        γ(f_e) = 0     ∀e ;
    Update a_e^(j+1) = a_e^(j) + h v_e, j = j + 1 ;
end
return a
```

where $h > 0$ is the step size, and v, with entries v_e, $e = 1, \dots, |\mathbb{E}|$, is the update direction. In practice, the update can either be instantaneous or gradual, e.g., using linear interpolation or higher order splines. We denote the eth entry of $\mathcal{E}^T\mathrm{y}$ as f_e and choose v in each iteration such that

$$v_e = \begin{cases} \frac{\mathrm{f}_e - \zeta^\star_e}{\gamma_e(\mathrm{f}_e)} & \gamma_e(\mathrm{f}_e) \neq 0 \\ 0 & \text{otherwise.} \end{cases}, \quad e \in \mathbb{E} \tag{6}$$

If k^{-1} and γ are differentiable functions, then we claim that $F(a) = ||\mathcal{E}^T\mathrm{y}(a) - \zeta^\star||^2$ decreases in the direction of v, i.e., $v^T\nabla F(a) < 0$. This leads to a multigain distributed control scheme, using (5) with (6), summarized in Algorithm 3. This multigain distributed control scheme is guaranteed to solve the practical formation problem after a finite number of iterations, and is summarized in the following theorem.

Theorem 3: Suppose that the functions k^{-1}, γ are differentiable, and that there exists an agent $i_0 \in \mathbb{V}$ such that $\frac{dk^{-1}_{i_0}}{d\mathrm{y}_{i_0}} > 0$ for any point $\mathrm{y}_{i_0} \in \mathbb{R}$. Moreover, assume that $\frac{d\gamma_e}{d\zeta_e} > 0$ for any $e \in \mathbb{E}, \zeta_e \in \mathbb{R}$. Then, $v^T\nabla F(a) \leq 0$, with v, F as defined in Algorithm 3 (and equality if and only if $\mathcal{E}^T\mathrm{y}(a) = \zeta^\star$). Furthermore, if the step size $h > 0$ is small enough, then Algorithm 3 halts after finite time, providing a gain vector that solves the practical formation control problem.

Sketch of Proof: The proof is based on showing that ∇F can be written as $-\mathrm{diag}(\gamma(\mathrm{f}))X(\mathrm{y}(a))(\mathrm{f} - \zeta^\star)$, where $X(\mathrm{y}(a))$ is a positive-definite matrix depending on $\mathrm{y}(a)$. We can now show that $v^T\nabla F = -(\mathrm{f} - \zeta^\star)^T X(\mathrm{y}(a))(\mathrm{f} - \zeta^\star) \leq 0$. The full proof of Theorem 3 is available in the Appendix. ∎

Algorithm 3 together with the theoretical results from Theorem 3 provides us with a very simple and distributed, iterative control scheme with theoretical guarantees. Note also that the steady states of the agents are independent of their initial condition. For each iteration, the agents can hence also start from the position they converged to at the last iteration. This can be interpreted similarly to Remark 3, where gains are updated on a slower time scale than convergence of the agents. However, instead of only the information whether practical formation control is achieved, we generally need the actual difference $\mathcal{E}^T\mathrm{y} - \zeta^\star$ that is achieved with the current controller in each iteration. In the special case of proportional controllers $\mu_e = \zeta_e - (\zeta^\star)_e$, yielding $v_e = 1$, we retrieve the exact controller scheme proposed in Remark 3.

An alternative gradient-free control scheme is the extremum seeking framework presented in [38]. Assuming that k^{-1} and γ are twice continuously differentiable, a step in the direction of steepest descent is approximated every $4|\mathbb{E}|$ steps (cf. [38, Theorem 1]). While the extremum seeking framework approximates the steepest descent (and the simple multigain approach only guarantees a descending direction), it also requires large amounts of experiments per approximated gradient step. Furthermore, the algorithm as presented in [38] cannot be computed in a purely distributed fashion. Therefore, the simple distributed control scheme in Algorithm 3 displays significant advantages in the present problem setup.

VI. Case Study: Velocity Coordination in Vehicles With Drag and Exogenous Forces

Consider a collection of 30 one-dimensional robot vehicles, each modeled by a double integrator $G(s) = \frac{1}{s^2}$. The robots try to coordinate their velocity. Each of them has its own drag profile $f(\dot{p})$, which is unknown to the algorithm, but it is known that f is increasing and $f(0) = 0$. Moreover, each vehicle experiences external forces (e.g., wind, and being placed on a slope). The velocity of the vehicles is governed by the equation

$$\Sigma_i : \begin{cases} \dot{x}_i &= -f_i(x_i) + u_i + w_i \\ y_i &= x_i \end{cases} \tag{7}$$

where x_i is the velocity of the ith vehicle, f_i is its drag model, w_i are the exogenous forces acting on it, u_i is the control input, and y_i is the measurement. In the simulation, the drag models f_i are given by $c_d|x|x$, where the drag coefficient c_d is chosen as a log uniformly distributed random variable. We assume that the vehicles are light, so the wind accelerates the vehicles by a non-negligible amount. Thus, w_i is randomly chosen between -2 and 2. We wish to achieve velocity consensus, with error no greater than $\epsilon = 0.2$. We consider a diffusive coupling of the agents with the cycle graph $\mathcal{G} = \mathcal{C}_{30}$, and take proportional controllers $\zeta_e = \mu_e$.

We apply the amplification scheme presented in Algorithm 2 and choose the consensus value $y_i^\star = 1.5_{m/sec}$ to use in the estimation algorithm. Note that the plants are MEIP, but not output-strictly MEIP, and use Algorithm 1 to estimate the required uniform gain α. The first two experiments are conducted with $\beta_i = 0.01$, and $y_{\mathrm{ref}} = \pm 100$. Based on their results, we run a third experiment on each of the agents for which this is required, this time with $\beta_i = 1$ and $y_{\mathrm{ref}} = \pm 10$, where the sign is chosen according to Algorithm 1. The experimental results are available in Fig. 5(a).

We estimate each m_i using Remark 4. For example, for agent 1, we get the three steady-state input–output pairs (0.9947,0.5203), $(-0.9687, -3.1294)$, and (3.4268,3.5732). Monotonicity implies that it has steady states $(u_1^\star, y_1^\star) = (u_1^\star, 1.5)$ and $(0, y_{1,0})$ with $0.9947 \leq u_1^\star \leq 3.4268$ and $-3.1294 \leq y_{1,0} \leq 0.5203$. We can thus estimate $m_2 \leq 3.4268 \cdot (1.5 - (-3.1294)) = 15.864$. Repeating this calculation for each of the agents and summing gives $m = 256.3658$.

As for estimating M, we have $\Gamma_e(\zeta_e) = \zeta_e^2$, so $\Gamma(\zeta) = \|\zeta\|^2$. The minimum is at $\zeta^\star = 0$, and by definition, we have $M =$

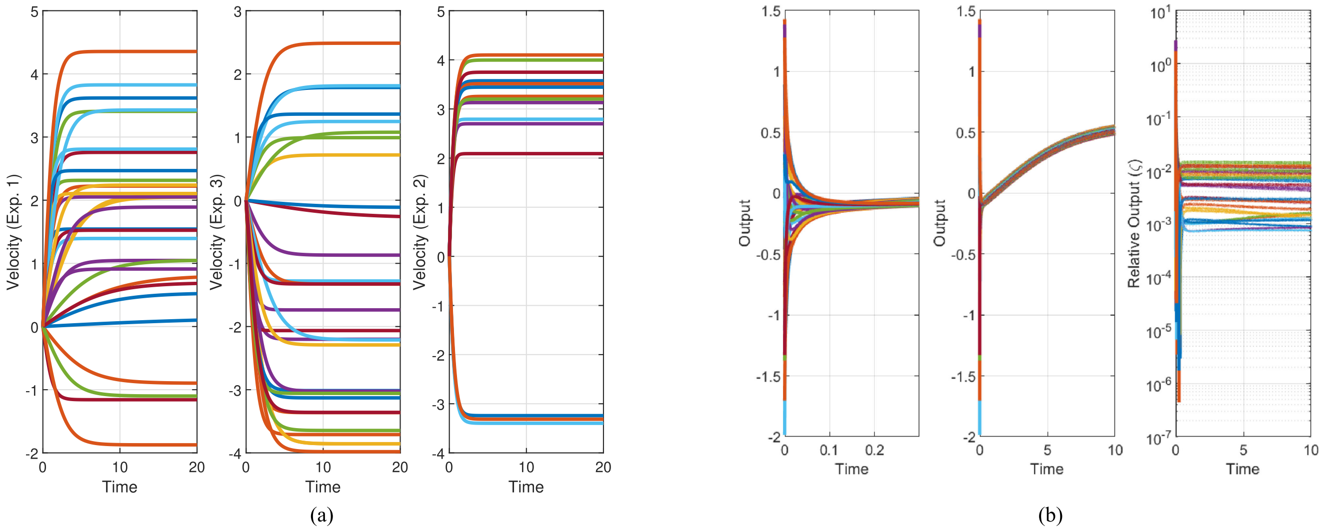

Fig. 5. Experiment results for vehicle case study. (a) Results of first set of experiments for the vehicles. Each plot corresponds to a different agent. (b) The closed loop with uniform gain $\alpha = 213.638$. The two leftmost graphs plot the agents' trajectories over 0.3 and over 10 s. The rightmost graph plots the relative outputs.

$\min_{\zeta\in \mathrm{Im}(\mathcal{E}^T):\ \|\zeta-\zeta^\star\|=\varepsilon} \Gamma(\zeta) = |\mathbb{E}|\varepsilon^2$. Thus, we get $\alpha = \frac{m}{M} = \frac{m}{30\varepsilon^2} = 213.64$. To verify the algorithm, we run the closed-loop system $(\mathcal{G}, \Sigma, \Pi, \alpha\mathbb{1})$ with the gain α we found. The results are available in Fig. 5(b). One can see that the overall control goal is achieved—the agents converge to a steady state which is ε-close to consensus. However, the agents actually converge to a much closer steady state than predicted by the algorithm. Namely, the distance of the steady-state output from consensus is roughly 0.04, much smaller than 0.2. Uncoincidentally, the true value of $m = K(0) + K^\star(y^\star)$ is 50.15, meaning we overestimate it (and hence m/M) by about 411%. One can mitigate this by using more experiments to improve the estimate m_i, as in Proposition 2 or in Remark 6. We follow this approach and conduct further experiments on each of the agents using $\beta_i = 10$ and choosing y_{ref} randomly. The resulting values of α, as well as the error from the true value of m/M, can be seen in the below table. It can be seen that even a single additional measurement per agent can significantly reduce the estimation error of m/M.

Measurements Per Agent	Value of α	Overestimation of m/M
3	213.638	411%
4	104.308	150%
10	67.316	61%
20	55.161	32%

Altogether we showed that Algorithm 2 manages to solve the practical consensus problem for vehicles, affected by drag and exogenous inputs, without using any model for the agents, while conducting very few experiments for each agent. However, it overestimates the required coupling, and thus has unnecessarily large energy consumption.

Let us now apply the iterative multigain control strategy. We start with $a^{(0)} = 0.1 \otimes 1_{|\mathbb{E}|}$, we choose the step size $h = 2$ and apply Algorithm 3. In fact, since $\zeta^\star = 0$ and $\zeta_e = \mu_e$, we receive $v = 1_{|\mathbb{E}|}$, which constitutes the special case of Remark 3. The corresponding norm of the gain vector and

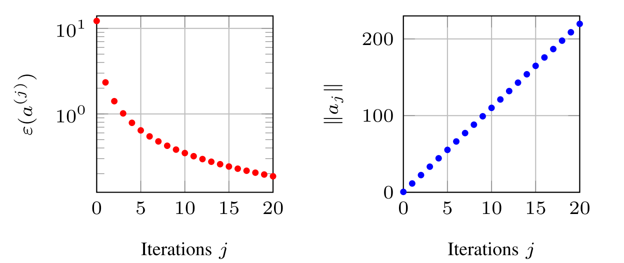

Fig. 6. Resulting ε and the norm of the gain vector $\|a^{(j)}\|$ over iterations j when applying the iterative multigain control strategy to the case study of velocity coordination in vehicles.

the resulting ε in each iteration are illustrated in Fig. 6. After 20 iterations, we already arrive at a vector, which solves the practical formation problem with $\|a^{(19)}\| = 208.68$, while $\varepsilon = 0.195 < 0.2$. Note that the controller with the uniform gain had $\|a\| = \sqrt{|\mathbb{E}|} \cdot 213.638 = 1170.1$, so the iterative scheme beats it by a factor of 5 in terms of energy.

VII. Conclusion

We presented an approach for model-free practical cooperative control for diffusively coupled systems only on the premise of passivity of the agents. The presented approach led to two control schemes: with additional two or three experiments on the agents, we can upper bound the controller gain which solves the practical formation problem, or we can iteratively adapt the adjustable gain vector until practical formation is reached. Both approaches are especially simple in their application, while still being scalable and providing theoretical guarantees. Future research might try and improve the presented methods, either by reducing the number of experiments needed on each agent, or by achieving faster practical convergence using iterations. One might also try to use very limited knowledge on the agents to achieve the said improvement. Other possible directions include data-driven solutions to more intricate problems using the network optimization framework, e.g., fault detection and isolation.

Appendix

We now give full proofs of Proposition 1 and Theorem 3.

A. Proving Proposition 1

In order to prove the proposition, we use the notion of cursive relations established in [26].

Definition 4 (Cursive Relations, [26]): A set $A \subset \mathbb{R}^2$ is called *cursive* if there exists a curve $\alpha : \mathbb{R} \to \mathbb{R}^2$ such that the following conditions hold.

i) The set A is the image of α.
ii) The map α is continuous.
iii) The map α satisfies $\lim_{|t|\to\infty} \|\alpha(t)\| = \infty$.
iv) $\{t \in \mathbb{R} :\ \exists s \neq t,\ \alpha(s) = \alpha(t)\}$ has measure zero.

A relation Υ is called cursive if the set $\{(p,q) :\ q \in \Upsilon(p)\}$ is cursive.

The notion of cursive relations is useful as it can help prove that systems are MEIP.

Theorem 4 ([26]): A monotone cursive relation is maximally monotone.

We can now prove Proposition 1.

Proof: Consider an arbitrary steady state of the system. As h is continuous and strictly monotone ascending, hence invertible, we must have $\dot{x} = 0$ for any steady-state input–output pair. Thus, we conclude that any steady-state input–output pair can be written as $(f(\sigma)/g(\sigma), h(\sigma))$ for some $\sigma \in \mathbb{R}$. We first show passivity with respect to every steady state, and then show that the steady-state input–output relation is maximally monotone. Take a steady state $(f(x_0)/g(x_0), h(x_0))$ of the system, and define $S(x) = d\sigma \int_{x_0}^{x} \frac{h(\sigma)-h(x_0)}{g(\sigma)}$. We claim that S is a storage function for the steady-state input–output pair $(f(x_0)/g(x_0), h(x_0))$. Indeed, $S(x) \geq 0$, with equality only at x_0, immediately follows from strict monotonicity of h and $g > 0$. As for the inequality defining passivity, we have

$$\begin{aligned}\frac{d}{dt}S(x) &= \frac{h(x)-h(x_0)}{g(x)}(-f(x)+g(x)u)\\ &= (h(x)-h(x_0))u - \frac{f(x)}{g(x)}(h(x)-h(x_0))\\ &= (h(x)-h(x_0))\left(u - \frac{f(x_0)}{g(x_0)}\right)\\ &\quad - \left(\frac{f(x)}{g(x)} - \frac{f(x_0)}{g(x_0)}\right)(h(x)-h(x_0))\end{aligned}$$

where the second term is negative as $\frac{f}{g}, h$ are monotone ascending, and the first term is $(y-h(x_0))(u-\frac{f(x_0)}{g(x_0)})$. Hence, the system is indeed passive with respect to any steady-state input–output pair. As for maximal monotonicity of the steady-state relation, we recall that it can be parameterized as $(f(\sigma)/g(\sigma), h(\sigma))$ for $\sigma \in \mathbb{R}$. We claim that this relation is both monotone and cursive, which will prove maximal monotonicity. Monotonicity holds as for any x_0, x_1

$$\frac{f(x_0)}{g(x_0)} > \frac{f(x_1)}{g(x_1)} \iff x_0 > x_1 \iff h(x_0) > h(x_1) \quad (8)$$

due to strict monotonicity. As for cursiveness, the map $\sigma \mapsto (f(\sigma)/g(\sigma), h(\sigma))$ is a curve whose image is the relation. Moreover, it is clear that the map is continuous, and also injective due to (8). Finally, we have

$$\lim_{|t|\to\infty} \left\| \left(\frac{f(t)}{g(t)}, h(t)\right) \right\| \geq \lim_{|t|\to\infty} \max\left\{ \left|\frac{f(t)}{g(t)}\right|, |h(t)| \right\} = \infty \quad (9)$$

so the proof is complete by Theorem 4. ∎

B. Proving Theorem 3

We first state and prove the following lemma.

Lemma 4: Suppose that the assumptions of Theorem 3 hold, and let $C > 0$ be any constant. Define $A_1 = \{\mathrm{y} \in \mathbb{R}^n :\ \|\mathcal{E}^T\mathrm{y} - \zeta^\star\| \leq C\}$ and $A_2 = \{\mathrm{y} \in \mathbb{R}^n :\ \sum_i k_i^{-1}(\mathrm{y}_i) = 0\}$. Then, the set $A_1 \cap A_2$ is bounded.

Proof: First, we note that the inequality $\|\mathcal{E}^T\mathrm{y} - \zeta^\star\| \leq C$ implies that for any edge $\{i,j\} \in \mathbb{E}$, we have $|\mathrm{y}_i - \mathrm{y}_j| \leq C + ||\zeta^\star||$ by the triangle inequality. We let $\omega = (C + ||\zeta^\star||)\mathrm{diam}(\mathcal{G})$, where $\mathrm{diam}(\mathcal{G})$ is the diameter of the graph $\mathcal{G}$, so that if there exists some $i,j \in \mathbb{V}$ such that $|\mathrm{y}_i - \mathrm{y}_j| > \omega$, then $\mathrm{y} \notin A_1$. Moreover, let $\mathrm{z} = k(0)$, so $\sum_i k_i^{-1}(\mathrm{z}_i) = 0$, so that if $\mathrm{y} \in \mathbb{R}^n$ satisfies $\forall i :\ \mathrm{y}_i > \mathrm{z}_i$, then $\mathrm{y} \notin A_2$. Indeed, for each i, we have $k_i^{-1}(\mathrm{y}_i) \geq k_i^{-1}(\mathrm{z}_i)$, and $k_{i_0}^{-1}(\mathrm{z}_{i_0}) > k_{i_0}^{-1}(\mathrm{y}_{i_0})$, meaning that $\sum_i k_i^{-1}(\mathrm{y}_i) > \sum_i k_i^{-1}(\mathrm{z}_i) = 0$. Similarly, if $\forall i, \mathrm{z}_i > \mathrm{y}_i$ then $\mathrm{y} \notin A_2$. We claim that for any $\mathrm{y} \in A_1 \cap A_2$ and any $i \in \mathbb{V}$, we have $C_1 < \mathrm{y}_i < C_2$, where $C_1 = \min_j \mathrm{z}_j - \omega - 1$ and $C_2 = \max_j \mathrm{z}_j + \omega + 1$. Indeed, take any $\mathrm{y} \in \mathbb{R}^n$, and suppose that $\mathrm{y}_i \geq C_2$ for some $i \in \mathbb{V}$. There are two possibilities.

- There is some $k \in \mathbb{V}$ such that $\mathrm{y}_k < \max_j \mathrm{z}_j + 1$. Then, $|\mathrm{y}_i - \mathrm{y}_k| > \omega$, implying that $\mathrm{y} \notin A_1$.
- For any $k \in \mathbb{V}$, $\mathrm{y}_k \geq \max_j \mathrm{z}_j + 1$, implying that $\mathrm{y} \notin A_2$.

Similarly, one shows that if there is some i such that $\mathrm{y}_i \leq C_1$, then $\mathrm{y} \notin A_1 \cap A_2$. This completes the proof. ∎

Proof of Theorem 3: Consider the solution $\mathrm{y}(a)$ of $0 = k^{-1}(\mathrm{y}) + \mathcal{E}\mathrm{diag}(a)\gamma(\mathcal{E}^T\mathrm{y})$ as a function of a. Then $\mathrm{y}(a)$ is a differentiable function by the inverse function theorem, and its differential is given by $\frac{d\mathrm{y}}{da} = -X(\mathrm{y}(a))\mathcal{E}\mathrm{diag}(\gamma(\mathcal{E}^T\mathrm{y}(a)))$, where the matrix $X(\mathrm{y})$ is given by

$$X(\mathrm{y}) = [\mathrm{diag}(\nabla k^{-1}(\mathrm{y})) + \mathcal{E}\mathrm{diag}(\nabla\gamma(\mathcal{E}^T\mathrm{y}))\mathcal{E}^T]^{-1}. \quad (10)$$

We note that $X(\mathrm{y})$ is positive-definite for any $\mathrm{y} \in \mathbb{R}^n$, by Proposition 2 in [36]. Thus, the gradient of F is given by

$$\nabla F(a) = -\mathrm{diag}(\gamma(\mathcal{E}^T\mathrm{y}(a)))\mathcal{E}^T X(\mathrm{y}(a))\mathcal{E}(\mathcal{E}^T\mathrm{y}(a) - \zeta^\star). \quad (11)$$

We note that $v^T\mathrm{diag}(\gamma(\mathcal{E}^T\mathrm{y}(a)) = \mathcal{E}^T\mathrm{y}(a) - \zeta^\star$, as $\gamma_e(x_e) = 0$ if and only if $x_e = \zeta_e^\star$ by strict monotonicity. Thus,

$$v^T\nabla F(a) = -(\mathcal{E}(\mathcal{E}^T\mathrm{y}(a) - \zeta^\star))^T X(\mathrm{y}(a))\mathcal{E}(\mathcal{E}^T\mathrm{y}(a) - \zeta^\star) \quad (12)$$

which is nonpositive as $X(\mathrm{y}(a))$ is a positive-definite matrix.

Now, we claim that $v^T\nabla F(a) = 0$ if and only if $\mathcal{E}^T\mathrm{y}(a) = \zeta^\star$. Indeed, $\zeta^\star \in \mathrm{Im}(\mathcal{E}^T)$, so we denote $\zeta^\star = \mathcal{E}^T\mathrm{y}_0$ for some y_0. As $X(\mathrm{y}(a))$ is positive definite, (12) implies that $v^T\nabla F(a) = 0$ if and only if $\mathcal{E}(\mathcal{E}^T\mathrm{y}(a) - \zeta^\star) = \mathcal{E}\mathcal{E}^T(\mathrm{y}(a) - \mathrm{y}_0)$ is the zero vector. The kernel of the Laplacian $\mathcal{E}\mathcal{E}^T$ is the span of the all-one vector $\mathbb{1}_n$, so $\mathrm{y}(a) - \mathrm{y}_0 = \kappa \mathbb{1}_n$ for some κ, hence $\mathcal{E}^T \mathrm{y}(a) = \mathcal{E}^T \mathrm{y}_0 = \zeta^\star$. This concludes the first part of the proof.

As for convergence, we know that if h is small enough, then $F(a^{(j+1)}) < F(a^{(j)})$. However, the value of h so that $F(a^{(j+1)}) < F(a^{(j)})$ can depend on $a^{(j)}$ itself, but it is obvious that if h is small enough, then for any j, we have $F(a^{(j)}) \leq F(a^{(0)})$. We let $C = F(a^{(0)}) = \|\mathcal{E}^T \mathrm{y}(a^{(0)}) - \zeta^\star\|$, and consider the sets $A_1 = \{\mathrm{y} : \|\mathcal{E}^T \mathrm{y} - \zeta^\star\| \leq C\}$ and $A_2 = \{\mathrm{y} : \sum_i k_i^{-1}(\mathrm{y}_i) = 0\}$. For any j, we know that $\mathrm{y}(a^{(j)}) \in A_1$ by above, and that $\mathrm{y}(a^{(j)}) \in A_2$ by the steady-state equation $0 = k^{-1}(\mathrm{y}(a)) + \mathcal{E}\mathrm{diag}(a)\gamma(\mathcal{E}^T \mathrm{y}(a))$. Thus, all steady-state outputs achieved during the algorithm are in the set $\mathcal{D} = A_1 \cap A_2$, which is bounded by Lemma 4. The map sending a matrix to its minimal singular value is continuous, meaning that $\underline{\sigma}(X(\mathrm{y}))$ achieves a minimum on the set $\mathcal{D}$ at some point y_1, and the minimum is positive at $X(\mathrm{y}_1)$ is positive-definite. We denote the minimum value by $\underline{\sigma}(\mathcal{D})$.

Now, consider (12). We get that $v^T \nabla F(a)$ is bounded by above $-\underline{\sigma}(\mathcal{D})||\mathcal{E}(\mathcal{E}^T \mathrm{y}(a) - \zeta^\star)||^2$. In turn, we saw above that unless $\mathcal{E}^T \mathrm{y}(a) = \zeta^\star$, $\mathcal{E}(\mathcal{E}^T \mathrm{y}(a) - \zeta^\star) \neq 0$, meaning that $\|\mathcal{E}(\mathcal{E}^T \mathrm{y}(a) - \zeta^\star|)\| \geq \varsigma||\mathcal{E}^T \mathrm{y}(a) - \zeta^\star||^2$, where ς is the minimal nonzero singular value of $\mathcal{E}$. Hence, at any time step j, $v^T \nabla F(a^{(j)}) < -\underline{\sigma}(\mathcal{D})\varsigma F(a^{(j)})$. In turn, we conclude that $F(a^{(j+1)}) = F(a^{(j)}) - h\underline{\sigma}(\mathcal{D})\varsigma F(a^{(j)}) + \mathcal{O}(h) = (1 - h\underline{\sigma}(\mathcal{D})\varsigma)F(a^{(j)}) + \mathcal{O}(h)$. Iterating this equation shows that eventually, $F(a^{(j)}) < \varepsilon$, completing the proof. ∎

References

[1] A. Franchi, P. R. Giordano, C. Secchi, H. I. Son, and H. H. Bulthoff, "Passivity-based decentralized approach for the bilateral teleoperation of a group of UAVs with switching topology," in *Proc. IEEE Int. Conf. Robot. Automatica*, 2011, pp. 898–905.

[2] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Dynamical model of traffic congestion and numerical simulation," *Phys. Rev. E*, vol. 51, pp. 1035–1042, Feb. 1995.

[3] D. Urban and T. Keitt, "Landscape connectivity: A graph-theoretic perspective," *Ecology*, vol. 82, no. 5, pp. 1205–1218, 2001.

[4] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annu. Rev. Control Robot., Auton. Syst.*, vol. 2, no. 1, pp. 253–279, 2019.

[5] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Foundations Comput. Math.*, vol. 20, pp. 633–679, 2020.

[6] D. Görges, "Distributed adaptive linear quadratic control using distributed reinforcement learning," in *Proc. 5th IFAC Conf. Intell. Control Automat. Sci.*, 2019, pp. 218–223.

[7] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *Proc. 18th Eur. Control Conf.*, 2019, pp. 307–312.

[8] S. Fattahi, N. Matni, and S. Sojoudi, "Efficient learning of distributed linear-quadratic control policies," *SIAM J. Control Optimization*, SIAM, vol. 58, no. 5, pp. 2927–2951, 2020.

[9] H. Jiang and H. He, "Data-driven distributed output consensus control for partially observable multiagent systems," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 848–858, Mar. 2019.

[10] F. Bianchini, G. Fenu, G. Giordano, and F. A. Pellegrino, "Model-free tuning of plants with parasitic dynamics," in *Proc. 56th IEEE Conf. Decis. Control*, Dec. 2017, pp. 499–504.

[11] D. D. Šiljak, "Decentralized control and computations: Status and prospects," *Annu. Rev. Control*, vol. 20, pp. 131–141, 1996.

[12] Y. Zheng, S. E. Li, K. Li, and W. Ren, "Platooning of connected vehicles with undirected topologies: Robustness analysis and distributed h-infinity controller synthesis," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, pp. 1353–1364, May 2018.

[13] W. Yu, P. DeLellis, G. Chen, M. di Bernardo, and J. Kurths, "Distributed adaptive control of synchronization in complex networks," *IEEE Trans. Automat. Control*, vol. 57, pp. 2153–2158, Aug. 2012.

[14] J. M. Montenbruck and F. Allgöwer, "Some problems arising in controller design from big data via input–output methods," in *Proc. 55th IEEE Conf. Decis. Control*, 2016, pp. 6525–6530.

[15] A. Romer, J. M. Montenbruck, and F. Allgower, "Sampling strategies for data-driven inference of passivity properties," in *Proc. IEEE 56th Annu. Conf. Decis Control (CDC)*, 2017, pp. 6389–6394, doi: 10.1109/CDC.2017.8264623.

[16] A. Romer, J. Berberich, J. Köhler, and F. Allgöwer, "One-shot verification of dissipativity properties from input-output data," *IEEE Control Syst. Lett.*, vol. 3, no. 3, pp. 709–714, Jul. 2019.

[17] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice hall, vol. 3, 2001.

[18] M. Arcak, "Passivity as a design tool for group coordination," *IEEE Trans. Automat. Control*, vol. 52, pp. 1380–1390, Aug. 2007.

[19] H. Bai, M. Arcak, and J. Wen, "Cooperative control design: A systematic, passivity-based approach," *in Communications and Control Engineering*. Berlin, Germany: Springer, 2011.

[20] A. Pavlov and L. Marconi, "Incremental passivity and output regulation," *Syst. Control Lett.*, vol. 57, no. 5, pp. 400–409, 2008.

[21] G. H. Hines, M. Arcak, and A. K. Packarda, "Equilibrium-independent passivity: A new definition and numerical certification," *Automatica*, vol. 47, no. 9, pp. 1949–1956, 2011.

[22] M. Bürger, D. Zelazo, and F. Allgöwer, "Duality and network theory in passivity-based cooperative control," *Automatica*, vol. 50, no. 8, pp. 2051–2061, 2014.

[23] M. Sharf and D. Zelazo, "A network optimization approach to cooperative control synthesis," *IEEE Control Syst. Lett.*, vol. 1, pp. 86–91, Jul. 2017.

[24] M. Sharf and D. Zelazo, "Analysis and synthesis of MIMO multi-agent systems using network optimization," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4512–4524, Nov. 2019.

[25] A. Jain, M. Sharf, and D. Zelazo, "Regulatization and feedback passivation in cooperative control of passivity-short systems: A network optimization perspective," *IEEE Control Syst. Lett.*, vol. 2, pp. 731–736, Oct. 2018.

[26] M. Sharf, A. Jain, and D. Zelazo, "A geometric method for passivation and cooperative control of equilibrium-independent passivity-short systems," in *IEEE Trans. Automatic Control*, to be published, doi: 10.1109/TAC.2020.3043390.

[27] M. Sharf and D. Zelazo, "A data-driven and model-based approach to fault detection and isolation in networked systems," 2019, *arXiv:1908.03588*.

[28] J. M. Montenbruck, M. Bürger, and F. Allgöwer, "Practical synchronization with diffusive couplings," *Automatica*, vol. 53, pp. 235–243, 2015.

[29] J. Kim, J. Yang, H. Shim, J. Kim, and J. H. Seo, "Robustness of synchronization of heterogeneous agents by strong coupling and a large number of agents," *IEEE Trans. Automat. Control*, vol. 61, no. 10, pp. 3096–3102, Oct. 2016.

[30] C. Godsil and G. Royle, "Algebraic graph theory," in *Graduate Texts in Mathematics*. New York, NY, USA: Springer, 2001.

[31] R. T. Rockafellar, "Convex Analysis," in *Princeton Landmarks in Mathematics and Physics*. Princeton, NJ, USA: Princeton University Press, 1997.

[32] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[33] A. Romer, J. M. Montenbruck, and F. Allgöwer, "Determining dissipation inequalities from input–output samples," in *Proc. 20th IFAC World Congress*, 2017, pp. 7789–7794.

[34] L. Holst, "On the lengths of the pieces of a stick broken at random," *J. Appl. Probability*, vol. 17, no. 3, pp. 623–634, 1980.

[35] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality and robustness," *IEEE Trans. Automat. Control*, vol. 65, no. 3, pp. 909–924, Mar. 2020.

[36] M. Sharf and D. Zelazo, "Network identification: A passivity and network optimization approach," in *Proc. 57th IEEE Conf. Decis. Control*, 2018, pp. 2107–2113.

[37] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.

[38] J. Feiling, A. Zeller, and C. Ebenbauer, "Derivative-free optimization algorithms based on non-commutative maps," *IEEE Control Syst. Lett.*, vol. 2, pp. 743–748, Oct. 2018.

Miel Sharf (Member, IEEE) received the B.Sc. and M.Sc. degrees in mathematics and the Ph.D. degree from the Aerospace Engineering Department, from the Technion–Israel Institute of Technology, Haifa, Israel, in 2013, 2016, and 2020, respectively.

Since September 2020, he has been a Postdoctoral Researcher with the Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden. His research interests include the relation between graph theory and algebraic graph theory to multiagent systems, nonlinear control and passivity theory, data-driven control, and security in networked systems.

Dr. Sharf is a recipient of the Springer Thesis Award.

Anne Koch (Graduate Student Member, IEEE) received the M.Sc. degree in engineering science and mechanics from the Georgia Institute of Technology, Atlanta, USA, in 2014, and the M.Sc. degree in engineering cybernetics, in 2016 from the University of Stuttgart, Germany, where she is currently working toward her Ph.D. degree.

In 2016, she joined the Institute for Systems Theory and Automatic Control, University of Stuttgart, as a Research and Teaching Assistant. Her research interests include data-based systems analysis and controller design.

Daniel Zelazo (Senior Member, IEEE) received the B.Sc. and M.Eng. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1999 and 2001, respectively, and the Ph.D. degree in aeronautics and astronautics from the University of Washington, Seattle, WA, USA, in 2009.

He is an Associate Professor of Aerospace Engineering at the Technion–Israel Institute of Technology, Haifa, Israel. From 2010–2012, he was a Postdoctoral Research Associate and Lecturer with the Institute for Systems Theory and Automatic Control, University of Stuttgart, Germany. His research interests include multiagent systems, optimization, and graph theory.

Frank Allgöwer (Member, IEEE) received the Ph.D. degree from the University of Stuttgart, Stuttgart, Germany.

He is Professor of mechanical engineering with the University of Stuttgart, Germany, and Director of the Institute for Systems Theory and Automatic Control (IST), University of Stuttgart. His research interests include predictive control, data-based control, networked control, cooperative control, and nonlinear control with application to a wide range of fields including systems biology.

Dr. Allgöwer has been President of the International Federation of Automatic Control (IFAC) from 2017 to 2020, Vice-President for Technical Activities of the IEEE Control Systems Society during 2013/14, and Editor of the journal *Automatica* from 2001 to 2015. From 2012 to 2020, he was Vice-President of the German Research Foundation (DFG).