ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

A Feedback-Based Distributed Method for Multiscale Optimal Control of Multi-Agent Systems

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Brumali, R., Carnevale, G., Notarstefano, G. (2025). A Feedback-Based Distributed Method for Multiscale Optimal Control of Multi-Agent Systems. New York : IEEE [10.1109/cdc57313.2025.11312373].

Availability:
This version is available at: https://hdl.handle.net/11585/1040251 since: 2026-01-29

Published:
DOI: http://doi.org/10.1109/cdc57313.2025.11312373

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

03 August 2026

A Feedback-Based Distributed Method for Multiscale Optimal Control of Multi-Agent Systems

Riccardo Brumali, Guido Carnevale, and Giuseppe Notarstefano

Abstract— **In this paper, we propose a distributed method for multiscale optimal control. Namely, we aim to optimize the macroscopic behavior of nonlinear multi-agent systems by acting on the microscopic dynamics of each agent using only local information. More specifically, we consider a finite-horizon optimal control problem in which the loss function is the sum of (i) local terms taking into account local inputs of each agent and (ii) a term depending on a macroscopic model of the network (e.g., the Kullback-Leibler divergence with respect to a target distribution). In particular, such a macroscopic model aggregates local kernels representing a probabilistic feature of a single agent (e.g., a local sensing model). The proposed method combines a local feedback-based action to optimize the macroscopic agents' behavior with a learning part aimed at locally reconstructing the macroscopic model for each instant of the optimal control problem. We theoretically prove that the proposed method asymptotically converges to the set of stationary points of the optimal control problem. We achieve this result by exploiting tools from system theory based on timescale separation and LaSalle's invariance principle. We test the proposed method with numerical simulations.**

I. Introduction

Large-scale systems are gaining attention in many fields, such as multi-robot systems [1], [2] and smart grids [3]. This trend has led to the emergence of optimal control in multi-agent contexts, see the surveys [4], [5]. An emerging approach considers the overall network as a continuum and describes the macroscopic behavior of the agents via Partial Differential Equations (PDEs) [6], [7] and the global agents' interaction via Probability Density Functions (PDFs) [8]–[12]. This framework has applications ranging from motion planning [13] to the control of large telescopes [14], and from sensor placement [15] to self-organization of robotic swarms [16]–[18]. A problem that has gained particular attention in the context of multi-agent systems is the so-called shepherding control problem [19], [20], where the goal is to act on a small group of agents to guide a larger population towards a desired region. In these contexts, particularly in optimal control, centralized approaches where a single entity has access to all data and controls all the agents, are often impractical or even infeasible, motivating the need for distributed strategies that rely on local computations and communication among neighboring nodes to collaboratively achieve a common goal. In this direction, authors in [21] propose a decentralized strategy to tackle the shepherding control problem. Further examples of distributed control schemes include those based on the Alternating Direction Method of Multipliers [22]–[26], Differential Dynamic Programming [27], and distributed Model Predictive Control [28]–[32]. Among these, the work [33] proposes a distributed optimal control method based on the concept of feedback immersion, originally introduced in [34] and later extended to the discrete-time setting [35].

Our main contribution is the development of the novel algorithm named Distributed methOd for MultIscale Networked OPTimal control (DOMINOPT). Specifically, DOMINOPT acts on the microscopic dynamics of each agent of the network to optimize their emerging macroscopic behavior. We address optimal control scenarios in which the local agents' dynamics are nonlinear, while the cost function depends on both (i) local terms accounting for the local inputs of each agent and (ii) a global term depending on a macroscopic model of the overall network (e.g., the so-called Kullback-Leibler divergence with respect to a target distribution). We model it as the aggregation of local kernels, each representing a probabilistic feature of a single agent. The design of DOMINOPT integrates two main components. First, it incorporates a feedback-based optimal control approach [33], [35] into this multiscale optimal control framework. Second, it integrates a consensus-based mechanism that enables each agent to locally reconstruct the macroscopic model. We theoretically prove that DOMINOPT asymptotically converges to the set of stationary points of the considered nonconvex optimal control problem. We achieve this result by analyzing the dynamical system describing the DOMINOPT evolution with tools from system theory based on timescale separation and LaSalle's invariance principle.

We organize the paper as follows. In Section II, we state the problem setup. In Section III, we present DOMINOPT and its convergence properties. In Section IV, we analyze the proposed method and, in Section V, we numerically test it.

Notation: Given N vectors $x_1, \dots, x_N$, their vertical stack is denoted by $\mathrm{col}(x_1, \dots, x_N)$. Given a function $\ell : \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_d} \to \mathbb{R}$, we denote its (total) gradient at a given point $(\bar{x}_1, \dots, \bar{x}_d)$ as $\nabla \ell(\bar{x}_1, \dots, \bar{x}_d) := \mathrm{col}(\nabla_1 \ell(\bar{x}_1, \dots, \bar{x}_d), \dots, \nabla_d \ell(\bar{x}_1, \dots, \bar{x}_d))$, where the partial gradient of ℓ with respect to its k-th argument, for all $k = 1, \dots, d$ is denoted as $\nabla_k \ell(\bar{x}_1, \dots, \bar{x}_d) \in \mathbb{R}^{n_k}$. For a given $T \in \mathbb{N}$, we denote the discrete-time horizon as $[0, T] := \{0, 1, 2, \dots, T\}$. Given $x \in \mathbb{R}^n$ and $X \subseteq \mathbb{R}^n$, we define $\mathrm{dist}(x, X) := \inf_{y \in X} \|x - y\|$. 1_N denotes the N-dimensional vector with 1 in each entry, while $\mathbf{1}_{N,d} := 1_N \otimes I_d$ where the symbol $\otimes$ denotes the Kronecker product.

Funded by the European Union - PRIN 2022 ECODREAM Energy COmmunity management: DistRibutEd AlgorithMs and toolboxes for efficient and sustainable operations code [202228CTKY 002] - CUP [J53D23000560006].

The authors are with the Department of Electrical, Electronic and Information Engineering, Alma Mater Studiorum - Universita‘ di Bologna, Bologna, Italy, e-mail: {name.surname}@unibo.it

II. Problem Setup and Preliminaries

A. Network Model

We consider a set $\mathcal{I}{:=}\{1,\dots,N\}$ of agents each evolving as

$$x_{i,t+1} = f_i(x_{i,t}, u_{i,t}), \tag{1}$$

for all $i \in \mathcal{I}$, with $f_i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}^{n_i}$, while $x_{i,t} \in \mathbb{R}^{n_i}$ and $u_{i,t} \in \mathbb{R}^{m_i}$ are the state and control input of agent i at time t. We stack the states and inputs of all the agents at t as

$$x_t \coloneqq \mathrm{col}(x_{1,t},\dots,x_{N,t}) \in \mathbb{R}^n, \quad u_t \coloneqq \mathrm{col}(u_{1,t},\dots,u_{N,t}) \in \mathbb{R}^m,$$

where $n = \sum_{i=1}^N n_i$ and $m = \sum_{i=1}^N m_i$. Given a time horizon $T \in \mathbb{N}$, for each agent $i \in \mathcal{I}$, $\mathbf{x}_i \in \mathbb{R}^{n_iT}$ and $\mathbf{u}_i \in \mathbb{R}^{m_iT}$ denote the stack of the states $x_{i,1},\dots,x_{i,T}$ and inputs $u_{i,0},\dots,u_{i,T-1}$. We denote the stack of all $\mathbf{x}_i$, $\mathbf{u}_i$ as

$$\mathbf{x} \coloneqq \mathrm{col}(\mathbf{x}_1,\dots,\mathbf{x}_N) \in \mathbb{R}^{nT}, \quad \mathbf{u} \coloneqq \mathrm{col}(\mathbf{u}_1,\dots,\mathbf{u}_N) \in \mathbb{R}^{mT}.$$

The agents' communication is ruled by a directed graph $\mathcal{G} = (\mathcal{I}, \mathcal{E}, \mathcal{W})$, where $\mathcal{E} \subseteq \mathcal{I} \times \mathcal{I}$ is the edge set and $\mathcal{W} \in \mathbb{R}^{N\times N}$ is the weighted adjacency matrix, with ij-entry denoted as $w_{ij} \in \mathbb{R}$. In detail, $w_{ij} > 0$ if $(j,i) \in \mathcal{E}$ and $w_{ij} = 0$ otherwise. Agent i can receive information from agent j if and only if $j \in \mathcal{N}_i \coloneqq \{j \in \{1,\dots,N\} \mid (j,i) \in \mathcal{E}\}$. The graphs considered in this work are characterized as follows.

Assumption 1. *The graph $\mathcal{G}$ is strongly connected and the matrix $\mathcal{W}$ is doubly stochastic.* □

B. Multiscale Optimal Control

The agents' goal is to cooperatively optimize a common index describing their emergent macroscopic behavior in a time interval $[0,T]$ and a finite space $\mathcal{C} \subset \mathbb{R}^\eta$ with $\eta \le \min_{i\in\mathcal{I}}\{n_i\}$. For instance, a team of mobile robots may optimize its position in space to track an ideal macroscopic distribution. Let $\rho_t : \mathbb{R}^n \times \mathbb{R}^\eta \to \mathbb{R}^\eta_+$ be a map associating the microscopic states x_t to their macroscopic representation in $c \in \mathcal{C}$. We formally define the macroscopic performance index at time t as the function $L_t : \mathbb{R}^n \to \mathbb{R}_+$ defined as

$$L_t(x_t) \coloneqq \int_{\mathcal{C}} h_t\left(\rho_t(x_t, c)\right) dc,$$

where $h_t : \mathbb{R}_+ \to \mathbb{R}$ is a performance criterion. For numerical reasons, we consider a discretized version of $L_t(x_t)$. Given a discretization interval $\delta > 0$, we divide the space into $C \in \mathbb{N}$ cells. Let $\psi : \mathbb{N} \to \mathbb{R}^\eta$ be the map associating the c-th cell to the coordinates of its center. We obtain a discretized cost

$$\ell_t^{\text{mac}}(p_t(x_t)) \coloneqq \textstyle\sum_{c=1}^C h_t\left(p_t^c(x_t)\right),$$

where $p_t^c(x) = \rho_t(x, \psi(c))$, $p_t(x) \coloneqq \mathrm{col}(p_t^1(x),\dots,p_t^C(x)) \in \mathbb{R}^C$, and $\mathbf{p}(\mathbf{x}) \coloneqq \mathrm{col}(p_1(x),\dots,p_T(x)) \in \mathbb{R}^{CT}$. Agents aim to minimize the overall cost $\ell : \mathbb{R}^{mT} \times \mathbb{R}_+^{CT} \to \mathbb{R}$ given by

$$\ell(\mathbf{u}, \mathbf{p}(\mathbf{x})) \coloneqq \textstyle\sum_{i=1}^N \ell_i^{\text{mic}}(\mathbf{u}_i) + \ell^{\text{mac}}(\mathbf{p}(\mathbf{x})), \tag{2}$$

in which, $\ell^{\text{mac}} : \mathbb{R}_+^{CT} \to \mathbb{R}$ depends on the macroscopic model $\mathbf{p}(\mathbf{x})$ of the network, while each local cost $\ell_i^{\text{mic}} : \mathbb{R}^{m_iT} \to \mathbb{R}$ depends only on the local input $\mathbf{u}_i$, specifically

$$\ell_i^{\text{mic}}(\mathbf{u}_i) \coloneqq \sum_{t=0}^{T-1} \ell_{i,t}^{\text{mic}}(u_{i,t}), \quad \ell^{\text{mac}}(\mathbf{p}(\mathbf{x})) \coloneqq \sum_{t=1}^{T} \ell_t^{\text{mac}}(p_t(x_t)),$$

where each $\ell_{i,t}^{\text{mic}} : \mathbb{R}^{m_i} \to \mathbb{R}$ and $\ell_t^{\text{mac}} : \mathbb{R}_+^C \to \mathbb{R}$ are the microscopic inputs and macroscopic model cost at time t, respectively. We aim to solve the multiscale optimal control problem

$$\min_{\substack{\mathbf{x}_1,\dots,\mathbf{x}_N \\ \mathbf{u}_1,\dots,\mathbf{u}_N}} \ \sum_{i=1}^N \sum_{t=0}^{T-1} \ell_{i,t}^{\text{mic}}(u_{i,t}) + \sum_{t=1}^{T} \ell_t^{\text{mac}}(p_t(x_t)) \tag{3a}$$

$$\text{subj. to} \ \ x_{i,t+1} = f_i(x_{i,t}, u_{i,t}) \quad t = 0,\dots,T-1 \tag{3b}$$

$$x_{i,0} = x_{i,\text{init}} \quad i = 1,\dots,N, \tag{3c}$$

where $x_{i,\text{init}} \in \mathbb{R}^{n_i}$ represents the initial condition of agent i.

C. Macroscopic Behaviors via Kernel Density Estimation

We focus on macroscopic models $p_t(x)$ being probability distributions over $\mathcal{C}$ depending on all the microscopic states $x_{i,t}$. By following the idea of Kernel Density Estimation (KDE) [36]–[38], we assume that $p_t(x)$ can be expressed as a function of microscopic models $p_{i,t} : \mathbb{R}^{n_i} \to \mathbb{R}_+^C$, namely

$$p_t(x_t) \coloneqq \frac{1}{N}\sum_{i=1}^N p_{i,t}(x_{i,t}),$$

for all $t \in [1,T]$. This definition allows us to rewrite $\mathbf{p}(\mathbf{x})$ as

$$\mathbf{p}(\mathbf{x}) \coloneqq \frac{1}{N}\sum_{i=1}^N p_i(\mathbf{x}_i),$$

where $p_i(\mathbf{x}_i) \coloneqq \mathrm{col}(p_{i,1}(x_{i,1}),\dots,p_{i,T}(x_{i,T})) \in \mathbb{R}^{\eta T}$.

Example 1. Gaussian kernels *may be a choice each $p_{i,t}(x_{i,t})$. Let $\iota_i : \mathbb{R}^{n_i} \to \mathbb{R}^\eta$ be a projection selecting η components of $x_{i,t}$. We can choose the c-th component of $p_{i,t}(x_{i,t})$ as*

$$p_{i,t}^c\left(x_{i,t}\right) \coloneqq \frac{e^{-\frac{1}{2}(\psi(c)-\iota_i(x_{i,t}))^\top \Sigma^{-1}(\psi(c)-\iota_i(x_{i,t}))}}{\sqrt{(2\pi)^\eta \left|\Sigma_i\right|}},$$

for all $c \in 1,\dots,C$ and a given $\Sigma_i \in \mathbb{R}^{\eta\times\eta}$. □

The considered functions satisfy the following assumption.

Assumption 2 (Dynamics and Cost)**.** *The functions f_i, $\ell_{i,t}^{\text{mic}}$, ℓ_t^{mac}, and p_i are $\mathcal{C}^2$ for all $i \in \mathcal{I}$ and $t \in [0,T]$. Further, $\ell(\cdot, \mathbf{p}(\cdot))$ is radially unbounded.* □

III. Distributed Multiscale Optimal Control

In this section, we present our distributed algorithm to solve the multiscale optimal control problem (3).

A. Distributed Feedback Policy

Before presenting our strategy, we clarify that we call *curve* every state-input sequence $(\boldsymbol{\alpha}_i, \boldsymbol{\mu}_i) \in \mathbb{R}^{n_iT} \times \mathbb{R}^{m_iT}$, hence a curve $(\boldsymbol{\alpha}_i, \boldsymbol{\mu}_i)$ does not have to satisfy the constraint (1) introduced by agent i dynamics. On the other hand, we call $(\mathbf{x}_i, \mathbf{u}_i) \in \mathbb{R}^{n_i} \times \mathbb{R}^{m_i}$ *trajectory* for agent i if it satisfies the constraint (1). By following the approach in [33]–[35], we tackle problem (3) introducing local feedback policies mapping local state-input curves into trajectories. This is achieved via the local feedback policy

$$\begin{aligned} u_{i,t} &= \mu_{i,t} + \kappa_{i,t}(\alpha_{i,t} - x_{i,t}) \\ x_{i,t+1} &= f_i(x_{i,t}, u_{i,t}), \end{aligned} \tag{4}$$

for all $i \in \mathcal{I}$, where $\kappa_{i,t} : \mathbb{R}^{n_i} \to \mathbb{R}^{m_i}$ is a given feedback policy designed to ensure numerical stability properties. The controllers (4) leads to the equivalent closed-loop formulation

$$\min_{\substack{\mathbf{x}_i,\mathbf{u}_i,\boldsymbol{\alpha}_i,\boldsymbol{\mu}_i\\ i=1,\ldots,N}} \sum_{i=1}^{N}\sum_{t=0}^{T-1} \ell_{i,t}^{\text{mic}}(u_{i,t}) + \sum_{t=1}^{T} \ell_t^{\text{mac}}(p_t(x_t)) \tag{5a}$$

$$\text{subj. to } x_{i,t+1} = f_i(x_{i,t}, u_{i,t}) \quad t = 0, \ldots, T-1 \tag{5b}$$

$$u_{i,t} = \mu_{i,t} + \kappa_{i,t}(\alpha_{i,t} - x_{i,t}) \tag{5c}$$

$$x_{i,0} = x_{i,\text{init}} \quad i = 1, \ldots, N. \tag{5d}$$

B. DOMINOPT

We report the overall description of DOMINOPT in Algorithm 1 by adopting the definitions

$$A_{i,t}^k \coloneqq \nabla_1 f_i(x_{i,t}^k, u_{i,t}^k)^\top, \qquad B_{i,t}^k \coloneqq \nabla_2 f_i(x_{i,t}^k, u_{i,t}^k)^\top,$$

$$K_{i,t}^k \coloneqq \nabla\kappa_{i,t}(\alpha_{i,t}^k - x_{i,t}^k)^\top, \quad g_t^{\text{mac}}(s) \coloneqq \frac{\partial \ell_t^{\text{mac}}(\tilde{s})}{\partial \tilde{s}}^\top \Bigg|_{\tilde{s}=s},$$

for all $i \in \mathcal{I}$ and $t \in [0, T-1]$. Specifically, for each

Algorithm 1 DOMINOPT – Agent i

for $k = 0, 1, 2 \ldots$ **do**

$$\lambda_{i,T}^k = \frac{\nabla p_{i,T}(x_{i,T}^k)}{N} g_T^{\text{mac}}(P_+[s_{i,T}^k]) \tag{6}$$

for $t = T-1, \ldots, 1$ **do**

Compute proxy variables based on $s_{i,t}^k$

$$\hat{a}_{i,t}^k = \frac{\nabla p_{i,t}(x_{i,t}^k)}{N} g_t^{\text{mac}}(P_+[s_{i,t}^k]) \tag{7a}$$

$$b_{i,t}^k = \nabla \ell_{i,t}^{\text{mic}}(u_{i,t}^k) \tag{7b}$$

$$\lambda_{i,t}^k = (A_{i,t}^k - B_{i,t}^k K_{i,t}^k)^\top \lambda_{i,t+1}^k + \hat{a}_{i,t}^k - K_{i,t}^{k,\top} b_{i,t}^k \tag{7c}$$

end for

for $t = 0, \ldots, T-1$ **do**

Update curve with $b_{i,0}^k = \nabla \ell_{i,0}^{\text{mic}}(u_{i,0}^k)$

$$\begin{aligned} \alpha_{i,t}^{k+1} &= \alpha_{i,t}^k + \gamma K_{i,t}^{k,\top}\left(-b_{i,t}^k - B_{i,t}^{k\top}\lambda_{i,t+1}^k\right) \\ \mu_{i,t}^{k+1} &= \mu_{i,t}^k + \gamma\left(-b_{i,t}^k - B_{i,t}^{k\top}\lambda_{i,t+1}^k\right) \end{aligned} \tag{8}$$

Compute new trajectory with $x_{i,0}^{k+1} = x_{i,\text{init}}$

$$u_{i,t}^{k+1} = \mu_{i,t}^{k+1} + \kappa_{i,t}(\alpha_{i,t}^{k+1} - x_{i,t}^{k+1}) \tag{9a}$$

$$x_{i,t+1}^{k+1} = f_i(x_{i,t}^{k+1}, u_{i,t}^{k+1}) \tag{9b}$$

Update trackers

$$s_{i,t+1}^{k+1} = \sum_{j\in\mathcal{N}_i} w_{ij} s_{j,t+1}^k + p_{i,t+1}(x_{i,t+1}^{k+1}) - p_{i,t+1}(x_{i,t+1}^k)$$

end for

end for

agent i, we first update the current curve $(\boldsymbol{\alpha}_i^k, \boldsymbol{\mu}_i^k)$ (cf. (8)) and, then, we recover the corresponding trajectory $(\mathbf{x}_i^k, \mathbf{u}_i^k)$ (cf. (9)) through the local feedback policy (4). The rationale behind the update mechanism of $(\boldsymbol{\alpha}_i^k, \boldsymbol{\mu}_i^k)$ (cf. (8)) is to approximate a (centralized) gradient update by replacing the global information (not locally available) with local proxies $s_{i,t} \in \mathbb{R}^C$ aimed at locally learning the macroscopic model $p_t(x_t^k)$ of the network at each time $t \in [1, T]$ via the consensus-based scheme ruling the update of each $s_{i,t}^k$. We also note that the proxies $s_{i,t}$ are used through the operator $P_+[\cdot]$ projecting onto the positive orthant $\mathbb{R}_+^{\mathcal{C}}$. Indeed, we recall that $p_t(x_t^k)$ is a PDF defined in the space $\mathcal{C}$. Thus, this projection allows us to replace it with the local estimate $s_{i,t}^k$ ensuring by construction that each $P_+[s_{i,t}^k] \geq 0$ for all $k \in \mathbb{N}$. Notice that the data stored by each agent do not grow with the number of agents, making the algorithm scalable.

C. Algorithm Convergence Result

In this section, we provide the convergence result characterizing Algorithm 1. For the sake of readability, let us introduce the stack vector $\mathbf{s}_i^k \coloneqq \text{col}(s_{i,1}^k, \ldots, s_{i,T}^k)$.

Theorem 1. *Consider Algorithm 1. Let Assumptions 1 and 2 hold. For all initial conditions $(\mathbf{x}_i^0, \mathbf{u}_i^0, \mathbf{s}_i^0) \in \mathbb{R}^{n_iT} \times \mathbb{R}^{m_iT} \times \mathbb{R}^{CT}$ and $(\boldsymbol{\alpha}_i^0, \boldsymbol{\mu}_i^0) \in \mathbb{R}^{n_iT} \times \mathbb{R}^{m_iT}$ such that $(\mathbf{x}_i^0, \mathbf{u}_i^0)$ is a trajectory and $\mathbf{s}_i^0 = p_i(\mathbf{x}_i^0)$ for all $i \in \mathcal{I}$, there exists $\bar{\gamma} > 0$ such that, for all $\gamma \in (0, \bar{\gamma})$, it holds*

$$\lim_{k\to\infty} \text{dist}\left((\mathbf{x}^k, \mathbf{u}^k), \Xi^\star\right) = 0,$$

where $\Xi^\star$ denotes the set of trajectories $(\mathbf{x}^\star, \mathbf{u}^\star)$ satisfying the first-order necessary conditions for optimality of (3). □

In Section IV-F, we provide a sketch of the proof of Theorem 1, while the complete proof will be provided in a forthcoming document. Specifically, in Section IV, we pursue a system-theoretical analysis of the dynamical system describing the evolution of DOMINOPT over the iterations k. More in detail, our analysis uses timescale separation to conveniently identify a *fast* subsystem aimed at locally learning the macroscopic state, and a *slow* one driving the microscopic agent dynamics toward an emerging behavior that is optimal with respect to problem (3). We remark that the algorithm is analyzed in a generic nonconvex setting. Consequently, the solution to problem (3) may not be unique, and only convergence to points satisfying the first-order necessary conditions for optimality can be guaranteed.

IV. Algorithm Analysis

The analysis of DOMINOPT relies on the following steps. First, in Section IV-A, the local controllers (9a) allow us to reformulate (5) as an unconstrained optimization problem. Second, in Section IV-B, we show that our strategy reads as an approximation of a gradient descent method interlaced with the evolution of the proxies for the macroscopic model. We conveniently analyze this interconnection as a two-time-scale system, where the fast subsystem describes the proxies' evolution, while the slow one describes the gradient-based part. Finally, based on the previous points, in Section IV-F, we provide a sketch of the proof of Theorem 1.

A. Reduced Reformulation of the Optimal Control Problem

Given an initial condition $x_{i,\text{init}} \in \mathbb{R}^{n_i}$, the space of trajectories $\mathcal{T}_{i,\text{init}} \subseteq \mathbb{R}^{n_iT} \times \mathbb{R}^{m_iT}$ of agent i is defined as

$$\mathcal{T}_{i,\text{init}} \coloneqq \Big\{(\mathbf{x}_i, \mathbf{u}_i) \in \mathbb{R}^{n_iT} \times \mathbb{R}^{m_iT} \mid x_{i,0} = x_{i,\text{init}}$$

$$\left.x_{i,t+1} = f_i(x_{i,t}, u_{i,t}) \quad \forall t \in [0, T]\right\},$$

for all $i \in \mathcal{I}$. For each agent i, the local policy (4) implements a (local) feedback-projection operator $\mathcal{P}_i : \mathbb{R}^{n_i T} \times \mathbb{R}^{m_i T} \to \mathcal{T}_{i,\text{init}}$, which maps (local) state-input curves into trajectories for agent i. Namely, $\mathcal{P}_i(\boldsymbol{\alpha}_i, \boldsymbol{\mu}_i)$ is designed such that for any $(\boldsymbol{\alpha}_i, \boldsymbol{\mu}_i) \in \mathbb{R}^{n_i T} \times \mathbb{R}^{m_i T}$ and $(\mathbf{x}_i, \mathbf{u}_i) \in \mathcal{T}_{i,\text{init}}$, it holds

$$\begin{bmatrix} \boldsymbol{\alpha}_i \\ \boldsymbol{\mu}_i \end{bmatrix} \longmapsto \begin{bmatrix} \mathbf{x}_i \\ \mathbf{u}_i \end{bmatrix} \coloneqq \mathcal{P}_i(\boldsymbol{\alpha}_i, \boldsymbol{\mu}_i) \coloneqq \begin{bmatrix} \mathcal{P}_i^{\mathbf{x}}(\boldsymbol{\alpha}_i, \boldsymbol{\mu}_i) \\ \mathcal{P}_i^{\mathbf{u}}(\boldsymbol{\alpha}_i, \boldsymbol{\mu}_i) \end{bmatrix},$$

where $\mathcal{P}_i^{\mathbf{x}}(\boldsymbol{\alpha}_i, \boldsymbol{\mu}_i)$ and $\mathcal{P}_i^{\mathbf{u}}(\boldsymbol{\alpha}_i, \boldsymbol{\mu}_i)$ are the state and input components of $\mathcal{P}_i(\boldsymbol{\alpha}_i, \boldsymbol{\mu}_i)$. For all $t \in [0, T-1]$ and $i \in \mathcal{I}$, we also use $x_{i,t} = \mathcal{P}_{i,t}^{\mathbf{x}}(\alpha_{i,t}, \mu_{i,t})$, $u_{i,t} = \mathcal{P}_{i,t}^{\mathbf{u}}(\alpha_{i,t}, \mu_{i,t})$, where $\mathcal{P}_{i,t}^{\mathbf{x}} : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}^{n_i}$ and $\mathcal{P}_{i,t}^{\mathbf{u}} : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}^{m_i}$. We denote as $\mathcal{P}_i^{\mathbf{x}}(\cdot, \cdot)$ and $\mathcal{P}_i^{\mathbf{u}}(\cdot, \cdot)$ the stack of all $\mathcal{P}_{i,t}^{\mathbf{x}}(\cdot, \cdot)$ and $\mathcal{P}_{i,t}^{\mathbf{u}}(\cdot, \cdot)$ for all $t \in [0, T-1]$ and as $\mathcal{P}^{\mathbf{x}}(\cdot, \cdot)$, $\mathcal{P}^{\mathbf{u}}(\cdot, \cdot)$ the stack of $\mathcal{P}_i^{\mathbf{x}}(\cdot, \cdot)$, $\mathcal{P}_i^{\mathbf{u}}(\cdot, \cdot)$ for all $i \in \mathcal{I}$. Thus, given $\boldsymbol{\xi}_i \coloneqq (\boldsymbol{\alpha}_i, \boldsymbol{\mu}_i)$ and $\boldsymbol{\xi} \coloneqq \text{col}(\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_N)$, we can reformulate (5) as

$$\min_{\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_N} \underbrace{\sum_{i=1}^{N} J_i(\boldsymbol{\xi}_i, \mathbf{p}(\boldsymbol{\xi}))}_{J(\boldsymbol{\xi}, \mathbf{p}(\boldsymbol{\xi}))}, \tag{10}$$

where $J_i(\boldsymbol{\xi}_i, \mathbf{p}(\boldsymbol{\xi})) \coloneqq \ell_i^{\text{mic}}(\mathcal{P}_i^{\mathbf{u}}(\boldsymbol{\xi}_i)) + \frac{1}{N} \ell^{\text{mac}}(\mathbf{p}(\mathcal{P}^{\mathbf{x}}(\boldsymbol{\xi})))$ and

$$\mathbf{p}(\boldsymbol{\xi}) \coloneqq \frac{1}{N} \sum_{i=1}^{N} \bar{\mathbf{p}}_i(\boldsymbol{\xi}_i),$$

with $\bar{\mathbf{p}}_i(\boldsymbol{\xi}_i) \coloneqq \text{col}(p_{i,1}(\mathcal{P}_{i,1}^{\mathbf{x}}(\boldsymbol{\xi}_i)), \dots, p_{i,T}(\mathcal{P}_{i,T}^{\mathbf{x}}(\boldsymbol{\xi}_i)))$.

B. Descent Direction and Local Proxies

In this section, we formally show that Algorithm 1 embeds a numerical routine for the local approximation of the gradient of the reduced cost J with respect to $\boldsymbol{\xi}_i$. To properly write this result, we introduce $g_i^J : \mathbb{R}^{(n+m)T} \to \mathbb{R}^{(m_i T + n_i T)}$ defined as

$$\begin{aligned} g_i^J(\boldsymbol{\xi}) &\coloneqq \left. \frac{\partial J(z_i, \mathbf{p}(z_1, \dots, z_N))}{\partial z_i} \right|_{z=\boldsymbol{\xi}}^{\top} \\ &= \nabla \ell_i^{\text{mic}}(\mathcal{P}_i^{\mathbf{u}}(\boldsymbol{\xi}_i)) + \tfrac{\nabla \bar{\mathbf{p}}_i(\boldsymbol{\xi}_i)}{N} g^{\text{mac}}(\mathbf{p}(\boldsymbol{\xi})), \end{aligned} \tag{11}$$

for all $i \in \mathcal{I}$, where $g^{\text{mac}}(s) \coloneqq \begin{bmatrix} g_0^{\text{mac}}(s)^\top & \dots & g_T^{\text{mac}}(s)^\top \end{bmatrix}^\top$.

Lemma 1. *Consider a state-input curve $\boldsymbol{\xi} = (\boldsymbol{\alpha}, \boldsymbol{\mu})$. For all $i \in \mathcal{I}$, the integration backward in time from $t = T-1, \dots, 1$ of the closed-loop costate equation reads as*

$$\lambda_{i,t} = (A_{i,t} - K_{i,t} B_{i,t})^\top \lambda_{i,t+1} + a_{i,t} - K_{i,t}^\top b_{i,t} \tag{12a}$$

$$A_{i,t} \coloneqq \nabla_1 f_i(x_{i,t}, u_{i,t})^\top, \; B_{i,t} \coloneqq \nabla_2 f_i(x_{i,t}, u_{i,t})^\top \tag{12b}$$

$$K_{i,t} \coloneqq \nabla \kappa_{i,t}(\alpha_{i,t} - x_{i,t})^\top \tag{12c}$$

$$a_{i,t} \coloneqq \frac{\nabla p_{i,t}(x_{i,t})}{N} g_t^{mac}(p_t(x_t)), \; b_{i,t} \coloneqq \nabla \ell_{i,t}^{mic}(u_{i,t}), \tag{12d}$$

with $\lambda_{i,T} \in \mathbb{R}^{n_i}$ and $b_{i,0} \in \mathbb{R}^{m_i}$ defined as

$$\lambda_{i,T} = \tfrac{\nabla p_{i,T}(x_{i,T})}{N} g_T^{mac}(p_T(x_T)), \; b_{i,0} \coloneqq \nabla \ell_{i,0}^{mic}(u_{i,0}). \tag{12e}$$

Then, for all $i \in \mathcal{I}$, the gradient of the reduced cost function (10) *with respect to $\mu_{i,t}$ and $\alpha_{i,t}$ read as*

$$[g_i^J(\boldsymbol{\xi})]_t = \begin{bmatrix} -b_{i,t} - B_{i,t} \lambda_{i,t+1} \\ K_{i,t}^\top \Delta \mu_{i,t} \end{bmatrix},$$

where the operator $[\cdot]_t$ denotes the t-th block of (11). □

The proof will be provided in a forthcoming document.

Notice that, the evaluation of (12d) and (12e) requires the local knowledge of the macroscopic model $p_t(\cdot)$. For this reason, in the computation of $\lambda_{i,T}^k$ and $a_{i,t}^k$, the macroscopic model is approximated via $P_+[s_{i,t}]$ (cf. (6)-(7a)), namely, Algorithm 1 updates each $[\boldsymbol{\xi}_i^k]_t = (\alpha_{i,t}^k, \mu_{i,t}^k)$ via $\gamma[d_i(\boldsymbol{\xi}_i^k, \mathbf{s}_i^k)]_t$, where $d_i : \mathbb{R}^{(m_i T + nT)} \times \mathbb{R}^{CT} \to \mathbb{R}^{(m_i T + n_i T)}$ is defined as

$$d_i(\boldsymbol{\xi}_i, s_i) \coloneqq \nabla \ell_i^{\text{mic}}(\mathcal{P}_i^{\mathbf{u}}(\boldsymbol{\xi}_i)) + \frac{\nabla \bar{\mathbf{p}}_i(\boldsymbol{\xi}_i)}{N} g^{\text{mac}}(s_i), \tag{13}$$

for all $i \in \mathcal{I}$. Indeed, by comparing the expressions (11) and (13), we note that if $\mathbf{s}_i^k = \mathbf{p}(\boldsymbol{\xi}^k)$, then $d_i(\boldsymbol{\xi}_i, \mathbf{p}(\boldsymbol{\xi}_i)) = g_i^J(\boldsymbol{\xi}_i)$ is the exact descent direction for all $i \in \mathcal{I}$.

C. Algorithm Reformulation as a Two-Time-Scales System

Next, we provide a reformulation of Algorithm 1 used to prove Theorem 1. Let $W \coloneqq \mathcal{W} \otimes I_{CT}$, $\mathbf{s}^k \coloneqq \text{col}(\mathbf{s}_1^k, \dots, \mathbf{s}_N^k)$, $\boldsymbol{\xi}^k \coloneqq \text{col}(\boldsymbol{\xi}_1^k, \dots, \boldsymbol{\xi}_N^k)$, $d(\boldsymbol{\xi}^k, \mathbf{s}^k) \coloneqq \text{col}(d_1(\boldsymbol{\xi}_1^k, \mathbf{s}_1^k), \dots, d_N(\boldsymbol{\xi}_N^k, \mathbf{s}_N^k))$, $\bar{\mathbf{p}} \coloneqq \text{col}(\mathbf{p}_1, \dots, \mathbf{p}_N)$, $\mathbf{w}^k \coloneqq \mathbf{s}^k - \bar{\mathbf{p}}(\boldsymbol{\xi}^k)$, and compactly rewrite Algorithm 1 as

$$\boldsymbol{\xi}^{k+1} = \boldsymbol{\xi}^k - \gamma d(\boldsymbol{\xi}^k, \mathbf{w}^k + \bar{\mathbf{p}}(\boldsymbol{\xi}^k)) \tag{14a}$$

$$\mathbf{w}^{k+1} = W \mathbf{w}^k + (W - I) \bar{\mathbf{p}}(\boldsymbol{\xi}^k). \tag{14b}$$

Now, we introduce the novel variables $\mathbf{w}_{\text{avg}} \coloneqq \frac{\mathbf{1}^\top}{N} \mathbf{w}$, $\boldsymbol{\eta} \coloneqq R^\top \mathbf{w}$, where $R \in \mathbb{R}^{NCT \times (N-1)(CT)}$ such that $R^\top R = I$, $R^\top \mathbf{1} = 0$. It is possible to show that $\mathbf{w}_{\text{avg}}^{k+1} = \mathbf{w}_{\text{avg}}^k$. This result combined with the choice of the initial conditions implies $\mathbf{w}_{\text{avg}}^k = 0$ for all $k \in \mathbb{N}$. Hence, from now on, we will ignore $\mathbf{w}_{\text{avg}}^k$ and rewrite (14) as the restricted system

$$\boldsymbol{\xi}^{k+1} = \boldsymbol{\xi}^k - \gamma d(\boldsymbol{\xi}^k, R\boldsymbol{\eta}^k + \bar{\mathbf{p}}(\boldsymbol{\xi}^k)) \tag{15a}$$

$$\boldsymbol{\eta}^{k+1} = R^\top W R \boldsymbol{\eta}^k + R^\top (W - I) \bar{\mathbf{p}}(\boldsymbol{\xi}^k). \tag{15b}$$

Subsystem (15b) has equilibria parametrized in $\boldsymbol{\xi}^k$ through

$$\boldsymbol{\eta}^{\text{eq}}(\boldsymbol{\xi}) \coloneqq -R^\top \bar{\mathbf{p}}(\boldsymbol{\xi}). \tag{16}$$

In light of this aspect and since we can arbitrarily tune the variations of (15a) via γ, we interpret the interconnection (15) as a two-time-scale system, where (15a) is the slow subsystem, while subsystem (15b) is the fast one. According to this interpretation, we now study the so-called boundary layer and reduced system, namely, two auxiliary schemes associated to the fast part (15b) and the slow one (15a), respectively.

D. Boundary Layer System

In this section, we study the boundary layer system associated to the two-time-scale system (15). Namely, we focus on the fast subsystem (15b) by considering an arbitrarily fixed slow state $\boldsymbol{\xi}^k = \boldsymbol{\xi} \in \mathbb{R}^{(n+m)T}$ for all $k \in \mathbb{N}$ and using the error coordinate $\tilde{\boldsymbol{\eta}}^k \coloneqq \boldsymbol{\eta}^k - \boldsymbol{\eta}^{\text{eq}}(\boldsymbol{\xi})$, thus obtaining

$$\tilde{\boldsymbol{\eta}}^{k+1} = R^\top W R \tilde{\boldsymbol{\eta}}^k. \tag{17}$$

Next, we ensure the existence of a Lyapunov function proving global exponential stability of the origin for (17).

Lemma 2. *There exists a continuous function* $U : \mathbb{R}^{(N-1)CT} \to \mathbb{R}$ *such that*

$$b_1\|\tilde{\boldsymbol{\eta}}\|^2 \leq U(\tilde{\boldsymbol{\eta}}) \leq b_2\|\tilde{\boldsymbol{\eta}}\|^2 \tag{18a}$$

$$U\big(R^\top W R\tilde{\boldsymbol{\eta}}\big) - U(\tilde{\boldsymbol{\eta}}) \leq -b_3\|\tilde{\boldsymbol{\eta}}\|^2 \tag{18b}$$

$$\big|U(\tilde{\boldsymbol{\eta}}) - U(\tilde{\boldsymbol{\eta}}')\big| \leq b_4\|\tilde{\boldsymbol{\eta}} - \tilde{\boldsymbol{\eta}}'\| \left(\|\tilde{\boldsymbol{\eta}}\| + \|\tilde{\boldsymbol{\eta}}'\|\right), \tag{18c}$$

for all $\tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{\eta}}' \in \mathbb{R}^{(N-1)CT}$ *and some* $b_1, b_2, b_3, b_4 > 0$. □

The proof will be provided in a forthcoming document.

E. Reduced System

Now we study the reduced system associated to (15). We focus on the slow part (15a) with $\boldsymbol{\eta}^k = \boldsymbol{\eta}^{\text{eq}}(\boldsymbol{\xi}^k)$ for all $k \in \mathbb{N}$. By observing the definitons of $\boldsymbol{\eta}^{\text{eq}}$ (cf. (16)) and the components d_i (cf. (13)) of d, the reduced system reads as

$$\boldsymbol{\xi}^{k+1} = \boldsymbol{\xi}^k - \gamma \nabla J_{\mathbf{p}}(\boldsymbol{\xi}^k), \tag{19}$$

where $J_{\mathbf{p}}(\boldsymbol{\xi}) := J(\boldsymbol{\xi}, \mathbf{p}(\boldsymbol{\xi}))$. Note that system (19) exactly recovers the gradient descent method applied to problem (10). Next, we ensure semi-global attractiveness properties of $\Xi^\star := \{\boldsymbol{\xi} \in \mathbb{R}^{(n+m)T} \mid \nabla J_{\mathbf{p}}(\boldsymbol{\xi}) = 0\}$ for the trajectories of (19).

Lemma 3. *For every compact set* $\mathcal{S} \subset \mathbb{R}^{(n+m)T}$*, there exists* $\bar{\gamma}_1 > 0$ *such that, for all* $\gamma \in (0, \bar{\gamma}_1)$*, it holds*

$$J_{\mathbf{p}}\left(\boldsymbol{\xi} - \gamma \nabla J_{\mathbf{p}}\left(\boldsymbol{\xi}\right)\right) - J_{\mathbf{p}}\left(\boldsymbol{\xi}\right) \leq -\gamma a_1 \left\|\nabla J_{\mathbf{p}}\left(\boldsymbol{\xi}\right)\right\|^2 \tag{20a}$$

$$\begin{aligned} J_{\mathbf{p}}\left(\boldsymbol{\xi} + \boldsymbol{\xi}_1\right) - J_{\mathbf{p}}\left(\boldsymbol{\xi} + \boldsymbol{\xi}_2\right) &\leq a_2 \left\|\nabla J_{\mathbf{p}}\left(\boldsymbol{\xi}\right)\right\| \left\|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\right\| \\ &\quad + a_2\left(\|\boldsymbol{\xi}_1\|^2 + \|\boldsymbol{\xi}_2\|^2\right), \end{aligned} \tag{20b}$$

for all $\boldsymbol{\xi}, \boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \in \mathcal{S}$ *and some* $a_1, a_2 > 0$. □

The proof will be provided in a forthcoming document.

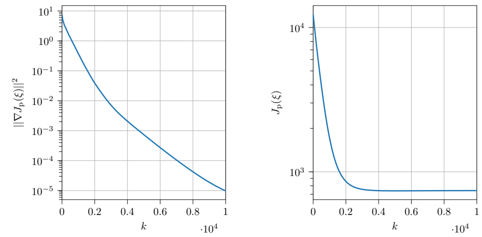
F. Sketch of the Proof of Theorem 1

Here, we provide a sketch of the proof of Theorem 1, while the complete one will be provided in a forthcoming document. Such a proof is based on two main steps. The first step consists in using the boundary layer (cf. Section IV-D) and reduced systems analysis (cf. Section IV-E) to invoke the timescale separation result [39, Th. 3] and ensure the attractiveness of the set $\mathcal{M} := \{(\boldsymbol{\xi}, \boldsymbol{\eta}) \in \mathbb{R}^{(n+m)T} \times \mathbb{R}^{(N-1)CT} \mid \boldsymbol{\xi} \in \Xi^\star, \boldsymbol{\eta} = \boldsymbol{\eta}^{\text{eq}}(\boldsymbol{\xi})\}$ for system (15). The second step consists in showing that the trajectory $(\mathbf{x}^\star, \mathbf{u}^\star)$ obtained by projecting $\boldsymbol{\xi}^\star \in \Xi^\star$ satisfies the first-order necessary conditions for optimality in correspondence of the costate sequence $\boldsymbol{\lambda}^\star$.

V. Numerical Simulations

In this section, we numerically test the effectiveness of DOMINOPT on a scenario where $N = 10$ agents navigate in a 2D space $\mathcal{C} := [0, 100] \times [0, 100]$ discretized in $C = 625$ cells. The agents are modeled as $N = 10$ discrete-time double integrators with $x_{i,t} \in \mathbb{R}^4$, $u_{i,t} \in \mathbb{R}^2$, and a sampling time of $\Delta t = 0.05$. They communicate over an Erdős-Rényi graph with parameter $r = 0.5$. Their macroscopic goal is to follow a time-varying Gaussian distribution described by

$$p_t^{\star,c} = \frac{1}{\Gamma_t 2\pi\sqrt{|\Sigma|}} e^{\frac{1}{2}(\psi(c)-\nu_t)^\top \Sigma^{-1}(\psi(c)-\nu_t)},$$



(a) (b)

Fig. 1: Evolution over k of the (a) reduced gradient squared norm $\|\nabla J(\boldsymbol{\xi}, \mathbf{p}(\boldsymbol{\xi}))\|^2$ and (b) reduced cost $J(\boldsymbol{\xi}, \mathbf{p}(\boldsymbol{\xi}))$.

for all $t \in [1, T]$ and $c \in [1, C]$, where $\Gamma_t \in \mathbb{R}$ is a normalization coefficient, $\Sigma = 80 I_2$ is the covariance matrix, and the sequence $\{\nu_t\}_{t=0}^T$ is given by $\nu_{t+1} = \nu_t - \text{col}(20\cos(\theta_t) + 50, 20\sin(\theta_t) + 50)$, in which θ_t varies according to $\theta_t = \frac{3\pi}{4e^{0.11(0.5T-t)}+4} - \frac{\pi}{4}$. Concerning the cost function, we consider

$$\ell_{i,t}^{\text{mic}}(u_{i,t}) := \tfrac{1}{2} u_{i,t}^\top R u_{i,t},$$

$$\ell_t^{\text{mac}}(p_t^c) := q_t \sum_{c \in C} p_t^c \log\left(\frac{p_t^c}{p_t^{\star,c}}\right),$$

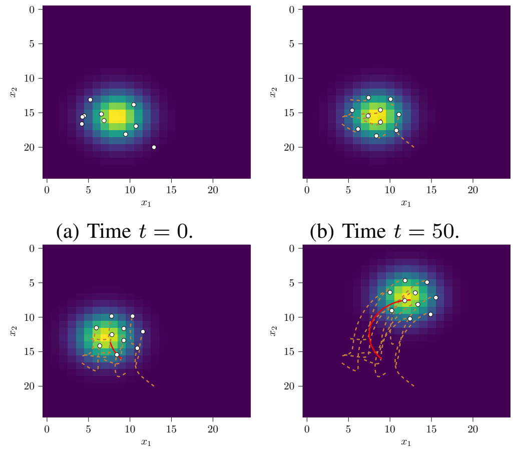
where $R = 10^{-3} I_2$ and $q_t = 10$, while $\ell_t^{\text{mac}}(p_t^c(x_t))$ is the Kullback-Leibler divergence between the macroscopic model and the target distribution, respectively. For each agent $i \in \mathcal{I}$, we select $p_{i,t}(x_{i,t})$ as in Example 1 with $\Sigma_i = 9$ and

$$\iota_i(x_{i,t}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_{i,t}.$$

Then, we design the local policies (9a) by solving an LQR problem problem involving the linearization of each agent dynamics about the current trajectory $(\mathbf{x}_i^k, \mathbf{u}_i^k)$. The cost matrices of the LQR are taken as $Q_{i,t}^\kappa = I$, $Q_{i,T}^\kappa = 10 I_2$, $R_{i,t}^\kappa = 10^{-3} I_2$. For all $i \in \mathcal{I}$ we generate $x_{i,\text{init}}$ by randomly exctracting it from $[\nu_0^0 - 20, \nu_0^1 - 20] \times [\nu_0^0 + 20, \nu_0^1 + 20] \times \{0\} \times \{0\}$ with uniform distribution. Then, for all $i \in \mathcal{I}$, we set $x_{i,t} = x_{i,\text{init}}$ for all $t \in [1, T]$, $\alpha_{i,t} = x_{i,\text{init}}$ for all $t \in [0, T-1]$ and $u_{i,t} = \mu_{i,t} = \text{col}(0, 0)$ for all $t \in [0, T-1]$. Lastly, we set $\gamma = 0.5$. Fig. 1 shows the evolution over k of the reduced gradient square norm $\|\nabla J(\boldsymbol{\xi}, \mathbf{p}(\boldsymbol{\xi}))\|^2$ and of reduced cost $J(\boldsymbol{\xi}, \mathbf{p}(\boldsymbol{\xi}))$ and confirms that DOMINOPT converges to a stationary point of the problem. Fig. 2 shows different snapshots of the agents' evolution achieved by DOMINOPT and the one of the mean ν_t of the target distribution.

VI. Conclusions

In this paper, we proposed DOMINOPT, a distributed method for multiscale optimal control of multi-agent systems. We considered macroscopic behaviors described by probability distributions over a finite time horizon and an optimal control problem with the cost function depending on both this macroscopic behavior and the individual control inputs of the agents. To address this problem, DOMINOPT combined a method based on feedback immersion to optimize the macroscopic behavior and a consensus-based one to concurrently learn it locally. We analyzed the algorithm with tools from system theory showing its convergence to the set of stationary points of the optimal control problem.



(a) Time $t = 0$. (b) Time $t = 50$. (c) Time $t = 80$. (d) Time $t = 180$.

Fig. 2: Evolution of the agents' position (white dots) and of the target distribution (in background). The agents' trajectories are represented by the brown dashed lines, while the red one represents the trajectory of the mean of the target distribution.

References

[1] M. Dorigo, G. Theraulaz, and V. Trianni, "Swarm robotics: Past, present, and future [point of view]," *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1152–1165, 2021.

[2] A. Testa, G. Carnevale, and G. Notarstefano, "A tutorial on distributed optimization for cooperative robotics: from setups and algorithms to toolboxes and research directions," *Proceedings of the IEEE*, 2025.

[3] N. Suhaimy, N. A. M. Radzi, W. S. H. M. W. Ahmad, K. H. M. Azmi, and M. Hannan, "Current and future communication solutions for smart grids: A review," *IEEE Access*, vol. 10, pp. 43 639–43 668, 2022.

[4] F. Chen, W. Ren *et al.*, "On the control of multi-agent systems: A survey," *Foundations and Trends® in Systems and Control*, vol. 6, no. 4, pp. 339–499, 2019.

[5] R. Hult, M. Zanon, S. Gros, H. Wymeersch, and P. Falcone, "Optimisation-based coordination of connected, automated vehicles at intersections," *Vehicle Systems Dynamics*, vol. 58, no. 5, pp. 726–747, 2020.

[6] R. M. D'Souza, M. di Bernardo, and Y.-Y. Liu, "Controlling complex networks with complex nodes," *Nature Reviews Physics*, vol. 5, no. 4, pp. 250–262, 2023.

[7] G. C. Maffettone, M. Porfiri, and M. Di Bernardo, "Continuification control of large-scale multiagent systems under limited sensing and structural perturbations," *IEEE Control Systems Letters*, vol. 7, pp. 2425–2430, 2023.

[8] K. Rudd, G. Foderaro, and S. Ferrari, "A generalized reduced gradient method for the optimal control of multiscale dynamical systems," in *52nd CDC*. IEEE, 2013, pp. 3857–3863.

[9] G. Foderaro, S. Ferrari, and T. A. Wettergren, "Distributed optimal control for multi-agent trajectory optimization," *Automatica*, vol. 50, no. 1, pp. 149–154, 2014.

[10] S. Ferrari, G. Foderaro, P. Zhu, and T. A. Wettergren, "Distributed optimal control of multiscale dynamical systems: a tutorial," *IEEE Control Systems Magazine*, vol. 36, no. 2, pp. 102–116, 2016.

[11] V. Krishnan and S. Martínez, "Distributed optimal transport for the deployment of swarms," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 4583–4588.

[12] V. Krishnan and S. Martinez, "Distributed control for spatial self-organization of multi-agent swarms," *SIAM Journal on Control and Optimization*, vol. 56, no. 5, pp. 3642–3667, 2018.

[13] T. Meurer and M. Krstic, "Finite-time multi-agent deployment: A nonlinear PDE motion planning approach," *Automatica*, vol. 47, no. 11, pp. 2534–2542, 2011.

[14] C. Sinigaglia, S. Bandyopadhyay, M. Quadrelli, and F. Braghin, "Optimal-transport-based control of particle swarms for orbiting rainbows concept," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 11, pp. 2108–2117, 2021.

[15] D. Ghimire and S. S. Kia, "Stein coverage: a variational inference approach to distribution-matching multisensor deployment," *IEEE Robotics and Automation Letters*, 2024.

[16] M. Todescato, A. Carron, R. Carli, G. Pillonetto, and L. Schenato, "Multi-robots gaussian estimation and coverage control: From client–server to peer-to-peer architectures," *Aut.*, vol. 80, pp. 284–294, 2017.

[17] G. Freudenthaler and T. Meurer, "PDE-based multi-agent formation control using flatness and backstepping: Analysis, design and robot experiments," *Automatica*, vol. 115, p. 108897, 2020.

[18] Y.-f. Chung and S. S. Kia, "A distributed service-matching coverage via heterogeneous agents," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4400–4407, 2022.

[19] G. C. Maffettone, A. Boldini, M. Porfiri, and M. di Bernardo, "Leader-follower density control of spatial dynamics in large-scale multi-agent systems," *IEEE Transactions on Automatic Control*, 2025.

[20] G. C. Maffettone, A. Boldini, M. di Bernardo, and M. Porfiri, "Density control of multi-agent swarms via bio-inspired leader-follower plasticity," *arXiv preprint arXiv:2507.15781*, 2025.

[21] B. Di Lorenzo, G. C. Maffettone, and M. di Bernardo, "Decentralized continuification control of multi-agent systems via distributed density estimation," *IEEE Control Systems Letters*, 2025.

[22] R. Van Parys and G. Pipeleers, "Distributed MPC for multi-vehicle systems moving in formation," *Robotics and Autonomous Systems*, vol. 97, pp. 144–152, 2017.

[23] A. D. Saravanos, A. Tsolovikos, E. Bakolas, and E. A. Theodorou, "Distributed covariance steering with consensus ADMM for stochastic multi-agent systems." in *Robotics: Science and Systems*, 2021.

[24] F. Rey, Z. Pan, A. Hauswirth, and J. Lygeros, "Fully decentralized ADMM for coordination and collision avoidance," in *2018 European Control Conference*, 2018, pp. 825–830.

[25] O. Shorinwa, T. Halsted, and M. Schwager, "Scalable distributed optimization with separable variables in multi-agent networks," in *American Control Conference*, 2020, pp. 3619–3626.

[26] T. Halsted, O. Shorinwa, J. Yu, and M. Schwager, "A survey of distributed optimization methods for multi-robot systems," *preprint arXiv:2103.12840*, 2021.

[27] O. So, K. Stachowicz, and E. A. Theodorou, "Multimodal maximum entropy dynamic games," *preprint arXiv:2201.12925*, 2022.

[28] M. A. Müller and F. Allgöwer, "Economic and distributed model predictive control: Recent developments in optimization-based control," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 2, pp. 39–52, 2017.

[29] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," *Aut.*, vol. 49, no. 3, pp. 829–833, 2013.

[30] B. Houska and J. Shi, "Distributed MPC with ALADIN — A tutorial," in *American Control Conference*, 2022, pp. 358–363.

[31] G. Stomberg, A. Engelmann, and T. Faulwasser, "Decentralized non-convex optimization via bi-level SQP and ADMM," in *IEEE Conference on Decision and Control*, 2022.

[32] G. Stomberg, H. Ebel, T. Faulwasser, and P. Eberhard, "Cooperative distributed MPC via decentralized real-time optimization: Implementation results for robot formations," *Control Engineering Practice*, vol. 138, p. 105579, 2023.

[33] L. Sforni, G. Carnevale, and G. Notarstefano, "A distributed feedback-based framework for nonlinear aggregative optimal control," *IEEE Transactions on Automatic Control*, 2024.

[34] J. Hauser, "A projection operator approach to the optimization of trajectory functionals," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 377–382, 2002.

[35] L. Sforni, S. Spedicato, I. Notarnicola, and G. Notarstefano, "Go-PRONTO: a feedback-based framework for nonlinear optimal control," *preprint arXiv:2108.13308*, 2021.

[36] Y.-C. Chen, "A tutorial on kernel density estimation and recent advances," *Biostatistics & Epid.*, vol. 1, no. 1, pp. 161–187, 2017.

[37] T. Zheng, Q. Han, and H. Lin, "PDE-based dynamic density estimation for large-scale agent systems," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 541–546, 2020.

[38] ——, "Distributed mean-field density estimation for large-scale systems," *IEEE TAC*, vol. 67, no. 10, pp. 5218–5229, 2021.

[39] G. Carnevale, N. Mimmo, and G. Notarstefano, "A unifying system theory framework for distributed optimization and games," *IEEE Transactions on Automatic Control*, 2025.