Multi-Agent Stochastic Control using Path Integral Policy Improvement

Peter Varnai and Dimos V. Dimarogonas[1]

***Abstract*— Path integral policy improvement (PI^2) is a data-driven method for solving stochastic optimal control problems. Both feedforward and feedback controls are calculated based on a sample of noisy open-loop trajectories of the system and their costs, which can be obtained in a highly parallelizable manner. The control strategy offers theoretical performance guarantees related to the expected cost achieved by the resulting closed-loop system. This paper extends the single-agent case to a multi-agent setting, where such theoretical guarantees have not been attained previously. We provide both a decentralized and a leader-follower scheme for distributing the feedback calculations under different communication constraints. The theoretical results are verified numerically through simulations.**

I. Introduction

As the deployment of robots in real-world scenarios becomes more and more widespread and technologically possible, the need for controlling systems where multiple agents must communicate and cooperate to solve complex problems effectively is increasing as well. Example applications are cooperative robots in industrial factories, unmanned aerial vehicles (UAVs) for surveillance and exploration, traffic control systems, and platooning. The size and complexity of these system leads to a state-space and thus computational complexity explosion when tackled with centralized, single-agent controllers, a problem known as the *curse of dimensionality*. This necessitates the development of multi-agent methods in order to distribute the workload by having the agents find (sub)optimal solutions using only locally available information under given communication constraints. The field has attracted much research, ranging from analytical [1] to deep reinforcement methods [2]. With growing levels of computational power and parallelization, the latter have become state-of-the-art for dealing with more complex systems in practice, but offer limited theoretical performance guarantees due to their learning aspect.

Policy improvement with path integrals (PI^2) [3] is a control method for solving stochastic optimal control problems in a simulation-driven manner while retaining theoretical guarantees related to the expected performance. The method relies on formulating a special form of feedback that allows control inputs to be calculated from the costs of open-loop trajectory samples (rollouts). Recent theoretical advances have both simplified this feedback calculation [4] and proposed to *apriori* find feedforward controls which optimize the expected closed-loop performance [5]. These improvements ease the implementation of the method and decrease the sampling effort required during its real-time implementation, improving its potential practical applicability. The computational reliance of PI^2 on our ability to generate and evaluate open-loop rollouts makes it of great interest, as this is highly parallelizable and thus the feasibility of the method can be expected to increase as technology advances.

While path integral control has enjoyed a broad range of research [6]–[8] and applications [9]–[13] in the single-agent case, there is very limited literature on its potential extension to multi-agent systems. Practical applications of PI^2 to multi-agent systems, such as UAV planning, essentially still work in a centralized manner [14]. In [15], graph inference techniques were used in the estimation of the joint system trajectory distribution. This improves the efficiency of the calculations, but still relies on global state information across the agents. In a more recent work [16], the single-agent theory was applied to an agent and its neighbors (*i.e.*, agents which contribute to its cost). Agents then compute optimal controls for their local group, extract their own input, and implement it in real-time. This loses theoretical guarantees and only works heuristically under the implicit assumption that the independently computed actions agree across the agents, *i.e.*, in a cooperative setting.

This paper extends PI^2 to a multi-agent setting where agents have independent dynamics and their costs depend on a set of neighboring agents. Our main contributions are:

- First, we show that in order for the feedbacks to be calculated via open-loop sampling, they must satisfy a certain linear equation. We give a decentralized scheme that allows the agents to agree on its solution by iteratively exchanging locally available information, as well as a leader-follower scheme that instead relies on a series of leader-to-follower communications. If the agent costs are non-conflicting and the system dynamics permit agents to correct for all costs simultaneously, the solution exists and theoretical performance guarantees are retained. Otherwise, the agents find a least-violating solution and achieve heuristically good performance.
- Second, we derive a (centralized) algorithm for finding the optimal feedforwards *apriori* by minimizing the sum of the expected agent costs.

An extended version of this paper with more detailed derivations, extended simulation results, and ideas for future work is provided separately in [17].

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, the Swedish Research Council (VR), the SSF COIN project, and the ERC LEAFHOUND project.

[1]Both authors are with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Brinellvgen 8, 114 28 Stockholm, Sweden.
Corresponding author: varnai@kth.se (P. Varnai)

The rest of this paper is structured as follows. Section II introduces functionals in order for the agents to handle path-dependent costs. Section III outlines the problem formulation, followed by the introduction of the multi-agent PI^2 control strategy in Section IV. Section V then discusses the feedback control calculations. The theory is verified using simulations in Section VI, and conclusions are given in Section VII.

II. Preliminaries

Let $T > 0$ denote the time horizon of an optimal control problem, and let Λ_t denote the set of all RCLL (right continuous, left limit) functions mapping each point $s \in [0, t]$ to $\mathbb{R}^p$ for any $t \in [0, T]$. The value of a trajectory (path) $\tau_t \in \Lambda_t$ of length t at time s is then denoted by the vector $\boldsymbol{\tau}_t(s) \in \mathbb{R}^p$. The set of all possible trajectories for all possible time intervals is given by $\Lambda := \bigcup_{t\in[0,T]} \Lambda_t$.

A *functional* $V : \Lambda \to \mathbb{R}$ assigns a real number to paths in the set Λ [18] and allows us to formulate optimal control problems involving path-dependent final costs. Its space and time derivatives are defined and operate similarly as those of traditional functions. To ease the definitions, let us introduce

$$\boldsymbol{\tau}_t^{\boldsymbol{h}}(s) = \begin{cases} \boldsymbol{\tau}_t(s), & s < t \\ \boldsymbol{\tau}_t(t) + \boldsymbol{h}, & s = t \end{cases} \quad \text{and} \quad \boldsymbol{\tau}_{t,\delta t}(s) := \begin{cases} \boldsymbol{\tau}_t(s), & s \le t \\ \boldsymbol{\tau}_t(t), & s \in (t, t+\delta t] \end{cases}$$

for a path $\tau_t \in \Lambda_t$, spacial shift $\boldsymbol{h} \in \mathbb{R}^p$, and temporal shift $\delta t > 0$. The (directional) space and time derivatives of the functional V can now be defined as $\Delta_{\boldsymbol{x}}^{\boldsymbol{h}} V(\tau_t) = \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left[V(\tau_t^{\epsilon\cdot\boldsymbol{h}}) - V(\tau_t)\right]$ and $\Delta_t V(\tau_t) = \lim_{\delta t\to 0^+} \frac{1}{\delta t}\left[V(\tau_{t,\delta t}) - V(\tau_t)\right]$. By choosing the direction $\boldsymbol{h}$ as the unit basis vectors $\boldsymbol{e}_i$ corresponding to the (i)-th dimensions of the space $\mathbb{R}^p$, the gradient of V can be formed as the vector $\boldsymbol{\Delta}_{\boldsymbol{x}} V(\tau_t) := \begin{bmatrix}\Delta_{\boldsymbol{x}}^{\boldsymbol{e}_1} V(\tau_t) & \dots & \Delta_{\boldsymbol{x}}^{\boldsymbol{e}_p} V(\tau_t)\end{bmatrix}^{\mathsf{T}}$. The Hessian is then given as the matrix $\boldsymbol{\Delta}_{\boldsymbol{xx}} V(\tau_t) := \begin{bmatrix}\boldsymbol{\Delta}_{\boldsymbol{x}}\left(\Delta_{\boldsymbol{x}}^{\boldsymbol{e}_1} V(\tau_t)\right) & \dots & \boldsymbol{\Delta}_{\boldsymbol{x}}\left(\Delta_{\boldsymbol{x}}^{\boldsymbol{e}_p} V(\tau_t)\right)\end{bmatrix}$.

III. Problem Formulation

Consider a group of $p = 1, \dots, P$ agents with independent, input-affine dynamics of the form

$$\dot{\boldsymbol{x}}_t^p = \boldsymbol{f}^p(\boldsymbol{x}_t^p, t) + \mathbf{g}^p(\boldsymbol{x}_t^p)\boldsymbol{u}_t^p + \left(\boldsymbol{\Sigma}_{x,t}^p\right)^{1/2} \boldsymbol{\epsilon}_{x,t}^p, \quad (1)$$

where $\boldsymbol{x}_t^p \in \mathbb{R}^{n_p}$ is the agent state, $\boldsymbol{u}_t^p \in \mathbb{R}^{m_p}$ is the agent input, $\boldsymbol{\epsilon}_{x,t}^p \in \mathbb{R}^{n_p}$ is zero-mean white noise included with covariance $\boldsymbol{\Sigma}_{x,t}^p \ge \mathbf{0}$, and the functions $\boldsymbol{f}^p(\cdot)$ and $\boldsymbol{g}^p(\cdot)$ describe the autonomous and input-dependent parts of the agent dynamics. Each agent will implement a control input composed of a feedforward and a feedback term as $\boldsymbol{u}_t^p := \boldsymbol{k}_t^p + \boldsymbol{\delta u}_t^p$, the former of which is generated as

$$\dot{\boldsymbol{k}}_t^p = \boldsymbol{\nu}_t^p + (\boldsymbol{\Sigma}_{\mathrm{k},t}^p)^{1/2} \boldsymbol{\epsilon}_{k,t}^p. \quad (2)$$

Here $\boldsymbol{\nu}_t^p \in \mathbb{R}^{m_p}$ is the nominal feedforward derivative and $\boldsymbol{\epsilon}_{k,t}^p \in \mathbb{R}^{m_p}$ is added white noise with covariance $\boldsymbol{\Sigma}_{\mathrm{k},t}^p \ge \mathbf{0}$. Together, the two equations can be combined into an extended state $\boldsymbol{z}_t^p \in \mathbb{R}^{n_p+m_p}$ to give the following state-space representation of agent p:

$$\dot{\boldsymbol{z}}_t^p = \begin{bmatrix}\dot{\boldsymbol{x}}_t^p \\ \dot{\boldsymbol{k}}_t^p\end{bmatrix} = \boldsymbol{F}_t^p(\boldsymbol{x}_t^p) + \mathbf{G}^p(\boldsymbol{x}_t^p)\boldsymbol{v}_t^p + (\boldsymbol{\Xi}_t^p)^{1/2}\boldsymbol{w}_t^p, \quad (3)$$

where the introduced quantities are defined as:

$$\boldsymbol{F}_t^p(\boldsymbol{x}_t^p) = \begin{bmatrix}\boldsymbol{f}^p(\boldsymbol{x}_t^p, t) \\ \mathbf{0}\end{bmatrix}, \quad \mathbf{G}^p(\boldsymbol{x}_t^p) = \begin{bmatrix}\mathbf{g}^p(\boldsymbol{x}_t^p) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}\end{bmatrix},$$
$$\boldsymbol{v}_t^p = \begin{bmatrix}\boldsymbol{u}_t^p \\ \boldsymbol{\nu}_t^p\end{bmatrix}, \quad \boldsymbol{\Xi}_t^p = \begin{bmatrix}\boldsymbol{\Sigma}_{x,t}^p & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{k,t}^p\end{bmatrix}, \quad \boldsymbol{w}_t^p = \begin{bmatrix}\boldsymbol{\epsilon}_{x,t}^p \\ \boldsymbol{\epsilon}_{k,t}^p\end{bmatrix}.$$

Note that $\boldsymbol{v}_t^p$ can be regarded as an abstracted input, and our goal will be to define a control policy for determining its value in time such that the entire closed-loop system achieves optimal performance.

To quantify the performance, we assume that each agent p aims to minimize a cost that depends on itself and a number N_p of neighboring agents. These are assembled as the set

$$\mathcal{N}^p := \{p, q_1^p, \dots, q_{N_p}^p\}, \quad (4)$$

where q_i^p denotes the index of the (i)-th neighbor of agent p. The joint states of these agents are defined as

$$\tilde{\boldsymbol{z}}_t^p := \begin{bmatrix}\left(\boldsymbol{z}_t^p\right)^{\mathsf{T}} & \left(\boldsymbol{z}_t^{q_1^p}\right)^{\mathsf{T}} & \dots & \left(\boldsymbol{z}_t^{q_{N_p}^p}\right)^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}, \quad (5)$$

and their values are assembled into trajectories of length t as $\tilde{\tau}_t^p := \{\tilde{\boldsymbol{z}}_s^p \mid 0 \le s \le t\}$. Considering a time horizon of length $T > 0$, each agent p is then assigned a cost

$$S_t^p(\tilde{\tau}_T^p) \equiv S^p(\tilde{\tau}_T^p, t) := \phi^p(\tilde{\tau}_T^p) + \int_t^T q^p(\tilde{\boldsymbol{z}}_s, s)\,\mathrm{d}s, \quad (6)$$

where $\phi^p : \Lambda_T \to \mathbb{R}$ is a path-dependent terminal cost and $q_t^p(\tilde{\boldsymbol{z}}_t) \equiv q^p(\tilde{\boldsymbol{z}}_t, t)$ is an instantaneous running cost. The expected value of this cost given the trajectories at time t then depends on the control strategy and is denoted by

$$V_{\tilde{\theta}_p}^p(\tilde{\tau}_t^p) = \mathbb{E}\left[S_t^p(\tilde{\tau}_T^p) \mid \tilde{\tau}_t^p\right], \quad (7)$$

where $\tilde{\theta}_p = \{\theta_q \mid q \in \mathcal{N}^p\}$ is assembled from a given set $\{\theta_p\}_{p=1}^P$ of parameters of each agent's control strategy.

Our goal is to follow the PI^2 control paradigm and design a feedforward/feedback law for each $\boldsymbol{v}_t^p$ in (3) of the form

$$\boldsymbol{v}_t^p \equiv \begin{bmatrix}\boldsymbol{u}_t^p \\ \boldsymbol{\nu}_t^p\end{bmatrix} := \begin{bmatrix}\boldsymbol{k}_t^p \\ \boldsymbol{\nu}_{0t}^p(\theta_p)\end{bmatrix} + \begin{bmatrix}\delta\boldsymbol{u}_t^p \\ \delta\boldsymbol{k}_t^p\end{bmatrix} := \boldsymbol{v}_{0t}^p(\theta_p) + \delta\boldsymbol{v}_t^p, \quad (8)$$

such that the expected closed-loop costs $V_{\tilde{\theta}_p}^p(\tilde{\tau}_t^p)$ for each agent can be approximated by sampling the neighboring agent dynamics in an open-loop manner. This will allow the feedbacks $\delta\boldsymbol{v}_t^p$ to also be calculated using such open-loop samples, enabling a computationally parallelizable real-time implementation. The joint parameters $\theta = \{\theta_1, \dots, \theta_P\}$ of the feedforwards $\boldsymbol{v}_{0t}^p(\theta_p)$ must be determined such that the expected joint cost

$$V_\theta(\tau_0^1, \dots, \tau_0^P) := \sum_{p=1}^P \alpha_p V_{\tilde{\theta}_p}^p(\tilde{\tau}_0^p) \quad (9)$$

is minimized for the system for given initial states $\{\tau_0^p\}_{p=1}^P$ and weighting coefficients $\alpha_p > 0$ such that $\sum \alpha_p = 1$.

IV. The multi-agent PI^2 control strategy

We derive a linear equation for the agent feedbacks $\delta\boldsymbol{v}_t^p$ in (8) whose satisfaction allows the expected closed-loop cost (7) of each agent p to be expressed using open-loop sampling. To this end, we first transform the partial differential equation (PDE) governing $V^p_{\tilde{\theta}_p}(\tilde{\tau}_t^p)$ in a way such that it can be linearized. The condition for this linearizability across *all* agents yields the sought-after linear equation for the feedbacks and enables the expression of each $V^p_{\tilde{\theta}_p}(\tilde{\tau}_t^p)$ using open-loop sampling via the Feynman-Kac theorem [18]. This allows us to outline a multi-agent PI^2 control strategy, whose details are discussed in the subsequently.

Since the agent dynamics are decoupled, we consider the derivations with respect to a given agent p without loss of generality, and drop the dependency of the value functional $V^p_{\tilde{\theta}_p}(\tilde{\tau}_t^p)$ on the feedforward parameterization $\tilde{\theta}_p$. Using (7), the PDE governing $V^p(\tilde{\tau}_t^p)$ can be derived from the dynamic programming equation

$$\begin{aligned} V^p(\tilde{\tau}_t^p) &= \mathbb{E}\left[S_t^p(\tilde{\tau}_T^p) \mid \tilde{\tau}_t^p \right] \\ &= \mathbb{E}\left[q_t^p \mathrm{d}t + \mathbb{E}\left[S_{t+\mathrm{d}t}^p(\tilde{\tau}_T^p) \mid \tilde{\tau}_{t+\mathrm{d}t}^p \right] \mid \tilde{\tau}_t^p \right] \\ &= q_t^p \mathrm{d}t + \mathbb{E}\left[V^p(\tilde{\tau}_{t+\mathrm{d}t}^p) \mid \tilde{\tau}_t^p \right] \end{aligned} \tag{10}$$

using the same methods as in previous work [3] to yield:

$$\begin{aligned} -\Delta_t V^p(\tilde{\tau}_t^p) = q_t^p + \sum_{q\in\mathcal{N}^p} \left(\boldsymbol{\Delta}_{\boldsymbol{z}^q} V^p(\tilde{\tau}_t^p)\right)^{\mathrm{T}} \left(\boldsymbol{F}_t^q(\boldsymbol{x}_t^q) + \mathbf{G}^q(\boldsymbol{x}_t^q)\boldsymbol{v}_t^q\right) \\ + \frac{1}{2}\sum_{q\in\mathcal{N}^p} \operatorname{tr}\left(\boldsymbol{\Delta}_{\boldsymbol{z}^q\boldsymbol{z}^q} V^p(\tilde{\tau}_t^p)\boldsymbol{\Xi}_t^q\right), \end{aligned} \tag{11}$$

with boundary condition $V^p(\tilde{\tau}_T^p) = S_T^p(\tilde{\tau}_T^p) = \phi^p(\tilde{\tau}_T^p)$. Compared to the single-agent case, the other agents affecting the cost appear through the summations over $q \in \mathcal{N}^p$ from (4). The complete derivation is provided in [17].

The PDE (11) will not be linear when substituting in a given control law for the inputs $\boldsymbol{v}_t^q$. The PI^2 approach aims to eliminate such nonlinearities in the PDE governing a logarithmic transformation of the value functional:

$$V^p(\tilde{\tau}_t^p) = -\lambda_p \log \Psi^p(\tilde{\tau}_t^p), \tag{12}$$

where $\lambda_p > 0$. The partial derivatives can be related as:

$$\Delta_t V^p(\tilde{\tau}_t^p) = -\lambda_p \frac{\Delta_t \Psi^p(\tilde{\tau}_t^p)}{\Psi^p(\tilde{\tau}_t^p)}, \quad \boldsymbol{\Delta}_{\boldsymbol{z}^q} V^p(\tilde{\tau}_t^p) = -\lambda_p \frac{\boldsymbol{\Delta}_{\boldsymbol{z}^q}\Psi^p(\tilde{\tau}_t^p)}{\Psi^p(\tilde{\tau}_t^p)} \tag{13a}$$

$$\boldsymbol{\Delta}_{\boldsymbol{z}^q\boldsymbol{z}^q} V^p(\tilde{\tau}_t^p) = \lambda_p \frac{\boldsymbol{\Delta}_{\boldsymbol{z}^q}\Psi^p(\tilde{\tau}_t^p)\left(\boldsymbol{\Delta}_{\boldsymbol{z}^q}\Psi^p(\tilde{\tau}_t^p)\right)^{\mathrm{T}}}{\Psi^p(\tilde{\tau}_t^p)^2} - \lambda_p \frac{\boldsymbol{\Delta}_{\boldsymbol{z}^q\boldsymbol{z}^q}\Psi^p(\tilde{\tau}_t^p)}{\Psi^p(\tilde{\tau}_t^p)}. \tag{13b}$$

To de-clutter the coming derivations, from this point we will simply write V^p for $V^p(\tilde{\tau}_t^p)$ and Ψ^p for $\Psi^p(\tilde{\tau}_t^p)$. Together with the feedforward/feedback structure (8) of the inputs, (12) transforms (11) into

$$\begin{aligned} &\lambda_p \frac{\Delta_t \Psi^p}{\Psi^p} = q_t^p + \frac{1}{2}\operatorname{tr}\Upsilon^p \\ &- \sum_{q\in\mathcal{N}^p} \lambda_p \frac{\left(\boldsymbol{\Delta}_{\boldsymbol{z}^q}\Psi^p\right)^{\mathrm{T}}}{\Psi^p}\left(\boldsymbol{F}^q(\boldsymbol{x}_t^q, t) + \mathbf{G}^q(\boldsymbol{x}_t^q)(\boldsymbol{v}_{0t}^q(\theta_q) + \delta\boldsymbol{v}_t^q)\right), \end{aligned} \tag{14}$$

where the term within the trace is given as $\Upsilon^p = \sum_{q\in\mathcal{N}^p} \lambda_p \left(\frac{\boldsymbol{\Delta}_{\boldsymbol{z}^q}\Psi^p (\boldsymbol{\Delta}_{\boldsymbol{z}^q}\Psi^p)^{\mathrm{T}}}{(\Psi^p)^2} - \frac{\boldsymbol{\Delta}_{\boldsymbol{z}^q\boldsymbol{z}^q}\Psi^p}{\Psi^p} \right) \boldsymbol{\Xi}_t^q$. We can achieve linearization by choosing the feedbacks $\delta\boldsymbol{v}_t^q$ for the agents $q \in \mathcal{N}^p$ affecting this cost such that they cancel out the quadratic noise component, *i.e.*, such that

$$\begin{aligned} &-\sum_{q\in\mathcal{N}^p} \lambda_p \frac{\left(\boldsymbol{\Delta}_{\boldsymbol{z}^q}\Psi^p\right)^{\mathrm{T}}}{\Psi^p} \mathbf{G}^q(\boldsymbol{x}_t^q)\delta\boldsymbol{v}_t^q \\ &\qquad = -\frac{1}{2}\sum_{q\in\mathcal{N}^p} \lambda_p \operatorname{tr}\left(\frac{\boldsymbol{\Delta}_{\boldsymbol{z}^q}\Psi^p \left(\boldsymbol{\Delta}_{\boldsymbol{z}^q}\Psi^p\right)^{\mathrm{T}}}{(\Psi^p)^2} \boldsymbol{\Xi}_t^q \right). \end{aligned} \tag{15}$$

Rearranging the above equation, we therefore require $\sum_{q\in\mathcal{N}^p} \left(-\lambda_p \frac{\boldsymbol{\Delta}_{\boldsymbol{z}^q}\Psi^p}{\Psi^p}\right)^{\mathrm{T}} \mathbf{G}^q(\boldsymbol{x}_t^q)\delta\boldsymbol{v}_t^q = -\frac{1}{2\lambda_p}\sum_{q\in\mathcal{N}^p} \left(-\lambda_p \frac{\boldsymbol{\Delta}_{\boldsymbol{z}^q}\Psi^p}{\Psi^p}\right)^{\mathrm{T}} \boldsymbol{\Xi}_t^q \left(-\lambda_p \frac{\boldsymbol{\Delta}_{\boldsymbol{z}^q}\Psi^p}{\Psi^p}\right)$, which can be rephrased from the gradients (13) as:

$$\begin{aligned} &\sum_{q\in\mathcal{N}^p} \left(\boldsymbol{\Delta}_{\boldsymbol{z}^q} V^p\right)^{\mathrm{T}} \mathbf{G}^q(\boldsymbol{x}_t^q)\delta\boldsymbol{v}_t^q \\ &\qquad = -\frac{1}{2\lambda_p}\sum_{q\in\mathcal{N}^p} \left(\boldsymbol{\Delta}_{\boldsymbol{z}^q} V^p\right)^{\mathrm{T}} \boldsymbol{\Xi}_t^q \boldsymbol{\Delta}_{\boldsymbol{z}^q} V^p. \end{aligned} \tag{16}$$

At this point, we introduce the shorthand notation

$$\boldsymbol{\delta}_t^{pq} := \boldsymbol{\Delta}_{\boldsymbol{z}^q} V^p(\tilde{\tau}_t^p). \tag{17}$$

As the cost $V^p(\tilde{\tau}_t^p)$ does not depend on $q \notin \mathcal{N}^p$, $\boldsymbol{\delta}_{pq} = \mathbf{0}$ if $q \notin \mathcal{N}^p$, and we can write (16) equivalently as:

$$\sum_{q=1}^{P} \left(\boldsymbol{\delta}_t^{pq}\right)^{\mathrm{T}} \mathbf{G}^q(\boldsymbol{x}_t^q)\delta\boldsymbol{v}_t^q = -\frac{1}{2\lambda_p}\sum_{q=1}^{P} \left(\boldsymbol{\delta}_t^{pq}\right)^{\mathrm{T}} \boldsymbol{\Xi}_t^q \boldsymbol{\delta}_t^{pq}. \tag{18}$$

Satisfying this equation will lead to the linearization of (14) for a single agent p. However, we need this cancellation to *jointly* occur in the PDEs of the transformed value functionals associated to *all* $p = 1, \ldots, P$ agents, as the feedbacks $\delta\boldsymbol{v}_t^q$ are the same for agent q throughout all the PDEs. This represents a system of P linear equations for the combined unknowns

$$\delta\boldsymbol{v}_t := \begin{bmatrix} (\delta\boldsymbol{v}_t^1)^{\mathrm{T}} & \ldots & (\delta\boldsymbol{v}_t^P)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{19}$$

which can be arranged into a linear matrix equation as

$$\mathbf{A}_t^{\mathrm{T}} \delta\boldsymbol{v}_t = -\boldsymbol{L}_t, \tag{20}$$

where the introduced quantities are

$$\mathbf{A}_t^{\mathrm{T}} = \begin{bmatrix} \left(\boldsymbol{\delta}_t^{11}\right)^{\mathrm{T}} \mathbf{G}^1(\boldsymbol{x}_t^1) & \ldots & \left(\boldsymbol{\delta}_t^{1P}\right)^{\mathrm{T}} \mathbf{G}^P(\boldsymbol{x}_t^P) \\ \vdots & \ddots & \vdots \\ \left(\boldsymbol{\delta}_t^{P1}\right)^{\mathrm{T}} \mathbf{G}^1(\boldsymbol{x}_t^1) & \ldots & \left(\boldsymbol{\delta}_t^{PP}\right)^{\mathrm{T}} \mathbf{G}^P(\boldsymbol{x}_t^P) \end{bmatrix} \tag{21a}$$

and

$$\boldsymbol{L}_t = \begin{bmatrix} \frac{1}{2\lambda_1}\sum_{q=1}^{P} \left(\boldsymbol{\delta}_t^{1q}\right)^{\mathrm{T}} \boldsymbol{\Xi}_t^q \boldsymbol{\delta}_t^{1q} \\ \vdots \\ \frac{1}{2\lambda_P}\sum_{q=1}^{P} \left(\boldsymbol{\delta}_t^{Pq}\right)^{\mathrm{T}} \boldsymbol{\Xi}_t^q \boldsymbol{\delta}_t^{Pq} \end{bmatrix}. \tag{21b}$$

Compared to the single-agent case, we now have a *system* of P linear equations that the feedbacks $\delta\boldsymbol{v}_t$ have to satisfy instead of a *single* equation. This was made possible by requiring the quadratic cancellations (15) independently for each expected cost $V^p(\tilde{\tau}_t^p)$, and relates to our goal for (9) to minimize the sum of expected agent costs instead of the expectation of their sums. The latter would also have led to a single equation and would not allow a decentralized solution.

Assuming the linear equation (20) is solvable, the cancellation (15) does indeed occur, and the transformed PDE (14) for all agents $p = 1, \dots, P$ finally simplifies to

$$-\partial_t \Psi^p = -\frac{\Psi^p}{\lambda_p} q_t^p - \sum_{q \in \mathcal{N}^p} \left(\boldsymbol{\Delta}_{\boldsymbol{z}^q} \Psi^p\right)^{\mathrm{T}} \left(\boldsymbol{F}_t^q(\boldsymbol{x}_t^q) + \mathbf{G}^q(\boldsymbol{x}_t^q)\boldsymbol{v}_{0t}^q(\theta_q)\right) - \frac{1}{2} \sum_{q \in \mathcal{N}^p} \operatorname{tr}\left(\boldsymbol{\Delta}_{\boldsymbol{z}^q \boldsymbol{z}^q} \Psi^p \boldsymbol{\Xi}_t^q\right) \quad (22)$$

with the boundary condition $\Psi^p(\tilde{\tau}_T^p) = \exp(-\frac{1}{\lambda_p}\phi^p(\tilde{\tau}_T^p))$. Matching this PDE with the Feynman-Kac theorem as in [5], one can see that the theorem is valid under the condition of sampling the system from continuations of the $\tilde{\tau}_t^p$ trajectories using the following open-loop dynamics of (3) for each neighboring agent $q \in \mathcal{N}^p$:

$$\dot{\boldsymbol{z}}_t^q = \boldsymbol{F}^q(\boldsymbol{x}_t^q, t) + \mathbf{G}^q(\boldsymbol{x}_t^q)\boldsymbol{v}_{0t}^q(\theta_q) + (\boldsymbol{\Xi}_t^q)^{1/2}\boldsymbol{w}_t^q. \quad (23)$$

Denoting this open-loop sampling method by OL, the theorem then states that each $\Psi^p(\tilde{\tau}_t^p)$ can be expressed as:

$$\Psi^p(\tilde{\tau}_t^p) = \mathbb{E}_{OL}\left[\exp\left(-\frac{1}{\lambda_p} S_t^p(\tilde{\tau}_T^p)\right) \mid \tilde{\tau}_t^p\right]. \quad (24)$$

From the definition (12), this implies that the expected closed-loop cost is

$$V_{\tilde{\theta}_p}^p(\tilde{\tau}_t^p) = -\lambda_p \log \mathbb{E}_{OL}\left[\exp\left(-\frac{1}{\lambda_p} S_t^p(\tilde{\tau}_T^p)\right) \;\middle|\; \tilde{\tau}_t^p, \tilde{\theta}_p\right], \quad (25)$$

where the dependency on the feedforward parameterization $\tilde{\theta}_p$ is indicated once again.

Notably, in this final result the sampling dynamics (23) are independent of the agent p whose value functional $V_{\tilde{\theta}_p}^p(\tilde{\tau}_t^p)$ another agent $q \in \mathcal{N}^p$ is helping to calculate using (25). Therefore, all agents $q = 1, \dots, P$ can simply sample their own dynamics using (23) and share the *same* results with all other agents who need it, *i.e.*, all agents p for which $q \in \mathcal{N}^p$. Alternatively, if communication is a bottleneck, an agent p only needs to know the feedforwards, states, and dynamics of its neighbors $q \in \mathcal{N}^p$ to perform the sampling itself. As we will see in the next section, determining $V_{\tilde{\theta}_p}^p(\tilde{\tau}_t^p)$ will enable the calculation of its gradients which appear in (21) and thereby allow us to assemble the linear equation (20) and solve it for the agent feedbacks $\delta\boldsymbol{v}_t$.

Based on the presented discussion, the two-stage multi-agent PI^2 control strategy can now be formulated as follows:

I. Determine the feedforward parameters θ_p of each $\boldsymbol{v}_{0t}^p(\theta_p)$ such that the joint expected cost (9) is minimized. This can be done by substituting the obtained agent costs (25) into (9) and optimizing for the parameters using natural gradient descent [19].

II. Assemble the elements of (20) and solve it using locally available information in order to implement the feedback controls $\delta\boldsymbol{v}_t^p$ during real-time operation.

In the next section, we discuss a decentralized and a leader-follower scheme to effectively tackle the second stage of this strategy, *i.e.*, the solution of (20). The details of the first stage are similar to [5] and are therefore omitted here due to space constraints and expanded upon separately in [17].

V. Feedback calculation schemes

In this section we discuss a decentralized and a leader-follower solution scheme to find the closed-loop feedbacks $\delta\boldsymbol{v}_t$ for the multi-agent system. To this end, we first express the unknown elements of the matrices and vectors involved in the linear equation (20), and then show how it can be solved under the communication assumptions of the two schemes.

A. Elements of the linear equation

The unknown elements of the $\mathbf{A}_t$ coefficient matrix and the $\boldsymbol{L}_t$ vector in (20) stem from the value functional gradients $\boldsymbol{\delta}_t^{pq} = \boldsymbol{\Delta}_{\boldsymbol{z}^q} V^p(\tilde{\tau}_t^p)$ for $p, q = 1, \dots, P$, as seen from (21). These gradients can be derived from the approximation (25) of $V^p(\tilde{\tau}_t^p)$ using N trajectory rollouts to yield[1]:

$$\boldsymbol{\delta}_t^{pq} = \sum_{i=1}^{N} w_{\mathrm{PI}^2}^{p(i)} \left(\boldsymbol{\Delta}_{\boldsymbol{z}_t^q} \phi^p(\tilde{\tau}_T^{p(i)}) - \lim_{\Delta t \to 0} \frac{\lambda_p}{\Delta t} (\boldsymbol{\Xi}_t^q)^{\dagger} \bar{\boldsymbol{w}}_k^{q(i)}\right), \quad (26)$$

where the PI^2 weights associated to each sample are

$$w_{\mathrm{PI}^2}^{p(i)} = \frac{\exp\left(-\frac{1}{\lambda_p} S_t^p(\tilde{\tau}_T^{p(i)})\right)}{\sum_{j=1}^{N} \exp\left(-\frac{1}{\lambda_p} S_t^p(\tilde{\tau}_T^{p(j)})\right)}. \quad (27)$$

In these expressions, $\dagger$ denotes the generalized matrix inverse, $\tilde{\tau}_T^{p(i)}$ is the (i)-th sampled joint trajectory assembled by agent p, and $\bar{\boldsymbol{w}}_k^{q(i)}$ is the (i)-th sampled noise for agent q with covariance $\boldsymbol{\Xi}_t^q \Delta t$. Note that for an agent p, $\boldsymbol{\delta}_t^{pq}$ can be calculated based on rollouts from the agents in $\mathcal{N}^p$ if $q \in \mathcal{N}^p$, while $\boldsymbol{\delta}_t^{pq} = 0$ otherwise.

B. Decentralized scheme

We first consider a decentralized communication scheme between the agents, as defined by the *undirected* communication graph $\mathcal{G} = (V, E)$ for vertex set $V = \{v_1, \dots, v_P\}$ and edge set $E = \{\{v_p, v_q\} \mid q \in \mathcal{N}^p \text{ or } p \in \mathcal{N}^q\}$ of *unordered* vertex pairs. This allows an agent p to communicate with agents q that affect its cost ($q \in \mathcal{N}^p$) and those whose cost it affects ($p \in \mathcal{N}^q$). We assume $\mathcal{G}$ is connected without loss of generality, and aim to solve (20) for the individual feedbacks $\delta\boldsymbol{v}_t^p$ under the communication constraints given by $\mathcal{G}$.

To begin, note that (20) is an underdetermined linear equation, because $\mathbf{A}_t^{\mathrm{T}} \in \mathbb{R}^{P \times (2m_1 + \dots + 2m_P)}$ is a wide matrix as it has at least $2P > P$ columns. In order to obtain a unique solution, we search for the one which satisfies (20) while minimizing its norm as measured by a block-diagonal penalty matrix

$$\mathbf{R}_{0t} = \operatorname{diag}\left(\mathbf{R}_{0t}^1, \mathbf{R}_{0t}^2, \dots, \mathbf{R}_{0t}^P\right), \quad (28)$$

[1]The formula presented herein is valid if $\boldsymbol{\Xi}_t^q$ does not depend on the state $\boldsymbol{x}_t^q$ of agent q; for the more general case and detailed derivations, see [17].

where $\mathbf{R}_{0t}^p = \begin{bmatrix} \mathbf{P}_{0t}^p & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{0t}^p \end{bmatrix}$ is constructed from user-defined positive definite matrices $\mathbf{P}_{0t}^p \in \mathbb{R}^{m_p}$ and $\mathbf{Q}_{0t}^p \in \mathbb{R}^{m_p}$ for each $p = 1, \dots, P$. This can be formulated as the optimization problem

$$\min_{\delta \boldsymbol{v}_t} \frac{1}{2} \delta \boldsymbol{v}_t^{\mathrm{T}} \mathbf{R}_{0t} \delta \boldsymbol{v}_t \quad \text{subject to} \quad \mathbf{A}_t^{\mathrm{T}} \delta \boldsymbol{v}_t = -\boldsymbol{L}_t. \tag{29}$$

Introducing the transformed feedbacks

$$\delta \hat{\boldsymbol{v}}_t := (\mathbf{R}_{0t})^{1/2} \delta \boldsymbol{v}_t, \tag{30}$$

this problem is seen to be equivalent as to finding the minimum-norm solution to the following linear equation:

$$\mathbf{A}_t^{\mathrm{T}} (\mathbf{R}_{0t})^{-1/2} \delta \hat{\boldsymbol{v}}_t = -\boldsymbol{L}_t. \tag{31}$$

The solution for $\delta \boldsymbol{v}_t$ can be calculated using the pseudo-inverse of the coefficient matrix in (31) and (30) to yield

$$\delta \boldsymbol{v}_t = -\mathbf{R}_{0t}^{-1} \mathbf{A}_t \left(\mathbf{A}_t^{\mathrm{T}} \mathbf{R}_{0t}^{-1} \mathbf{A}_t \right)^{-1} \boldsymbol{L}_t, \tag{32}$$

and our goal is to obtain this solution using only locally available information. To this end, let us define the transformed inputs for each agent p as

$$\delta \hat{\boldsymbol{v}}_t^p := (\mathbf{R}_{0t}^p)^{1/2} \delta \boldsymbol{v}_t^p. \tag{33}$$

and write the coefficient matrix $\mathbf{A}_t^{\mathrm{T}} (\mathbf{R}_{0t})^{-1/2}$ of (31) using the derived elements (21) as

$$\begin{bmatrix} \left(\boldsymbol{\delta}_t^{11}\right)^{\mathrm{T}} \mathbf{G}^1(\boldsymbol{x}_t^1)(\mathbf{R}_{0t}^1)^{1/2} & \dots & \left(\boldsymbol{\delta}_t^{1P}\right)^{\mathrm{T}} \mathbf{G}^P(\boldsymbol{x}_t^P)(\mathbf{R}_{0t}^P)^{1/2} \\ \vdots & \ddots & \vdots \\ \left(\boldsymbol{\delta}_t^{P1}\right)^{\mathrm{T}} \mathbf{G}^1(\boldsymbol{x}_t^1)(\mathbf{R}_{0t}^1)^{1/2} & \dots & \left(\boldsymbol{\delta}_t^{PP}\right)^{\mathrm{T}} \mathbf{G}^P(\boldsymbol{x}_t^P)(\mathbf{R}_{0t}^P)^{1/2} \end{bmatrix}$$

Now note that for a given agent p, the vector $\boldsymbol{\delta}_t^{pq} = \boldsymbol{\Delta}_{\boldsymbol{z}^q} V^p(\tilde{\tau}_t^p) = \mathbf{0}$ if its cost is not influenced by q, *i.e.*, if $q \notin \mathcal{N}^p$. Therefore, the nonzero entries in the (p)-th row of the coefficient matrix only contain the quantities $\boldsymbol{\delta}_t^{pq}$, $\mathbf{G}^q(\boldsymbol{x}_t^q)$, $\mathbf{R}_{0t}^q$, and $\boldsymbol{\Xi}_t^q$ for $q \in \mathcal{N}^p$. This is similarly the case for the (p)-th row of the right-hand side vector $\boldsymbol{L}_t$. Thus, the (p)-th row of (31) can be constructed by agent p using local information from agents $q \in \mathcal{N}^p$, which is permitted under the constraints of the communication graph $\mathcal{G}$.

Optimization methods for iteratively agreeing on the least-violating solution to a linear equation based on separate knowledge of the rows of the equation are well studied in the literature [20]. These allow a decentralized solution of (31) and thus finding $\delta \hat{\boldsymbol{v}}_t^p$ for each $p = 1, \dots, P$. In particular, we apply the results of [21] for our case and within the communication constraints defined by $\mathcal{G}$. The method also returns the least-norm solution if initialized at the origin. The final feedback controls can then be recovered individually by each agent from (33) as $\delta \boldsymbol{v}_t^p = (\mathbf{R}_{0t}^p)^{-1/2} \delta \hat{\boldsymbol{v}}_t^p$.

C. Leader-follower scheme

Let us now consider a leader-follower hierarchy between the agents in order to find a solution to (20) through a series of successive calculations. We assume that the indices $p = 1, \dots, P$ of the agents are ordered such that if $p < q$, then $q \notin \mathcal{N}^p$. For example, the cost for agent $q = 1$ only depends on itself; for $q = 2$ it can also depend on agent $p = 1$, and so on. The leaders of an agent p are then chosen as the set $\mathcal{N}^p$. The communication flow is described by the *directed* graph $\mathcal{G} = (V, E)$ for vertex set $V = \{v_1, \dots, v_P\}$ and edge set $E = \{(v_q, v_p) \mid q \in \mathcal{N}^p\}$ of *ordered* vertex pairs, allowing an agent p to receive information from all agents $q \in \mathcal{N}^p$.

The above imposed cost assumption implies that the terms $\boldsymbol{\delta}_t^{pq} = \mathbf{0}$ if $q > p$. The matrix $\mathbf{A}_t$ as defined in (21) therefore has a block lower triangular structure, which potentially allows the individual agent feedbacks $\delta \boldsymbol{v}_t^p$ in (20) to be solved for row by row. Agent p has to solve

$$\begin{aligned} &\left(\boldsymbol{\delta}_t^{pp}\right)^{\mathrm{T}} \mathbf{G}^p(\boldsymbol{x}_t^p) \delta \boldsymbol{v}_t^p \\ &= -\sum_{q \in \mathcal{N}^p} \frac{1}{2\lambda_p} \left(\boldsymbol{\delta}_t^{pq}\right)^{\mathrm{T}} \boldsymbol{\Xi}_t^q \boldsymbol{\delta}_t^{pq} - \sum_{\substack{q \in \mathcal{N}^p \\ q \neq p}} \left(\boldsymbol{\delta}_t^{pq}\right)^{\mathrm{T}} \mathbf{G}^q(\boldsymbol{x}_t^q) \delta \boldsymbol{v}_t^q, \end{aligned}$$

i.e., the the (p)-th row of the equation. Denoting the block in row p and column q of a matrix $\mathbf{M}$ by $[\mathbf{M}]_{pq}$ and the (p)-th row of a vector $\boldsymbol{v}$ by $[\boldsymbol{v}]_p$, this can be compactly written as:

$$[\mathbf{A}_t^{\mathrm{T}}]_{pp} \delta \boldsymbol{v}_t^p = -[\boldsymbol{L}_t]_p - \sum_{\substack{q \in \mathcal{N}^p \\ q \neq p}} [\mathbf{A}_t^{\mathrm{T}}]_{pq} \delta \boldsymbol{v}_t^q. \tag{34}$$

Similarly to the decentralized case, this equation is in general underdetermined and we instead solve it in a least-norm sense as measured by the penalty matrix $\mathbf{R}_{0t}^p$:

$$[\mathbf{A}_t^{\mathrm{T}}]_{pp} (\mathbf{R}_{0t}^p)^{-1/2} \delta \hat{\boldsymbol{v}}_t^p = -[\boldsymbol{L}_t]_p - \sum_{\substack{q \in \mathcal{N}^p \\ q \neq p}} [\mathbf{A}_t^{\mathrm{T}}]_{pq} \delta \boldsymbol{v}_t^q, \tag{35}$$

where $\delta \hat{\boldsymbol{v}}_t^p = (\mathbf{R}_{0t}^p)^{1/2} \delta \boldsymbol{v}_t^p$ is defined as in (33). If the agents solve these equations successively in the order $q = 1, 2, \dots$, then the right hand side is completely known for the coming agent p from the solutions $\delta \boldsymbol{v}_t^q$ previously calculated by its possible leaders. Solving (35) and inserting (33) allows the feedback $\delta \boldsymbol{v}_t^p$ to then be obtained directly as:

$$\begin{aligned} \delta \boldsymbol{v}_t^p = (\mathbf{R}_{0t}^p)^{-1} [\mathbf{A}_t]_{pp} \left([\mathbf{A}_t^{\mathrm{T}}]_{pp} (\mathbf{R}_{0t}^p)^{-1} [\mathbf{A}_t]_{pp} \right)^{-1} \\ \cdot \left(-[\boldsymbol{L}_t]_p - \sum_{\substack{q \in \mathcal{N}^p \\ q \neq p}} [\mathbf{A}_t^{\mathrm{T}}]_{pq} \delta \boldsymbol{v}_t^q \right). \end{aligned} \tag{36}$$

The found solution can then be sent to the following agents in order for them to calculate their feedbacks, and so on.

Note that even if (20) has a solution for the system as a whole, it is possible that by successively calculating and fixing the leaders' feedbacks in a least-norm manner using (36), a follower will encounter an unsolvable row equation (35). It will still solve it in a least-violating manner.

Compared to the decentralized scheme examined previously, the presented leader-follower scheme avoids the need for an iterative agreement on the feedbacks between the agents when solving (20), thereby reducing the required communication between them. In exchange, there is some conservatism introduced for the solvability, as well as the optimality of the solution, since the leaders are not helping the followers correct for the noise impacting their costs.

VI. Simulation study

We present a sample optimal control problem to numerically verify the correctness of our theoretical results in the context of the decentralized feedback calculation scheme.

Consider $P = 3$ single-integrator agents, each with state $\boldsymbol{x}_t^p \in \mathbb{R}^2$, feedforward $\boldsymbol{k}_t^p \in \mathbb{R}^2$, and added noise with covariance $\boldsymbol{\Sigma}_{x,t}^p = 0.004\mathbf{I}$. The control problem has a horizon $T = 10s$ and is simulated with $\Delta t = 0.02$.

Each agent is assigned a cost with equal weight $\alpha_p = 1/3$ depending on their target behaviors. Agent 1 aims to reach a goal position $\boldsymbol{g}_1 = [3.0 \quad 1.0]^\mathrm{T}$ at time $t_1 = T/2$, while minimizing its input effort according to the cost $S_t^1(\tau_T^1) := 2\left\|\boldsymbol{x}_{t_1}^1 - \boldsymbol{g}_1\right\|^2 + 0.2\int_t^T \left\|\boldsymbol{k}_s^1\right\|^2 \mathrm{d}s$. Agent 2 aims to maintain connectivity via a constant distance imposed between itself and the others, and achieve a triangular formation at $t_F = T$ while minimizing its energy with the cost definition $S_t^2(\tau_T^1, \tau_T^2, \tau_T^3) := 0.2\int_t^T \left\|\boldsymbol{k}_s^2\right\|^2 \mathrm{d}s + 5\left\|\boldsymbol{x}_{t_F}^2 - \boldsymbol{x}_{t_F}^3 - \Delta\boldsymbol{x}_{23}\right\|^2 + 5\left\|\boldsymbol{x}_{t_F}^2 - \boldsymbol{x}_{t_F}^1 - \Delta\boldsymbol{x}_{12}\right\|^2 + 8\int_t^T \left(\left\|\boldsymbol{x}_s^1 - \boldsymbol{x}_s^2\right\| - d\right)^2 + \left(\left\|\boldsymbol{x}_s^3 - \boldsymbol{x}_s^2\right\| - d\right)^2 \mathrm{d}s$, where $\Delta\boldsymbol{x}_{12} = [-1.0 \quad 0.0]^\mathrm{T}$, $\Delta\boldsymbol{x}_{23} = [-0.5 \quad -0.866]^\mathrm{T}$, and $d = 1$. Finally, agent 3 aims to reach a goal position $\boldsymbol{g}_3 = [1.0 \quad 3.5]^\mathrm{T}$ at time $t_F = T$, while minimizing its input effort using $S_t^3(\tau_T^3) := 2\left\|\boldsymbol{x}_{t_F}^3 - \boldsymbol{g}_3\right\|^2 + 0.2\int_t^T \left\|\boldsymbol{k}_s^3\right\|^2 \mathrm{d}s$.

The decentralized PI^2 control strategy is then implemented with $\lambda_1 = \lambda_2 = 0.1$, $\lambda_3 = 0.05$ using a different number of N roll-outs for feedback calculation. Table I shows the comparison with the theoretically expected values from (25), while Figure 1 shows sample closed-loop trajectories for the $N = 10000$ case. The results tend towards the prediction as $N \to \infty$. Further details about the scenario and additional simulation examples are given in [17].

TABLE I: Achieved average closed-loop costs for the simulation case study. The results are approximated from 100 sample runs and include 95% confidence intervals.

p	1	2	3
$\mathbb{E}_{OL}\left[S^p\right]$	0.67 ± 0.00	10.1 ± 0.06	0.53 ± 0.00
$\mathbb{E}_{CL}\left[S^p, N = 100\right]$	0.72 ± 0.02	2.17 ± 0.15	0.32 ± 0.02
$\mathbb{E}_{CL}\left[S^p, N = 10000\right]$	0.65 ± 0.02	1.75 ± 0.12	0.37 ± 0.04
$\mathbb{E}_{CL}\left[S^p, theoretical\right]$	0.62 ± 0.00	1.39 ± 0.01	0.32 ± 0.00

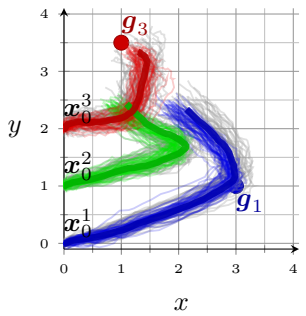

Fig. 1: Sample open-loop (gray) and closed-loop (colored) trajectories with the nominal, noiseless solution drawn thicker for the simulation case study. The robots aim to reach respective goal regions at different times while maintaining a constant distance and ending in a triangle formulation.

VII. Conclusions

We proposed the first multi-agent extension to PI^2 with theoretical performance guarantees. The resulting control strategy is simple to implement and can readily benefit from parallelization, giving potential for practical applications. Future work could decentralize the offline feedforward calculations and improve the sample efficiency of the method.

References

[1] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010, vol. 33.

[2] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3826–3839, 2020.

[3] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *Journal of Machine Learning Research (JMLR)*, vol. 11, pp. 3137–3181, 2010.

[4] P. Varnai and D. V. Dimarogonas, "The two-stage PI^2 control strategy," *IEEE Control Systems Letters*, vol. 6, pp. 2072–2077, 2022.

[5] ——, "Path integral policy improvement: an information-geometric approach," *Journal of Machine Learning Research (JMLR)*, submitted for publication, preprint available online on ResearchGate DOI: 10.13140/RG.2.2.13969.76645.

[6] F. Stulp and O. Sigaud, "Path integral policy improvement with covariance matrix adaptation," in *Proc. 29th International Conference on Machine Learning (ICML)*, 2012, pp. 1547–1554.

[7] S. Satoh, H. J. Kappen, and M. Saeki, "An iterative method for nonlinear stochastic optimal control based on path integrals," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 262–276, 2016.

[8] K. Yamamoto, R. Ariizumi, T. Hayakawa, and F. Matsuno, "Path integral policy improvement with population adaptation," *IEEE Transactions on Cybernetics*, pp. 1–11, 2020.

[9] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *IEEE Int. Conf. on Rob. and Aut. (ICRA)*, 2016, pp. 1433–1440.

[10] F. Ficuciello, D. Zaccara, and B. Siciliano, "Synergy-based policy improvement with path integrals for anthropomorphic hands," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1940–1945.

[11] Y. Chebotar, M. Kalakrishnan, A. Yahya, A. Li, S. Schaal, and S. Levine, "Path integral guided policy search," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2017, pp. 3381–3388.

[12] W. Zhu, X. Guo, Y. Fang, and X. Zhang, "A path-integral-based reinforcement learning algorithm for path following of an autoassembly mobile robot," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4487–4499, 2020.

[13] M. Hibbard, Y. Wasa, and T. Tanaka, "Path integral control for stochastic dynamic traffic routing problems," in *Proc. 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 261–267.

[14] V. Gómez, S. Thijssen, A. Symington, S. Hailes, and H. J. Kappen, "Real-time stochastic optimal control for multi-agent quadrotor systems," in *Twenty-Sixth International Conference on Automated Planning and Scheduling*, 2016.

[15] B. Van Den Broek, W. Wiegerinck, and B. Kappen, "Graphical model inference in optimal control of stochastic multi-agent systems," *Journal of Artificial Intelligence Research*, vol. 32, pp. 95–122, 2008.

[16] N. Wan, A. Gahlawat, N. Hovakimyan, E. A. Theodorou, and P. G. Voulgaris, "Cooperative path integral control for stochastic multi-agent systems," in *Proc. 2021 American Control Conference (ACC)*, 2021, pp. 1262–1267.

[17] P. Varnai and D. V. Dimarogonas, "Multi-agent stochastic control using path integral policy improvement," uploaded online on ResearchGate.

[18] B. Dupire, "Functional Itô calculus," *Quantitative Finance*, vol. 19, no. 5, pp. 721–729, 2019.

[19] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen, "Information-geometric optimization algorithms: A unifying picture via invariance principles," *The Journal of Machine Learning Research (JMLR)*, vol. 18, no. 1, pp. 564–628, 2017.

[20] P. Wang, S. Mou, J. Lian, and W. Ren, "Solving a system of linear equations: From centralized to distributed algorithms," *Annual Reviews in Control*, vol. 47, pp. 306–322, 2019.

[21] X. Wang, J. Zhou, S. Mou, and M. J. Corless, "A distributed linear equation solver for least square solutions," in *Proc. IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 5955–5960.