Proceedings of Machine Learning Research vol 242:1–12, 2024 6th Annual Conference on Learning for Dynamics and Control

Distributed On-the-Fly Control of Multi-Agent Systems With Unknown Dynamics: Using Limited Data to Obtain Near-optimal Control

Shayan Meshkat Alsadat SMESHKA1@ASU.EDU
Nasim Baharisangari NBAHARIS@ASU.EDU
Zhe Xu XZHE1@ASU.EDU
USA, Tempe, AZ 85281.

Editors: A. Abate, K. Margellos, A. Papachristodoulou

Abstract

We propose a method called ODMU for "on-the-fly control of distributed multi-agent systems with unknown nonlinear dynamics" and with (a)synchronous communication between the agents where data from a single finite-horizon trajectory is used, possibly in conjunction with side information. ODMU can be applied to real-time scenarios when the dynamics of the system are unknown or suddenly change such that a priori known model cannot be applied. In our proposed algorithm, the agents communicate their states using (a)synchronous communication and exploit the side information, e.g., regularities of the system, states, agents' communication scheme, algebraic limitations, and coupling in the system states. We provide ODMU for over-approximating the reachable sets and to control the agents under conditions with severely limited data. ODMU creates differential inclusion sets that calculate the over approximations of the reachable sets containing the unknown vector field. We show that ODMU calculates the near-optimal control and calculates an upper bound (suboptimality bound) for the error between the optimal trajectory and the trajectory calculated by ODMU. We use convex-optimization-based control to obtain the guaranteed near-optimal solution. We demonstrate the effect of side information on obtaining smaller bounds on suboptimality by applying ODMU on a system of unicycles. Additionally, we present a case study where a multi-agent system of unicycles with unknown dynamics is controlled via ODMU. Moreover, we have developed two baselines, SINDYcMulti and CGP-LCBMulti to compare our method with them.

Keywords: Nonlinear dynamical systems, multi-agent systems, optimization, reachability analysis, uncertain systems

1. Introduction

When sudden and abrupt changes are experienced by a system such that it alters its dynamics severely, a priori known model cannot be exploited; therefore, we need to learn the dynamics of the system on the fly Djeumou et al. (2021). These scenarios are even more involved for systems with multiple agents where task completion depends on all agents, not just one agent. In such cases, the system must exploit the information from its current trajectory to maintain a certain degree of control. We develop ODMU "on-the-fly control of distributed multi-agent systems with unknown nonlinear dynamics" to control multi-agent systems (MAS) with unknown dynamics and under conditions where data is severely *limited*, only data from trajectory is available. We use this framework as it enables robust control of the MAS when the underlying dynamics are unknown.

ODMU over approximates the reachable set of the system and allows for reachability analysis. ODMU is a one-step optimization algorithm that enables convex optimization to obtain a near-optimal solution to the control input for an on-the-fly control problem. Several methods could approximate the reachable sets, such as Hamilton-Jacobi-based Mitchell et al. (2005) or methods based on interval Taylor such as Berz and Makino (1998) Goubault and Putot (2019). These methods are used when the dynamics are known; however, in our scenario, the system's dynamics become unknown, and for such scenarios, there has been limited work done Djeumou et al. (2021) and even less for MAS, especially when the data is severely limited. `ODMU` uses side information, if available, such as regularities in the dynamics or bounds on vector fields to obtain a narrower reachable set. The obtained trajectory of the system for each agent contains the state of the agent, derivatives of the state of the agent, and the applied control input of the agent. **Contributions.** We have the following contributions to the field: (1) We propose a novel distributed data-driven control algorithm for on-the-fly control of MAS and implement two approaches with synchronous and asynchronous communications for the agents to coordinate their actions. (2) We provide a theoretical guarantee that `ODMU` obtains a near-optimal solution for MAS, and we compute an upper bound for this near-optimality of the proposed algorithm `ODMU`. (3) Our proposed algorithm is developed for MAS where the system's dynamics are unknown or suddenly become unknown, and it is capable of controlling the system under conditions where data is severely limited. We demonstrate the efficiency of `ODMU` by simulating a multi-agent system consisting of unicycles where the dynamics are unknown and data is severely limited also compare it with two baselines that we have developed called `SINDYcMulti` and `CGP-LCBMulti`. **Related Work.** Limited work is done for on-the-fly data-driven control of single-agent systems and even more limited for on-the-fly control of MAS Hou and Wang (2013) Djeumou et al. (2021). These methods combine system identification and model predictive control Korda and Mezić (2018)-Vinod et al. (2020). For instance, Korda and Mezić (2018) uses Koopman theory for linear system identification, and `SINDYc` Kaiser et al. (2018) employs sparse regression for nonlinear system identification in single-agent systems, requiring substantial data. We propose `SINDYcMulti`, a multi-agent baseline based on `SINDYc`. Researchers in Van Waarde et al. (2020); van Waarde et al. (2020); Markovsky and Dörfler (2022); Berberich et al. (2020, 2022) have proposed methods that work without system identification and mostly assume linear time-invariant dynamical systems; however, our `ODMU` works with general nonlinear unknown dynamics. Another method, `DMDc`, obtains a linear approximation using spectral properties but requires more data than `ODMU`. Myopic control Ornik et al. (2019) builds a local linear model with finite perturbations but cannot integrate side information. Contextual optimization-based methods like Gaussian processes (`CGP-LCB`) Krause and Ong (2011) face high computational costs. `C2Opt` Vinod et al. (2020) overcomes these costs but relies on the gradient of the one-step cost, and it could only use limited forms of side information.

2. Preliminaries

We denote the set of real numbers by $\mathbb{R}$, natural numbers by $\mathbb{N}$, n-dimensional vector x by $x \in \mathbb{R}^n$, matrix by $X \in \mathbb{R}^{n\times m}$ where $n, m \in \mathbb{R}$, $|\cdot|$ set cardinality, 1-norm by $\|x\|_1$, and 2-norm by $\|x\|_2$. We show the over-approximation of a set $A \subset \mathbb{R}^n$ by $\overline{A}$. An interval is denoted by $[a, b] = \{r \in \mathbb{R} | a \leq r \leq b\}$ where $a, b \in \mathbb{R}$ and $a \leq b$, set $\{i, \ldots, j\}$ is denoted by $\mathbb{N}_{[i,j]}$ where $i, j \in \mathbb{N}$ and $i \leq j$. k^{th} component of vector x and (k, j) component of matrix X are shown by $(x)_k$ and $(X)_{k,j}$, respectively. The Lipshitz constant of $f : \mathcal{X} \to \mathbb{R}$ by $L_f = \sup\{L \in \mathbb{R} \mid |f(x) - f(y)| \leq L\|x - y\|_2, x, y \in \mathcal{X}, x \neq y\}$ where $\mathcal{X} \subseteq \mathbb{R}^n$, and the Jacobian of function f by $\frac{\partial f}{\partial x}$. Function $f \in \mathscr{C}^\alpha(\mathcal{X})$ with $\alpha \geq 0$ is called $\mathscr{C}^\alpha$ continuous if the function is continuous on $\mathcal{X} \subseteq \mathbb{R}^n$ and all the partial derivatives up to order $1, \ldots, \alpha$ do exist and are continuous on the domain $\mathcal{X}$, f is $piecewise - \mathscr{C}^\alpha$ if there is a set A such that $A \subseteq \mathcal{X}$ and f is $\mathscr{C}^\alpha$ on A Djeumou et al. (2021).

2.1. Interval Analysis

We show sets of intervals by $\mathbb{IR} = \{\mathcal{A} = [\underline{\mathcal{A}}, \overline{\mathcal{A}}] \,|\underline{\mathcal{A}}, \overline{\mathcal{A}} \in \mathbb{R}, \underline{\mathcal{A}} \leq \overline{\mathcal{A}}\}$, set of n-dimensional interval vectors by $\mathbb{IR}^n$, $n \times m$-dimensional matrices by $\mathbb{IR}^{n\times m}$, the absolute value of an interval $\mathcal{A}$ by $|A| = \max\{|\overline{\mathcal{A}}|, |\underline{\mathcal{A}}|\}$. We apply the arithmetic operations to set inclusion and intersections of intervals to interval vectors and matrices component-wise. We denote the infinity norm of $A \in \mathbb{IR}^n$ by $|A|_\infty = \sup_{i\in\mathbb{N}_{[1,n]}}|(A)_i|$, Cartesian product as $A \otimes B = \left[[\underline{\mathcal{A}}, \overline{\mathcal{A}}], [\underline{\mathcal{B}}, \overline{\mathcal{B}}]\right] \in \mathbb{IR}^2$ for any $\mathcal{A}, \mathcal{B} \in \mathbb{IR}$, and the Cartesian product of any interval $\mathcal{A} \in \mathbb{IR}$ with itself n times is shown by $\mathcal{A}^n$. The term "interval" refers to either an interval vector or interval matrix, depending on the context. Interval-valued functions are essential in interval analysis; therefore, we introduce an interval-valued function as follows.

Given $f : \mathcal{X} \to \mathcal{Y}$ with $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{Y} \subseteq \mathbb{R}^m$ (or $\mathcal{Y} \subseteq \mathbb{R}^{n\times m}$), then we can show the interval-valued function $\boldsymbol{f} : \mathbb{IR}^n \to \mathbb{R}^n$ is an interval extension of function f if

$$\boldsymbol{f}(\mathcal{A}) \supseteq \mathscr{R}(f, \mathcal{A}) = \{f(x)|x \in \mathcal{A}\} \; \forall \mathcal{A} \subseteq \mathcal{X} \tag{1}$$

Hence, given an interval $\mathcal{A}$, $\boldsymbol{f}(\mathcal{A})$ is an interval-valued function that over-approximates the range ($\mathscr{R}$) of values given to f over the set $\mathcal{A}$. As per the general notation of interval-valued functions, we demonstrate interval-valued vector functions with bold lowercase and interval-valued matrix functions with bold uppercase symbols.

2.2. Multi-agent System

The set of all agents is $\mathcal{N} = \{1, \ldots, N\}$ and neighbor's for agent k is denoted by $\mathcal{N}_k$ where $k \leq N \in \mathbb{N}$. Each agent has its own dynamics and we model the communication structure of the multi-agent system using an *undirected graph* defined as follows.

Definition 1 *We show an udirected graph by $G = (\mathcal{C}, \mathcal{E})$, where $\mathcal{C} = \{c_1, c_2, ..., c_{n_C}\}$ is a finite set of nodes, $\mathcal{E} \subseteq \mathcal{E}' = \{e_{1,2}, e_{1,3}, ..., e_{1,n_\mathcal{E}}, e_{2,3}, ..., e_{n_{\mathcal{E}-1},n_\mathcal{E}}\}$ is a finite set of edges where $e_{\nu,\varsigma} \in \mathcal{E}$ if nodes c_ν and c_ς are connected by an edge in the graph G, and $n_\mathcal{C}, n_\mathcal{E} \in \mathbb{N}$.*

Each node c_i of the undirected graph G represents an agent in the system. Each edge $e_{\nu,\varsigma}$ connecting the nodes ν and ς represents the fact that agents ν and ς are neighbors, i.e., agents ν and ς can communicate with each other. For quantities with two superscripts such as $x^{k,\kappa}$, the first superscript k refers to the agent that is communicating, and the second superscript κ refers to the agent k's estimation of the other agent (agent κ) for that quantity.

2.3. Reachable Set In Multi-agent Systems

Let us consider a nonlinear dynamical system for each of the agents as

$$\dot{x}^k = \mathcal{T}^k(x^k, u^k) \tag{2}$$

where the state $x^k : \mathbb{R}_+ \mapsto \mathcal{X}$ of agent k is a continuous-time signal evolving in $\mathcal{X} \in \mathbb{IR}^n$, control input $u^k \in \mathbb{U}$ of agent k is evolving in the control set $\mathcal{U} \in \mathbb{IR}^m$ with $\mathbb{U} = \{v^k : \mathbb{R}_+ \mapsto \mathcal{U} \mid v^k \text{ is } piecewise - \mathscr{C}^{D_u}\}$ for $D_u \geq 0$. The dynamics of agent k is $\mathcal{T}^k : \mathcal{X} \times \mathcal{U} \mapsto \mathcal{Y}$ which is $\mathscr{C}^{D_\mathcal{T}}$ for $D_\mathcal{T} \geq 1$ and $\mathcal{Y} \in \mathbb{IR}^n$. Evolution of the states of the agent k considering the initial state $x_i^k = x^k(t_i)$ at time $t_i \in \mathbb{R}_{\geq 0}$ where $i \in \mathbb{Z}_{\geq 0}$ and control input signal $u^k \in \mathbb{U}$ will create a sequence of states that will be a part of the trajectory of the system. A trajectory of (2) is a function of time $x^k(\cdot, x_i^k, u^k) : [t_i, \infty) \mapsto \mathcal{X}$ and it satisfies Equation (2).

Definition 2 (Reachable set) *Consider a set $\mathcal{Q}_i \subseteq \mathcal{X}$ at time t_i and a set of controls, $\mathbb{V} \subseteq \mathbb{U}$, then the reachable set of dynamics as shown in Equation (2) at times $t \geq t_i$ is obtained by*

$$R(t, \mathcal{Q}_i, \mathbb{V}) = \{z \in \mathcal{X} \mid \exists x_i \in \mathcal{Q}_i, \exists \Lambda \in \mathbb{V}, z = x(t; x_i, \Lambda)\} \tag{3}$$

Considering a set $\mathbb{V} \subseteq \mathbb{U}$ and an initial set $\mathcal{Q}_0^k \subseteq \mathcal{X}$ of states at time t_0 for agent k, we can calculate the over-approximations of $R^k(t, \mathcal{Q}_0^k, \mathbb{V})$ at time $t \geq t_0$ by exploiting interval Taylor-based methods. We consider a discrete time set $t_0 < \ldots < t_\mu$ ($\mu \in \mathbb{N}$) such that for all $\Lambda^k \in \mathbb{V}$, Λ^k is $\mathscr{C}^{D_u}$ for each of the interval $[t_i, t_{i+1})$. To calculate $R_i^{k,+} \in \mathbb{IR}^n$ sets considering the initial set $R_0^{k,+} = \mathcal{Q}_0^k$ such that for all $i \in \mathbb{Z}_{[0,\mu-1]}$ we have $R_{i+1}^k = R^k(t_{i+1}, R_i^{k,+}, \mathbb{V}) \subseteq R_{i+1}^{k,+}$. We define agent k's interval Taylor expansion $\mathcal{T}^k$ by denoting the d^{th} derivative as $\mathcal{T}^{k,[d]}$ ($\mathcal{T}^{k,[1]} = \mathcal{T}^k$) as:

$$\mathcal{T}^{k,[d+1]} = \frac{1}{1+d}\left(\frac{\partial \mathcal{T}^{k,[d]}}{\partial x}\mathcal{T}^k + \sum_{l=0}^{d-1} \frac{\partial \mathcal{T}^{k,[d]}}{\partial u^{(l)}} u^{(l+1)}\right) \tag{4}$$

Substituting the initial set with $R_i^{k,+} = \mathcal{Q}_0^k$ results in the reachable set of the next time step as

$$R_{i+1}^{k,+} = R_i^{k,+} + \sum_{d=1}^{D-1} (t_{i+1} - t_i)^d \left(\mathcal{T}^{k,[d]}(R_i^{k,+}, \mathbf{\Lambda}^k)\right)(t_i) + (t_{i+1} - t_i)^D \left(\mathcal{T}^{k,[D]}(\Gamma_i^k, \mathbf{\Lambda}^k)\right)([t_i, t_{i+1}]) \tag{5}$$

where D must be $D \leq \min(D_u + 1, D_{\mathcal{T}})$, $\mathbf{\Lambda}^{k,(0)}(\mathcal{A})$ is $\mathbf{\Lambda}^k(\mathcal{A})$, and the intervals $\mathbf{\Lambda}^{k,(d)}(\mathcal{A})$ for all $d \in \mathbb{N}_{[0,D_u]}$ are such that $\bigcup_{\Lambda^k \in \mathbb{V}} \mathscr{R}(\Lambda^{k,(d)}, \mathcal{A}) \subseteq \mathbf{\Lambda}^{k,(d)}(\mathcal{A})$ with $\mathcal{A} \subseteq R^+$ or $\mathbf{\Lambda}^{k,(d)}(\mathcal{A})$ over-approximates the range of the d^{th} derivative of all $\Lambda^k \in \mathbb{V}$ on the interval $\mathcal{A}$. We calculate agent k's *a priori rough enclosure* set $\Gamma_i^k \subseteq \mathcal{X}$ of $R^k(t, R_i^{k,+}, \mathbb{V})$ for all $t \in [t_i, t_{i+1}]$ recursively by solving:

$$R_i^{k,+} + [0, t_{i+1} - t_i]\mathcal{T}^k(\Gamma_i^k, \mathbf{\Lambda}^k([t_i, t_{i+1}])) \subseteq \Gamma_i^k. \tag{6}$$

2.4. Constrained Receding-horizon Control For Multi-agent Systems

We define the cost function for agent k as C^k to solve the optimization problem as *one-step cost function* $C : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \to \mathbb{R}$. We assign a constant time step $\Delta t \geq 0$ where the next state would be $x_{i+1}^k = x^k(t_{i+1}^k, x_i^k, u_i^k)$ where $t_{i+1} = t_i + \Delta t$ (subscript i is a short-hand for t_i), agent's initial state is $x_i^k \in \mathcal{X}$, and the control input signal $u_i^k \in \mathcal{U}$ is constant between two time steps. The control input at time t_{i+1} for the initial state x_i^k of agent k is the solution of

$$\begin{aligned} &\text{minimize} \quad \sum_{t_i=i}^{i+H-1} C_i^k(x_i^1, u_i^1, x_{i+1}^1, x_i^2, u_i^2, x_{i+1}^2, \ldots, x_i^N, u_i^N, x_{i+1}^N) \\ &\text{subject to} \quad u_i^1, \ldots, u_i^N \in \mathcal{U};\ u_{i+H-1}^1, \ldots, u_{i+H-1}^N \in \mathcal{U};\ \dot{x}_i^k = \mathcal{T}^k(x_i^k, u_i^k),\ i, k \in \mathbb{N} \end{aligned} \tag{7}$$

where the planning horizon is $H \geq 1$. The optimization problem (7) is not a convex optimization problem due to the possible nonlinearities of the system dynamics (2) Kocijan et al. (2004).

3. Problem Statement

We consider a *control-affine* nonlinear dynamics for each agent k as

$$\dot{x}^k = f^k(x^k) + G^k(x^k)u^k, \tag{8}$$

where the *unknown* vector-valued function is $f^k : \mathcal{X} \mapsto \mathbb{R}^n$ and the matrix-valued function is $G^k : \mathcal{X} \mapsto \mathbb{R}^{n \times m}$ for $\mathcal{X} \in \mathbb{IR}^n$.

Assumption 1 (Lipschitz system) *f^k and G^k are locally Lipshitz-continuous functions on $x^k \in \mathbb{R}$ and their components have a finite Lipshitz constant on all the subsets of $\mathbb{R}^n$.*

f^k and G^k for each agent k are globally Lipshitz-continuous on $\mathcal{X}$ due to the bounded domain, $\mathcal{X} \in \mathbb{IR}^n$. We know the upper bounds on the Lipshitz constants $L_f^k \in \mathbb{R}_+^n$ and $L_G^k \in \mathbb{R}_+^{n \times m}$. L_f^k stand for the Lipshitz constant of the function f^k for agent k, this description can be extended to the Lipshitz constant of the function G^k. The known upper bounds on the Lipshitz constants of $\left(f^k\right)_q$ and $\left(G^k\right)_{q,l}$ are $(L_f^k)_q = L_{f_q}^k$ and $(L_G^k)_{q,l} = L_{G_{q,l}}^k$ for all $q \in \mathbb{N}_{[1,n]}$ and $l \in \mathbb{N}_{[1,m]}$ Chakrabarty et al. (2019); Calliess (2017); Zabinsky et al. (2003).

Lemma 3 (f^k and G^k over-approximations using a sample trajectory $\mathscr{I}_\mu^k$) *We have a set* $\epsilon_\mu^k = \{(x^k, \eta_{\mathcal{F}_i}^k, \eta_{\mathcal{G}_i}^k) | f^k(x_i) \in \eta_{\mathcal{F}_i}^k, G(x_i^k) \in \eta_{\mathcal{G}_i}^k\}_{i=0}^{\mu}$ *obtained from trajectory* $\mathscr{I}_\mu^k$. *Using Lipschitz bounds* L_f^k *and* L_G^k *we can calculate the intervals* $\boldsymbol{f} : \mathcal{X} \to \mathbb{IR}^n$ *and* $\boldsymbol{G} : \mathcal{X} \to \mathbb{IR}^{n\times m}$ *for all the components* $q \in \mathbb{N}_{[1,n]}$ *and* $l \in \mathbb{N}_{[1,m]}$ *as following (proof can be found in Djeumou et al. (2022))*

$$(\boldsymbol{f}^k(x^k))_q = \{\bigcap_{(x_i^k, \eta_{\mathcal{F}_i}^k, \cdot)\in\epsilon_\mu^k} (\eta_{\mathcal{F}_i}^k)_q + L_{f_q}^k \|x^k - x_i^k\|_2[-1, 1] \mid f^k(x^k) \in \boldsymbol{f}^k(x^k)\ \forall x^k \in \mathcal{X}\} \tag{9}$$

$$(\boldsymbol{G}^k(x^k))_{q,l} = \{\bigcap_{(x_i^k, \eta_{\mathcal{G}_i}^k, \cdot)\in\epsilon_\mu^k} (\eta_{\mathcal{G}_i}^k)_{q,l} + L_{G_{q,l}}^k \|x^k - x_i^k\|_2[-1, 1] \mid G^k(x^k) \in \boldsymbol{G}^k(x^k)\ \forall x^k \in \mathcal{X}\} \tag{10}$$

where the intervals $\eta_{\mathcal{F}_i}^k$ and $\eta_{\mathcal{G}_i}^k$ are the tightest intervals that contain $f^k(x_i^k)$ and $G^k(x_i^k)$, respectively. A priori rough enclosure Γ_i^k for agent k can be calculated using (6),

$$R_i^{k,+} + [0, \Delta t](\boldsymbol{f}^k(\Gamma_i^k) + \boldsymbol{G}^k(\Gamma_i^k)\mathcal{U}) \subseteq \Gamma_i^k \tag{11}$$

We use Lemma 3 for each agent k to obtain the intervals for the dynamics of the control-affine system (8), f^k and G^k, then we apply the Theorem 4 to obtain the reachable set $R_{i+1}^{k,+}$.

Theorem 4 (Reachable set over-approximation of an agent) *For agent k considering Lipschitz bounds* L_f^k *and* L_G^k, *piecewise* $-\mathscr{C}^{D_u}$ *control signals with continuity condition* $D_u \geq 1$ *on a set* $\mathbb{V} \subseteq \mathbb{U}$, *a trajectory* $\mathscr{I}_\mu^k$, *and time step* Δt, *then the reachable set* R_i^k *at time* t_{i+1} *can be calculated for the dynamics defined by differential inclusion* $\dot{x}_{t_i}^k \in \boldsymbol{f}^k(x) + \boldsymbol{G}^k(x)$ *as*

$$\begin{aligned} R_{i+1}^{k,+} =& R_i^{k,+} + (\boldsymbol{f}^k(R_i^{k,+}) + \boldsymbol{G}^k(R_i^{k,+})\boldsymbol{\Lambda}^k(t_i))\Delta t + \\ & (J_f^k + J_G^k\boldsymbol{\Lambda}_i^k)(\boldsymbol{f}^k(\Gamma_i^k) + \boldsymbol{G}^k(\Gamma_i^k)\Phi_i^k)\frac{\Delta t^2}{2} + \boldsymbol{G}^k(\Gamma_i^k)\Phi_i^{k,(1)}\frac{\Delta t^2}{2} \end{aligned} \tag{12}$$

where the control input interval is $\boldsymbol{\Lambda}^k(t_i)$, $\Phi_i^k = \boldsymbol{\Lambda}^k([t_i, t_i + \Delta t])$, and control input derivative is $\Phi_i^{(1)} = \boldsymbol{\Lambda}^{k,(1)}([t_i, t_i + \Delta t])$ for all the control signals in $\mathbb{V}$. Theorem 4 can be proved using Taylor expansion when $D_u = 2$ (second-order Taylor expansion). Equation (12) when the control input $u_i^k \in \mathcal{U}$ is constant in the time interval of $[t_i, t_{i+1})$ becomes

$$R_{i+1}^k \subseteq (\mathcal{B}_i^k + \mathcal{A}_i^{k,+}u_i^k) \cap (\mathcal{B}_i^k + \mathcal{A}_i^{k,-}u_i^k) \tag{13}$$

such that intervals $\mathcal{B}_i^k$, $\mathcal{A}_i^{k,-}$, and $\mathcal{A}_i^{k,+}$ are calculated as follows

$$\mathcal{B}_i^k = R_i^{k,+} + \boldsymbol{f}^k(R_i^{k,+})\Delta t + J_f^k\boldsymbol{f}^k(\Gamma_i^k)\frac{\Delta t^2}{2} \tag{14}$$

$$\mathcal{A}_i^{k,-} = \boldsymbol{G}^k(R_i^{k,+})\Delta t + (J_G^k\boldsymbol{G}^k(\Gamma_i^k) + J_G^{k^T}(\boldsymbol{f}^k(\Gamma_i^k) + \boldsymbol{G}^k(\Gamma_i^k)\mathcal{U}))\frac{\Delta t^2}{2} \tag{15}$$

$$\mathcal{A}_i^{k,+} = \boldsymbol{G}^k(R_i^{k,+})\Delta t + ((J_f^k + J_G^k\mathcal{U})\boldsymbol{G}^k(\Gamma_i^k) + J_G^{k^T}(\boldsymbol{f}^k(\Gamma_i^k)))\frac{\Delta t^2}{2} \tag{16}$$

Each agent k uses Equations (14)-(16) to obtain the intervals of other agents. We denote the concatenation of these intervals by agent k as $\zeta_i^{k,+} = \left[\mathcal{A}_i^{1,+}, \ldots, \mathcal{A}_i^{N,+}\right]^T$ and $\sigma_i^k = \left[\mathcal{B}_i^1, \ldots, \mathcal{B}_i^N\right]^T$.

Example 1 *We study a unicycle system with* $x_i^k = \begin{bmatrix}\rho_x & \rho_y & \phi\end{bmatrix}$, *representing position in the x and y planes and angular orientation, respectively. Control input* $u^k = \begin{bmatrix}\varkappa & \Im\end{bmatrix}$ *adjusts the unicycle's speed and heading, control bounds* $\mathcal{U} = [-3, 3] \times [-\pi, \pi]$, $f^k = 0$, G^k *with* $\left(G^k\right)_{1,1} = \cos\phi$, $\left(G^k\right)_{2,1} = \sin\phi$, $\left(G^k\right)_{3,2} = 1$, *and other entries* 0, *Lipschitz bounds of* $L_f^k = [0.01, 0.01, 0.01]$, *and* $L_{G_{1,1}}^k = L_{G_{2,1}}^k = 1.1$, $L_{G_{3,2}}^k = 1$, *and other entries* 0. *We utilize Example 1 with a* $\Delta t = 0.1[s]$ *time step as a motivating illustration.*

4. Control Synthesis For Multi-agent Systems

The multi-agent system needs to collaborate, i.e., agents are collaborating to carry out a system-level task in a cooperative setting to synthesize the control input vector $U_i = \left[u_{i+1}^1, \ldots, u_{i+1}^N\right]$ at time step t_i by solving the following optimization problem (17).

$$\begin{aligned} &\text{minimize} \quad \sum_{k=1}^{N} C_i^k(x_i^1, u_i^1, x_{i+1}^1, x_i^2, u_i^2, x_{i+1}^2, \ldots, x_i^N, u_i^N, x_{i+1}^N) \\ &\text{subject to} \quad u_i^k \in \mathcal{U}; |x^k - x^\kappa| \geq d_{col}, \forall \kappa \in \mathcal{N} \setminus \{k\};\ \dot{x}_i^k \in \boldsymbol{f}_i^k + \boldsymbol{G}_i^k u_i^k,\ \forall k \in \mathcal{N} \end{aligned} \tag{17}$$

where C_i^k is the agent k's cost function, x_i^k is the state of agent k at time step t_i, u_i^k represents the control input at time t_i, $d_{col} \in \mathbb{R}$ is collision avoidance value, $\boldsymbol{f}_i^\kappa$ is the interval-valued function of over-approximated set of f^k at time t_i, and $\boldsymbol{G}_i^\kappa$ is the interval-valued function of over-approximated set of G^k at time t_i. By applying the relaxation in the control input plane, the optimization problem becomes convex Djeumou et al. (2022), meaning that a constant control input will be applied in the continuous time interval $[t_i, t_{i+1})$.

Assumption 2 (Quadratic cost function) *We apply quadratic cost for one-step cost function C, which is a restricted strongly convex quadratic function,*

$$C(x, u, \dot{x}) = \begin{bmatrix} \dot{x} \\ u \end{bmatrix}^T \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \begin{bmatrix} \dot{x} \\ u \end{bmatrix} + \begin{bmatrix} q \\ r \end{bmatrix}^T \begin{bmatrix} \dot{x} \\ u \end{bmatrix} \tag{18}$$

where $Q = Q^T \in \mathbb{R}^{n \times n}$, $S \in \mathbb{R}^{n \times m}$, $R = R^T \in \mathbb{R}^{m \times n}$, $q \in \mathbb{R}^n$, and $r \in \mathbb{R}^m$. Please refer to Zhang and Cheng (2015) for more details about restricted strongly convex functions.

Theorem 5 provides an upper bound on the difference between the optimal cost function and the suboptimal cost function of optimization problem (17) where the suboptimal and optimal cost functions correspond to cases where the agents' dynamics is unknown and known, respectively.

Theorem 5 (Suboptimality bound) *The optimal and suboptimal cost functions of agent k at time t_i are C_i^{*k} and C_i^k, respectively.*

$$|C_i^{*k} - C_i^k| \leq \max\left(\|w(\sigma_i^k) + w(\zeta_i^{k,+})|\mathcal{U}|\|_2 Z(\zeta_i^{k,+}), \|w(\sigma_i^k) + w(\zeta_i^{k,+})|\mathcal{U}|\|_2 Z(\zeta_i^{k,+})\right) \tag{19}$$

where $Z\left(\zeta^{k,+}\right) = \min\left(\|2|\Gamma_i^k \mathcal{U}| + q + 2|Q\left(\sigma_i^k + \zeta_i^{k,+}\mathcal{U}\right)|\|_2, \|2|\Gamma_i^k\mathcal{U}| + q + 2|Q\mathcal{X}|\|_2\right)$ for any $\zeta^{k,+} \in \mathbb{IR}^{n \times m}$ and $w\left(\zeta^{k,+}\right) = \overline{\zeta}^{k,+} - \underline{\zeta}^{k,+}$.

Proof We consider two arbitrary states of the system for agent k; therefore, we have:

$$x_{i+1}^{k,+} = b_i^k + A_i^{k,+} u_i^k \in \mathcal{B}_i^k + \mathcal{A}_i^{k,+} u_i^k, \quad u_i^k \in \mathcal{U} \tag{20}$$

$$\hat{x}_{i+1}^{k,+} = \hat{b}_i^k + \hat{A}_i^{k,+} u_i^k \in \mathcal{B}_i^k + \mathcal{A}_i^{k,+} u_i^k, \quad u_i^k \in \mathcal{U} \tag{21}$$

Substituting Equations (20) and (21) into Equation (19) using Equation (18) format and by exploiting the width of the interval operator, w, to obtain $|b_i^k - \hat{b}_i^k| \leq w(\mathcal{B}_i^k)$ and $|A_i^{k,+} - \hat{A}_i^{k,+}| \leq w(\mathcal{A}_i^{k,+})$ knowing that the intervals of the over-approximation sets must stay in the set of states $\mathcal{X}$ (Equation (22)), then we obtain Equation (23):

$$(b_i^k + \hat{b}_i^k) + (A_i^{k,+} + \hat{A}_i^{k,+}) u_i \in 2(\mathcal{B}_i^k + 2\mathcal{A}_i^{k,+} u_i^k) \cap 2\mathcal{X} \tag{22}$$

$$\left\| 2S \begin{bmatrix} u_i^1 \\ \vdots \\ u_i^N \end{bmatrix} + q + Q\left(\begin{bmatrix} b_i^1 + \hat{b}_i^1 \\ \vdots \\ b_i^N + \hat{b}_i^N \end{bmatrix} + \begin{bmatrix} A_i^{1,+} + \hat{A}_i^{1,+} \\ \vdots \\ A_i^{N,+} + \hat{A}_i^{N,+} \end{bmatrix}^T \begin{bmatrix} u_i^1 \\ \vdots \\ u_i^N \end{bmatrix} \right) \right\|_2 \leq Z(\zeta_i^{k,+}) \tag{23}$$

Now, we can write the cost function inequality as:

$$
\begin{aligned}
&|C^k(\cdot, u_i^1, x_{i+1}^{1,+}, \dots, \cdot, u_i^N, x_{i+1}^{N,+}) - C^k(\cdot, u_i^1, \hat{x}_{i+1}^{1,+}, \dots, \cdot, u_i^N, \hat{x}_{i+1}^{N,+})| \\
&\leq \left\| \begin{bmatrix} w(\mathcal{B}_i^1) \\ \vdots \\ w(\mathcal{B}_i^N) \end{bmatrix} + \begin{bmatrix} w(\mathcal{A}_i^{1,+}) \\ \vdots \\ w(\mathcal{A}_i^{N,+}) \end{bmatrix} |\mathcal{U}| \right\|_2 Z(\zeta_i^{k,+})
\end{aligned} \tag{24}
$$

Considering that the optimal solution is u_i^{*k} and the next unknown state of the system will be $x_{i+1}^k(u_i^{*k}) \in \mathcal{B}_i^k + \mathcal{A}_i^{k,+} u_i^{*k}$. Then we can rewrite the Equation (24) as:

$$
\begin{aligned}
&|C_i^{*k} - C_i^k| = |C^{*k}(x_i^1, u_i^{*1}, x_{i+1}^1(u_i^{*1}), \dots, x_i^N, u_i^{*N}, x_{i+1}^N(u_i^{*N})) - \\
&C^k(x_i^1, u_i^1, \hat{x}_{i+1}^1(\hat{u}_i^1), \dots, x_i^N, \hat{u}_i^N, \hat{x}_{i+1}^N(\hat{u}_i^N))| \leq \left\| \begin{bmatrix} w(\mathcal{B}_i^1) \\ \vdots \\ w(\mathcal{B}_i^N) \end{bmatrix} + \begin{bmatrix} w(\mathcal{A}_i^{1,+}) \\ \vdots \\ w(\mathcal{A}_i^{N,+}) \end{bmatrix} |\mathcal{U}| \right\|_2 Z(\zeta_i^{k,+})
\end{aligned} \tag{25}
$$

The optimal control solution, u^{*k}, will correspond to $x_{i+1}^k \in \mathcal{B}_i^k + \mathcal{A}_i^{k,+} u^{*k}$ next unknown state. Let $\hat{u}_i^k$ be the optimal solution of problem (17), then the next known state is obtained by $\hat{x}_i^k(\hat{u}_i^k) \in \mathcal{B}_i^k + \mathcal{A}_i^{k,+} u_i^{*k}$; therefore $|C_i^{*k} - C_i^k|$ becomes

$$
\begin{aligned}
&|C^{*k}(x_i^1, u_i^{*1}, x_{i+1}^1(u_i^{*1}), \dots, x_i^N, u_i^{*N}, x_{i+1}^N(u_i^{*N})) - \\
&C^k(x_i^1, \hat{u}_i^1, \hat{x}_{i+1}^1(\hat{u}_i^1), \dots, x_i^N, \hat{u}_i^N, \hat{x}_{i+1}^N(\hat{u}_i^N))| \leq |C^{*k}(x_i^1, \hat{u}_i^1, x_{i+1}^1(\hat{u}_i^1), \dots, x_i^N, \hat{u}_i^N, x_{i+1}^N(\hat{u}_i^N)) - \\
&C^k(x_i^1, \hat{u}_i^1, \hat{x}_{i+1}^1(\hat{u}_i^1), \dots, x_i^N, \hat{u}_i^N, \hat{x}_{i+1}^N(\hat{u}_i^N))| \leq \|w(\sigma_i) + w(\zeta_i^{k,+})|\mathcal{U}|\|_2 Z(\zeta_i^{k,+})
\end{aligned} \tag{26}
$$

Similarly we could prove $|C_i^{*k} - C_i^k| \leq \|w(\sigma_i^k) + w(\zeta_i^{k,-})|\mathcal{U}|\|_2 Z(\zeta_i^{k,-})$ for $\zeta_i^{k,-}$, the suboptimality bounds for the optimistic control problem. We calculate the upper bound on $|\sum_{k=0}^N C_i^{*k} - \sum_{k=0}^N C_i^k|$ using triangle inequality and denote the calculated upper bound error for each agent k as Ξ^k (right-hand side of Eq. (19)). We extend this representation to problem (17).

$$
|\sum_{k=1}^N C_i^{*k} - \sum_{k=1}^N C_i^k| = |C_i^{*1} + \ldots + C_i^{*N} - \left(C_i^1 + \ldots + C_i^N\right)| \leq \tag{27}
$$

$$
|C_i^{*1} - C_i^1| + \ldots + |C_i^{*N} - C_i^N| \leq \Xi^1 + \ldots + \Xi^N. \tag{28}
$$

■

5. On-the-fly Control of Multi-agent Systems With Unknown Dynamics

`ODMU` (Algorithm 2) synthesizes control input vector U for each time interval $[t_i, t_i + \Delta t)$. `ODMU` uses the continuity of the unknown dynamical system to obtain the $R_i^{k,+}$. Using its sampled trajectory, it calculates the reachable set (Line 19) for agent k. Then agent k communicates with its neighbors $\mathcal{N}_k$ (one or more, depending on communication scheme) to receive the state of the neighbors $x_i^\kappa, \kappa \neq k$ (Line 21). We then calculate the over-approximations for $\boldsymbol{f}^k$ and $\boldsymbol{G}^k$, then calculate Γ^k set (Lines 23-24). Once the intervals are calculated for the system dynamics (Line 29), we solve the one-step optimization problem to calculate the control input u_i^k for agent k. We demonstrated the convergence of `ODMU` in Corollary 6. We propose two algorithms to solve (17).

5.1. Graph-Based Constrained ADMM Optimization for Multi-agent Systems

We introduce a distributed graph-based alternating direction method of multipliers (ADMM) constrained optimization for MAS with unknown dynamics `GM-ADMM`, which is a generalization of Khatana and Salapaka (2023), a distributed constrained optimization that can handle both equality and inequality constraints. `GM-ADMM` combines differential inclusion with the methods introduced in Khatana and Salapaka (2023) and Melbourne et al. (2020) for on-the-fly control synthesis. In Algorithm 1 (`GM-ADMM`), control input $U_{i'}^k$ (primal variable) is calculated by agent k and the consensus

Algorithm 1 Distributed on-the-fly control for MAS with consensus check, `GM-ADMM` algorithm.

Input: Hyper-parameter γ, $\{\eta_{i'}\}_{i'\geq 0}$ consensus tolerance, $Iter'_{max} \in \mathbb{Z}_{\geq 0}$, upper bound for convergence diameter $\mathcal{D} \in \mathbb{Z}_{\geq 0}$
Output: Updated consensus primal variable $y^k_{i'+1}$

1 $y^k_{i'} \in \mathcal{U}, \lambda^k_{i'} \in \mathbb{R}^n, \Upsilon^k_{i'} = \mathbb{R}^m, \mathcal{Z}^k_{i'} \in \mathbb{R}^{n\times m}, \mathscr{B}^k_{i'} \in \mathbb{R}^m$;
2 **function** `GM-ADMM()`**:**
3     $U^k_{i'+1} \leftarrow \arg\min_{U^k\in\mathcal{U}} \left\{ C^k_{i'} + \lambda^{k^T}_{i'} \left(U^k_{i'} - y^k_{i'}\right) + \frac{\gamma}{2}\|U^k_{i'} - y^k_{i'}\|_2^2 + \Upsilon^{k^T}_{i'} \left(\mathcal{Z}^k_{i'}U^k_{i'} - \mathscr{B}^k_{i'}\right) + \frac{\gamma}{2}\|\mathcal{Z}^k_{i'}U^k_{i'} - \mathscr{B}^k_{i'}\|_2^2 \right\}$
4     $\varpi^k_{i'=0} = \mathscr{U}^k_{i'=0}$; $\mathscr{V}^k_{i'=0} = 1$; $\mathscr{H}^k_{i'=0} = 0$; $\mathscr{M} := 1$
5   **for** $i' = 0, \ldots, Iter'_{max}$ **do**
6         $\mathscr{U}^k_{i'+1} \leftarrow p_{k,k}\mathscr{U}^k_{i'} + \sum_{\kappa\in\mathcal{N}} p_{k,\kappa}\mathscr{U}^\kappa_{i'}$, $\mathscr{V}^k_{i'+1} \leftarrow p_{k,k}\mathscr{V}^k_{i'} + \sum_{\kappa\in\mathcal{N}} p_{k,\kappa}\mathscr{V}^\kappa_{i'}$, $\varpi^k_{i'+1} \leftarrow \frac{1}{\mathscr{V}^k_{i'+1}}\mathscr{U}^k_{i'+1}$
7         $\mathscr{H}^k_{i'} \leftarrow \max_{\kappa\in\mathcal{N}} \left\{\|\varpi^k_{i'+1} - \varpi^k_{i'}\| + \mathscr{H}^\kappa_{i'}\right\}$
8   **if** $i' = \mathscr{M}\mathcal{D} - 1$ **then**
9             $\hat{\mathscr{H}}^k_{\mathscr{M}-1} = \mathscr{H}^k_{i'+1}$
10   **if** $\hat{\mathscr{H}}^k_{\mathscr{M}-1} < \eta_{i'}$ **then**
11                 $y^k_{i'+1} \leftarrow \left(U^k_{i'+1} + \frac{1}{\gamma}\lambda^k_{i'}\right)$, **break** (consensus reached)
12             $\mathscr{H}^k_{i'+1} = 0$, $\mathscr{M} = \mathscr{M} + 1$
13   **end**
14     $\lambda^k_{i'+1} \leftarrow \lambda^k_{i'} + \gamma\left(U^k_{i'+1} - y^k_{i'+1}\right)$, $\Upsilon^k_{i'+1} \leftarrow \Upsilon^k_{i'} + \gamma\left(\mathcal{Z}^k_{i'}U^k_{i'+1} - \mathscr{B}^k_{i'+1}\right)$

primal variable $y^k_{i'}$ which ensures $U^k_{i'} = U^\kappa_{i'}, \forall k, \kappa \in \mathcal{N}$. In `GM-ADMM`, each agent k communicates only with its neighbors to acquire information from them at each time step t_i. Hence, we relax the exact consensus requirement to a $\eta_{i'+1}$ closeness among the variables of all the agents, i.e., $\|y^k_{i'} - y^\kappa_{i'}\| \leq 2\eta_{i'}, \forall k, \kappa \in \mathcal{N}$ where $\eta_{i'} \in \mathbb{R}_{>0}$. $\lambda^k_{i'}$ and $\Upsilon^k_{i'}$ are the dual variables in the `GM-ADMM`. Algorithm 1 ensures at $t_{i'}$ agents reach consensus for U^k where agent k solves the Lagrangian of the optimization problem by initializing the consensus primal variable $y^k_{i'}$, dual variables $\lambda^k_{i'}, \Upsilon^k_{i'}$, and constraints $\mathcal{Z}^k_{i'}, \mathscr{B}^k_{i'}$, subsequently, agent k obtains $U^k_{i'+1}$ (Lines 3-4). Then the consensus variables $\varpi^k_{i'}, \mathscr{U}^k_{i'}$ ($\mathscr{U}^k_{i'=0} = U^k_{i'+1} + \lambda^k_{i'}$), $\mathscr{V}^k_{i'}, \mathscr{H}^k_{i'}$, and iterator $\mathscr{M}$ are initialized so that agent k can update its consensus regarding $y^k_{i'}$ based on its communication with all of its neighbors where $p_{k,\kappa}$ denotes the element (k, κ) in adjacency matrix p (Lines 4-6). Agent k then calculates the largest radius $\mathscr{H}^k_{i'}$ that encompasses all the consensus primal variables of itself and its neighbors and consensus is reached when $\mathscr{H}^k_{i'}$ is less than the consensus tolerance $\eta_{i'}$ leading to the update of $y^k_{i'+1}$ (Lines 7-12). Then, the dual variables are updated (Line 14).

Corollary 6 (`ODMU` convergence) *`ODMU` at each time step t_i converges to a suboptimal solution $U^*_i = [u^{*1}_i, \ldots, u^{*N}_i]$ for the optimization problem (17).*

Proof (1) Theorem 5 guarantees $|C^{*k}_i - C^k_i|$ is bounded at each time step t_i. (2) at each time step t_i, Lemma 1, Lemma 2, and Theorem 3 in Khatana and Salapaka (2023) guarantee that Algorithm 1 converges to the optimal cost function C^{*k}_i when the number of iterations in Algorithm 2 goes to infinity, i.e., Algorithm 1 is guaranteed to converge to a suboptimal cost function within finite number of iterations. Hence, Algorithm 1 converges to the optimal cost function C^{*k}_i when $i \to \infty$. ■

5.2. Data Driven Control of Multi-agent Systems With Decentralized On-the-Fly Control

We propose Data-driven control of MAS with Decentralized On-the-Fly control (`DMDO`) where we skip the consensus check compared to `GM-ADMM` and use asynchronous communication, so that

Algorithm 2 `ODMU`: one-step near-optimal on-the-fly control solution at time $t_i > t_\mu$

Input: Trajectory $\mathscr{J}_i^k$, cost threshold C_{th}, maximum iteration $Iter_{max} \in \mathbb{Z}_{\geq 0}$, control set $\mathcal{U}$, and if available any of the side information.

Output: Constant control signal obtained for time interval $[t_i, t_i + \Delta t)$

function `ComputeControlInput():`
  for $i = 1, \ldots, Iter_{max}$ **do**
    if $C^k > C_{th}$ **then**
      for $k = 1, \ldots, N$ *(Number of agents)* **do**
        $R_i^{k,+} \leftarrow \{x_i^k\}$
        for $\kappa \in \mathcal{N}_k, \kappa \neq k$ **do**
          $\{x_i^{k,\kappa}\} \leftarrow GetAgentsState()$ (Agent k communicates with its neighbors $\mathcal{N}_k$)
        end
        $\boldsymbol{f}^k(R_i^{k,+}) \leftarrow ComputeF(R_i^{k,+})$ using (9) , $\boldsymbol{G}^k(R_i^{k,+}) \leftarrow ComputeG(R_i^{k,+})$ using (10)
        $\Gamma_i^k \leftarrow \boldsymbol{f}^k(R_i^{k,+})$, $\boldsymbol{G}^k(R_i^{k,+})$, (11), $R_i^{k,+}$, and $\mathcal{U}$
        $\boldsymbol{f}^k(\Gamma_i^k) \leftarrow ComputeF(\Gamma_i^k)$ using (9), $\boldsymbol{G}^k(\Gamma_i^k) \leftarrow ComputeG(\Gamma_i^k)$ using (10)
        if $SideInformationAvailable()$ **then**
          Obtain tighter bounds on $\boldsymbol{f}^k(\Gamma_i^k)$ and $\boldsymbol{G}^k(\Gamma_i^k)$ using any side information from 1-4
        Compute the J_f^k using L_f^k and Compute the J_G^k using L_G^k
        $\mathcal{B}_i^k, \mathcal{A}_i^{k,+}, \mathcal{A}_i^{k,-} \leftarrow R_i^{k,+}, \Gamma_i^k, \mathcal{U}, \boldsymbol{f}^k, \boldsymbol{G}^k, J_f^k$, and J_G^k
        $u_{i+1}^k \leftarrow$ `GM-ADMM` (Algorithm 1) or `DMDO` (Subsection 5.2)
      end
  end
  return u_{i+1}^k

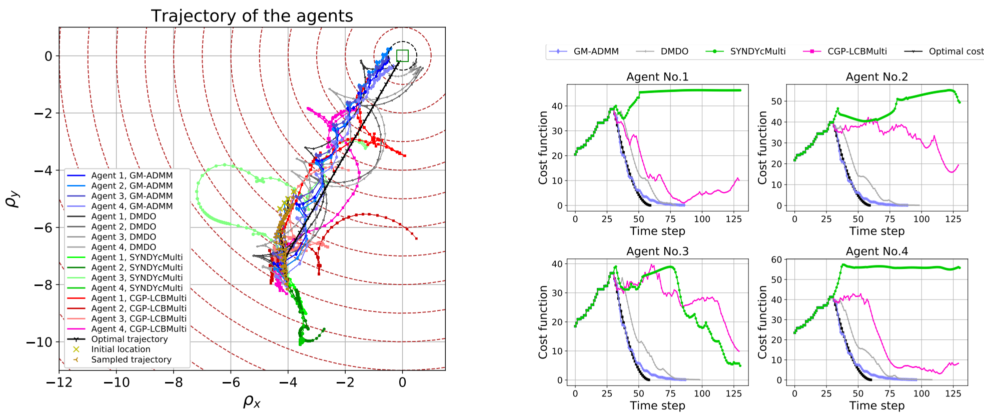
agent k at each time step t_i can communicate with only one of its neighbors $\mathcal{N}_k$; thus, $x_i^k = \Theta(x_{i-\tau}^\kappa), \kappa \neq k$ where Θ is the extrapolation function to update the estimated state (x^k) of agent k's from agent κ with time delay of $\tau \in \mathbb{Z}_{\geq 0}$ due to communication scheme. In `DMDO`, each agent calculates a near-optimal solution due to Theorem 5. This method is guaranteed to converge to an ϵ-Nash equilibrium due to the existence of the Ξ^k for each agent k (it can be proved using Theorem 5 and convexity of the optimization).

5.3. Side Information

We use side information to tighten approximations $R_i^{k,+}$ Djeumou et al. (2021). **Field Bounds (1):** Bounds on $\boldsymbol{f}^k$ and $\boldsymbol{G}^k$ for agent k are $\mathcal{R}^{k,f_\mathcal{A}} \in \mathbb{IR}^n$ and $\mathcal{R}^{k,G_\mathcal{A}} \in \mathbb{IR}^{n\times m}$ for a given set $\Gamma \subseteq \mathcal{A}$. Tighter field approximations are obtained as $\boldsymbol{f}^k(\Gamma) \leftarrow \boldsymbol{f}^k(\Gamma) \cap \mathcal{R}^{k,f_\mathcal{A}}$. **Gradients bounds(2):** Information on some components of the gradient for agent k, e.g., Jacobian bound J_f^k of f^k on set $\mathcal{A}$ leads to $(J_f^k)_{q,l} \leftarrow (J_f^k)_{q,l} \cap \mathbb{R}_+$ if $\Gamma_i^k \subseteq \mathcal{A}$. **Algebraic constraints (3):** Constraints on differentiable function $\mathcal{H}$, e.g., $\mathcal{H}(\dot{x}^k(\cdot), x^k(\cdot)) \geq 0$ for all $x^k \in \mathcal{X}$. **Dynamics partial knowledge(4):** Dynamics for agent k may include *known* and *unknown* components, e.g., $\boldsymbol{f}^k = \boldsymbol{f}_{\text{kn}}^k + \boldsymbol{f}_{\text{ukn}}^k$. This enables a tighter over-approximation of R_{i+1}^k by $J_f^k = \frac{\partial \boldsymbol{f}_{\text{kn}}^k}{\partial x^k}(\Gamma_i) + J_{f_{\text{ukn}}}^k$.

6. Simulation

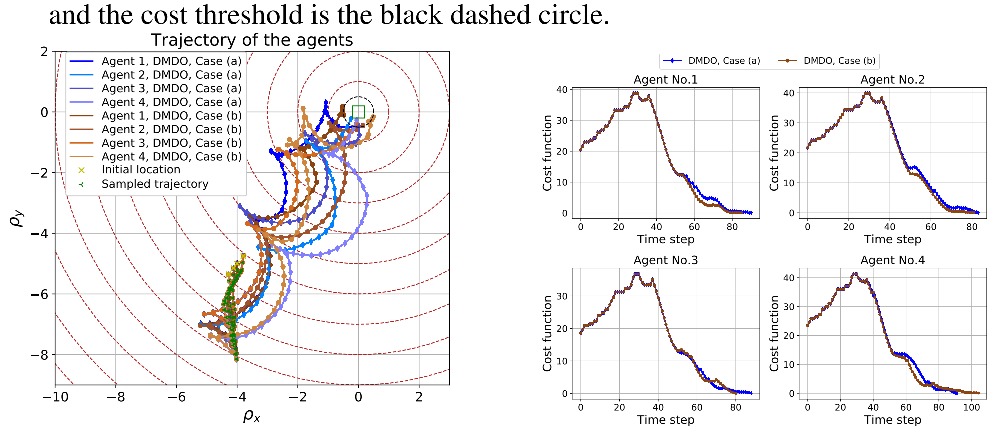
We consider $C^k = \frac{1}{2}\|x_{i+1}^k\|_2^2 + \frac{1}{2}\|x_{i+1}^k - \bar{x}_{i+1}^k\|_2^2$ where $\bar{x}^k = \frac{1}{|\mathcal{N}_k|}\sum_{\kappa\in\mathcal{N}_k} x^{k,\kappa}$; for each agent k we apply the step size of $\Delta t = 0.1[s]$ and $|\mathcal{N}_k| = 2$. We use $\mathscr{J}_{30}^k$ to sample 30 data points (one trajectory per agent) for R_i^k. Figure 1 which demonstrates the efficiency of our proposed methods `GM-ADMM` and `DMDO` in comparison to `SINDYcMulti` (an extension of `SINDYc` Kaiser et al. (2018) to MAS that uses sparse regression for nonlinear system identification; needs substantial data) and `CGP-LCBMulti` (a MAS extension of Krause and Ong (2011) that is based on Gaussian process). Our algorithms reach the cost threshold, while the `CGP-LCBMulti` algorithm cannot properly guide the agents towards the center, and `SINDYcMulti` struggles to coordinate the agents' inputs to achieve a correct heading. The cost associated with the optimization problem (17) for each agent is shown in Figure 1. `SINDYcMulti` is reaching a plateau after some steps, and `CGP-LCBMulti` cost is reducing but cannot reach the threshold. We consider the $L^k_{f,\text{kn}} = \begin{bmatrix} 0.15 & 0.15 & 0.18 \end{bmatrix}$ and $(J^k_f)_{q,l} = 0.05$ additionally to Example 1 (Case (b)) and the setup of the Example 1 (Case (a)) to demonstrate the effect of side information on the `DMDO` where Figure 2 shows agents in Case (b) reach the target state sooner in comparison with Case (a).



(a) MAS consisting of 4 agents driven to the target state.

(b) Cost associated with each agent.

Figure 1: Trajectories and optimization cost for Example 1, target state is the green square at $(0, 0)$ and the cost threshold is the black dashed circle.



(a) `DMDO` for cases (a) and (b).

(b) Cost for cases (a) and (b).

Figure 2: Effect of side information on `ODMU` convergence to target state.

7. Conclusion

We developed a distributed on-the-fly control algorithm for MAS. Our approach calculates the over-approximations of the reachable sets for each agent and exploits any available side information to compute a tighter over-approximation set. We showed that `ODMU` is guaranteed to calculate the near-optimal control input for the MAS. The experiment demonstrates the efficiency of our proposed method. For future work, we plan to investigate the effect of communication schemes and expand the work to swarm systems.

Acknowledgments

We would like to express our sincere appreciation to Professor Franck Djeumou for his input to this project. This research is partially supported by the National Science Foundation under grant NSF CNS 2304863 and the Office of Naval Research under grant ONR N00014-23-1-2505.

References

Julian Berberich, Johannes Köhler, Matthias A Müller, and Frank Allgöwer. Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, 66(4):1702–1717, 2020.

Julian Berberich, Carsten W Scherer, and Frank Allgöwer. Combining prior knowledge and data for robust controller design. *IEEE Transactions on Automatic Control*, 2022.

Martin Berz and Kyoko Makino. Verified integration of odes and flows using differential algebraic methods on high-order taylor models. *Reliable computing*, 4(4):361–369, 1998.

Jan-Peter Calliess. Lipschitz optimisation for lipschitz interpolation. In *2017 American Control Conference (ACC)*, pages 3141–3146. IEEE, 2017.

Ankush Chakrabarty, Devesh K Jha, and Yebin Wang. Data-driven control policies for partially known systems via kernelized lipschitz learning. In *2019 American Control Conference (ACC)*, pages 4192–4197. IEEE, 2019.

Franck Djeumou, Abraham P Vinod, Eric Goubault, Sylvie Putot, and Ufuk Topcu. On-the-fly control of unknown smooth systems from limited data. In *2021 American Control Conference (ACC)*, pages 3656–3663. IEEE, 2021.

Franck Djeumou, Abraham P Vinod, Eric Goubault, Sylvie Putot, and Ufuk Topcu. On-the-fly control of unknown systems: From side information to performance guarantees through reachability. *IEEE Transactions on Automatic Control*, 2022.

Eric Goubault and Sylvie Putot. Inner and outer reachability for the verification of control systems. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, pages 11–22, 2019.

Zhong-Sheng Hou and Zhuo Wang. From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, 235:3–35, 2013.

Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A*, 474(2219): 20180335, 2018.

Vivek Khatana and Murti V. Salapaka. Dc-distadmm: Admm algorithm for constrained optimization over directed graphs. *IEEE Transactions on Automatic Control*, 68(9):5365–5380, 2023. doi: 10.1109/TAC.2022.3221856.

J. Kocijan, R. Murray-Smith, C.E. Rasmussen, and A. Girard. Gaussian process model based predictive control. In *Proceedings of the 2004 American Control Conference*, volume 3, pages 2214–2219 vol.3, 2004. doi: 10.23919/ACC.2004.1383790.

Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

Andreas Krause and Cheng Ong. Contextual gaussian process bandit optimization. *Advances in neural information processing systems*, 24, 2011.

Ivan Markovsky and Florian Dörfler. Data-driven dynamic interpolation and approximation. *Automatica*, 135:110008, 2022.

James Melbourne, Govind Saraswat, Vivek Khatana, Sourav Patel, and Murti V Salapaka. On the geometry of consensus algorithms with application to distributed termination in higher dimension. *IFAC-PapersOnLine*, 53(2):2951–2956, 2020.

Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on automatic control*, 50(7):947–957, 2005.

Melkior Ornik, Steven Carr, Arie Israel, and Ufuk Topcu. Control-oriented learning on the fly. *IEEE Transactions on Automatic Control*, 65(11):4800–4807, 2019.

Henk J van Waarde, M Kanat Camlibel, and Mehran Mesbahi. From noisy data to feedback controllers: Nonconservative design via a matrix s-lemma. *IEEE Transactions on Automatic Control*, 67(1):162–175, 2020.

Henk J Van Waarde, Jaap Eising, Harry L Trentelman, and M Kanat Camlibel. Data informativity: a new perspective on data-driven analysis and control. *IEEE Transactions on Automatic Control*, 65(11):4753–4768, 2020.

Abraham P Vinod, Arie Israel, and Ufuk Topcu. Convexified contextual optimization for on-the-fly control of smooth systems. In *2020 American Control Conference (ACC)*, pages 2004–2011. IEEE, 2020.

Zelda B Zabinsky, Robert L Smith, and Birna P Kristinsdottir. Optimal estimation of univariate black-box lipschitz functions with upper and lower error bounds. *Computers & Operations Research*, 30(10):1539–1553, 2003.

Hui Zhang and Lizhi Cheng. Restricted strong convexity and its applications to convergence analysis of gradient-type methods in convex optimization. *Optimization Letters*, 9:961–979, 2015.