Topology-preserving flocking of nonlinear agents using optimistic planning*

Lucian Buşoniu, Irinel-Constantin Moruarescu†

Abstract

We consider the generalized flocking problem in multiagent systems, where the agents must drive a subset of their state variables to common values, while communication is constrained by a proximity relationship in terms of another subset of variables. We build a flocking method for general nonlinear agent dynamics, by using at each agent a near-optimal control technique from artificial intelligence called optimistic planning. By defining the rewards to be optimized in a well-chosen way, the preservation of the interconnection topology is guaranteed, under a controllability assumption. We also give a practical variant of the algorithm that does not require to know the details of this assumption, and show that it works well in experiments on nonlinear agents.

1 Introduction

Multi-agent systems such as robotic teams, energy and telecommunication networks, collaborative decision support systems, data mining, etc. appear in many areas of technology. Their component agents usually only have a local, limited view, which means decentralized approaches are necessary to control the overall system. In this decentralized setting, often consensus between the agents is desired, meaning that they must reach agreement on certain controlled variables of interest [1, 2, 3]. Inspired by the behavior of flocks of birds, researchers studied the *flocking* variant of consensus, which only requires consensus on velocities while also using position measurements [3, 4]. Flocking is highly relevant in e.g. mobile robot teams [5].

In this paper we consider a generalized version of the flocking problem, in which agreement is sought for a subset of agent variables, while other variables define the interconnection topology between the agents. These two subsets may,

* The work of L. Buşoniu was supported by a grant of the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PNII-RU-TE-2012-3-0040. The work of I.-C Moruarescu was partially funded by the ANR project ”Computation Aware Control Systems”, ANR-13-BS03-004-02.

† L. Buşoniu is with Department of Automation, Technical University of Cluj-Napoca, Memorandumului 28, 400114 Cluj-Napoca, Romania. I.-C. Moruarescu is with Université de Lorraine, CRAN, UMR 7039 and CNRS, CRAN, UMR 7039, 2 Avenue de la Forêt de Haye, Vandœuvre-lès-Nancy, France.

but need not represent velocities and positions. The communication connections between agents are based on a proximity relationship, in which a connection is active when the agents are closer than some threshold in terms of the connectivity variables. Each agent finds control actions (inputs) using the optimistic planning (OP) algorithm from artificial intelligence [6]. OP works with discrete actions, like the consensus method of [7], and finds sequences of actions that are near-optimal with respect to general nonquadratic reward functions, for general nonlinear dynamics. The first major advantage of our technique is this inherited generality: it works for any type of nonlinear agents. A controllability property is imposed that, for any connected state, roughly requires the existence of an input sequence which preserves connectivity. We define agent reward functions with separate agreement and connectivity components, and our main analytical result shows that if the connectivity rewards are sufficiently large, the algorithm will preserve the interconnection topology. In interesting cases, the computational complexity of the flocking problem is not larger than if the agent would solve the agreement-only problem. The theoretical algorithm is restrictive in requiring to know the length of action sequences satisfying the controllability property. We therefore also provide a practical algorithm variant which does not use this knowledge, and validate it in simulation to nonholonomic agents and robot arms [8]. In the second problem we illustrate that despite our focus on flocking, the method also works in the full-state consensus case.

The main novelty of the OP approach compared to existing methods is that it is agnostic to the specific agent dynamics, and so it works uniformly for general nonlinear agents. In particular, our analysis shows that when a solution that preserves the topology exists (in a sense that will be formalized later), then irrespective of the details of the dynamics the algorithm will indeed maintain the topology. Existing topology preservation results are focused on specific types of agents, mainly linear [9, 10], [11, Ch. 4], or sometimes nonlinear as in e.g. [12] where the weaker requirement of connectivity is considered. Our practical flocking algorithm exhibits the same generality, whereas existing methods exploit the structure of the specific dynamics targeted to derive predefined control laws, e.g. for linear double integrators [3], agents with nonlinear acceleration dynamics [13, 14], or nonholonomic robots [15, 12]. The technical contribution allowing us to achieve these results is the exploitation of the OP algorithm, and of its strong near-optimality guarantees.

The approach presented here is a significant extension of our earlier work [16]: it introduces a new algorithm that is theoretically shown to preserve the topology, and also includes new empirical results for nonholonomic agents. Also related is our optimistic-*optimization* based approach of [17], which only handles consensus on a fixed graph rather than flocking, and directly optimizes over fixed-length action sequences rather than using planning to exploit the dynamical structure of the control problem.

The remainder of this paper is organized as follows. After formalizing the problem in Section 2 and explaining OP in Section 3, the two variants of the consensus algorithm and the analysis of the theoretical variant are given in Section 4. Section 5 presents the experimental results and Section 6 concludes the paper.

List of symbols and notations

$\lvert\cdot\rvert$	cardinality of argument set
n	number of agents
$\mathcal{G}, \mathcal{V}, \mathcal{E}, \mathcal{N}$	graph, vertices, edges, neighbors
i, j	agent indices
x, u, f	state, action, dynamics
$x^{\mathrm{a}}, x^{\mathrm{c}}$	agreement states, connectivity states
P	communication range
$\tilde{u}, \tilde{f}$	extended action, extended dynamics
k	absolute time step
K	length of sequence ensuring connectivity
$\boldsymbol{u}_d$	action sequence of length d
$\boldsymbol{u}_i^k$	action sequence of agent i at k
$\boldsymbol{x}_i^k$	state sequence of agent i at k
$\widehat{\boldsymbol{x}}_j^{i,k}$	state sequence prediction for agent j, built by agent i at step k
$u_{i,d}^k, x_{i,d}^k, \widehat{x}_{j,d}^{i,k}$	the dth element of respective sequence
ρ, v	reward function, value (return)
γ	discount factor
Δ, Γ	agreement reward, connectivity reward
β	weight of connectivity reward
T	optimistic planning budget
$\mathcal{T}, \mathcal{T}^*, \mathcal{L}$	tree, near-optimal tree, leaves
d	depth in the tree (relative time step)
b, ν	upper and lower bound on return
κ	branching factor of near-optimal tree

2 Problem statement

Consider a set of n agents with decoupled nonlinear dynamics $x_{i,k+1} = f_i(x_{i,k}, u_{i,k})$, $i = 1, \ldots, n$, where x_i and u_i denote the state and action (input) of the ith agent, respectively. The agents can be heterogeneous: they can have different dynamics and state or input dimensionality. An agent only has a local view: it can receive information only from its neighbors on an interconnection graph $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$, which can be time-varying. The set of nodes $\mathcal{V} = \{1, \ldots, n\}$ represents the agents, and the edges $\mathcal{E}_k \subseteq \mathcal{V} \times \mathcal{V}$ are the communication links. Denote by $\mathcal{N}_{i,k} = \{j \mid (i,j) \in \mathcal{E}_k\}$ the set of neighbors of node i at step k. A path through the graph is a sequence of nodes $i_1, \ldots, i_N$ so that $(i_l, i_{l+1}) \in \mathcal{E}_k, 1 \leq l < N$. The graph is connected if there is a path between any pair of nodes i, j.

The objective can be formalized as:

$$\lim_{k \to \infty} \|x_{i,k}^{\mathrm{a}} - x_{j,k}^{\mathrm{a}}\| = 0 \quad \forall i, j = 1, \ldots, n$$

where x^{a} selects only those state variables for which *agreement* is desired, and $\|\cdot\|$ denotes an arbitrary norm. We require of course that the selection produces a vector with the same dimensionality for all agents. When all agents have the same state dimension, $x^{\mathrm{a}} = x$, and $\mathcal{E}_k = \mathcal{E}$ (a fixed communication graph) we obtain the standard full-state consensus problem [1, 2]. While our technique can be applied to this case, as will be illustrated in the experiments, in the analytical development we will focus on the flocking problem, where the communication network varies with the *connectivity* state variables x^{c}. Usually, x^{a} and x^{c} do not overlap, being e.g., the agent's velocity and position [3], so that velocities must be synchronized under communication constraints dependent on the position. Specifically, we consider the case where a link is active when the connectivity states of two agents are close:

$$\mathcal{E}_k = \left\{(i,j) \;\middle|\; i \neq j, \|x^{\mathrm{c}}_{i,k} - x^{\mathrm{c}}_{j,k}\| \leq P \right\} \tag{1}$$

For example when x^{c} is a position this corresponds to the agents being physically closer than some transmission range P.

Our approach requires discretized agent actions.

Assumption 1 *Agent actions are discretized: $u_i \in \{U_i\}$ with $|U_i| = M_i$.*

Remark: Certain systems have inherently discrete and finitely-many actions, because they are controlled by switches. When the actions are originally continuous, discretization reduces performance, but the loss is often manageable. Other authors showed interest in multiagent coordination with discretized actions, e.g. [7]. □

Further, to develop our connectivity analysis, we require the following controllability condition. Denote $\boldsymbol{u}_{i,K} = (u_{i,0}, u_{i,1}, \ldots, u_{i,K-1}) \in U_i^K$ a sequence of K actions of agent i, and $\tilde{f}_i(x_i, \boldsymbol{u}_{i,K})$ the result of applying this sequence: the agent's state after K steps, with $\tilde{f}_i$ the extended dynamics.

Assumption 2 *There exists K so that for any agent i, and any states x_i, x_j, $\forall j \in \mathcal{N}_{i,k}$ so that $\|x^{\mathrm{c}}_i - x^{\mathrm{c}}_j\| \leq P$, there exists some sequence $\boldsymbol{u}_{i,K}$ so that $\|\tilde{f}^{\mathrm{c}}_i(x_i, \boldsymbol{u}_{i,K}) - x^{\mathrm{c}}_j\| \leq P$, $\forall j \in \mathcal{N}_{i,k}$.*

Remark: This is a feasibility assumption: it is difficult to preserve the topology without requiring such a condition. The condition simply means that for any joint state of the system in which an agent is connected to some neighbors, this agent has an action sequence by which it is again connected after K steps, if its neighbors do not move. So if the assumption does not hold and the problem is such that the neighbors do stay still, the agent will indeed lose some connections and topology cannot be preserved. Of course, in general the neighbors will move, but as we will show Assumption 2 is nevertheless sufficient to ensure connectivity.

K-step controllability properties are thoroughly studied in the literature, e.g. [18] provide Lie-algebraic conditions to guarantee them. We make a similar assumption in our previous paper [17], where it is however much stronger, requiring that the control is able to move the agent between *any* two arbitrary states in a bounded region. With a sufficiently fine action discretization, such an assumption would locally imply Assumption 2 in the present paper.

When making the assumption, we could also use the following definition for the links:

$$\mathcal{E}_k = \{(i,j)|i \neq j, \|x^{\mathrm{c}}_{i,k} - x^{\mathrm{c}}_{j,k}\| \leq P, \text{ and if } k > 0, (i,j) \in \mathcal{E}_{k-1}\} \tag{2}$$

so that the agents never gain new neighbors, and only need to stay connected to their initial neighbors. The analysis will also hold in this case, which is important because with (1), as k grows many or all the agents may become interconnected. For simplicity we use (1) in the sequel. □

3 Background: Optimistic planning for deterministic systems

Consider a (single-agent) optimal control problem for a deterministic, discrete-time nonlinear system $x_{d+1} = f(x_d, u_d)$ with states x and actions u. Define an infinitely-long action sequence $\boldsymbol{u}_\infty = (u_0, u_1, \dots)$ and its truncation to d initial actions, $\boldsymbol{u}_d = (u_0, \dots, u_{d-1})$. Given an initial state x_0, the return of a sequence is:

$$v(\boldsymbol{u}_\infty) = \sum_{d=0}^{\infty} \gamma^d \rho_{d+1}(\boldsymbol{u}_{d+1}) \tag{3}$$

where $\rho_d : U^d \to [0, 1]$ gives the reward after d steps and $\gamma \in [0, 1)$ is a discount factor, which given the bounded rewards ensures bounded returns. For example, an approximately quadratic problem is obtained if $\rho_d(\boldsymbol{u}_d) = 1 - \max\{{x_d}^\top Q x_d, 1\}$, where x_d is the result of applying $\boldsymbol{u}_d$ from initial state x_0 and Q is properly chosen so that the rewards are sensitive to the interesting region of x. Denote $v^* = \sup_{\boldsymbol{u}_\infty} v(\boldsymbol{u}_\infty)$ the optimal return. Note that reinforcement learning [19] and adaptive dynamic programming [20] also aim to solve this type of optimal control problem.

Optimistic planning for deterministic systems (OP) [6, 21] explores a tree representation of the possible action sequences from the current system state, as illustrated in Figure 1. It requires a discrete action space $U = \{u^1, \dots, u^M\}$; recall Assumption 1, which ensures this is true for our agents. OP starts with a root node representing the empty sequence, and iteratively expands T well-chosen nodes. Expanding a node adds M new children nodes for all possible discrete actions. Each node at some depth d is reached via a unique path through the tree, associated to a unique action sequence $\boldsymbol{u}_d$ of length d. We will denote the nodes by their corresponding action sequences. Denote also the current tree by $\mathcal{T}$, and its leaves by $\mathcal{L}(\mathcal{T})$.

For a leaf node $\boldsymbol{u}_d$, the following gives an upper bound on the returns of all

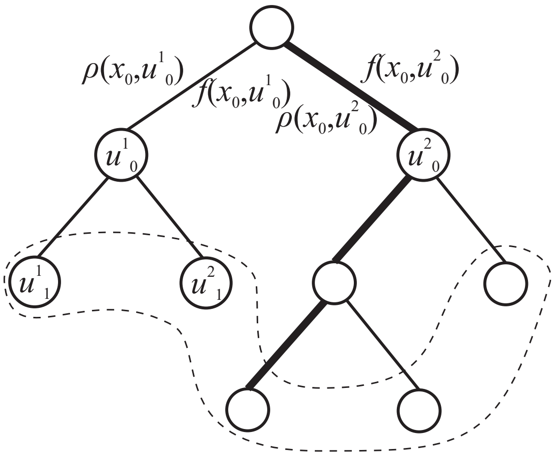

Figure 1: Illustration of an OP tree $\mathcal{T}$. Nodes are labeled by actions, arcs represent transitions and are labeled by the resulting states and rewards. Subscripts are depths, superscripts index the M possible actions/transitions from a node (here, $M = 2$). The leaves are enclosed in a dashed line, while the thick path highlights a sequence.

infinite action sequences having in common the initial subsequence up to $\boldsymbol{u}_d$:

$$b(\boldsymbol{u}_d) = \sum_{e=0}^{d-1} \gamma^e \rho_{e+1}(\boldsymbol{u}_{e+1}) + \frac{\gamma^d}{1-\gamma} =: \nu(\boldsymbol{u}_d) + \frac{\gamma^d}{1-\gamma}$$

where $\nu(\boldsymbol{u}_d)$ is a lower bound. These properties hold because all the rewards at depths larger than d are in $[0, 1]$.

OP *optimistically* explores the space of action sequences, by always expanding further the most promising sequence: the one with the largest upper bound, $\boldsymbol{u}^\dagger = \arg\max_{\boldsymbol{u}\in\mathcal{L}(\mathcal{T})} b(\boldsymbol{u})$. After T node expansions, a sequence that maximizes ν among the leaves is returned, intuitively seen as a safe choice, see Algorithm 1.

Algorithm 1 Optimistic planning for deterministic systems.

```
initialize tree: $\mathcal{T} \leftarrow \{$empty sequence $\boldsymbol{u}_0\}$
for $t = 1, \ldots, T$ do
    find optimistic leaf: $\boldsymbol{u}^\dagger \leftarrow \arg\max_{\boldsymbol{u}\in\mathcal{L}(\mathcal{T})} b(\boldsymbol{u})$
    add to $\mathcal{T}$ the children of $\boldsymbol{u}^\dagger$,
        labeled by $u^1, \ldots, u^M$
end for
return $\boldsymbol{u}^*_{d^*}$, where $\boldsymbol{u}^*_{d^*} = \arg\max_{\boldsymbol{u}\in\mathcal{L}(\mathcal{T})} \nu(\boldsymbol{u})$
```

Usually OP and its analysis are developed for time-invariant reward functions [6, 21], such as the quadratic reward exemplified above. However, this fact is not used in the development, which therefore entirely carries over to the time-varying case explained here. We provide the algorithm and results directly in the time-varying case, since this will be useful in the consensus context.

To characterize the complexity of finding the optimal sequence from a given state x, we use the asymptotic branching factor of the near-optimal subtree:

$$\mathcal{T}^* = \{\boldsymbol{u}_d \,|\, d \geq 0, v^* - v(\boldsymbol{u}_d) \leq \frac{\gamma^d}{1-\gamma}\} \tag{4}$$

where the value of a finitely long sequence is defined as $v(\boldsymbol{u}_d) = \sup_{\boldsymbol{u}_\infty \in \boldsymbol{u}_d} v(\boldsymbol{u}_\infty)$ and $\boldsymbol{u}_\infty \in \boldsymbol{u}_d$ means that $\boldsymbol{u}_\infty$ starts with $\boldsymbol{u}_d$. Let $\mathcal{T}_d^*$ be the set of nodes at depth d on $\mathcal{T}^*$ and $|\cdot|$ denote set cardinality, then the asymptotic branching factor is defined as $\kappa = \limsup_{d\to\infty} |\mathcal{T}_d^*|^{1/d}$.

A sequence $\boldsymbol{u}_d$ is said to be ε-optimal when $v^* - v(\boldsymbol{u}_d) \leq \varepsilon$. The upcoming theorem is a consequence of the analysis in [6, 21]. It is given here in a form that brings out the role of the sequence length, useful later. Part (i) of the theorem shows that OP returns a long, near-optimal sequence, while part (ii) quantifies this length and near-optimality, via branching factor κ.

Theorem 3 *When OP is called with budget T:*

(i) The length d^* of the sequence $\boldsymbol{u}_{d^*}^*$ returned is at least $d(\mathcal{T}) - 1$ where $d(\mathcal{T})$ is the depth of the tree developed. This sequence is $\frac{\gamma^{d^*}}{1-\gamma}$-optimal.

(ii) If $\kappa > 1$ OP will reach a depth of $d^* = \Omega(\frac{\log T}{\log \kappa})$, and its near-optimality will be $O(T^{-\frac{\log 1/\gamma}{\log \kappa}})$. If $\kappa = 1$, $d^* = \Omega(T)$ and near-optimality is $O(\gamma^{cT})$, where c is a problem-dependent constant. □

*Proof:*Part (i) follows from the proof of Theorem 2 in [6], and (ii) from the proofs of Theorems 2 and 3 in [6]. A sketch for part (ii) is given here, since it will be useful later in our analysis. A core property of OP is that it only expands nodes in $\mathcal{T}^*$. According to item (i), performance is dominated by the depth reached. Thus the worst case is when nodes in $\mathcal{T}^*$ are expanded in the order of their depth. Now, $\mathcal{T}^*$ contains $T = O(\kappa^d)$ nodes up to depth d when $\kappa > 1$, and $T = O(d)$ otherwise. Inverting these relationships obtains the formulas for d^* in the Theorem statement, and replacing these expressions for d^* into $\frac{\gamma^{d^*}}{1-\gamma}$ provides the corresponding near-optimality bounds.

The smaller κ, the better OP does. The best case is $\kappa = 1$, obtained e.g. when a single sequence always obtains rewards of 1, and all the other rewards on the tree are 0. In this case the algorithm must only develop this sequence, and suboptimality decreases exponentially. In the worst case, $\kappa = M$, obtained e.g. when all the sequences have the same value, and the algorithm must explore the complete tree in a uniform fashion, expanding nodes in order of their depth.

4 Flocking algorithm and analysis

The OP-based approach to the flocking problem in Section 2 works as follows. At every time step k, a local optimal control problem is defined for each agent i, using information locally available to it. The goal in this problem is to align the agreement states x^{a} with those of the neighbors $\mathcal{N}_{i,k}$, while maintaining the connection topology by staying close to them in terms of x^{c}. OP is used to near-optimally solve this control problem, and an initial subsequence of the sequence returned is applied by the agent. Then the system evolves, and the procedure is applied again, for the new states and possibly changed graph.

To construct its optimal control problem, each agent needs the predicted behavior of its neighbors. Here, agents will exchange the predicted state sequences resulting from the near-optimal action sequences returned by OP. Because the agents must act at the same time, how they exchange predictions is nontrivial. If predictions do not match, a *coordination* problem may arise where mismatching actions are applied. Coordination is a difficult challenge in multi-agent systems and is typically solved in model-predictive control by explicit, iterative negotiation over successive local solutions, e.g. [22]. However, it is unlikely that the agents can computationally afford to repeatedly communicate and reoptimize their solutions at every step. Thus we adopt a sequential communication procedure in which agents optimize once per step, similar to the procedure for distributed MPC in [23]. We show in Section 4.1 that connectivity can be guaranteed despite this one-shot solution.

To implement the sequential procedure, each agent needs to know its index i as well as the indices of its neighbors. One way to ensure this is an initial, centralized assignment of indices to the agents. Agent i waits until the neighbors j with $j < i$ have solved their local optimal control problems and found their predicted state sequences. These agents communicate their predictions to i. For $j > i$, agent i constructs other predictions as described later. Agent i optimizes its own behavior while coordinating with the predictions. It then sends its own, newly computed prediction to neighbors $j > i$.

To formalize the approach, first note that when Algorithm 1 is called, we internally relabel time k to 0, so that indices/depths d in OP and the analysis of Section 3 are relative to k. Externally, we denote quantities that depend on the time step by superscript k. Then, the planner of some agent i returns at step k an action sequence denoted $\boldsymbol{u}_i^k = (u_{i,0}^k, u_{i,1}^k, ..., u_{i,d-1}^k)$, which leads to predicted state sequence $\boldsymbol{x}_i^k = (x_{i,0}^k, x_{i,1}^k, \dots, x_{i,d}^k)$. Here $x_{i,0}^k = x_{i,k}$ is the state measurement, and the other states are predictions. Sequences found at different time steps may have different actions at corresponding positions (e.g., $u_{i,1}^k$ may be different from $u_{i,0}^{k+1}$ even though they correspond to the same time index, $k+1 = (k+1)+0$).

Consider now a specific agent i. At every step k, it receives the states $x_{j,k}$ of its neighbors $j \in \mathcal{N}_{i,k}$. For neighbors $j \in \mathcal{N}_{i,k},\ j < i$, it directly receives their prediction at k and uses this as an estimation of their future behavior: $\widehat{\boldsymbol{x}}_j^{i,k} = (\widehat{x}_{j,0}^{i,k}, \widehat{x}_{j,1}^{i,k}, \dots) = \boldsymbol{x}_j^k$. For $j \in \mathcal{N}_{i,k},\ j > i$, updated predictions are not available, instead a different prediction $\widehat{\boldsymbol{x}}_j^{i,k}$ is formed in a way that we specify later. We add i to the superscript to highlight that the predictions are from the point of view of agent i.

The local optimal control problem of agent i is then defined using the reward function:

$$\rho_{i,d}^k(\boldsymbol{u}_{i,d}) = (1-\beta)\Delta_{i,d}^k(\boldsymbol{u}_{i,d}) + \beta\Gamma_{i,d}^k(\boldsymbol{u}_{i,d}) \tag{5}$$

where $\Delta_{i,d}^k : U_i^d \to [0,1]$ rewards the alignment between agreement states and $\Gamma_{i,d}^k : U_i^d \to [0,1]$ rewards the preservation of neighbor connections, with β weighing the relative importance of these terms. Typically, $\beta \geq 1-\beta$ so that connectivity is given priority. Recall that depth d in the planning tree is equivalent to a time index relative to k. Both Δ and Γ may use the predictions $\widehat{\boldsymbol{x}}_j^{i,k}$. Note that d may exceed the length of the available predictions; when that happens the predictions are heuristically kept constant at the last value available.

In the implementation, if the agents have their neighbors' models, they could also exchange predicted action sequences instead of states. Since actions are discrete and states usually continuous, this saves some bandwidth at the cost of extra computation to resimulate the neighbor's transitions up to the prediction length. In any case, it should be noted that agents do *not* optimize over the actions of their neighbors, so complexity does not directly scale with the number of neighbors.

So far, we have deliberately left open the specific form of the rewards and predictions for neighbors $j > i$. Next, we instantiate them in a theoretical algorithm for which we guarantee the preservation of the interconnection topology and certain computational properties. However, this theoretical variant has shortcomings, so we additionally present a different instantiation which is more suitable in practice and which we later show works well in experiments.

4.1 A theoretical algorithm with guaranteed topology preservation

Our aim in this section is to exploit Assumption 2 to derive an algorithm that preserves the communication connections. We first develop the flocking protocol for each agent, shown as Algorithm 2. Our analysis proceeds by showing that, if sequences preserving the connections exist at a given step, the rewards can be designed to ensure that the algorithm will indeed find one such sequence (Lemma 4). This property is then used to prove topology preservation in closed loop, in Theorem 5. Finally, Theorem 6 shows an interesting computational property of the algorithm: under certain conditions the extra connectivity reward does not increase the complexity from the case where only agreement would be required.

Define a prediction for agents $j > i$ held constant to the latest exchanged state, $\widehat{\boldsymbol{x}}_j^{i,k} = (x_{j,k}, x_{j,k}, \dots)$. Then, the connectivity reward for agent i is an indicator function that becomes 0 only if agent i breaks connectivity with some neighbor(s) after K steps:

$$\Gamma^k_{i,d}(\boldsymbol{u}_{i,d}) = \begin{cases} 0 & \text{if } d = K \text{ and} \\ & \exists j \in \mathcal{N}_{i,k}, \|\widehat{x}^{i,k,\mathrm{c}}_{j,d} - x^{k,\mathrm{c}}_{i,d}\| > P \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

The agreement reward is left general, but to fix ideas, it could be for instance:

$$\Delta^k_{i,d}(\boldsymbol{u}_{i,d}) = 1 - \frac{1}{|\mathcal{N}_{i,k}|} \sum_{j \in \mathcal{N}_{i,k}} \max\{\|\widehat{x}^{i,k,\mathrm{a}}_{j,d} - x^{k,\mathrm{a}}_{i,d}\|, 1\} \tag{7}$$

where the distance measure $\|\cdot\|$ (which may be a norm or more general) is properly weighted to be sensitive to the relevant regions of x^{a}. Then, the agents always apply in open loop the first K actions from their computed sequences, after which they close the loop, measure the state, and repeat the procedure, see Algorithm 2.

Algorithm 2 OP flocking at agent i – theoretical variant.

```
set initial prediction $\boldsymbol{x}_i^{-1}$ to an empty sequence
for $\ell = 0, 1, 2, \dots$ do
    current step is $k \leftarrow \ell K$
    exchange state at $k$ with all neighbors $j \in \mathcal{N}_{i,k}$
    send $\boldsymbol{x}_i^{k-1}$ to $j < i$
    wait to receive new predictions $\widehat{\boldsymbol{x}}_j^{i,k}$ from all $j < i$
    form predictions $\widehat{\boldsymbol{x}}_j^{i,k}$ for $j > i$
    run OP with (5) and (6), obtaining $\boldsymbol{u}_i^k$ and $\boldsymbol{x}_i^k$
    send $\boldsymbol{x}_i^k$ to $j > i$
    execute $K$ actions $u^k_{i,0}, \dots, u^k_{i,K-1}$ in open loop
end for
```

The reader may wonder why we do not simply redefine the optimal control problem in terms of the multistep dynamics $\tilde{f}_i$. The answer is that this would introduce exponential complexity in K: instead of M_i actions, we would have M_i^K, and this would also be the number of children created with each node expansion in OP. In contrast, applying OP directly to the 1-step problem leads to significantly decreased computation – in some cases no more than solving a 1-step problem without connectivity constraints, as shown in Theorem 6 below.

Moving on to the analysis now, we first show that when it is possible, each agent preserves connectivity with respect to the predicted states of its neighbors.

Lemma 4 *Take $\beta \geq \frac{1/(1-\gamma)+\varepsilon}{1/(1-\gamma)+\gamma^{K-1}}$ for some $\varepsilon \in (0, \gamma^{K-1})$. Assume that there exists a sequence that preserves connectivity with the neighbors at step K, i.e. $\Gamma^k_{i,K}(\boldsymbol{u}_{i,K}) = 1$. Then for any agent i, given a sufficiently large budget T, the solution returned by OP contains at least K actions and does indeed preserve connectivity.*

Proof: The value of a solution that preserves connectivity at step K is at least $v_1 = \frac{\beta}{1-\gamma}$, while for a solution that does not it is at most $v_2 = \frac{1}{1-\gamma} - \beta\gamma^{K-1}$, since the β reward is not received at step K. We have:

$$v_1 - v_2 \geq \frac{\beta}{1-\gamma} - \frac{1}{1-\gamma} + \beta\gamma^{K-1} \geq \varepsilon$$

obtained by replacing the value of β. Therefore, the optimal value satisfies $v^* \geq v_1$, and as soon as the OP reaches depth $d+1$ for which $\frac{\gamma^d}{1-\gamma} < \varepsilon$, due to Theorem 3(i) it will return a solution that is closer than ε to v^* and which therefore preserves connectivity. For sufficiently large T, depth $\max\{d, K\}+1$ is reached which guarantees both that the precision is ensured and that the length of the solution at least K. The proof is complete.

Putting the local guarantees together, we have topology preservation for the entire system, as follows.

Theorem 5 *Take β and T as in Lemma 4, then under Assumption 2 and if the graph is initially connected, Algorithm 2 preserves the connections at any step $k = \ell K$.*

Proof: The intuition is very simple: each agent i will move so as to preserve connectivity with the *previous* state of any neighbor $j > i$, and then in turn j will move while staying connected with the *updated* state of i, which is what is required. However, since Assumption 2 requires connectivity to hold globally for all neighbors, the formal proof is somewhat technical.

To make it easier to understand, define relation $\mathcal{C}(i, j_1, \dots, j_{N_i})$, where indices j_l are all the neighbors $\mathcal{N}_{i,k}$ at k sorted in ascending order, and $N_i = |\mathcal{N}_{i,k}|$. This relation means that i is connected with all j_l, i.e. $\|x^{\mathrm{c}}_{i,k} - x^{\mathrm{c}}_{j_l,k}\| \leq P$, for $l = 1, \dots, N_i$. When some agents have superscript '+' in the relation, this means that the relation holds with their *updated* states after K steps.

Assume the agents are connected via edges $\mathcal{E}_k$ at step k, a multiple of K. We will show by induction that $\mathcal{C}(i^+, j_1^+, \dots, j_{l(i)}^+, j_{l(i)+1}, j_{N_i})$ where $l(i)$ is the last neighbor smaller than i. For the base case $i = 1$, we have $\mathcal{C}(1, j_1, \dots, j_{N_1})$ by the fact that $(1, j_l) \in \mathcal{E}_k$. Hence the conditions of Assumption 2 are satisfied and there exists some $\boldsymbol{u}_{1,K}$ that preserves connectivity with the previous states of all neighbors. By Lemma 4 the algorithm finds and applies such a sequence, which implies $\mathcal{C}(1^+, j_1, \dots, j_{N_1})$. For the general case, we have that $\mathcal{C}(i, j_1^+, \dots, j_{l(i)}^+, j_{l(i)+1}, \dots, j_{N_i})$ by simply looking at earlier cases stated where the first argument of relation $\mathcal{C}$ is $m = j_1, \dots, j_{l(i)}$ (they are earlier cases since $j_{l(i)} < i$). As above, this means the conditions of Assumption 2 and therefore Lemma 4 are satisfied for the *updated* states of $j_1^+, \dots, j_{l(i)}^+$, and therefore that $\mathcal{C}(i^+, j_1^+, \dots, j_{l(i)}^+, j_{l(i)+1}, \dots, j_{N_i})$ which completes the induction.

Take any $(i, j) \in \mathcal{E}_k$ for which $i > j$, which is sufficient since the graph is undirected. Then, $j \leq j_{l(i)}$ and the already shown relation $\mathcal{C}(i^+, j_1^+, \dots, j_{l(i)}^+, j_{l(i)+1}, \dots, j_{N_i})$ implies $(i, j) \in \mathcal{E}_{k+K}$. So all the links are preserved, and since the derivation holds for arbitrary k, they are preserved in closed loop.

Theorem 5 guarantees that the *topology* is preserved when the initial agent states correspond to a connected network. However, this result does not concern the stability of the *agreement*. In practice, we solve the agreement problem by choosing appropriately the rewards Δ, such as in (7), so that by maximizing the discounted returns the agents achieve agreement. In Section 5, we illustrate that this approach performs well in experiments. Note that Theorem 5 holds whether the graph is defined with (1) or (2).

It is also interesting to study the following result about the performance of OP. Consider some agent i at step k. Since we need to look into the details of OP for a single agent i at fixed step k, for readability we suppress these indices in the sequel, so we write $\rho_d(\boldsymbol{u}_d) = (1-\beta)\Delta_d(\boldsymbol{u}_d) + \beta\Gamma_d(\boldsymbol{u}_d)$ for reward function (5). We define two optimal control problems derived from this reward function. The first removes the connectivity constraint, so that $\rho_{d,\mathrm{u}}(\boldsymbol{u}_d) = (1-\beta)\Delta_d(\boldsymbol{u}_d) + \beta$. The second is the *agreement* (only) problem with $\rho_{d,\mathrm{a}}(\boldsymbol{u}_d) = \Delta_d(\boldsymbol{u}_d)$, i.e. for $\beta = 0$. Denote $v^*_\mathrm{u} = \sup_{\boldsymbol{u}_\infty} v_\mathrm{u}(\boldsymbol{u}_\infty)$ and $v^*_\mathrm{a} = \sup_{\boldsymbol{u}_\infty} v_\mathrm{a}(\boldsymbol{u}_\infty)$ where v_u and v_a are the discounted returns under the new reward functions.

We will compare performance in the original problem with that in the agreement problem.

Theorem 6 *Assume $v^* = v^*_\mathrm{u}$. For OP applied to the* original *problem, the near-optimality bounds of Theorem 3(ii) hold with the branching factor κ_a of the* agreement *problem.*

Proof: We start with a slight modification to the analysis of OP. For any problem, define the set:

$$\tilde{\mathcal{T}} = \{\boldsymbol{u}_d \mid d \geq 0, v^* \leq b(\boldsymbol{u}_d)\}$$

Note that $\tilde{\mathcal{T}} \subseteq \mathcal{T}^*$ of (4), since:

$$v(\boldsymbol{u}_d) + \frac{\gamma^d}{1-\gamma} \geq \nu(\boldsymbol{u}_d) + \frac{\gamma^d}{1-\gamma} = b(\boldsymbol{u}_d)$$

and so the condition in $\tilde{\mathcal{T}}$ implies the one in (4). Further, OP only expands nodes in $\tilde{\mathcal{T}}$, since in any tree considered, there always exists some sequence $\boldsymbol{u}$ with $b(\boldsymbol{u}) \geq v^*$ (e.g., the initial subsequence of an optimal sequence), and OP always expands a sequence that maximizes b.

Denote now $\tilde{\mathcal{T}}_\mathrm{u}$ and $\tilde{\mathcal{T}}_\mathrm{a}$ the corresponding sets for the unconstrained and agreement cases. Take a sequence $\boldsymbol{u}_d \in \tilde{\mathcal{T}}$, the set in the original problem. By assumption $v^* = v^*_\mathrm{u}$, and by construction $b(\boldsymbol{u}_d) \leq b_\mathrm{u}(\boldsymbol{u}_d)$, so $v^* \leq b(\boldsymbol{u}_d)$ implies $v^*_\mathrm{u} \leq b_\mathrm{u}(\boldsymbol{u}_d)$, and $\tilde{\mathcal{T}} \subseteq \tilde{\mathcal{T}}_\mathrm{u}$. Next, $v^*_\mathrm{u} = (1-\beta)v^*_\mathrm{a} + \frac{\beta}{1-\gamma}$ and $b_\mathrm{u}(\boldsymbol{u}_d) = \nu_\mathrm{u}(\boldsymbol{u}_d) + \frac{\gamma^d}{1-\gamma} = (1-\beta)\nu_\mathrm{a}(\boldsymbol{u}_d) + \frac{\gamma^d}{1-\gamma} = (1-\beta)b_\mathrm{a}(\boldsymbol{u}_d) + \frac{\beta\gamma^d}{1-\gamma}$. Replacing these in condition $v^*_\mathrm{u} \leq b_\mathrm{u}(\boldsymbol{u}_d)$, we obtain:

$$(1-\beta)v^*_\mathrm{a} + \beta\frac{1-\gamma^d}{1-\gamma} \leq (1-\beta)b_\mathrm{a}(\boldsymbol{u}_d)$$

which implies $v^*_{\mathrm{a}} \leq b_{\mathrm{a}}(\boldsymbol{u}_d)$, and so $\tilde{\mathcal{T}}_{\mathrm{u}} \subseteq \tilde{\mathcal{T}}_{\mathrm{a}}$.

Therefore, finally:

$$\tilde{\mathcal{T}} \subseteq \tilde{\mathcal{T}}_{\mathrm{u}} \subseteq \tilde{\mathcal{T}}_{\mathrm{a}} \subseteq \mathcal{T}^*_{\mathrm{a}}$$

Given budget T, the smallest possible depth reached by OP in the original problem is that obtained by exploring the set $\tilde{\mathcal{T}}$ uniformly, in the order of depth. Due to the inclusion chain above, this depth is at least as large as that obtained by exploring $\mathcal{T}^*_{\mathrm{a}}$ uniformly. The latter depth is $\Omega(\log T / \log \kappa_{\mathrm{a}})$ if $\kappa_{\mathrm{a}} > 1$, or else $\Omega(T)$. The bounds follow immediately as in the proof of Theorem 3.

Theorem 6 can be interpreted as follows. If the unconstrained optimal solution would have naturally satisfied connectivity (which is not unreasonable), adding the constraint does not harm the performance of the algorithm, so that flocking is as easy as solving only the agreement problem. This is a nice property to have.

4.2 A practical algorithm

Algorithm 2 has an important shortcoming in practice: it requires knowing a value of K for which Assumption 2 is satisfied. Further, keeping predictions constant for $j > i$ is safe, but conservative, since better predictions are usually available: those made by the neighbors at previous steps, which may not be expected to change much, e.g. when a steady state is being approached.

Next, we present a more practical variant that does not have these issues. It works in increments of 1 step (rather than K), and at step k, it forms the predictions for neighbors $j > i$ as follows: $\widehat{\boldsymbol{x}}^{i,k}_j = (x_{j,k}, x^{k-1}_{j,2}, ..., x^{k-1}_{j,d})$; Thus for the present step $x_{j,k}$ is used since it was already measured and exchanged, while for future steps the previously communicated trajectories are used.

Since K is unknown, the agent will try preserving connectivity at *every* step, with as many neighbors as possible:

$$\Gamma^k_{i,d}(\boldsymbol{u}_{i,d}) = \frac{1}{|\mathcal{N}_{i,k}|} \sum_{j \in \mathcal{N}_{i,k}} \begin{cases} 1 & \text{if } \|x^{k,\mathrm{c}}_{i,d} - \widehat{x}^{i,\mathrm{c}}_{j,d}\| \leq P \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

For the links, definition (1) is used, since old neighbors may be lost but the graph may still remain connected due to new neighbors. So the aim here is only connectivity, weaker than topology preservation. For the agreement component, (7) is employed. Algorithm 3 summarizes the resulting protocol for generic agent i.

Algorithm 3 OP flocking at agent i – practical variant.

```
set initial prediction $\boldsymbol{x}^{-1}_i$ to an empty sequence
for step $k = 0, 1, 2, \ldots$ do
    exchange state at $k$ with all neighbors $j \in \mathcal{N}_{i,k}$
    send $\boldsymbol{x}^{k-1}_i$ to $j < i$, receive $\boldsymbol{x}^{k-1}_j$ from $j > i$
    wait to receive new predictions $\widehat{\boldsymbol{x}}^{i,k}_j$ from all $j < i$
    form predictions $\widehat{\boldsymbol{x}}^{i,k}_j$ for $j > i$
    run OP with (5) and (8), obtaining $\boldsymbol{u}_i^k$ and $\boldsymbol{x}_i^k$
    send $\boldsymbol{x}_i^k$ to $j > i$
    execute action $u_{i,0}^k$
end for
```

The main advantage of our approach, in both Algorithm 2 and Algorithm 3, is the generality of the agent dynamics it can address. This generality comes at the cost of communicating sequences of states, introducing a dependence of the performance on the action discretization, and a relatively computationally involved algorithm. The time complexity of each individual OP application is between $O(T \log T)$ and $O(T^2)$ depending on κ. The overall complexity for all agents, if they run OP in parallel as soon as the necessary neighbor predictions become available, is larger by a factor equal to the length of the longest path from any i to any $j > i$. Depending on the current graph this length may be significantly smaller than the number of agents n.

5 Experimental results

The proposed method is evaluated in two problems with nonlinear agent dynamics. The first problem concerns flocking for a simple type of nonholonomic agents, where we also study the influence of the tuning parameters of the method. In the second experiment, full-state consensus for two-link robot arms is sought. This experiment illustrates that the algorithm can on the one hand handle rather complicated agent dynamics, and on the other hand that it also works for standard consensus on a fixed graph, even though our analytical focus was placed on the flocking problem.

While both types of agents have continuous-time underlying dynamics, they are controlled in discrete time, as is commonly done in practical computer-controlled systems. The discrete-time dynamics are then the result of integrating the continuous-time dynamics with zero-order-hold inputs. Then, in order for the analysis to hold for Algorithm 2, Assumption 2 must be satisfied by these discretized dynamics. Note that in practice we apply Algorithm 3, and the numerical integration technique introduces model errors that our analysis does not handle.

5.1 Flocking of nonholonomic agents

Consider homogeneous agents that evolve on a plane and have the state vector $x = [X, Y, v, \theta]$ with X, Y the position on the plane [m], v the linear velocity [m/s], and θ the orientation [rad]. The control inputs are the rate of change a

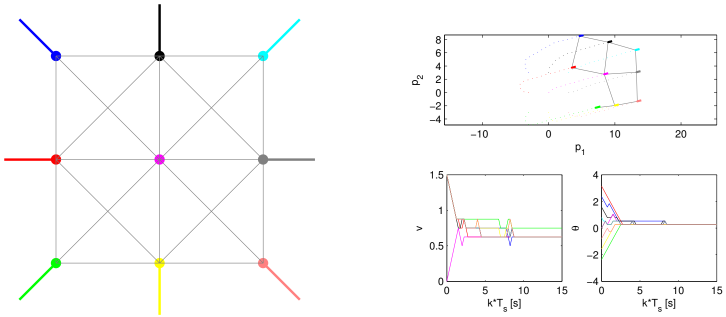

Figure 2: Results for nonholonomic agents. Top: initial configuration, with the agents shown as colored dots, their initial velocities and orientations symbolized by the thick lines, and their initial graph with thin gray lines. Middle: trajectories on the plane, also showing the final configuration of the agents. Bottom: evolution of agreement variables.

of the velocity and ω of the orientation. The discrete-time dynamics are:

$$\begin{aligned} X_{k+1} &= X_k + T_{\rm s} v_k \cos\theta_k \\ Y_{k+1} &= Y_k + T_{\rm s} v_k \sin\theta_k \\ v_{k+1} &= v_k + T_{\rm s} a_k \\ \theta_{k+1} &= \theta_k + T_{\rm s} \omega_k \end{aligned}$$

where Euler discretization with sampling time $T_{\rm s}$ was employed. The aim is to agree on $x^{\rm a} = [v, \theta]^\top$, which represent the velocity vector of the agent, while maintaining connectivity on the plane by keeping the distances between the connectivity states $x^{\rm c} = [X, Y]^\top$ below the communication range P.

The specific multiagent system we experiment with consists of 9 agents initially arranged on a grid with diverging initial velocities, see Figure 2, top. Their initial communication graph has some redundant links. In the reward function, $\beta = 0.5$ so that agreement and connectivity rewards have the same weight, and the agreement reward is (7) with the distance measure being a 2-norm weighted so that it saturates to 1 at a distance 5 between the agreement states. The range is $P = 5$. The sampling time is $T_{\rm s} = 0.25\,$s.

Figure 2 shows that the OP method preserves connectivity while achieving flocking, up to errors due mainly to the discretized actions. The discretized action set was $\{-0.5, 0, 0.5\}$ m/s$^2 \times \{-\pi/3, 0, \pi/3\}$ rad/s, and the planning budget of each agent is $T = 300$ node expansions. For all the experiments, the discount factor γ is set to 0.95, so that long-term rewards are considered with significant weight.

Next, we study the influence of the budget T and a *cutoff length* L for the communicated state predictions, a crucial parameter for the communication requirements of the algorithm. With a finite L, even if OP provides a longer sequences of predicted states, only the first L values are communicated

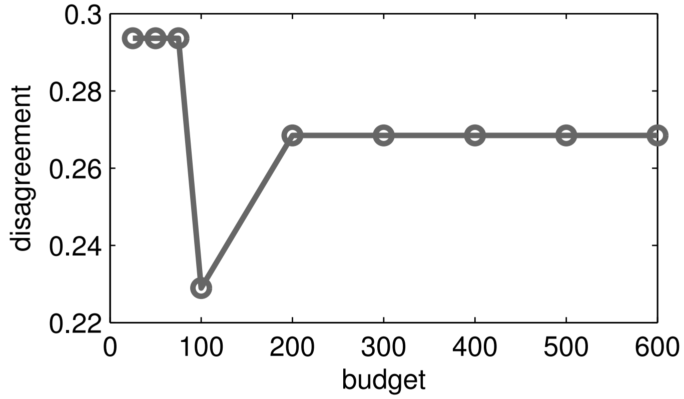

Figure 3: Influence of the expansion budget.

to the neighbors, and they set subsequent state predictions constant at the last known values. To characterize performance in each experiment with a single number, a mean inter-agent disagreement is computed at every step: $\delta_k = \frac{2}{n(n-1)} \sum_{i<j} \|x^{\mathrm{a}}_{i,k} - x^{\mathrm{a}}_{j,k}\|$, and the average of δ_k across all steps in the trajectory is reported.

The following budgets are used: $T = 25, 50, 75, 100, 200, \dots, 600$, and the length of the predictions is not limited. As shown in Figure 3 and as expected from the theoretical guarantees of OP, disagreement largely decreases with T although the decrease is not monotonic. [1]

The influence of the prediction length is studied for fixed $T = 300$, by taking $L = 0, 1, 3, 4$ and then allowing full predictions.[2] Figure 4 indicates that performance is not monotonic in L, and medium-length predictions are better in this experiment. While it is expected that too long predictions will not increase performance since they will rarely be actually be followed, the good results for communicating just the current state without any prediction are more surprising, and need to be studied further.

5.2 Consensus of robotic arms

Consider next the full-state consensus of two-link robotic arms operating in a horizontal plane. The state variables for each agent are the angles and angular velocities of the two links, $x_i = [\theta_{i,1}, \dot{\theta}_{i,1}, \theta_{i,2}, \dot{\theta}_{i,2}]$, and the agreement variables comprise the entire state, $x^{\mathrm{a}}_i = x_i$, without a connectivity state or reward component. The actions are the torques of the motors actuating the two links $u_i = [\tau_{i,1}, \tau_{i,2}]$. The model is standard so we omit the details and just note that the sampling time is $T_{\mathrm{s}} = 0.05\,\mathrm{s}$; the other parameters can be found in [25]. Ap-

[1] See [24], footnote 4 for an example showing how nonmonotonicity can happen.

[2] In effect, predictions with this budget do not exceed length 4 so the last two results will be identical.

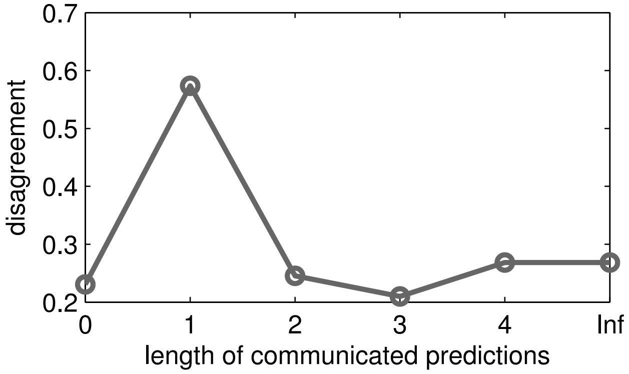

Figure 4: Influence of the maximal prediction length ("Inf" means it is not limited).

plications of this type of consensus problem include decentralized manipulation and teleoperation.

Three robots are connected on a fixed undirected communication graph in which robot 1 communicates with both 2 and 3, but 2 and 3 are not connected. The initial angular positions are taken random with zero initial velocities, see Figure 5. The distance measure is the squared Euclidean distance, weighted so that the angular positions are given priority. The discretized actions are $\{-1.5, 0, 1.5\}$ Nm × $\{-1, 0, 1\}$ Nm, and the budget of each agent is $T = 400$. Consensus is achieved without problems.

6 Conclusions

We have provided a flocking technique based on optimistic planning (OP), which under appropriate conditions is guaranteed to preserve the connectivity topology of the multiagent system. A practical variant of the technique worked well in simulation experiments.

An important future step is to develop guarantees also on the agreement component of the state variable. This is related to the stability of the near-optimal control produced by OP, and since the objective function is discounted such a stability property is a big open question in the optimal control field [26]. It would also be interesting to apply optimistic methods to other multiagent problems such as gossiping or formation control.

References

[1] Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

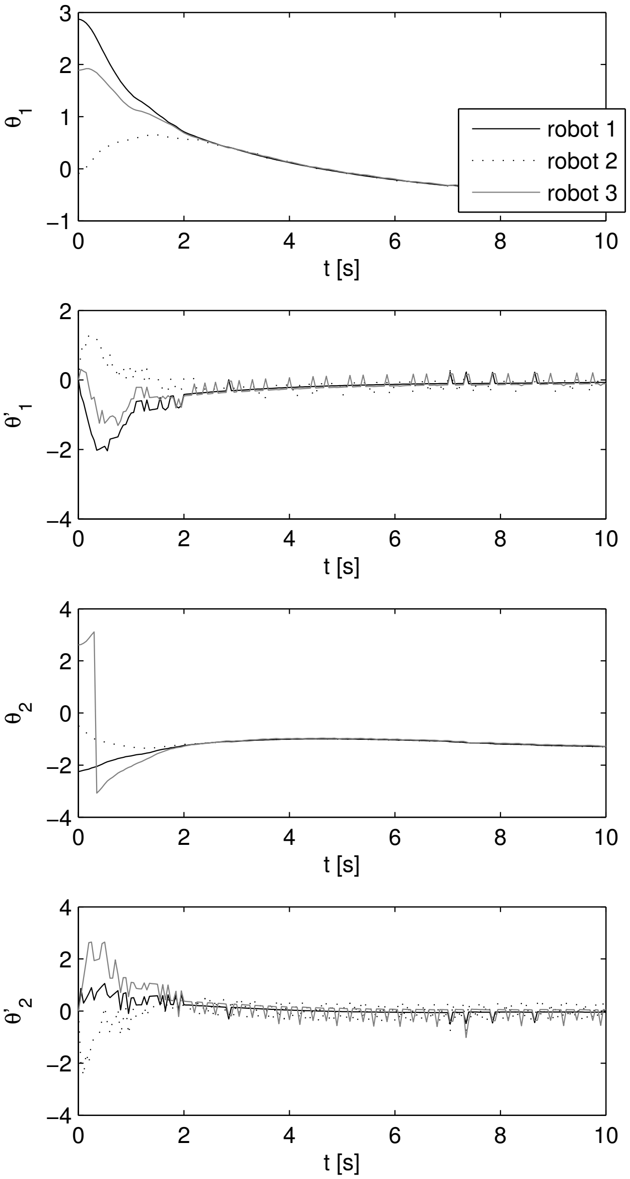

Figure 5: Leaderless consensus of multiple robotic arms: angles and angular velocities for the two links, overimposed for all the robots. Angles wrap around in the interval $[-\pi, \pi)$.

[2] Wei Ren and Randal W. Beard. *Distributed Consensus in Multi-Vehicle Cooperative Control: Theory and Applications.* Communications and Control Engineering. Springer, 2008.

[3] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[4] H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, 52(5):863–868, 2007.

[5] Wenjie Dong. Flocking of multiple mobile robots based on backstepping. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41(2):414–424, 2011.

[6] Jean-François Hren and Rémi Munos. Optimistic planning of deterministic systems. In *Proceedings 8th European Workshop on Reinforcement Learning (EWRL-08)*, pages 151–164, Villeneuve d'Ascq, France, 30 June – 3 July 2008.

[7] Claudio De Persis and Paolo Frasca. Robust self-triggered coordination with ternary controllers. *IEEE Transactions on Automatic Control*, 58(12):3024–3038, 2013.

[8] Jie Mei, Wei Ren, and Guangfu Ma. Distributed coordinated tracking with a dynamic leader for multiple Euler-Lagrange systems. *IEEE Transactions on Automatic Control*, 56(6):1415–1421, 2011.

[9] M.M. Zavlanos and G.J. Pappas. Distributed connectivity control of mobile networks. *IEEE Transactions on Robotics*, 24(6):1416–1428, 2008.

[10] M. Fiacchini and I.-C. Moruarescu. Convex conditions on decentralized control for graph topology preservation. *IEEE Transactions on Automatic Control*, 59(6):1640–1645, 2014.

[11] F. Bullo, J. Cortés, and S. Martinez. *Distributed Control of Robotic Networks. A Mathematical Approach to Motion Coordination Algorithms.* Princeton University Press, 2009.

[12] Jiandong Zhu, Jinhu Lu, and Xinghuo Yu. Flocking of multi-agent nonholonomic systems with proximity graphs. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(1):199–210, 2013.

[13] Housheng Su, Guanrong Chen, Xiaofan Wang, and Zongli Lin. Adaptive second-order consensus of networked mobile agents with nonlinear dynamics. *Automatica*, 47(2):368–375, 2011.

[14] Jin Zhou, Xiaoqun Wu, Wenwu Yu, Michael Small, and Junan Lu. Flocking of multi-agent dynamical systems based on pseudo-leader mechanism. *Systems & Control Letters*, 61(1):195–202, 2012.

[15] Herbert Tanner, Ali Jadbabaie, and George Pappas. Flocking in teams of nonholonomic agents. In Vijay Kumar, Naomi Leonard, and A. Morse, editors, *Cooperative Control*, volume 309 of *Lecture Notes in Control and Information Sciences*, pages 458–460. Springer, 2005.

[16] Lucian Buşoniu and Constantin Moraرescu. Optimistic planning for consensus. In *Proceedings American Control Conference 2013 (ACC-13)*, Washington, DC, 17–19 June 2013.

[17] Lucian Buşoniu and Constantin Morarescu. Consensus for black-box nonlinear agents using optimistic optimization. *Automatica*, 50(4):1201–1208, 2014. .

[18] Bronislaw Jakubczyk and Eduardo D. Sontag. Controllability of nonlinear discrete-time systems: A lie-algebraic approach. *SIAM Journal of Control and Optimization*, 28:1–33, 1990.

[19] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.

[20] Frank Lewis and Derong Liu, editors. *Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control.* Wiley, 2012.

[21] Remi Munos. The optimistic principle applied to games, optimization and planning: Towards foundations of Monte-Carlo tree search. *Foundations and Trends in Machine Learning*, 7(1):1–130, 2014.

[22] Rudy R. Negenborn, Bart De Schutter, and Hans Hellendoorn. Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. *Engineering Applications of Artificial Intelligence*, 21(3):353–366, 2008.

[23] Jinfeng Liu, Xianzhong Chen, David Munoz de la Peña, and Panagiotis D. Christofides. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. *American Institute of Chemical Engineers (AIChE) Journal*, 56(8):2137–2149, 2010.

[24] Lucian Buşoniu, Rémi Munos, and Robert Babuška. A review of optimistic planning in Markov decision processes. In Frank Lewis and Derong Liu, editors, *Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control.* Wiley, 2012.

[25] Lucian Buşoniu, Damien Ernst, Bart De Schutter, and Robert Babuška. Approximate dynamic programming with a fuzzy parameterization. *Automatica*, 46(5):804–814, 2010.

[26] Bahare Kiumarsi, Frank Lewis, Hamidreza Modares, Ali Karimpour, and Mohammad-Bagher Naghibi-Sistani. Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. *Automatica*, 50(4):1167–1175, 2014.