Consensus for Black-Box Nonlinear Agents Using Optimistic Optimization

Lucian Busoniu [a]   Irinel-Constantin Morarescu [b]

[a] *Automation Department, Technical University of Cluj-Napoca, Memorandumului 28, 400114 Cluj-Napoca, Romania*

[b] *Université de Lorraine, CRAN, UMR 7039 and CNRS, CRAN, UMR 7039, 2 av. Forêt de Haye, Vandœuvre-lès-Nancy, France*

Abstract

An important problem in multiagent systems is consensus, which requires the agents to agree on certain controlled variables of interest. We focus on the challenge of dealing in a generic way with nonlinear agent dynamics, represented as a black-box with unknown mathematical form. Our approach designs a reference behavior with a classical consensus method. The main novelty is using optimistic optimization (OO) to find controls that closely follow the reference behavior. The first advantage of OO is that it only needs to sample the black-box model of the agent, and so achieves our goal of handling unknown nonlinearities. Secondly, a tight relationship is guaranteed between computation invested and closeness to the reference behavior. Our main results exploit these properties to prove practical consensus. An analysis of representative examples builds additional insight and shows that in some nontrivial problems the optimization is easy to solve by OO. Simulations on these examples accompany the analysis.

Key words: Multiagent systems; consensus; optimistic optimization; nonlinear systems.

1 Introduction

Multi-agent systems have applications in a wide variety of domains such as robotic teams, energy and telecommunication networks, collaborative decision support systems, data mining, etc. Each agent typically has only a local, limited view of the overall system, which means decentralized control approaches are necessary. Requirements on the coherent behavior of the agents are often expressed in terms of *consensus*, in which the agents must reach agreement on controlled variables of interest [21,25,20].

Classical consensus algorithms are designed for agents with linear dynamics, for which the behavior is well understood. In this setting, the literature considers fixed and time-varying communication topologies [24,16], directed or undirected graphs [21,25], synchronous or asynchronous information exchange [31,5], delayed or immediate transmissions [19,15], etc. Consensus approaches also exist for nonlinear agent dynamics, such as second-order systems with nonlinear acceleration dynamics [28,35], nonholonomic robots [30], and Euler-Lagrange dynamics [14]. These works usually require an explicit mathematical model of the agents, the form of which is exploited to derive tailored control laws (often via Lyapunov synthesis).

Our goal here is different: a generic consensus method, applicable without changes to a wide class of nonlinear agents. To achieve this, we propose an approach that only requires black-box models [27,29] of the agents, thereby eliminating the dependence on the particular mathematical form of their dynamics. Neural networks are a typical category of black-box models, e.g. [9,13], but nonstandard models like computer programs are also allowed; such models are common e.g. in planning [4].

A first ingredient of the approach is a classical consensus algorithm, used to design reference next states for the agents. The core novelty is the use of *optimistic optimization* (OO) [17] to solve the control problem of reaching

---

close to these states, formulated as optimal predictive control. OO allows our approach to deal with highly general, black-box dynamics, since it performs global nonconvex optimization by only simulating the dynamics for some inputs. Furthermore, the properties of OO allow a tight characterization of the relation between closeness to the reference states and computation invested, and in particular guarantee arbitrarily small errors as computation increases. Exploiting all this, our main results prove practical consensus, i.e. that the agents converge close to each other, as follows.

The main technical requirement is that each agent is controllable in a small number of K steps. The communication graph is connected, possibly directed, and to keep the focus on the main challenge of general nonlinear agents, we do not consider for now time-varying graphs, noise, or network effects such as delays or packet dropouts. Then, under some additional technical assumptions related to optimization, our analysis ensures that the agents converge to a small region and then return to this region once every K steps. Under stronger regularity conditions including invertibility, we further guarantee the agents remain close to each other at *every* step. A measure of problem complexity called near-optimality dimension [17] modulates the relationship between computation and approximation error. We analyze this complexity measure and build insight into the meaning of the assumptions for the entire class of linear systems, as well as for a nonlinear example; and provide simulation results for both cases.

Note that, given controllability and invertibility, *if* the specific form of the dynamics were known they could be feedback-linearized before designing the consensus controls, like in [23]. However, here the model is a black box and its form is unknown, so explicit linearization is impossible. Nevertheless, our approach does obtain an approximately linear overall behavior (at multiples of K), similarly in this sense to [23]; it is the job of OO to find the controls achieving this behavior. The approach is also related to model-predictive control, which was applied to the consensus of linear agents in [12,6]. Our existing work includes [2], where we gave a different approach without analysis; and [1], which is a preliminary version of the present work. With respect to [1], here we introduce novel analysis that includes: using the deterministic OO algorithm, handling directed graphs, and a detailed investigation of the computational aspects; while the examples are also new.

Next, we formalize the problem in Section 2, introduce OO in Section 3, and give our approach in Section 4. Section 5 provides the examples and Section 6 concludes.

2 Consensus problem

We consider a set of M agents with decoupled nonlinear dynamics $x_{i,k+1} = f_i(x_{i,k}, u_{i,k})$, $i = 1, \ldots, M$. The dimensionality of x (the order of the dynamical system f), denoted n, is the same for every agent, and we have $x_i \in \mathbb{R}^n$. Similarly, for every agent u_i belongs to a compact set U. Define also the collective state $\boldsymbol{x} = [x_1^\top, \cdots, x_M^\top]^\top \in \mathbb{R}^{Mn}$, containing the states of all the agents. The ideal goal is to achieve full-state consensus:

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

where $\|\cdot\|$ is the Euclidean 2-norm, here as well as in the sequel. We employ the more pragmatic requirement of practical consensus, defined as the existence, for any given disagreement $\Delta > 0$, of a finite step k_0 so that for all $k \geq k_0$, $\|x_{i,k} - x_{j,k}\| \leq \Delta$ for all i, j.

An agent can receive information only from its neighbors on an interconnection graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The set of nodes $\mathcal{V} = \{1, \ldots, M\}$ represents the agents, and the edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are the communication links. Denote by $\mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}\}$ the set of neighbors of node i. The graph may be directed (for some edges $(i, j) \in \mathcal{E}$, $(j, i) \notin \mathcal{E}$), but we require that it is time-invariant and strongly connected. The latter property means a path exists between any pair of nodes i, j: a sequence of nodes $i = i_1, \ldots, i_L = j$ so that $(i_l, i_{l+1}) \in \mathcal{E}, 1 \leq l < L$.

Let $X_0 \subset \mathbb{R}^n$ be a user-defined, bounded set of interesting initial agent states, from which consensus might be required. Let $\mathbf{B}(x_c, R)$ be the smallest Euclidean ball containing X_0, which has center x_c and radius R. Then, define an enlarged set $\mathcal{X} = \mathbf{B}(x_c, 3R + \Delta^+)$, where $\Delta^+ > 0$ is an arbitrarily small constant. Finally, denote by $\tilde{u}_{i,k} = [u_{i,k}^\top, \ldots, u_{i,k+K-1}^\top]^\top$ a sequence of K actions for agent i starting at step k, and by $\tilde{f}_i(x_{i,k}, \tilde{u}_{i,k})$ the resulting state of the agent after applying $\tilde{u}_{i,k}$ (K steps later). We impose a controllability assumption on f_i in the set $\mathcal{X}$.

Assumption 1 (Controllability) *There exists a finite K such that, for any $x, x^* \in \mathcal{X}$ and any agent i, there exists a sequence $\tilde{u}_i \in U^K$ so that:*

$$\tilde{f}_i(x, \tilde{u}_i) = x^*$$

Remarks: This type of controllability property is thoroughly studied by e.g. [11], where Lie-algebraic conditions are provided for nonlinear systems, including a characterization of the size of reachable sets. The controllable set is enlarged to $\mathcal{X}$ to deal with the errors made by the algorithm in some special, unfavorable initial conditions, such as when most agents are close together at one edge of X_0 and one agent is diametrally opposite. In fact, for most initial conditions, we expect the agent to stay close to the convex hull of X_0. □

Under this assumption and some other requirements related to the optimization algorithm, our results guarantees practical consensus at steps that are multiples of K. To guarantee the stronger result of practical consensus at every step, an additional assumption is imposed. Define an *invertibility* property of dynamics $\tilde{f}_i$, which requires that for every pair $(x, x^*) \in \mathcal{X}^2$ the sequence $\tilde{u}_i$ achieving $\tilde{f}_i(x, \tilde{u}_i) = x^*$ is unique. The *inverse* $\tilde{f}_i^{-1} : \mathcal{X}^2 \to U^K$ is defined as the mapping between state pairs and their corresponding action sequences.

Assumption 2 (Homogeneity and invertibility) *The multiagent system is homogeneous: $f_i = f_j =: f$, $\forall i, j$ with common dynamics f that are Lipschitz-continuous in $\mathcal{X} \times U$. Furthermore, the K-step dynamics $\tilde{f}$ has a unique inverse $\tilde{f}^{-1}$, which is also Lipschitz continuous in its domain $\mathcal{X}^2$.*

Remarks: Invertibility conditions are an important topic in nonlinear system theory, and several types of conditions for discrete-time systems are provided e.g. by [8,34,7]. Since the set of states $\mathcal{X}$ and the action space U were already taken compact, the Lipschitz conditions are not significantly more restrictive. Homogeneity (together with invertibility) is needed to guarantee the agents behave similarly for similar desired transitions, see Section 5.2 for a simulation illustrating this. □

It is essential to note that the dynamics f_i are only available to the agents as black-box, simulation models, the inverse dynamics $\tilde{f}^{-1}$ (if they exist) are not available at all, and the sequences $\tilde{u}_i$ of Assumption 1 are unknown; indeed, the main task of our algorithm will be to approximate them.

3 Background: Optimistic optimization

Our method will use a predictive approach that finds the control actions with some recent global optimization methods, called *optimistic optimization*. Two such methods are used, deterministic and simultaneous optimistic optimization – DOO and SOO. They have some essential advantages in our framework: they only require to sample the black-box agent model, and they provide a directly implementable procedure with a tight analysis of the relation between computation invested and near-optimality.

For now, we introduce DOO and SOO generically, following [17], for the maximization of any function $r : U \to \mathbb{R}$. We reuse notations u and U, while keeping in mind the framework remains generic in this section. The optimization proceeds by hierarchically partitioning the domain U. This partitioning procedure is an input into the algorithms, and is represented by a tree structure $\mathcal{T}$ in which each node (d, j) is labeled by a point $u_{d,j}$ and represents a subset of U denoted $U_{d,j} \ni u_{d,j}$. Here, $d \geq 0$ is the depth in the tree and j is the node index at a given depth. The root of the tree represents the entire domain U, and the tree is defined so that the children of a node form a partition of the set represented by their parent. The collection of leaves of the currently explored tree is denoted by $\mathcal{L}$, and the leaves at depth d by $\mathcal{L}_d$.

The core requirements of the algorithms are stated in terms of a semimetric $\ell : U \times U \to [0, \infty)$ (a semimetric has all the properties of a metric except perhaps the triangle inequality).

Assumption 3 (Optimization) *The objective function and the partitioning satisfy the following conditions:*

3.i There exists an optimum u^* so that:

$$r(u^*) - r(u) \leq \ell(u, u^*) \quad \forall u \in U \tag{1}$$

3.ii There exist $c > 0$ and $\gamma \in (0,1)$ such that for any d, $\delta_{d,j} \leq c\gamma^d$ for all nodes j at depth d, where $\delta_{d,j} := \sup_{u \in U_{d,j}} \ell(u_{d,j}, u)$, a pseudo-diameter of $U_{d,j}$.

3.iii There exists a constant ν such that any subset $U_{d,j}$ contains a ball with center $u_{d,j}$ and radius $\nu c\gamma^d$ in the semimetric ℓ.

Remarks: Intuitively, **??**.i is a local, one-sided Lipschitz property. In our upcoming consensus algorithm, it does not restrict the dynamics much more than the Lipschitz requirement of Assumption 2. The other two conditions define the properties of a good partitioning procedure, which should provide well-shaped sets (**??**.iii) that shrink with the depth in the tree (**??**.ii).[1] □

Denote $\delta(d) = c\gamma^d$, the maximal diameter at depth d. DOO works by always partitioning further a set that is likely to contain the optimum of r. It does this by assigning upper bounds to all leaf sets $U_{d,j}, (d,j) \in \mathcal{L}$:

$$b(U_{d,j}) = r(u_{d,j}) + \delta(d) \tag{2}$$

so that $b(U_{d,j}) \geq r(u), \ \forall u \in U_{d,j}$. Then at each iteration, an *optimistic* leaf set, which maximizes the upper bound, is expanded (partitioned into its children sets). After n such expansions, DOO returns the sample with the largest value $r(u_{d,j})$.

DOO assumes knowledge of ℓ by using $\delta(d)$ in the upper bounds. The alternative, SOO, does not require this knowledge. Instead, at each round, SOO simultaneously expands all potentially optimal leaf sets: those for which the upper bound could be the largest under *any* semimetric ℓ. With a little thought, a set can only contain the largest upper bound if its sample value is at least as good as the values of all sets with diameters larger than its own; we say that the set is not dominated by larger sets. Since further, $\delta(d)$ decreases with d, we only have to compare with leaves higher up the tree. At each iteration t, the algorithm expands at most one leaf set at each depth. If we define $\mathcal{L}_{\leq d}$ as the set of leaf nodes having depths $d' \leq d$, then a leaf (d,j) is only expanded if $r(u_{d,j}) = \max_{(d',j') \in \mathcal{L}_{\leq d}} r(u_{d',j'})$; if there are several such leaves at d, one is chosen arbitrarily. SOO additionally limits the tree depth at each iteration t with a function $d_{\max}(t)$, a parameter of the algorithm that controls the tradeoff between deeper or more uniform exploration. A typical schedule is $d_{\max}(t) = t^a$ with $a \in (0,1)$. For clarity, Algorithm 1 summarizes SOO.

Algorithm 1 Simultaneous Optimistic Optimization

Input: function r, depth function $d_{\max}$,
      budget N, partitioning of U
1: initialize $\mathcal{T} \leftarrow U_{0,0}$; $t \leftarrow 1$
2: **loop**
3:     $r_{\max} \leftarrow -\infty$
4:     **for** $d = 0$ to $\min(\text{depth}(\mathcal{T}), d_{\max}(t))$ **do**
5:         $j^\dagger \leftarrow \arg\max_{j \in \mathcal{L}_d} r(u_{d,j})$
6:         **if** $r(u_{d,j^\dagger}) \geq r_{\max}$ **then**
7:             expand $(d, j^\dagger)$, add children to $\mathcal{T}$
8:             $r_{\max} \leftarrow r(u_{d,j^\dagger})$
9:             $t \leftarrow t + 1$
10:           **if** $t > N$ **then** stop the algorithm
11:           **end if**
12:         **end if**
13:     **end for**
14: **end loop**
Output: $\widehat{u} = \arg\max_{(d,j) \in \mathcal{T}} r(u_{d,j})$

The convergence of DOO and SOO depends on how smooth the function is in the semimetric ℓ. To formalize this, define first the near-optimal sets $U_\varepsilon = \{u \in U \mid r(u^*) - r(u) \leq \varepsilon\}$. For any ε, the packing number of U_ε is defined as the maximal number of disjoint ℓ-balls with centers in U_ε and equal radii $\nu\varepsilon$ (recall ν from Assumption **??**.iii). Finally, the *near-optimality dimension* is the smallest $\beta > 0$ so that there exists a positive constant C, such that the packing number is smaller than $C\varepsilon^{-\beta}$. The optimization problem is easier to solve when the semimetric ℓ captures more closely the smoothness of the objective function r around u^*, in which case β is smaller, with the ideal β being

---

[1] Note that the guarantees can be generalized to the case of diameters that decrease other than exponentially.

0. In some contrived cases β may be infinite, e.g. when r is constant and a 'malicious' semimetric is defined that shrinks superexponentially around u^*. In order to eliminate these edge cases, a final, technical assumption is needed.

Assumption 4 *A finite β exists.*

Then, the following results hold [17].

Proposition 1 (Near-optimality) *DOO returns a solution $\widehat{u}$ satisfying:*[2]

$$r(u^*) - r(\widehat{u}) = \begin{cases} O(N^{-1/\beta}) & \text{if } \beta > 0 \\ O(\gamma^{N/C}) & \text{if } \beta = 0 \end{cases}$$

where C is the constant from the near-optimality dimension. SOO returns a solution $\widehat{u}$ satisfying:

$$r(u^*) - r(\widehat{u}) = \begin{cases} O(N^{-(1-a)/\beta}) & \text{if } \beta > 0 \\ O(\gamma^{\sqrt{N}/C'}) & \text{if } \beta = 0 \end{cases}$$

where $C' \geq 1$ is a constant, and we choose $d_{\max}(t) = t^a$ (with $a > 0$) when $\beta > 0$ and $d_{\max}(t) = \sqrt{t}$ when $\beta = 0$.

Thus, sub-optimality decreases polynomially with N, depending on problem complexity as expressed by β; and when $\beta = 0$, the decrease is faster – exponential, confirming that the problem is 'easy' and solved efficiently by the algorithm. Note that DOO and SOO have similar guarantees. Since it does not use the semimetric, SOO must expand more sets, and it converges at a somewhat slower rate. However, for the same reason, SOO has a surprising advantage: it converges at the fastest rate allowed by any valid semimetric.

4 Consensus for black-box nonlinear agents

Our approach to the consensus problem in Section 2 works as follows. Once every K steps, each agent i computes a reference (desired) next state, which is a weighted average of the current states of its neighbors $\mathcal{N}_i$. Then, a sequence of actions is optimized with OO so that after K steps, the reference state is approached as closely as possible. All agents apply their sequences and the whole process is repeated. The main idea is therefore intuitive: by emulating a classical consensus algorithm [21], an approximately linear overall agent behavior is obtained, similarly to [23]. The main novelty is using OO to find the controls, which only requires sampling a black-box model, without knowledge about its internal structure. The properties of OO tightly connect computation invested with approximation error. In our analysis, we focus on these computational aspects and on the impact of the error on the consensus guarantees.

4.1 Algorithm

At step k – a multiple of K – each agent i computes a reference state with:

$$x^*_{i,k+K} = \sum_{j=1}^{M} p_{i,j} x_{j,k} \tag{3}$$

$$\text{where } p_{i,j} \in [0,1] \; \forall i,j, \quad \sum_{j=1}^{M} p_{i,j} = 1 \; \forall i,$$

$$p_{i,j} \neq 0 \text{ iff } i = j \text{ or } (i,j) \in \mathcal{E}$$

Since each agent only uses information about its neighbors, these reference states can be computed in a decentralized way. Matrix $P = [p_{i,j}]$ should be freely chosen to impose a desired consensus behavior. Several decentralized strategies to select it are given in [19,33,22].

Then, each agent solves with DOO or SOO the optimal control problem:

$$\tilde{u}^*_{i,k} = \arg\max_{\tilde{u}_{i,k}} -\|\tilde{f}_i(x_{i,k}, \tilde{u}_{i,k}) - x^*_{i,k+K}\| \tag{4}$$

---

[2] 'Big-O' notation $g = O(h)$ means that g asymptotically grows at most as fast as h.

(recall the definitions of K-step actions and dynamics). This is a type of dynamic inversion control, see e.g. [10], and similar ideas are widely used among others in adaptive inverse control [32]. A near-optimal extended action $\widehat{\tilde{u}}_{i,k}$ is found, which will reach close to the reference state. Algorithm 2 summarizes the procedure.

Algorithm 2 Nonlinear consensus at agent i

Input: row P_i of P, budget N, partitioning of U,
    OO algorithm to use, if DOO: ℓ, if SOO: $d_{\max}$
1: $k \leftarrow 0$
2: **loop**
3:     measure $x_{i,k}$ and send it to $j \in \mathcal{N}_i$
4:     receive $x_{j,k}$ from $j \in \mathcal{N}_i$
5:     compute reference state $x^*_{i,k+K}$ using P_i
6:     $r_{i,k}(\tilde{u}_{i,k}) := -\|\tilde{f}_i(x_{i,k}, \tilde{u}_{i,k}) - x^*_{i,k+K}\|$
7:     **if** using DOO **then**
8:         $\widehat{\tilde{u}}_{i,k} \leftarrow \text{DOO}(r_{i,k}, \ell, N, \text{partitioning of } U)$
9:     **else if** using SOO **then**
10:        $\widehat{\tilde{u}}_{i,k} \leftarrow \text{SOO}(r_{i,k}, d_{\max}, N, \text{partitioning of } U)$
11:     **end if**
12:     apply $\widehat{\tilde{u}}_{i,k}$ in open loop
13:     $k \leftarrow k + K$
14: **end loop**

While the number of parameters of OO may seem large, most can be chosen in a standard way. The partitioning of U is typically defined as a collection of hyperboxes in a Km-dimensional space, where m denotes the number of action variables. A large hyperbox contains the whole action space, and each node expansion corresponds to splitting the parent hyperbox into 2^{Km} subboxes, in half along all dimensions. Since this leads to exponential complexity in K, a practical requirement of our method is that K *must be small*; the limit depends on the agent's computational power, but to give an idea, say that the product Km should be less 10 (Section 6 points out how future extensions might avoid this).

For DOO, the semimetric ℓ is a less obvious choice. For some problems, its form may be available from prior knowledge. In general, defining ℓ can be difficult, in which case SOO should be used. In SOO, a default choice of $d_{\max}(t)$ is $\sqrt{t}$, expecting that a semimetric that yields a near-optimality dimension of 0 exists; if the optimization is expected to be difficult, a better schedule is t^a with small a, to increase the asymptotic convergence rate, see Proposition 1. We provide more insight about the semimetric and near-optimality dimension in the examples. The main parameter is however N, and we dedicate the next section to analyze its effect.

4.2 Computational considerations

Simulating the nonlinear dynamics will usually dominate computation, making the time complexity at each agent roughly linear in the budget N.

A more interesting question is what value of N is required to achieve a uniformly small approximation error ε^*. By essentially inverting the relationships in Proposition 1, we find the following budgets at agent i in collective state $\boldsymbol{x}$, where the optimization complexity is characterized by $\beta(i, \boldsymbol{x})$. In DOO, $N = O(\varepsilon^{*-\beta(i,\boldsymbol{x})})$, polynomial in $1/\varepsilon^*$, when $\beta(i, \boldsymbol{x})$ is positive. When $\beta(i, \boldsymbol{x}) = 0$, $N = O(\log(1/\varepsilon^*))$, logarithmic (much lower). In SOO, taking for simplicity $d_{\max}(t) = \sqrt{t}$, $N = O(\varepsilon^{*-2\beta(i,\boldsymbol{x})})$ when $\beta(i, \boldsymbol{x}) > 0$ and $N = O(\log^2(1/\varepsilon^*))$ when $\beta(i, \boldsymbol{x}) = 0$. So the complexity behaves similarly but is larger – the polynomial is of double degree (which can be improved by better choosing $d_{\max}$), while the logarithm is squared.

In practice, the agents will use the same N everywhere. In that case, any desired ε^* can be achieved by taking a sufficiently large N, which is related to the largest β (of course, this large N is conservative where β is smaller, leading to errors smaller than ε^* in those situations). This uniform error property is essential to our analysis below.

4.3 Consensus guarantees

Theorem 2 (Consensus every K steps) *Under Assumptions 1, 3, and 4, for large enough N, Algorithm 2 achieves practical consensus at steps multiple of K.*

Proof: Denote by Λ the left eigenvector of P associated with the eigenvalue 1 (which will always exist due to the structure of P), $\bar{x} = \Lambda^\top \boldsymbol{x}_0 \in \mathbb{R}^n$, and $\bar{\boldsymbol{x}} = [\bar{x}, \dots, \bar{x}]^\top$. We will first show (i) that $\exists l_0, \Delta$ so that for all $l \geq l_0$, $\|x_{i,lK} - \bar{x}\| \leq \Delta$ for all i, while assuming that the agents remain controllable. Then, it is proven that (ii) they actually do remain controllable.

(i) P is a row stochastic matrix associated to a strongly connected graph. By iterating the ideal protocol $x^*_{i,k+K} = \sum_{j=1}^M p_{i,j} x^*_{j,k}$ starting from $\boldsymbol{x}_0$, all agents would converge exponentially to $\bar{x}$. Next, the effect of the deviation of the real states from this ideal trajectory is analyzed. Let λ be the largest eigenvalue of P that is different from 1. Then, $0 < |\lambda| < 1$ and:

$$\|\boldsymbol{x}^*_{(l+1)K} - \bar{\boldsymbol{x}}\| = |\lambda| \cdot \|\boldsymbol{x}_{lK} - \bar{\boldsymbol{x}}\| \tag{5}$$

Since each agent reaches its reference state with at most ε^* error:

$$\|\boldsymbol{x}_{lK} - \boldsymbol{x}^*_{lK}\| \leq \sqrt{M}\varepsilon^*, \forall l \tag{6}$$

Combining (6) and (5), for arbitrary $\Delta > 0$ and $\rho \in (|\lambda|, 1)$, Δ-practical consensus with a convergence rate ρ will be ensured. To this end, choose a large enough budget N to guarantee $\varepsilon^* \leq (\rho - |\lambda|)\Delta/\sqrt{M}$.

Assume first that the initial disagreement is larger than Δ. Then, for all $\boldsymbol{x}_{lK}$ such that $\|\boldsymbol{x}_{lK} - \bar{\boldsymbol{x}}\| > \Delta$ one has:

$$\begin{aligned}
\|\boldsymbol{x}_{(l+1)K} - \bar{\boldsymbol{x}}\| &\leq \|\boldsymbol{x}_{(l+1)K} - \boldsymbol{x}^*_{(l+1)K}\| + \|\boldsymbol{x}^*_{(l+1)K} - \bar{\boldsymbol{x}}\| \\
&\leq \sqrt{M}\varepsilon^* + |\lambda| \cdot \|\boldsymbol{x}_{lK} - \bar{\boldsymbol{x}}\| \\
&\leq (\rho - |\lambda|)\Delta + |\lambda| \cdot \|\boldsymbol{x}_{lK} - \bar{\boldsymbol{x}}\| \\
&< \rho\|\boldsymbol{x}_{lK} - \bar{\boldsymbol{x}}\|
\end{aligned}$$

This exponential decay ensures that after a finite $l_0 = \lceil \log_\rho \Delta/\|\boldsymbol{x}_0 - \bar{\boldsymbol{x}}\| \rceil$, the distance to $\bar{x}$ will drop below Δ. Once this is true, i.e., if $\|\boldsymbol{x}_{lK} - \bar{\boldsymbol{x}}\| < \Delta$, then:

$$\begin{aligned}
\|\boldsymbol{x}_{(l+1)K} - \bar{\boldsymbol{x}}\| &\leq \|\boldsymbol{x}_{(l+1)K} - \boldsymbol{x}^*_{(l+1)K}\| + \|\boldsymbol{x}^*_{(l+1)K} - \bar{\boldsymbol{x}}\| \\
&\leq \sqrt{M}\varepsilon^* + |\lambda| \cdot \|\boldsymbol{x}_{lK} - \bar{\boldsymbol{x}}\| \\
&\leq (\rho - |\lambda|)\Delta + |\lambda|\Delta = \rho\Delta < \Delta
\end{aligned}$$

so the state will remain at distance Δ from $\bar{x}$. If the initial disagreement is already below Δ, then $l_0 = 0$ and the derivation of the exponential decay is no longer needed.

(ii) To ensure controllability, the states of all the agents must remain inside set $\mathcal{X}$ at each step lK (recall that $X_0 \subseteq \mathbf{B}(x_c, R)$ and $\mathcal{X} = \mathbf{B}(x_c, 3R + \Delta^+)$). Define $\Delta_0 = \|\boldsymbol{x} - \bar{\boldsymbol{x}}\|$, the *collective* initial disagreement, and $\Delta_0^{\mathrm{a}} = \max_i \|x_{i,0} - \bar{x}\|$, the *per-agent* initial disagreement. Take a desired collective disagreement of $\Delta \leq 2R$; part (i) allows imposing any $\Delta > 0$, and since $2R$ upper-bounds the diameter of X_0, a larger value makes little sense.

If $\Delta \geq \Delta_0$, then already, for any $l \geq 0$, $\|x_{i,lK} - \bar{x}\| \leq \|\boldsymbol{x}_{lK} - \bar{\boldsymbol{x}}\| \leq \Delta$. Thus the agents remain in $\mathbf{B}(\bar{x}, \Delta)$ and since $\bar{x} \in \mathbf{B}(x_c, R)$, they remain in $\mathbf{B}(x_c, 3R) \subset \mathcal{X}$. Now if $\Delta < \Delta_0$, then for any l and i we have:

$$\begin{aligned}
\|x_{i,(l+1)K} - \bar{x}\| &\leq \|x_{i,(l+1)K} - x^*_{i,(l+1)K}\| + \|x^*_{i,(l+1)K} - \bar{x}\| \\
&\leq \varepsilon^* + \max_j \|x_{j,lK} - \bar{x}\| \\
&\leq (l+1)\varepsilon^* + \Delta_0^{\mathrm{a}} \tag{7}
\end{aligned}$$

where the second inequality implies the third by induction. Furthermore, by part (i), as long as the collective disagreement is larger than Δ it decreases exponentially:

$$\|x_{i,lK} - \bar{x}\| \leq \|\boldsymbol{x}_{lK} - \bar{\boldsymbol{x}}\| \leq \rho^l \Delta_0$$

This allows taking a finite l_1 so that $\rho^{l_1}\Delta_0 \leq \max\{\Delta, \Delta_0^{\mathrm{a}}\}$, from which $\|x_{i,lK} - \bar{x}\| \leq \max\{\Delta, \Delta_0^{\mathrm{a}}\}\ \forall i, l \geq l_1$. Combining this with (7), we have $\|x_{i,lK} - \bar{x}\| \leq \max\{\Delta, l_1\varepsilon^* + \Delta_0^{\mathrm{a}}\}, \forall i, l \geq 0$. Therefore, finally imposing $\varepsilon^* \leq \Delta^+/l_1$ and noticing that $\Delta_0^{\mathrm{a}} \leq 2R$, the states are guaranteed to remain in $\mathcal{X}$. ■

Theorem 3 (Consensus) *Under Assumptions 1–4, for sufficiently large N Algorithm 2 achieves practical consensus at every step.*

Proof: In Theorem 2, take $l \geq l_0$. Then $\forall i, j, \|x_{i,lK} - x_{j,lK}\| \leq 2\Delta$, due to $\|\boldsymbol{x}_{lK} - \bar{\boldsymbol{x}}\| \leq \Delta$ and the triangle inequality. Due to Assumption 2, we have $\tilde{u}_{i,lK} = \tilde{f}^{-1}(x_{i,lK}, x_{i,(l+1)K})$, $\tilde{u}_{j,lK} = \tilde{f}^{-1}(x_{j,lK}, x_{j,(l+1)K})$ and denoting the Lipschitz constant of $\tilde{f}^{-1}$ by L_{-1}:

$$
\begin{aligned}
\|\tilde{u}_{i,lK} &- \tilde{u}_{j,lK}\| \\
&\leq L_{-1}(\|x_{i,lK} - x_{j,lK}\| + \|x_{i,(l+1)K} - x_{j,(l+1)K}\|) \\
&\leq 4L_{-1}\Delta
\end{aligned}
\tag{8}
$$

Then, at steps $lK + k$ with $k = 1, \ldots, K - 1$, we have:

$$
\begin{aligned}
\|u_{i,lK+k} - u_{j,lK+k}\| &\leq \sqrt{m}\,\|\tilde{u}_{i,lK} - \tilde{u}_{j,lK}\|_\infty \\
&\leq 4\sqrt{m}L_{-1}\Delta
\end{aligned}
$$

Denote $L' = 4\sqrt{m}L_{-1}$, and L the Lipschitz constant of f. Finally, by a straightforward derivation we get the desired result:

$$
\|x_{i,lK+k} - x_{j,lK+k}\| \leq (L^k + L'\sum\nolimits_{k'=1}^{k} L^{k'})\Delta \qquad\qquad ■
$$

Remarks: From the proof of Theorem 2, a rather strong type of practical consensus is ensured, where agents exponentially approach and then remain inside a region of size Δ around $\bar{x}$ – but only at steps multiple of K. The size Δ can be controlled by ε^*, and so indirectly by the budget N, as seen in Section 4.2. In-between these steps and above l_0K, Theorem 3 ensures, under the additional Assumption 2, a weaker practical consensus where agents are only close to each other and not necessarily to $\bar{x}$. Indeed, the region containing the states can travel in the state space, with the constraint that it must always return around $\bar{x}$ at each multiple of K. Note that the region may grow somewhat in-between the multiples of K, but its size remains proportional to Δ. Figure 1 illustrates these guarantees. □
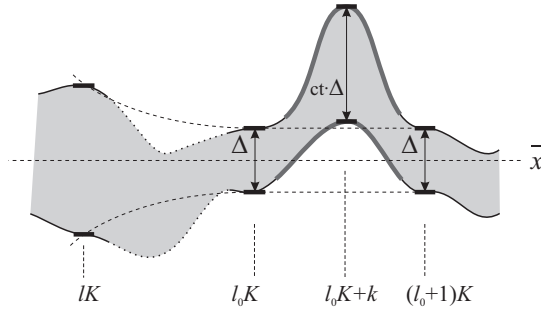

Fig. 1. Symbolic illustration of consensus guarantees. The agent states will stay within the shaded area. Dotted contour: analysis does not constrain the states; solid black contour: guarantees of Theorem 2; thick gray contour: guarantees of Theorem 3.

5 Analysis and simulations in representative cases

We next exemplify linear agents as a special case of our framework, together with a type of nonlinear agents (although the method targets nonlinear agents, understanding how it works in the familiar, linear case builds useful insight). For each type of agents, we analyze the assumptions and present simulation results. It must be emphasized that these examples are analyzed here in detail for illustrative purposes, using 'white-box', full mathematical models of the agent dynamics. Indeed, the whole point of our algorithm is to avoid the need to use such models; instead, they are a black box supplied to the algorithm, and it is precisely this which lends the approach its generality. In practice,

the user simply chooses a sufficiently large $\tilde{U}$ and—provided of course the assumptions hold—does not need to worry about the other details.

5.1   Linear agents

Consider an agent with linear controllable dynamics:

$$x_{k+1} = Ax_k + Bu_k$$

Different agents may have different dynamics if we are only interested in consensus every K steps; otherwise, agents must be homogeneous due to Assumption 2. In both cases, we drop the agent index since the development is generally valid for any linear dynamics.

Analysis.   The set of interesting initial states considered is $X_0 = \mathbf{B}(0, R)$. It follows that $\mathcal{X} = \mathbf{B}(0, 3R + \Delta^+)$ for an arbitrary $\Delta^+ > 0$. Denote $R_{\mathcal{X}} = 3R + \Delta^+$. For now, K is taken equal to the order n of the dynamics, and the K-step dynamics are:

$$x_{k+K} = \tilde{A}x_k + \tilde{B}\tilde{u}_k = A^K + [A^{K-1}B, A^{K-2}B, \dots, B]\tilde{u}_k$$

Since the system is controllable $\tilde{B}$ is full row rank, and for any x_k at least one $\tilde{u}_k^*$ exists that reaches an arbitrary x_{k+K}^* (there may be more):

$$\tilde{u}_k^* = \tilde{B}^{\ddagger}(x_{k+K}^* - \tilde{A}x_k) \tag{9}$$

where $\ddagger$ denotes left pseudoinverse. Then $\|\tilde{u}_k^*\| \leq \|\tilde{B}^{\ddagger}\|(1 + \|\tilde{A}\|)\|x_k\|$, and an action space $\tilde{U}$ including a ball of radius $R_U = \|\tilde{B}^{\ddagger}\|(1 + \|\tilde{A}\|)R_{\mathcal{X}}$ ensures Assumption 1 ($\|\cdot\|$ denotes the induced norm for matrices).

The optimal control at step k must maximize:

$$r(\tilde{u}_k) = -\|x_{k+K} - x_{k+K}^*\| = -\|\tilde{B}\tilde{u}_k - \tilde{B}\tilde{u}_k^*\|$$

where $\tilde{u}_k^*$ is the (unknown) action that exactly reaches x_{k+K}^*. We have:

$$r(\tilde{u}_k^*) - r(\tilde{u}_k) = 0 + \|\tilde{B}\tilde{u}_k - \tilde{B}\tilde{u}_k^*\| \leq \|\tilde{B}\|\|\tilde{u}_k - \tilde{u}_k^*\|$$

So, a semimetric $\ell(\tilde{u}, \tilde{u}') = \overline{\gamma}\|\tilde{u} - \tilde{u}'\|$ with $\overline{\gamma} \geq \|\tilde{B}\|$ (in fact a metric) ensures the weak-Lipschitz property **??**.i. Note that $r(\tilde{u}_k^*) = 0$, and also that $\|\tilde{B}\|$ plays the role of Lipschitz constant for r. If the prior knowledge that ℓ is a good metric in the linear problem is available, then DOO should be applied with this metric. Otherwise, SOO should be used. Further, it is clear that if we take a hyperbox U^K and create a natural, exponential partitioning by recursively splitting it into smaller hyperboxes along each dimension (2^{Km} new hyperboxes at each step), Assumptions **??**.ii and **??**.iii hold.

Next, the near-optimality dimension β is investigated (omitting the proof). If $mK > n$, there are multiple optimal solutions and β is upper-bounded by $\varphi = mK - n$, the number of extra degrees of freedom in the solution (9). This can intuitively be understood as OO having to uniformly explore the φ-dimensional subspace of equally good near-optimal solutions, and provides yet another incentive for having K as small as possible. Since β is bounded, Assumption 4 holds.

So far, we only imposed Assumption 1. Note now that the direct and inverse dynamics are Lipschitz. Further requiring Assumption 2, invertibility in K steps, boils down to rank$(\tilde{B}) = mK$ [26]. Since due to controllability rank$(\tilde{B}) = n$, we must have $K = n/m$, and controllability must hold in $K < n$ steps. In effect, all this leads to a square, invertible $\tilde{B}$. So, the condition is significantly more restrictive than Assumption 1. However, due to invertibility β is now 0, and the optimization problem is easy to solve.

Simulation results.   Consider the classical example of double-integrator agents [20]:

$$x_{i,k+1}^{\mathrm{p}} = x_{i,k}^{\mathrm{p}} + T_{\mathrm{s}}x_{i,k}^{\mathrm{p}}, \quad x_{i,k+1}^{\mathrm{v}} = x_{i,k}^{\mathrm{v}} + T_{\mathrm{s}}u_{ik}$$

9

where x^{p} is the position, x^{v} the velocity, the input is u_{ik}, and Euler discretization is used with sampling time $T_{\mathrm{s}} = 0.5\,\mathrm{s}$. The system is controllable in $K = 2$ steps (note that $n = Km$, leading to $\beta = 0$).

Five agents are considered, having the (undirected) connection graph from Figure 2 (top), initial positions $0, 1, 2, 3, 4$, and initial velocities $-3, -1, 0, 1, 3$. In all our experiments, the neighbor weights are set with the method of [19]: the agents first agree on a value $\alpha \in (0, \frac{1}{\max_i |\mathcal{N}_i|})$, and then agent i sets $p_{i,i} = 1 - \alpha\,|\mathcal{N}_i|$ and $p_{i,j} = \alpha$, $\forall j \in \mathcal{N}_i$. We set $\alpha = 0.2$. The action space is taken large enough to ensure controllability (specifically, $\tilde{U} = [-200, 200]^2$). To characterize performance in each experiment, an inter-agent disagreement is computed at every step: $\Delta_k = \sum_{i<j} \|x_{i,k} - x_{j,k}\|$, and the average of Δ_k over the entire trajectory is reported.

We will study the performance for varying budget $N = 15, 25, 50, 75, 100, 200, \ldots, 700$, in three cases. First, DOO is applied, using the 'ideal' metric with constant $\overline{\gamma} = \|\tilde{B}\|$. In the second, more realistic case, $\overline{\gamma}$ is overestimated as $4\|\tilde{B}\|$. Third, the metric is no longer assumed known and SOO is applied. As seen in Figure 2 (middle), performance largely increases with N for all optimization methods, although monotonicity is not guaranteed. Moreover, as expected, DOO with the ideal metric performs best, obtaining near-optimal performance for small budgets. Since it does not use the metric, SOO lags slightly behind, but quickly reaches near-optimal performance as well. DOO with the overestimated metric performs poorly, falling behind SOO. Figure 2 (bottom) shows a trajectory for SOO and $N = 200$. Practical consensus is obtained in under $20\,\mathrm{s}$. Since the desired agent state $x^*_{i,k+2}$ is designed without knowing the structure of the dynamics, the resulting trajectory is not smooth.
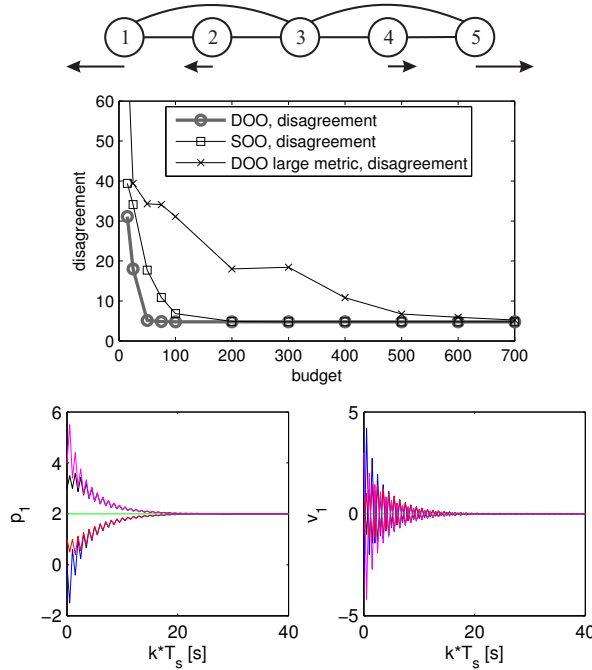

Fig. 2. Top: Connection graph for double integrators, with initial velocities symbolized by arrows. Middle: Results for varying N. Bottom: A representative trajectory.

5.2 Hammerstein nonlinear agents

To illustrate the framework in the nonlinear case, we consider an example where linear dynamics are affected by an algebraic input nonlinearity – a Hammerstein system, taken from the literature [3].[3] The example is adapted to our

---

[3] For analysis and experiments on agents with nonlinearities in the state transitions, see [1].

framework by transforming it in the state-space form and eliminating the noise, leading to:

$$x_{k+1} = Ax_k + Bz_k = \begin{bmatrix} 1.6 & -0.8 \\ 1 & 0 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0 \end{bmatrix} z_k$$

$$z_k = g(u_k) = u_k + 0.5u_k^2 + 0.25u_k^3$$

(10)

where again we do not include the agent index.

Analysis. Like before, $X_0 = \mathbf{B}(0, R)$, leading to $\mathcal{X} = \mathbf{B}(0, R_{\mathcal{X}})$ with $R_{\mathcal{X}} = 3R + \Delta^+$. Note that the input nonlinearity g is strictly increasing and bijective. From the 2-step dynamics:

$$x_{k+2} = A^2 x_k + \tilde{B}\tilde{z}_k = A^2 x_k + [AB, B][z_k, z_{k+1}]^\top$$

with a full-rank $\tilde{B}$, the linear system is controllable in $K = 2$ steps, and due to the bijective nonlinearity the same remains true of the nonlinear system. Furthermore, the inverse dynamics are:

$$\tilde{u}_k = [g^{-1}(z_k), g^{-1}(z_{k+1})]^\top, \tilde{z}_k = \tilde{B}^{-1}(x_{k+2}^* - A^2 x_k)$$

From the fact that $0.5|u| \le |g(u)|$, we have $|u| \le 2\|\tilde{z}_k\| \le 2\|\tilde{B}^{-1}\|(1 + \|A^2\|)R_{\mathcal{X}}$, where the second part is taken from the linear case above. So, a compact input box of this size is sufficient to make the system controllable in $\mathcal{X}$. Since the direct and inverse dynamics are also Lipschitz, Assumptions 1 and 2 have been verified.

The objective function of the optimal control problem is $r(\tilde{u}_k) = -\|x_{k+2} - x_{k+2}^*\|$. To find a semimetric ℓ satisfying **??**.i, note that:

$$r(\tilde{u}_k^*) - r(\tilde{u}_k) = \|x_{k+2} - x_{k+2}^*\| \le \|\tilde{B}(\tilde{z}_k - \tilde{z}_k^*)\|$$

$$\le \|\tilde{B}\| \left\| [g(u_k) - g(u_k^*), g(u_{k+1}) - g(u_{k+1}^*)]^\top \right\|_1$$

$$\le (1 + R_{\mathcal{X}} + 0.75R_{\mathcal{X}}^2)\|\tilde{B}\| \|\tilde{u}_k - \tilde{u}_k^*\|_1$$

(11)

where the last step is due to $|g(u) - g(u^*)| = |u - u^* + 0.5(u^2 - u^{*2}) + 0.25(u^3 - u^{*3})| \le |u - u^*| \cdot |1 + 0.5(u + u^*) + 0.25(u^2 + u^{*2} + uu^*)| \le (1 + R_{\mathcal{X}} + 0.75R_{\mathcal{X}}^2)|u - u^*|$. Therefore, an appropriate metric is $\ell(\tilde{u}_k, \tilde{u}_k^*) = \overline{\gamma} \|\tilde{u}_k - \tilde{u}_k^*\|_1$ with a 'Lipschitz constant' $\overline{\gamma}$ greater than the multiplying factor above. Noticing that under this metric the natural partitioning into squares satisfies the remaining conditions **??**.ii and **??**.iii, Assumption 3 is validated.

Finally, to find the near-optimality dimension β, we study the sets:

$$\tilde{U}_\varepsilon = \{\tilde{u} \mid r(\tilde{u}^*) - r(\tilde{u}) \le \varepsilon\} \subseteq \left\{\tilde{u} \mid \underline{\gamma}\|\tilde{u}_k - \tilde{u}_k^*\|_1 \le \varepsilon\right\}$$

The inclusion holds due to $r(\tilde{u}^*) - r(\tilde{u}) = \|\tilde{B}(\tilde{z}_k - \tilde{z}_k^*)\| \ge \sqrt{\lambda_{\min}(\tilde{B}^\top \tilde{B})/2} \|\tilde{z}_k - \tilde{z}_k^*\|_1$. Further, by the monotonicity of g, around any u^* we have $|z - z^*| = |g(u) - g(u^*)| \ge \alpha|u - u^*|$ with some positive α. Thus, $\tilde{U}_\varepsilon$ is included in a ℓ-ball of radius proportional to ε, and therefore the packing number is constant, leading to a near-optimality dimension of 0: Assumption 4 is satisfied and the optimization problem is easy to solve.

Simulation results. Five agents with dynamics (10) are connected with the directed graph shown at the top-left of Figure 3. We apply SOO consensus with $N = 200$ and neighbor weight $\alpha = 0.2$, from uniformly random initial states $x_{i,0} \in [-5, 5]^2$ (the action space is then taken sufficiently large, $\tilde{U} = [-300, 300]^2$). Figure 3 (bottom) shows the resulting trajectories of the agents. The algorithm easily deals with these nonlinear dynamics. Note that, unlike for the double integrator, here the consensus state is not an equilibrium, so the states no longer reach a *constant* ball around it; instead, they synchronously travel in the state space as predicted by the analysis, see Figure 3 (top-right) and again Figure 1.
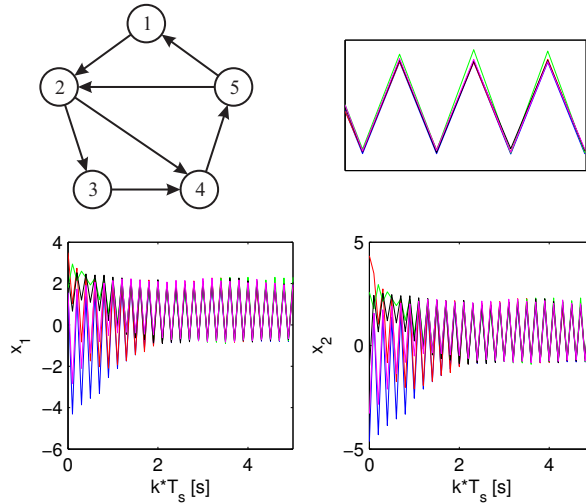
Fig. 3. Consensus of Hammerstein systems. Top-left: Communication graph. Bottom: State trajectories (the sampling time is taken 0.1 s). Top-right: zoom-in on the last portion of the x_2 trajectory; note the agents are synchronized.

6 Conclusions and perspectives

We presented a generic consensus method for nonlinear agents. Its main component is an optimistic optimization algorithm used to find the control actions, which works by sampling a black-box model of the agent, and can therefore be applied without changes to a large class of nonlinear dynamics. An analysis of representative examples clarified the meaning of the assumptions, showed that in certain problems the optimization is easy to solve, and was complemented by simulations.

The method works in increments of a small number K of steps, which is its main limitation; a natural extension to large or unknown K is to apply long-horizon optimistic *planning* methods [18,13, Ch. 22]. Future work also includes extending the approach, by exploiting similar analysis techniques, to other open problems in nonlinear consensus, e.g. maintaining connectivity with limited communication range and dealing with perturbations such as noise, time delays or packet dropouts.

References

[1] L. Buşoniu and C. Morarescu, "Consensus for agents with general dynamics using optimistic optimization," in *Proceedings 52nd Conference on Decision and Control (CDC-13)*, Florence, Italy, 10–13 December 2013.

[2] ——, "Optimistic planning for consensus," in *American Control Conference 2013 (ACC-13)*, Washington, DC, 17–19 June 2013.

[3] F. Ding and T. Chen, "Identification of hammerstein nonlinear armax systems," *Automatica*, vol. 41, no. 9, pp. 1479–1489, 2005.

[4] S. Edelkamp and S. Schrödl, *Heuristic Search: Theory and Applications*. Morgan Kauffman, 2012.

[5] L. Fang, P. Antsaklis, and A. Tzimas, "Asynchronous consensus protocols: Preliminary results, simulations and open questions," in *Proceedings of the Joint 44th IEEE Conference on Decision and Control, and the European Control Conference*, 2005, pp. 2194–2199.

[6] G. Ferrari-Trecate, L. Galbusera, M. Marciandi, and R. Scattolini, "Model predictive control schemes for consensus in multi-agent systems with single- and double-integrator dynamics," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2560–2572, 2009.

[7] M. Fliess, "Reversible linear and nonlinear discrete-time dynamics," *IEEE Transactions on Automatic Control*, vol. 37, no. 8, pp. 1144–1153, 1992.

[8] J. Grizzle, "A linear algebraic framework for the analysis of discrete-time nonlinear systems," *SIAM Journal of Control and Optimization*, vol. 31, no. 4, pp. 1026–1044, 1993.

[9] K. Hunt, D. Sbarbaro, R. Zbikowski, and P. Gawthrop, "Neural networks for control systems – a survey," *Automatica*, vol. 28, no. 6, pp. 1083–1112, 1992.

[10] L. Hunt and G. Meyer, "Stable inversion for nonlinear systems," *Automatica*, vol. 33, no. 8, pp. 1549–1554, 1997.

[11] B. Jakubczyk and E. D. Sontag, "Controllability of nonlinear discrete-time systems: A lie-algebraic approach," *SIAM Journal of Control and Optimization*, vol. 28, pp. 1–33, 1990.

12

[12] T. Keviczky and K. Johansson, "A study on distributed model predictive consensus," in *Proceedings 17th IFAC World Congress (IFAC-08)*, Seoul, Korea, 6–11 July 2008, pp. 1516–1521.

[13] F. Lewis and D. Liu, Eds., *Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control*.   Wiley, 2012.

[14] J. Mei, W. Ren, and G. Ma, "Distributed coordinated tracking with a dynamic leader for multiple Euler-Lagrange systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1415–1421, 2011.

[15] W. Michiels, I.-C. Morarescu, and S.-I. Niculescu, "Consensus problems with distributed delays, with application to traffic flow models," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 77–101, 2009.

[16] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, pp. 169–182, 2005.

[17] R. Munos, "Optimistic optimization of a deterministic function without the knowledge of its smoothness," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, Eds., 2011, pp. 783–791.

[18] ——, "From bandits to Monte-Carlo tree search: The optimistic principle applied to optimization and planning," INRIA, Tech. Rep., 2013.

[19] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transaction on Automatic Control*, vol. 49, pp. 1520–1533, 2004.

[20] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[21] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[22] A. Olshevsky and J. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Journal of Control and Optimization*, vol. 48, no. 1, pp. 33–55, 2009.

[23] Z. Qu, J. Wang, and R. Hull, "Cooperative control of dynamical systems with application to autonomous vehicles," *IEEE Transaction on Automatic Control*, vol. 53, no. 4, pp. 894–911, 2008.

[24] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, pp. 655–661, 2005.

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

[26] M. K. Sain and J. L. Massey, "Invertibility of linear time-invariant dynamical systems," *IEEE Transaction on Automatic Control*, vol. 14, no. 2, pp. 141–149, 1969.

[27] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[28] H. Su, G. Chen, X. Wang, and Z. Lin, "Adaptive second-order consensus of networked mobile agents with nonlinear dynamics," *Automatica*, vol. 47, no. 2, pp. 368–375, 2011.

[29] J. Suykens and J. Vandewalle, Eds., *Nonlinear Modeling: Advanced Black-Box Techniques*.   Springer, 1998.

[30] H. Tanner, A. Jadbabaie, and G. Pappas, "Flocking in teams of nonholonomic agents," in *Cooperative Control*, ser. Lecture Notes in Control and Information Sciences, V. Kumar, N. Leonard, and A. Morse, Eds.   Springer, 2005, vol. 309, pp. 458–460.

[31] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, pp. 803–812, 1986.

[32] B. Widrow and E. Walach, *Adaptive Inverse Control: A Signal Processing Approach, Reissue Edition*.   Wiley, 2008.

[33] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *System and Control Letters*, vol. 53, pp. 65–78, 2004.

[34] Y. Zheng and R. J. Evans, "Minimal order discrete-time nonlinear system inversion," in *Proceedings 15th IFAC World Congress*, Barcelona, Spain, 21–26 July 2002, pp. 1119–1125.

[35] J. Zhou, X. Wu, W. Yu, M. Small, and J. Lu, "Flocking of multi-agent dynamical systems based on pseudo-leader mechanism," *Systems & Control Letters*, vol. 61, no. 1, pp. 195–202, 2012.