An active learning method for solving competitive multi-agent decision-making and control problems

Filippo Fabiani and Alberto Bemporad, *Fellow, IEEE*

Abstract—To identify a stationary action profile for a population of competitive agents, each executing private strategies, we introduce a novel active-learning scheme where a centralized external observer (or entity) can probe the agents' reactions and recursively update simple local parametric estimates of the action-reaction mappings. Under very general working assumptions (not even assuming that a stationary profile exists), sufficient conditions are established to assess the asymptotic properties of the proposed active learning methodology so that, if the parameters characterizing the action-reaction mappings converge, a stationary action profile is achieved. Such conditions hence act also as certificates for the existence of such a profile. Extensive numerical simulations involving typical competitive multi-agent control and decision-making problems illustrate the practical effectiveness of the proposed learning-based approach.

Index Terms—Multi-agent systems, Active learning, Competitive decision-making, Generalized Nash equilibria.

I. INTRODUCTION

Multi-agent system applications are rapidly growing in complexity due to the widespread use of embedded sensors and distributed control systems [1]. The deployed agents can make autonomous decisions to maximize some private performance index while, at the same time, interacting with their peers. The competitive or cooperative nature of such interactions yield emerging behaviours, which in addition to the possibly large-scale structure and potential privacy issues, considerably complicate the analysis of the multi-agent system at hand. Besides, those systems made of interacting decision-makers typically feature desirable collective outcomes meant as solutions of the interaction process, which can be gathered under some (loose) stationarity notion, for example, a Nash equilibrium in noncooperative game theory [2].

The considerations above hence create a fertile ground for the problem investigated in this paper. Consider, for instance, a population of competitive agents that interact with each other through private *action-reaction mappings*, which may (or may not) lead to a particular collective outcome, here identified as a *stationary action profile*. By leveraging tools from the machine learning and system identification literature [3]–[5], our work aims at answering the following question: given an agnostic scenario in which an external observer (or entity) is only allowed to query the action-reaction mappings, is it possible *to learn* a stationary action profile of the underlying multi-agent interaction process?

A. Motivating example: forecasting price-responses to control the aggregated electricity consumption in smart grids

Among the *indirect control* strategies enabling energy flexibility offered by end-users [6], energy retailers or distribution system operators (DSOs) design price signals to which price-sensitive users adjust their consumption/storage profile to meet their private needs [7]–[9]. Leveraging insights into user price-responsiveness, the price-generation process thus serves as a mechanism for managing electricity consumption. Such an indirect control method, however, rely on accurate predictions of the users' consumption responses to broadcasted price signals for activating flexibility. It is then crucial to develop tools to accomplish such a prediction task under limited knowledge about the decision process driving users' price-responsiveness.

As an example, consider a set of N electric vehicles (EVs) populating a distribution grid [10]–[12], where every user wants to determine an optimal EV charging schedule over a certain discrete time interval $\{1, \ldots, T\}$ by controlling the energy injection $x_i \in \mathbb{R}^T_{\geq 0}$ of its own EV charger. Specifically, each user $i \in \{1, \ldots, N\}$ aims at minimizing a *private* cost in the form $J_i(x_i, \sigma(\boldsymbol{x})) = \|x_i\|^2_{Q_i} + c_i^\top x_i + (a(\sigma(\boldsymbol{x}) + d) + b\mathbf{1}_T)^\top x_i$, where $\boldsymbol{x} \in \mathbb{R}^{NT}_{\geq 0}$ represents the vector stacking all the users' decision variables, $\|x_i\|^2_{Q_i} + c_i^\top x_i$ models the battery degradation cost, and $(a(\sigma(\boldsymbol{x}) + d) + b\mathbf{1}_T)^\top x_i$ the electricity pricing. In particular, $\sigma(\boldsymbol{x})$ denotes the *aggregate demand* of the overall population of EVs, usually defined as $\sigma(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^N x_i \in \mathbb{R}^T_{\geq 0}$, $a > 0$ represents the inverse of the price elasticity of demand, $b > 0$ the baseline price, and $d \in \mathbb{R}^T_{\geq 0}$ the normalized average inflexible demand. In addition, each user has to satisfy both local and shared constraints due for instance to a minimum charging amount over the interval, $\mathbf{1}_T^\top x_i \geq \gamma_i \geq 0$, a cap on the power injection $x_i \in [0, \bar{x}_i]^T$, or accounting for intrinsic grid limitations, e.g., $\sigma(\boldsymbol{x}) + d \in [0, \bar{c}]^T$.

This formulation thus originates a so-called generalized Nash game in aggregative form, whose solution coincides to a Nash equilibrium [2], i.e., a collective charging profile, say $\boldsymbol{x}^\star$, in which none of the user can gain by unilaterally deviating from its current strategy. On the other hand, such an equilibrium $\boldsymbol{x}^\star$, which yields the aggregate consumption $\frac{1}{N}\sum_{i=1}^N x_i^\star$, heavily depends on the price signals a and b affecting the *private* cost function of each agent, J_i, i.e., $\boldsymbol{x}^\star = \boldsymbol{x}^\star(a, b)$. It is then clear how a suitable design of a and b, based on an accurate prediction of the resulting $\boldsymbol{x}^\star(a, b)$, allows for an efficient (for DSOs) and profitable (for retailers) usage of the distribution grid. In §II-B, we will derive an active learning-based approach enabling an exact prediction of the outcome of the multi-agent interaction process on which little knowledge is available.

B. Related work

Our results can be positioned within those branches of literature referring to the learning of equilibria in an empirical game-theoretic or black-box setting. However, our framework can encompass a broader class of multi-agent decision-making and control problems than is usually considered.

The pioneering work [13] focused on congestion games, showing how one can learn the agents' cost functions while querying only a portion of decision spaces. This work then originated a series of contributions investigating query and/or communication complexity of algorithms for specific classes of games [14]–[17]. Along the same line, in [18], several schemes were devised with provable bounds on the best-response query complexity for computing approximate equilibria of two-player games with a finite number of strategies. Few other works belonging to the simulation-based game literature, instead, took a probably approximately correct learning perspective to reconstruct an analytical representation of normal form games (i.e., matrix games) for which a black-box simulator provides noisy samples of agents' utilities, devising algorithms that uniformly approximate the original games and associated equilibria with finite-sample guarantees [19]–[21]. Stochastic [22] or sample-average approximation [23] of simulated games, and Bayesian optimization-based methods [22], [24], [25] have also been adopted. In the former cases, the authors investigated matrix games with finite decision sets, providing an asymptotic analysis of Nash equilibria obtained from simulation-based models, along with probabilistic bounds on their approximation quality. Bayesian approaches are, instead, empirical and typically rely on Gaussian processes used as emulators of the black-box cost functions. Posterior distributions provided by the Gaussian process are successively adopted to design acquisition functions tailored to solve Nash games, which are based on the probability of achieving an equilibrium. More recently, [26] proposed an active-set-based first-order algorithm to learn the rationality parameters of the agents taking part in a potential game from historical observations of Nash equilibria.

Suitable examples of works tackling standard problems in game theory through machine learning techniques can be found in [27]–[30]. In particular, [27] addressed payoff-function learning as a standard regression problem. The methodology presented there, however, focused on games in normal form and came with no theoretical guarantees. Combining proximal-point iterations and ordinary least-squares (LS) estimators, [28] designed a distributed algorithm with probabilistic convergence guarantees to an equilibrium in stochastic games where the agents learn their own cost functions. In [29], [30], instead, a coordinator aims at reconstructing private information held by the agents to enable the computation of an equilibrium by designing personalized incentives affecting the cost functions.

C. Summary of contributions and paper organization

Unlike the works above, we design an active learning scheme that allows an agnostic external entity to reproduce faithful approximations of action-reaction mappings, privately held by a population of agents, in order to predict a stationary profile of the underlying multi-agent interaction process exactly.
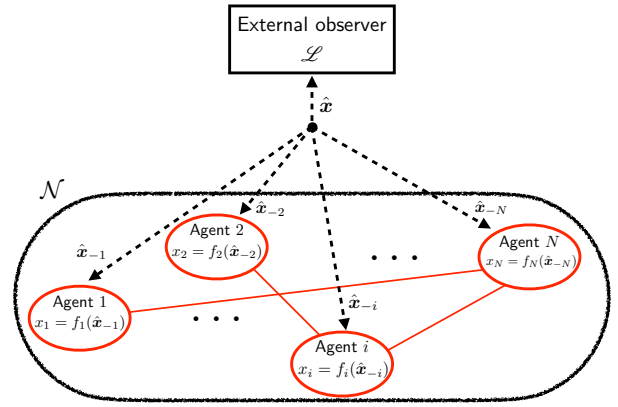

Fig. 1: An external observer with learning procedure $\mathscr{L}$ makes queries $\hat{\boldsymbol{x}}$ and observes each reaction to $\hat{\boldsymbol{x}}_{-i}$ (dashed black lines) taken by agent $i \in \mathcal{N}$ (red circles) through its private action-reaction mapping, $f_i(\cdot)$, which may depend on the decision of any other agent in $\mathcal{N}$ (solid red lines), with the goal of predicting an outcome of the multi-agent interaction process.

Specifically, we consider the scenario illustrated in Fig. 1 and mathematically formalized in §II-A, where such an external observer, endowed with a learning procedure $\mathscr{L}$, can get samples of the action-reaction mappings $f_i(\cdot)$ of each agent i, for all $i = 1, \ldots, N$; in particular, the external observer iteratively proposes to agent i a configuration $\hat{\boldsymbol{x}}_{-i}$ of possible other agents' decisions and gets back the reaction $x_i = f_i(\hat{\boldsymbol{x}}_{-i})$ of the agent to that configuration (see also the different tasks reported in Algorithm 1). The proposed scheme is of active-learning nature in that the queries $\hat{\boldsymbol{x}}_{-i}$ are generated based on the current surrogate models $\hat{f}_i(\cdot)$ of the agents' responses.

In our framework the low-level interaction among agents is not particularly relevant, since they are left free to possibly interact with each other. As such, we do not assume any particular communication pattern underlying the information exchange among the agents and topology of interactions. In fact, we only assume that the decision of some agent i may be affected by the decision of any other agent within $\mathcal{N}$. On the contrary, we focus on the external entity, which iteratively makes queries to estimate private action-reaction mappings with the ultimate goal of predicting a a stationary profile $\boldsymbol{x}^\star$ *exactly*. To this end, the proposed learning mechanism allows the central entity to collect and exploit high-quality data that are informative with respect to (w.r.t.) such a stationary point prediction task. We summarize our contributions as follows:

i) Formalize a novel problem setting involving an external entity that aims at predicting a possible outcome in multi-agent decision-making and control problems in which the decision policies of the agents are private (§II, V);

ii) Devise an active learning algorithm allowing the external entity to collect sensible data and update parametric estimates of the agents' action-reaction mappings (§IV);

iii) Prove sufficient conditions to assess the asymptotic properties of our active learning scheme so that, if convergence happens, it can only be towards a stationary action profile (§III, IV). This fact entails two main implications:

a) Learning locally exact surrogates of the action-reaction mappings allows the external observer to succeed in its prediction task;

b) Working with very general assumptions, the established conditions also serve as certificates for the existence of a stationary action profile.

iv) A demonstration of the practical effectiveness of our methodology through extensive numerical simulations on the indirect control method for smart grids in §I-A, and typical multi-agent control and decision-making problems, including generalized Nash games and competitive linear feedback design problems (§VI).

The proofs of the technical results of the paper are all deferred to Appendix A, while Appendix B reports mathematical derivations adopted in the implementation of our approach. The Python package gnep-learn, implementing the proposed active learning-based scheme, is publicly available at https://github.com/bemporad/gnep-learn.

To the best of our knowledge, this work represents a first attempt integrating traditional machine learning paradigms within a smart query process, with the goal of predicting a possible outcome in multi-agent problems in which the decision policies of the agents are kept private. In contrast with the cognate literature reviewed in §I-B, we design a deterministic, active learning-based mechanism that exploits action-reaction samples to learn private action-reaction mappings, a setting that covers a wide variety of problems involving competitive agents with continuous (i.e., possibly infinite) decision sets.

Notation: $\mathbb{N}$, $\mathbb{R}$ and $\mathbb{R}_{\geq 0}$ denote the set of natural, real, and nonnegative real numbers, respectively. $\mathbb{N}_0 \coloneqq \mathbb{N} \cup \{0\}$, while $\bar{\mathbb{R}} \coloneqq \mathbb{R} \cup \{+\infty\}$. The transpose of a matrix $A \in \mathbb{R}^{n \times n}$ is $A^\top$, while $A \succ 0$ ($\succeq 0$) denotes its positive (semi)definiteness. For a vector $v \in \mathbb{R}^n$ and a matrix $A \succ 0$, $\|v\|_2$ denotes the standard Euclidean norm, while $\|\cdot\|_A$ the A–induced norm $\|v\|_A \coloneqq \sqrt{v^\top A v}$. The n-dimensional ball centred around $\bar{x}$ with radius $\theta > 0$ is $\mathcal{B}_\theta(\bar{x}) \coloneqq \{x \in \mathbb{R}^n \mid \|x - \bar{x}\|_2 \leq \theta\}$. I_n, $\mathbf{1}_n$, and $\mathbf{0}_n$ denote the $n \times n$ identity matrix, the vector of all 1, and 0, respectively (we omit the dimension n whenever clear). Given a set $\mathcal{X} \subseteq \mathbb{R}^n$, $\iota_{\mathcal{X}} : \mathbb{R}^n \to \bar{\mathbb{R}}$ denote the associated indicator function, i.e., $\iota_{\mathcal{X}}(x) = 0$ if $x \in \mathcal{X}$, $\iota_{\mathcal{X}}(x) = +\infty$ otherwise. The operator $\mathrm{col}(\cdot)$ stacks its arguments in column vectors or matrices of compatible dimensions. For example, given vectors $x_1, \ldots, x_N$ with $x_i \in \mathbb{R}^{n_i}$ and $\mathcal{N} = \{1, \ldots, N\}$, we denote $\boldsymbol{x} \coloneqq (x_1^\top, \ldots, x_N^\top)^\top = \mathrm{col}((x_i)_{i \in \mathcal{N}}) \in \mathbb{R}^n$, $n \coloneqq \sum_{i \in \mathcal{N}} n_i$, and $\boldsymbol{x}_{-i} \coloneqq \mathrm{col}((x_j)_{j \in \mathcal{N} \setminus \{i\}})$. With a slight abuse of notation, we sometimes use also $\boldsymbol{x} = (x_i, \boldsymbol{x}_{-i})$. The uniform distribution on the interval $[a, b]$ is denoted by $\mathcal{U}(a, b)$.

II. LEARNING PROBLEM

We start by formalizing the learning problem addressed, which is successively motivated by two multi-agent control and decision-making problems that fit the framework we consider.

A. Mathematical formulation

We consider the scenario illustrated in Fig. 1. There, an external observer (or entity) is allowed to make queries and observe the reactions taken by a set of N (possibly competitive) agents,

indexed by $\mathcal{N} \coloneqq \{1, \ldots, N\}$, which mutually influence each other. Specifically, we assume that each agent controls a vector of variables $x_i \in \mathbb{R}^{n_i}$ to react, by means of a private *action-reaction mapping* $f_i : \mathbb{R}^{n_{-i}} \to \mathbb{R}^{n_i}$, to the other agents' actions $\boldsymbol{x}_{-i} \in \mathbb{R}^{n_{-i}}$, $n_{-i} \coloneqq \sum_{j \in \mathcal{N} \setminus \{i\}} n_j$. Within the multi-agent interaction process, the whole population of agents has also to meet collective constraints, i.e., $\boldsymbol{x} \coloneqq (x_i, \boldsymbol{x}_{-i}) \in \Omega \subseteq \mathbb{R}^n$, $n \coloneqq n_i + n_{-i}$, which are known to every agent and the external entity, and whose portion involving the i-th agent can be embedded directly within the action choice $f_i(\cdot)$. Specifically, for any given $\boldsymbol{x}_{-i}$, $f_i(\boldsymbol{x}_{-i})$ is so that $(f_i(\boldsymbol{x}_{-i}), \boldsymbol{x}_{-i}) \in \Omega$, for all $i \in \mathcal{N}$. We assume that possible additional local constraints $x_i \in \mathcal{X}_i$, such as lower and upper bounds on x_i, are either private (i.e., unknown to the other agents and the external entity), or shared. In both cases, it always holds that the resulting values $f_i(\boldsymbol{x}_{-i}) \in \mathcal{X}_i$. In the former case, this is a *de facto* situation that the external observer and the other agents take cognizance of. In the latter case, we assume that $x_i \in \mathcal{X}_i$ is already embedded in the collective constraint $\boldsymbol{x} \in \Omega$.

Being completely agnostic on the overall multi-agent interaction process and main quantities involved, we assume the underlying external entity endowed with some parametric learning procedure $\mathscr{L}$, which exploits non-private data (i.e., the decisions $f_i(\boldsymbol{x}_{-i})$) collected iteratively from the agents through a query process. Specifically, the main goal of the external entity is to learn, at least locally, faithful surrogates $\hat{f}_i$ (formally defined later in this section) of the unknown mappings $f_i(\cdot)$ in order to predict a desirable outcome of the multi-agent interaction process, here identified in accordance with the following definition:

Definition 2.1. (Stationary action profile) *A collective action profile* $\boldsymbol{x}^\star \in \Omega$ *is* stationary *if, for all* $i \in \mathcal{N}$, $x_i^\star = f_i(\boldsymbol{x}_{-i}^\star)$. $\square$

Given a stationary profile, which coincides with a fixed point of the action-reaction mappings, none of the agents has incentive to deviate from the action currently taken. We will discuss in §II-B (and §V-B when introducing a generalization of the problem investigated) suitable multi-agent control and decision-making applications calling for a stationarity condition as the one in Definition 2.1.

We now state some conditions on the formalized problem that we assume will hold true throughout the paper:

Standing Assumption 2.2 (Mappings and constraints)**.** *For all* $i \in \mathcal{N}$, *the single-valued mapping* $f_i : \mathbb{R}^{n_{-i}} \to \mathbb{R}^{n_i}$ *is continuous. In addition, the set* $\Omega \subseteq \mathbb{R}^n$ *of feasible collective actions is nonempty.* $\square$

While the non-emptiness of Ω is an obvious requirement for the problem to be solvable, continuity of f_i is a technical, yet not very restrictive, condition needed to prove our results.

As mentioned above, we endow the external observer with a learning procedure $\mathscr{L}$ to produce faithful proxies of the action-reaction mappings $f_i(\cdot)$ executed by the agents. Specifically, we let $\hat{f}_i : \mathbb{R}^{n_{-i}} \times \mathbb{R}^{p_i} \to \mathbb{R}^{n_i}$ denote the mapping of agent i as estimated by the external entity, which is parametrized by $\theta_i \in \mathbb{R}^{p_i}$ to be updated iteratively by integrating data obtained from the agents through a query process, as described in §IV.

Since our technical analysis will only require $\hat{f}_i(\cdot, \theta_i)$ to approximate $f_i(\cdot)$ *locally* around a possible stationary action, for simplicity, we parameterize $\hat{f}_i(\cdot, \theta_i)$ as the affine mapping

$$\hat{f}_i(\boldsymbol{x}_{-i}, \theta_i) = \Lambda_i \begin{bmatrix} \boldsymbol{x}_{-i} \\ 1 \end{bmatrix}, \tag{1}$$

for a matrix of coefficients $\Lambda_i \in \mathbb{R}^{n_i \times (n_{-i}+1)}$ (in this case, θ_i is the vectorization of Λ_i, with $p_i = n_i(n_{-i}+1)$). In §IV-D we will show that adopting such simple surrogate mappings comes with no restrictions, in the sense that they actually guarantee the external entity to succeed in its prediction task. In addition, they bring significant simplifications in the procedure discussed in Algorithm 1, thus fully motivating their use. The case involving a generic $\hat{f}_i(\cdot, \theta_i)$, instead, will be discussed in §V-C.

Finally, we remark that the conditions postulated in Standing Assumption 2.2 on the multi-agent interaction process are very general and, as a result, they do not even guarantee the existence of a stationary action profile in the sense of Definition 2.1. As a distinct feature of our approach, we will show that sufficient conditions can be established to assess the asymptotic properties of the proposed learning-based algorithm so that, if convergence happens, it can only be towards a stationary action profile, thus also providing certificates for the existence of the latter.

B. Applications in multi-agent control and decision-making

1) Learning equilibria in generalized Nash games: A generalized Nash equilibrium problem (GNEP) [2] typically involves a population of N agents where each one of them aims at minimizing some cost function $J_i : \mathbb{R}^n \to \mathbb{R}$ that depends both on its own (locally constrained) decision $x_i \in \mathcal{X}_i \subseteq \mathbb{R}^{n_i}$, as well as on the decisions of all the other agents $\boldsymbol{x}_{-i}$. In addition, the selfish agents also compete for shared resources, and are thus required to satisfy coupling constraints, i.e., $(x_i, \boldsymbol{x}_{-i}) \in \Omega \subseteq \mathbb{R}^n$. The resulting game is hence defined by a collection of mutually coupled optimization problems:

$$\forall i \in \mathcal{N} : \begin{cases} \min_{x_i \in \mathcal{X}_i} & J_i(x_i, \boldsymbol{x}_{-i}) \\ \text{s.t.} & (x_i, \boldsymbol{x}_{-i}) \in \Omega. \end{cases} \tag{2}$$

A key notion is represented here by the generalized Nash equilibrium (GNE), which typically identifies a desirable outcome of the noncooperative multi-agent decision-making process. In particular, by introducing $f_i(\boldsymbol{x}_{-i}) = \arg\min_{y_i \in \mathcal{X}_i(\boldsymbol{x}_{-i})} J_i(y_i, \boldsymbol{x}_{-i})$ as the so-called best-response (BR) mapping, single-valued under strict convexity of $x_i \mapsto J_i(x_i, \boldsymbol{x}_{-i})$, and $\mathcal{X}_i(\boldsymbol{x}_{-i}) := \{x_i \in \mathcal{X}_i \mid (x_i, \boldsymbol{x}_{-i}) \in \Omega\}$, a GNE $\boldsymbol{x}^\star$ amounts to a fixed point of the stack of the BR mappings of the agents, i.e., $x_i^\star = f_i(\boldsymbol{x}_{-i}^\star)$ for all $i \in \mathcal{N}$. Note that numerous standard problems in robust and model predictive control [31]–[33] based on min-max optimization $\min_{x_1} \max_{x_2} J(x_1, x_2)$ can be recast in the form in (2), where $J_1(x_1, x_2) = J(x_1, x_2)$, $J_2(x_2, x_1) = -J(x_1, x_2)$, and the set Ω is the Cartesian product between the set of feasible inputs x_1 and the set of possible disturbances x_2.

In this way, a GNE coincides with a stationary action profile as in Definition 2.1, and an external entity aims at reconstructing the BR mappings of the agents to predict a possible outcome of the observed GNEP, i.e., a GNE. Also

in this case, we note that Standing Assumption 2.2 is not sufficient to guarantee the existence of a GNE profile.

2) Multi-agent feedback controller synthesis: Consider a competitive version of traditional decentralized control synthesis problems [34], [35], in which each of the N agents wants to stabilize some prescribed output $y_i \in \mathbb{R}^{n_{y_i}}$ of a global dynamical system, a goal that may be shared with other agents, by manipulating a subvector $u_i \in \mathbb{R}^{n_{u_i}}$ of control inputs. The overall linear time-invariant (LTI) dynamics thus reads as:

$$\begin{cases} z(k+1) = Az(k) + \sum_{i \in \mathcal{N}} B_i u_i(k), \\ y_i(k) = C_i z(k), \text{ for all } i \in \mathcal{N}, \end{cases} \tag{3}$$

where $z \in \mathbb{R}^{n_z}$ denotes the full state vector, $A \in \mathbb{R}^{n_z \times n_z}$ and $B = [B_1 \ \ldots \ B_N] \in \mathbb{R}^{n_z \times n_u}$, $n_u := \sum_{i \in \mathcal{N}} n_{u_i}$, the matrices of the state-update function, with $B_i \in \mathbb{R}^{n_z \times n_{u_i}}$, and $C_i \in \mathbb{R}^{n_{y_i} \times n_z}$ the i-th output matrix. We assume each pair (A, B_i) controllable and (A, C_i) observable, for all $i \in \mathcal{N}$. Note that while the input vector $u = \text{col}((u_i)_{i \in \mathcal{N}}) \in \mathbb{R}^{n_u}$ is partitioned in N subvectors, the output vectors y_i are combinations of the full state z, for example y_i and y_j might contain overlapping subvectors of z. In case each agent adopts a linear feedback controller $u_i(z(k)) = \kappa_i z(k)$, the overall control design then reduces to synthesizing gains $\kappa_i \in \mathbb{R}^{n_{u_i} \times n_z}$. From the i-th agent's perspective, the state evolution in (3) hence turns into:

$$z(k+1) = \underbrace{\left(A + \sum_{j \in \mathcal{N} \setminus \{i\}} B_j \kappa_j \right)}_{=: A_i(\boldsymbol{\kappa}_{-i}) = A_i} z(k) + B_i \kappa_i z(k).$$

In case each agent has full knowledge of the global model (A, B) and is interested in minimizing a standard infinite-horizon cost as local performance index $J_i(\kappa_i, \boldsymbol{\kappa}_{-i}) = \sum_{k=0}^{\infty} (y_i(k)^\top Q_i y_i(k) + u_i^\top(k) R_i u_i(k))$, $R_i \succ 0$ and $Q_i \succcurlyeq 0$, the i-th action-reaction mapping in this case reads as:

$$f_i(\boldsymbol{\kappa}_{-i}) = \Big\{ \kappa_i \in \mathbb{R}^{n_{u_i} \times n_z} \mid \kappa_i = \left(R_i + B_i^\top P_i B_i \right)^{-1} B_i^\top P_i A_i,$$
$$A_i^\top P_i A_i - A_i^\top P_i B_i \left(R_i + B_i^\top P_i B_i \right)^{-1} B_i^\top P_i A_i + \bar{Q}_i = 0,$$
$$P_i \succcurlyeq 0 \Big\} \tag{4}$$

where $\bar{Q}_i := C_i^\top Q_i C_i$. Each agent thus aims at designing a standard linear quadratic regulator (LQR) trying to stabilize only the particular output of interest y_i so that (3) is steered to the origin. Note that whenever a solution to each algebraic Riccati equation (ARE) in (4) exists and is unique, every action-reaction mapping $f_i(\cdot)$ happens to be single-valued.

However, since the evolution of each y_i heavily depends on the linear gains κ_j, $j \in \mathcal{N} \setminus \{i\}$, the outputs y_i's can be partially overlapping, and the weights Q_i are individually chosen, a conflicting scenario among agents' decisions may be triggered. The solution of the underlying multi-agent feedback controller synthesis thus calls for a collective action profile in the spirit of Definition 2.1. Note that, in case such a profile does exist, then it is necessarily stabilizing since (4) requires that, for fixed $\boldsymbol{\kappa}_{-i}$, $(A_i(\boldsymbol{\kappa}_{-i}) + B_i \kappa_i)$ has eigenvalues strictly inside the unit circle for all $i \in \mathcal{N}$. Establishing the existence of a collective action profile a-priori, however, is less obvious.

III. PRELIMINARY RESULTS

We now provide general convergence results under certain assumptions, which will be then further specialized in §IV-D to establish the properties of our active learning scheme.

Let us then consider a generic function $r : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$, evaluated as $r(x, \theta)$ with $x \in \mathbb{R}^n$ and $\theta \in \mathbb{R}^p$, and a feasible set $\mathcal{X} \subseteq \mathbb{R}^n$ satisfying the following conditions:

Assumption 3.1. *The set $\mathcal{X}$ is convex. Moreover, for all $\theta \in \mathbb{R}^p$, $x \mapsto r(x, \theta)$ is a convex function.* $\square$

Assumption 3.2. *The set $\mathcal{X}$ is bounded and nonempty. Moreover, it holds that:*

(i) *For all $x \in \mathcal{X}$, $\theta \mapsto r(x, \theta)$ is convex and differentiable;*
(ii) *For all $\theta \in \mathbb{R}^p$, $x \mapsto r(x, \theta)$ is continuous;*
(iii) *For all $\theta \in \mathbb{R}^p$, the vector $\partial r(x, \theta)/\partial \theta \in \mathbb{R}^p$ of partial derivatives is bounded w.r.t. x.* $\square$

Lemma 3.3. *Let Assumption 3.1 hold true. Then, given any $\theta \in \mathbb{R}^p$, the set $\mathcal{M}(\theta) := \operatorname{argmin}_{y \in \mathcal{X}} r(y, \theta)$ is convex. If also Assumption 3.2 holds true, then the value function $r^\star : \mathbb{R}^p \to \mathbb{R}$, $r^\star(\theta) := \min_{y \in \mathcal{X}} r(y, \theta)$, is continuous.* $\square$

Before proceeding further, we recall some basic definitions from [36] useful for the remainder of this section:

Definition 3.4 (Lower semicontinuity, [36, Def. 1.5]). *A function $h : \mathbb{R}^n \to \bar{\mathbb{R}}$ is* lower semicontinuous *if, for all $\tilde{x} \in \mathbb{R}^n$, $\liminf_{x \to \tilde{x}} h(x) := \lim_{\varepsilon \to 0} \left(\inf_{y \in \mathcal{B}_\varepsilon(\tilde{x})} h(y) \right) \geq h(\tilde{x})$.* $\square$

Definition 3.5 (Level-boundedness, [36, Def. 1.16]). *A function $g : \mathbb{R}^n \times \mathbb{R}^p \to \bar{\mathbb{R}}$ is* level-bounded in x locally uniformly in θ *if, for all $\tilde{\theta} \in \mathbb{R}^p$ and $\alpha \in \mathbb{R}$, there exists a neighbourhood $\mathcal{I} = \mathcal{I}(\tilde{\theta}) \subseteq \mathbb{R}^p$ and a bounded set $\mathcal{S} \subset \mathbb{R}^n$ such that $\{x \in \mathbb{R}^n \mid g(x, \theta) \leq \alpha\} \subset \mathcal{S}$ for all $\theta \in \mathcal{I}$.* $\square$

Lemma 3.6. *Let Assumption 3.1 and 3.2 hold true. Moreover, let $\mathcal{X}$ be a closed set. Then, the function $\bar{r} : \mathbb{R}^n \times \mathbb{R}^p \to \bar{\mathbb{R}}$, defined as $\bar{r}(x, \theta) := r(x, \theta) + \iota_{\mathcal{X}}(x)$ is lower semicontinuous and level-bounded in x locally uniformly in θ.* $\square$

We conclude this section with the following result establishing the convergence of the sequence of minimizers for r with minimum norm, $\{x^k\}_{k \in \mathbb{N}}$. Specifically, given some $\theta^k \in \mathbb{R}^p$ the corresponding element x^k is defined as follows:

$$x^k := \operatorname{argmin}_{y \in \mathbb{R}^n} \left\{ \tfrac{1}{2} \|y\|_2^2 \mid y \in \mathcal{M}(\theta^k) \right\}. \tag{5}$$

Lemma 3.7. *Let Assumption 3.1 and 3.2 hold true, and assume that $\mathcal{X}$ is also closed. Let $\{\theta^k\}_{k \in \mathbb{N}}$ be a sequence so that $\lim_{k \to \infty} \theta^k = \tilde{\theta}$, and let $\mathcal{M}(\tilde{\theta}) = \{\tilde{x}\}$. Then, the sequence $\{x^k\}_{k \in \mathbb{N}}$ generated by (5) is feasible, i.e., $x^k \in \mathcal{X}$ for all $k \in \mathbb{N}$, and satisfies $\lim_{k \to \infty} x^k = \tilde{x}$.* $\square$

The technical results just introduced will be instrumental to establish the asymptotic properties of the active learning scheme presented in the next section. Specifically, we will see in §IV-D how the proposed method meets Assumption 3.1 and 3.2 to exploit the convergence of the sequence of minimum-norm minimizers of a suitable collection of parametric programs.

Algorithm 1: Active learning-based method

Initialization: $\boldsymbol{x}^0 \in \Omega$, $\theta_i^0 \in \mathbb{R}^{p_i}$ for all $i \in \mathcal{N}$

Iteration ($k \in \mathbb{N}_0$)**:**

- $\theta_i^{k+1} \in \operatorname*{argmin}_{\xi_i \in \mathbb{R}^{p_i}} \frac{1}{k} \sum_{t=1}^k \ell_i(x_i^t, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^t, \xi_i)), \forall i \in \mathcal{N}$

- $\hat{\boldsymbol{x}}^{k+1} = \operatorname*{argmin}_{\boldsymbol{y} \in \mathbb{R}^n} \frac{1}{2} \|\boldsymbol{y}\|_2^2$ s.t. $\boldsymbol{y} \in \mathcal{M}(\theta^{k+1})$

- ○ For all $i \in \mathcal{N}$: $x_i^{k+1} = f_i(\hat{\boldsymbol{x}}_{-i}^{k+1})$

IV. ACTIVE LEARNING PROCEDURE AND MAIN RESULTS

We now introduce, discuss and analyze the convergence property of our iterative scheme based on a active learning.

A. Algorithm description

The main steps of our active learning scheme are summarized in Algorithm 1[1], where the black-filled bullets refer to the tasks the external observer is required to perform, while the empty bullet to the one performed by the agents in $\mathcal{N}$.

Thus, at the generic k-th iteration of Algorithm 1, the external entity updates the affine surrogate mappings $\hat{f}_i(\cdot, \theta_i^k)$ including the most recent samples according to the following rule:

$$\theta_i^{k+1} \in \operatorname*{argmin}_{\xi_i \in \mathbb{R}^{p_i}} \frac{1}{k} \sum_{t=1}^k \ell_i(x_i^t, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^t, \xi_i)), \tag{6}$$

where $\ell_i : \mathbb{R}^n \times \mathbb{R}^{p_i} \to \mathbb{R}$ is some *loss function*, typically dictated by the function approximation type adopted $\mathscr{L}$ and the learning problem at hand. Referring to Algorithm 1, we preliminarily define, for all $i \in \mathcal{N}$, $\operatorname{argmin}_{\xi_i \in \mathbb{R}^{p_i}} (1/k) \sum_{t=1}^k \ell_i(x_i^t, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^t, \xi_i)) =: \Theta_i(\{(x_i^t, \hat{\boldsymbol{x}}_{-i}^t)\}_{t=1}^k) = \Theta_i^k$. By referring to Fig. 1, the vector $\hat{\boldsymbol{x}}^t$ hence denotes the query point employed by the external entity at the t-th iteration to gather all the best responses $f_i(\hat{\boldsymbol{x}}_{-i}^t)$.

Successively, by relying on the updated proxies for the agents' action-reaction mappings $\hat{f}_i$, the external entity chooses the next query point $\hat{\boldsymbol{x}}^{k+1}$ as the minimum norm strategy profile in the set $\mathcal{M} : \mathbb{R}^p \to 2^\Omega$, $p := \sum_{i \in \mathcal{N}} p_i$, formally defined as:

$$\mathcal{M}(\theta^{k+1}) := \operatorname*{argmin}_{\boldsymbol{y} \in \Omega} \sum_{i \in \mathcal{N}} \left\| y_i - \hat{f}_i(\boldsymbol{y}_{-i}, \theta_i^{k+1}) \right\|_2^2, \tag{7}$$

which contains all collective profiles that are the closest (according to the squared Euclidean norm, although different metrics could be used) to a fixed point of each $\hat{f}_i(\cdot, \theta_i^{k+1})$. This step is motivated by the notion of stationary action profile in Definition 2.1, which coincides with a fixed point of the stack of the agents' mappings, i.e., $x_i^\star = f_i(\boldsymbol{x}_{-i}^\star)$ for all $i \in \mathcal{N}$. Indeed, if $\hat{f}_i$ were exactly equal to f_i and the minimum in (7) were zero, any $\boldsymbol{x}^\star \in \mathcal{M}(\theta^{k+1})$ would be a stationary action profile. Note that, by referring to (7), θ^{k+1} identifies the whole collection of parameters $\{\theta_i^{k+1}\}_{i \in \mathcal{N}}$ characterizing the surrogate mappings, which at every iteration represent the argument of the corresponding parameter-to-query mapping $\mathcal{M}(\cdot)$. In the spirit of the results developed in §III, the cost function in (7)

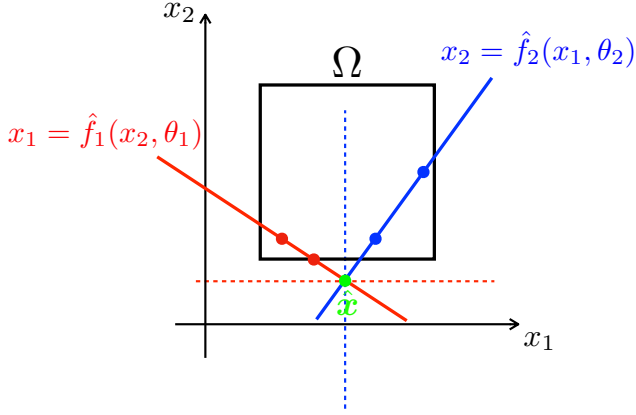[1]A Python package with its implementation is publicly available at `https://github.com/bemporad/gnep-learn`.

Fig. 2: With $N = 2$, $n_1 = n_2 = 1$, and affine $\hat{f}_i$'s constructed using feasible samples (red and blue dots), solving (9) would return the green point as a unique minimizer outside Ω (black box). While agent 2 can still provide a feasible reaction to this infeasible query point (decision to be made along the dashed blue line), agent 1 can not (decision along the dashed red line).

then takes on the same role as the generic function $r(\cdot)$ in that section, thus establishing an immediate connection between (7) and the point-to-set mapping $\mathcal{M}(\cdot)$ defined in Lemma 3.3.

By considering the minimization problem in (7), we note that the special choice for the surrogate mappings in (1) is particularly convenient, since (7) turns out to be a constrained LS problem, which is convex whenever Ω (or a conservative subset of it) is so. To simplify notation, assume $n_i = 1$ for all $i \in \mathcal{N}$ so that $\Lambda_i = [\nu_i^\top \; c_i]$ in (1) (the extension to the case $n_i \geq 2$ follows readily). Then, the program in (7) becomes:

$$\min_{\boldsymbol{y} \in \Omega} \sum_{i \in \mathcal{N}} \left\| y_i - (\nu_i^{k+1})^\top \boldsymbol{y}_{-i} - c_i^{k+1} \right\|_2^2. \tag{8}$$

Moreover, in the absence of constraints ($\Omega = \mathbb{R}^n$), we can further characterize the stationary action profile we wish to compute. In fact, problem (7) can be solved by imposing $y_i - \hat{f}_i(\boldsymbol{y}_{-i}, \theta_i) = y_i - \nu_i^\top \boldsymbol{y}_{-i} - c_i = 0$ for all $i \in \mathcal{N}$, therefore getting the following system of N equations with N unknowns:

$$\begin{bmatrix} 1 & & -\nu_1^\top \\ \vdots & \ddots & \vdots \\ -\nu_N^\top & & 1 \end{bmatrix} \boldsymbol{y} = \begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix}, \tag{9}$$

where the superscript $^{k+1}$ has been omitted for simplicity. In particular, if (9) is solvable, every solution lies on a subspace $\mathcal{M}(\theta^{k+1})$ of dimension smaller than N (a single point when it is unique), and the vector $\hat{\boldsymbol{x}}^{k+1}$ corresponds to the minimum-norm solution of (9). When $\Omega \subset \mathbb{R}^n$, one might still be tempted to solve (9) to get the next query point $\hat{\boldsymbol{x}}^{k+1}$. However, further queries may return infeasible reactions by some of the agents, as shown in the pictorial representation in Fig. 2.

Once obtained the minimum norm vector $\hat{\boldsymbol{x}}^{k+1}$ in (7), this serves as a query profile to collect the next agents' reaction data, $x_i^{k+1} = f_i(\hat{\boldsymbol{x}}_{-i}^{k+1})$. Within the final step of the procedure, indeed, $\hat{\boldsymbol{x}}^{k+1}$ is broadcasted to the agents, which on their side react to $\hat{\boldsymbol{x}}^{k+1}$ by computing $f_i(\cdot)$, and finally return the results

to the external entity. Therefore, each x_i^{k+1} enriches the training dataset used in (6) to improve the affine model $\hat{f}_i(\cdot, \theta_i^{k+1})$.

B. Initialization

As commonly done in most active learning algorithms [3], [4], [37], the execution of Algorithm 1 is preceded by a (passive) random data collection phase to make an initial estimate of the affine surrogates $\hat{f}_i$. During this initialization procedure, the external entity draws iteratively a random feasible point $\boldsymbol{x}^k \in \Omega$ as follows. First, it forms a random vector $\boldsymbol{x}_r^k \in \mathbb{R}^n$ by extracting its components $x_{r_i}^k$ from distribution $\mathcal{U}(m_i^-, m_i^+)$ where $m^- \leq \boldsymbol{x} \leq m^+$ define a range of plausible values of $\boldsymbol{x}$. Then, the following constrained LS problem

$$\boldsymbol{x}^k = \operatorname*{argmin}_{\boldsymbol{y} \in \Omega} \|\boldsymbol{y} - \boldsymbol{x}_r^k\|_2^2, \tag{10}$$

is solved for $k = 1, \ldots, K_{\text{in}}$ to define $\boldsymbol{x}^k$ and pass it in parallel to each agent, which in turn responds through the private action-reaction mapping f_i. The resulting actions are collected by the external entity to update its estimates θ_i^k. The remaining $K - K_{\text{in}}$ iterations of Algorithm 1 are then executed, where K is the total number of queries performed to the agents. Alternatively, more sophisticated initialization strategies are possible, such as Latin hypercube sampling [38], [39] as in, e.g., [4], however, significant differences are not expected in our framework as we only learn *linear* proxies rather than global surrogates of the agents' responses. Nevertheless, given its key role we will discuss in detail the random initialization procedure within each numerical example reported in §VI.

C. Complexity analysis

We analyze the complexity of Algorithm 1 in the affine setting (1) along with the mean squared error (MSE) loss

$$\ell_i(x_i^t, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^t, \xi_i)) = \|x_i^t - \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^t, \xi_i)\|_2^2, \tag{11}$$

which make (6) a linear LS problem and the acquisition problem (7) either a linear system as in (9) ($\Omega = \mathbb{R}^n$), or a constrained linear LS as in (8) ($\Omega \subset \mathbb{R}^n$ is a polyhedron), both having N unknowns. Hence, the complexity of solving (7) or (10) is either the complexity of solving a linear system of N equations or, in the constrained case, a linear LS problem with N residuals and as many inequalities as in the hyperplane representation of Ω, multiplied by the number K of iterations. The memory size required to store the model parameters and form problem (7) is therefore $\sum_{i=1}^n n_i(n_{-i} + 1)$.

Due to the monotonically increasing number of training data k, we consider simple recursive methods to update the estimate θ_i^{k+1} in (6), as described for instance in [40], [41], such as recursive LS [42]. To control the learning rate of the algorithm, we adopt the following linear Kalman filter iterations [43] tailored for $n_i = 1$ (see Appendix B for the derivation):

$$\phi_i^k = \begin{bmatrix} \hat{\boldsymbol{x}}_{-i}^k \\ 1 \end{bmatrix}, \; a_i^k = P_i^k \phi_i^k \tag{12a}$$

$$P_i^{k+1/2} = P_i^k - \frac{a_i^k(a_i^k)^\top}{1 + (\phi_i^k)^\top a_i^k} \tag{12b}$$

$$\theta_i^{k+1} = \theta_i^k + P_i^{k+1/2}\phi_i^k(x_i^k - (\phi_i^k)^\top \theta_i^k) \tag{12c}$$

$$P_i^{k+1} = P_i^{k+1/2} + \beta I \tag{12d}$$

For $n_i > 1$, a bank of Kalman filters as in (12) is run in parallel for each of the n_i components of $\hat{f}_i$. The complexity of solving (6) to update the parameter vector θ is hence $K \sum_{i=1}^{n} n_i O(n_{-i}^2)$ and requires $\sum_{i=1}^{n} n_i(n_{-i}+1)(n_{-i}+2)/2$ additional numbers to store the symmetric covariance matrices.

In (12) $\beta \geq 0$ defines the learning rate of the filter, and the initial value of the N-by-N covariance matrix $P_i^0 = \alpha I$ the amount of ℓ_2-regularization on θ_i (cf. [44, Eq. (12)]). In particular, the larger the β, the faster is the convergence of the linear estimator (we obtain standard recursive LS updates for $\beta = 0$ [40], [41]), while the smaller the α, the larger is the ℓ_2-regularization. We will adopt the above setting in all the numerical simulations reported in §VI.

D. Asymptotic properties

Before characterizing the asymptotic properties of the active learning procedure in Algorithm 1, we first postulate some assumptions on the learning procedure $\mathscr{L}$ the external observer is endowed with and then establish some results that will be instrumental for the main statement of the paper.

Standing Assumption 4.1 (Training loss). *For all $i \in \mathcal{N}$, the loss function ℓ_i is continuous in both arguments and, for all $(x_i, \boldsymbol{x}_{-i}, \theta_i) \in \Omega \times \mathbb{R}^{p_i}$, $0 \leq \ell_i(x_i, \hat{f}_i(\boldsymbol{x}_{-i}, \theta_i)) < \infty$, with $\ell_i(x_i, \hat{f}_i(\boldsymbol{x}_{-i}, \theta_i)) = 0$ if and only if $x_i = \hat{f}_i(\boldsymbol{x}_{-i}, \theta_i)$.* $\square$

The conditions in Standing Assumption 4.1 allow us to characterize the choice of the loss functions adopted in (6).

Our technical analysis will be based on the possibility of matching *pointwise* (and, therefore, not necessarily globally nor even locally) the action-reaction mapping f_i of each agent, which is a key property possessed by the affine surrogates in (1), which can hence be used with no restrictions. Specifically, for all $i \in \mathcal{N}$ and for all $(x_i, \boldsymbol{x}_{-i}) \in \Omega$, there exists a set $\mathcal{A}_i = \mathcal{A}_i(x_i, \boldsymbol{x}_{-i}) \subseteq \mathbb{R}^{p_i}$ such that $\ell_i(x_i, \hat{f}_i(\boldsymbol{x}_{-i}, \tilde{\theta}_i)) = 0$, for all $\tilde{\theta}_i \in \mathcal{A}_i$. While this condition is intrinsic for affine surrogates, it shall therefore be assumed to hold true in case of generic proxies $\hat{f}_i(\cdot, \theta_i)$ within the discussion in §V-C. Moreover, in view of the specific structure (1) we consider, the affine proxies are also continuous with respect to their second argument, i.e., $\hat{f}_i(\boldsymbol{x}_{-i}, \cdot)$ is continuous for all $\boldsymbol{x}_{-i} \in \mathbb{R}^{n-i}$.

The crucial result stated next requires the external entity to solve (6) at the global minimum at every iteration. In case there exist multiple (global) minimizers we then tacitly assume further the external observer being endowed with a tailored *tie-break rule*, i.e., some single-valued mapping $\mathcal{T} : 2^{\mathbb{R}^{p_i}} \to \mathbb{R}^{p_i}$ used consistently across iterations, such as picking up the solution θ_i^k with minimum norm at every $k \in \mathbb{N}$, to single-out one of those minimizers, for all $i \in \mathcal{N}$. With this tie-break rule $\mathcal{T}$ in place, which takes a set of vectors in $\mathbb{R}^{p_i}$ as argument and returns one element from it according to some criterion, the first step in Algorithm 1 performed by the external observer turns into an equality rather than an inclusion.

Lemma 4.2. *For all $i \in \mathcal{N}$, let $\{(x_i^t, \hat{\boldsymbol{x}}_{-i}^t)\}_{t=1}^{k}$ be a collection of samples to be employed in the update rule for the parameter θ_i in (6), where each $(x_i^t, \hat{\boldsymbol{x}}_{-i}^t) \in \Omega$. Assume that, for every $k \in \mathbb{N}$, (6) is solved at the global minimum and a tie-break rule $\mathcal{T}$ is applied to single-out a minimizer. If $\lim_{k\to\infty} x_i^k = \bar{x}_i$,*

$\lim_{k\to\infty} \hat{\boldsymbol{x}}_{-i}^k = \bar{\boldsymbol{x}}_{-i}$ *so that $(\bar{x}_i, \bar{\boldsymbol{x}}_{-i}) \in \Omega$, and $\lim_{k\to\infty} \theta_i^k = \bar{\theta}_i$, then $\bar{\theta}_i \in \mathcal{A}_i(\bar{x}_i, \bar{\boldsymbol{x}}_{-i})$, for all $i \in \mathcal{N}$.* $\square$

Lemma 4.2 shows a sort of "consistency property" of the affine surrogates $\hat{f}_i$, meaning that if all the ingredients involved in Algorithm 1 happen to converge, then the pointwise approximation of f_i shall be exact at $\boldsymbol{x}_{-i}$, i.e., $\hat{f}_i(\boldsymbol{x}_{-i}, \tilde{\theta}_i) = f_i(\boldsymbol{x}_{-i})$.

Standing Assumption 4.3 (Feasible set Ω). *The set Ω is a nonempty polytope.* $\square$

We now characterize the properties of the sequence of query points $\{\hat{\boldsymbol{x}}^k\}_{k\in\mathbb{N}}$ produced by the central entity in the second step of Algorithm 1 by computing the optimal solution to:

$$\hat{\boldsymbol{x}}^k := \operatorname{argmin}_{\boldsymbol{y}\in\mathbb{R}^n} \left\{ \tfrac{1}{2}\|\boldsymbol{y}\|_2^2 \mid \boldsymbol{y} \in \mathcal{M}(\theta^k) \right\}, \quad (13)$$

for all $k \in \mathbb{N}$, thus mirroring the strategy to generate x^k in (5).

Proposition 4.4. *Let $\{\theta^k\}_{k\in\mathbb{N}}$ be a sequence so that $\lim_{k\to\infty} \theta_i^k = \tilde{\theta}_i$ for all $i \in \mathcal{N}$, and let $\mathcal{M}(\tilde{\theta}) = \{\tilde{\boldsymbol{x}}\}$. Then the sequence $\{\hat{\boldsymbol{x}}^k\}_{k\in\mathbb{N}}$ generated by (13) is feasible and satisfies $\lim_{k\to\infty} \hat{\boldsymbol{x}}^k = \tilde{\boldsymbol{x}}$.* $\square$

We are now ready to state the asymptotic properties of the active learning methodology described in Algorithm 1:

Theorem 4.5. *Let $\{\theta^k\}_{k\in\mathbb{N}}$ be a sequence so that $\lim_{k\to\infty} \theta_i^k = \tilde{\theta}_i$ for all $i \in \mathcal{N}$, and let $\mathcal{M}(\tilde{\theta}) = \{\tilde{\boldsymbol{x}}\}$. For every $k \in \mathbb{N}$, let (6) be solved at the global minimum and a tie-break rule $\mathcal{T}$ applied to single-out a minimizer. Then, $\lim_{k\to\infty} \|\hat{\boldsymbol{x}}^k - x^k\|_2 = 0$, and the sequence $\{x^k\}_{k\in\mathbb{N}}$ and $\{\hat{\boldsymbol{x}}^k\}_{k\in\mathbb{N}}$ generated by Algorithm 1 converge to the same stationary action profile.* $\square$

As a consequence of Theorem 4.5, in case the convergence of the parametric estimates happens, it can only be towards the true (at least pointwise) values. This fact, together with $\lim_{k\to\infty} \|\hat{\boldsymbol{x}}^k - x^k\|_2 = 0$, yields $\sum_{i\in\mathcal{N}} \|\tilde{x}_i - \hat{f}_i(\tilde{\boldsymbol{x}}_{-i}, \tilde{\theta}_i)\|_2^2 = \sum_{i\in\mathcal{N}} \|\tilde{x}_i - f_i(\tilde{\boldsymbol{x}}_{-i})\|_2^2 = 0 = \sum_{i\in\mathcal{N}} \|f_i(\bar{\boldsymbol{x}}_{-i}) - \bar{x}_i\|_2^2 = 0$, i.e., $\|f_i(\bar{\boldsymbol{x}}_{-i}) - \bar{x}_i\|_2 = 0$, i.e., the external entity has in its hands both a stationary action profile $\tilde{\boldsymbol{x}}$, and pointwise-exact surrogates of the action-reaction mappings $\hat{f}_i(\tilde{\boldsymbol{x}}_{-i}, \tilde{\theta}_i) = f_i(\tilde{\boldsymbol{x}}_{-i})$.

Note that the assumptions postulated in §II-A are not sufficient to guarantee the existence of a vector $\boldsymbol{x}^\star$ such that $\|x_i^\star - f_i(\boldsymbol{x}_{-i}^\star)\|_2 = 0$ for all $i \in \mathcal{N}$. The following corollary offers a certificate for the existence of at least a stationary action profile, and it is an immediate consequence of Theorem 4.5.

Corollary 4.6. *Under the conditions stated in Theorem 4.5, the multi-agent interaction process at hand admits a stationary action profile $\boldsymbol{x}^\star$ in the sense of Definition 2.1.* $\square$

V. TECHNICAL DISCUSSION AND POSSIBLE EXTENSIONS

Next, we elaborate around the conditions granting the asymptotic properties in Theorem 4.5, along with possible extensions to a more general framework or surrogate mappings.

A. Discussion on Theorem 4.5

Assuming that $\lim_{k\to\infty} \theta_i^k = \tilde{\theta}_i$ for all $i \in \mathcal{N}$, along with the existence of a single minimizer contained in $\mathcal{M}(\tilde{\theta})$ (7), appear rather strong requirements to achieve the convergence of the proposed active learning scheme that, to the best of

our knowledge, can not be guaranteed beforehand, not even by imposing further structure on the learning procedure $\mathscr{L}$ or the multi-agent problem at hand. Both can indeed be verified only a posteriori or in practice after a large enough number of iterations of Algorithm 1. As a further motivation for adopting the affine surrogates in (1), note that checking whether $\mathcal{M}(\theta)$ is a singleton translates into verifying the positive definiteness of the Hessian matrix associated to the cost function in (8).

We note that, however, the technical conditions assumed in Standing Assumption 2.2 are so general that they do not even seem to be sufficient to establish the existence of a stationary action profile for the underlying problem. This should be put together with the fact that the external observer: i) has very little knowledge of such a process, ii) can be endowed with any learning procedure $\mathscr{L}$ satisfying Standing Assumption 4.1, and iii) could succeed in its prediction task even with a simple, i.e., affine, parametrization for the surrogate mappings $\hat{f}_i$.

We point to §VI for many numerical results in which the a-posteriori verification of the convergence of the parametric estimates yielding a single minimizer in $\mathcal{M}(\cdot)$ succeeds, thus demonstrating the practical effectiveness of our methodology.

B. Learning composition of action-reaction mappings

Consider the case in which the decision variable x_i does not follow directly as a result of some reaction process through $f_i(\cdot)$, but it coincides with a function of some other inner variable, say $w_i \in \mathcal{W}_i \subseteq \mathbb{R}^{m_i}$ (in this new framework, f_i should be suitably redefined taking $\mathbb{R}^{m_i}$ as codomain). By introducing $g_i : \mathbb{R}^{m_i} \to \mathbb{R}^{n_i}$ as an additional, private mapping we have that x_i is obtained through the composition $(g_i \circ f_i)(\boldsymbol{x}_{-i})$, i.e.,

$$x_i = g_i(w_i) = g_i(f_i(\boldsymbol{x}_{-i})).$$

This different perspective makes the reaction process to $\boldsymbol{x}_{-i}$ indirect. As a supporting example for this scenario, consider a population of agents obeying to the following dynamics:

$$\forall i \in \mathcal{N} : \begin{cases} z_i(k+1) = d_i(z_i(k), u_i(k)) + s_i(\boldsymbol{v}_{-i}(k), \boldsymbol{u}_{-i}(k)), \\ v_i(k) = t_i(z_i(k), u_i(k)), \end{cases}$$
(14)

where $z_i \in \mathbb{R}^{n_{z_i}}$, $u_i \in \mathbb{R}^{n_{u_i}}$, and $v_i \in \mathbb{R}^{n_{v_i}}$ denote the state vector, the control input, and the controlled output of each agent, respectively, whose evolution (except for u_i, which has to be designed) is determined by mappings $d_i : \mathbb{R}^{n_{z_i}} \times \mathbb{R}^{n_{u_i}} \to \mathbb{R}^{n_{z_i}}$ and $t_i : \mathbb{R}^{n_{z_i}} \times \mathbb{R}^{n_{u_i}} \to \mathbb{R}^{n_{v_i}}$. The behaviour of the state variables, however, is also affected by some function $s_i : \mathbb{R}^{n_{u_{-i}}} \times \mathbb{R}^{n_{v_{-i}}}$ of the other agents' control inputs and outputs. Planning an optimal control strategy over some horizon of length $T \geq 1$ traditionally requires each agent to find a solution to the following optimal control problem:

$$\forall i \in \mathcal{N} : \begin{cases} \min_{w_i} & \Psi_i(z_i(T)) + \sum_{k=0}^{T-1} \Phi_i(x_i(k), \boldsymbol{x}_{-i}(k), k) \\ \text{s.t.} & z_i(0) = z_{i,0}, \ (14), \ \forall k = 1, \ldots, T-1, \\ & x_i(k) \in X_i, \ \forall k = 0, \ldots, T-1, \end{cases}$$
(15)

where $z_{i,0} \in \mathbb{R}^{n_{z_i}}$ is the measured initial state, $w_i(k) = \mathrm{col}(u_i(k), z_i(k+1))$, $w_i := \mathrm{col}((w_i(k))_{k=0}^{T-1}) \in \mathbb{R}^{n_{w_i}}$, $n_{w_i} :=$

$T(n_{u_i} + n_{z_i})$, $x_i(k) := \mathrm{col}(u_i(k), v_i(k)) \in \mathbb{R}^{n_{u_i}+n_{v_i}}$, with $x_i := g(w_i) = \mathrm{col}((x_i(k))_{k=0}^{T-1}) \in \mathbb{R}^{n_i}$, $n_i := T(n_{u_i} + n_{v_i})$, $g : \mathbb{R}^{n_{w_i}} \to \mathbb{R}^{n_i}$, $X_i \subseteq \mathbb{R}^{n_{u_i}} \times \mathbb{R}^{n_{v_i}}$ denotes the input-output constraint set, $\Phi_i : \mathbb{R}^{n_{u_i}+n_{v_i}} \times \mathbb{R}^{n_{x-i}} \times \mathbb{N} \to \mathbb{R}$ is the i-th stage cost that depends on k to include a possible reference output signal to track, and $\Psi_i : \mathbb{R}^{n_{z_i}} \to \mathbb{R}$ is a terminal cost.

The exogenous term s_i in (14) thus makes (15) a collection of optimization problems with coupling terms dependent on $\boldsymbol{x}_{-i}$ both in the cost and constraints, whose overall solution, if any, naturally calls for a collective action profile in the spirit of Definition 2.1, which in this generalized setting reads as:

$$\boldsymbol{x}^\star \in \Omega \text{ s.t., for all } i \in \mathcal{N}, \ x_i^\star = g_i(f_i(\boldsymbol{x}_{-i}^\star)).$$

The agnostic external entity will therefore collect data by querying the agents with tailored opponents' input/output profiles, and then update proxies for each composition $(g_i \circ f_i)$ to predict the optimal control actions adopted by the set of agents over the T-steps long horizon.

From a merely technical perspective, to let our machinery work also in this more general setting, the conditions on $f_i(\cdot)$ in Standing Assumption 2.2 shall be replaced with the following:

i) Each mapping $(g_i \circ f_i)(\cdot)$ is continuous and single-valued;

ii) For all $\boldsymbol{x}_{-i} \in \mathbb{R}^{n-i}$, $f_i(\boldsymbol{x}_{-i})$ produces some $w_i \in \mathcal{W}_i$ so that $(g_i(w_i), \boldsymbol{x}_{-i}) \in \Omega$.

Then, nothing prevents the external entity from adopting the same affine surrogate mappings (1) to estimate each composition $(g_i \circ f_i)(\cdot)$, whereas the final step of the active learning scheme in Algorithm 1 will require to simply collect $x_i^{k+1} = g_i(f_i(\hat{\boldsymbol{x}}_{-i}^{k+1}))$ for all $i \in \mathcal{N}$. The technical analysis on the asymptotic properties of the variant of Algorithm 1 just discussed then mimics the one carried out in §IV-D.

C. Generic surrogate mappings $\hat{f}_i$

Adopting the affine parametrization in (1) for the action-reaction proxies has been shown beneficial to satisfy the assumptions postulated so far, as well as to simplify the tasks in Algorithm 1 the central entity has to perform. More importantly, in view of the analysis in §IV-D, such simple surrogates can be used with no restrictions, since they actually guarantee the external entity to succeed in its prediction task.

In case, however, one is interested in more flexible surrogate mappings $\hat{f}_i$, some further conditions have to be imposed. Specifically, by mirroring the discussion in §IV-D tailored for affine proxies, one shall necessarily integrate Standing Assumption 4.1 with a requirement on the continuity of the mapping $\theta_i \mapsto \hat{f}_i(\boldsymbol{x}_{-i}, \theta_i)$ for all $\boldsymbol{x}_{-i} \in \mathbb{R}^{n-i}$. Nevertheless, the proxies $\hat{f}_i$ shall also be able to match at least pointwise the action-reaction mapping f_i of each agent, thus requiring that for all $(x_i, \boldsymbol{x}_{-i}) \in \Omega$, there exists a set $\mathcal{A}_i = \mathcal{A}_i(x_i, \boldsymbol{x}_{-i}) \subseteq \mathbb{R}^{p_i}$ such that $\ell_i(x_i, \hat{f}_i(\boldsymbol{x}_{-i}, \tilde{\theta}_i)) = 0$, for all $\tilde{\theta}_i \in \mathcal{A}_i$. These two conditions together are key to prove the consistency property in Lemma 4.2 also with generic surrogate mappings. In addition, to rely on Proposition 4.4 one should also ensure that the chosen proxies $\hat{f}_i$ allow meeting Assumption 3.1 and Assumption 3.2, which characterize the second task performed by the external entity in Algorithm 1, so that the arguments made in the proof of Theorem 1 could be applied also to this more generic case.

TABLE I: Charging EVs – Simulation parameters

Parameters	Description	Value
T	Time interval	14
N	Number of EVs	10
q_i	Degradation cost – quadratic term	$\sim \mathcal{U}(0.006, 0.01)$
c_i	Degradation cost – affine term	$\sim \mathcal{U}(0.055, 0.095)^T$
d	Normalized inflexibility demand	from [10, Fig. 1]
γ_i	Local charging requirement	$\sim \mathcal{U}(1.2, 1.8)$
$\bar{c}_i$	Upper bound - power injection	0.25
$\bar{c}$	Grid capacity	0.2
K	Number of iterations (Alg. 1)	200
K_{in}	% of K - initialization of Alg. 1	20
β	Learning rate (Kalman filter)	1
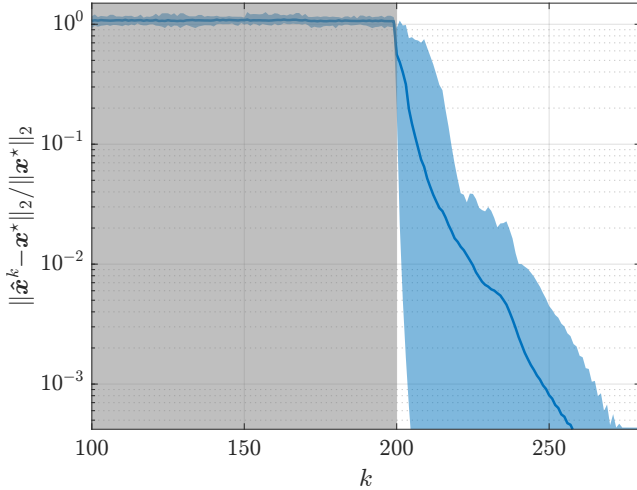
Fig. 3: Sequence $\{\hat{\boldsymbol{x}}^k\}_{k\in\mathbb{N}}$, averaged over 25 numerical instances (solid blue line). The shaded black region corresponds to the passive random data collection phase, which is reported for the interval $k \in [100, 200]$ only for illustrative purposes.

A main drawback of using nonlinear surrogates is that the cost function in problem (7) may easily become nonconvex, unless combined with suitable loss functions. For example, a convex program equivalent to (7) might be obtained by adopting softmax surrogate mappings and cross-entropy losses, in place of the Euclidean norm, to model categorical decision variables. This is however a topic of current investigation.

VI. NUMERICAL RESULTS

We now test the practical effectiveness of the active learning scheme in Algorithm 1 on several numerical instances of the multi-agent control and decision-making problems described in §I-A, §II, along with comparing its performance with the state-of-the-art centralized method for GNE computation in [45]. We adopt the affine setting (1) with MSE loss (11) detailed in §IV-C. All simulations are run in MATLAB on a laptop with an Apple M2 chip featuring an 8-core CPU and 16 GB RAM. Some of the examples analyzed in this section are also included in the Python package gnep-learn, publicly available at https://github.com/bemporad/gnep-learn.

A. Motivating example revisited

Consider again the indirect control problem for smart grids presented in §I-A. A DSO or an energy retailer (i.e., an
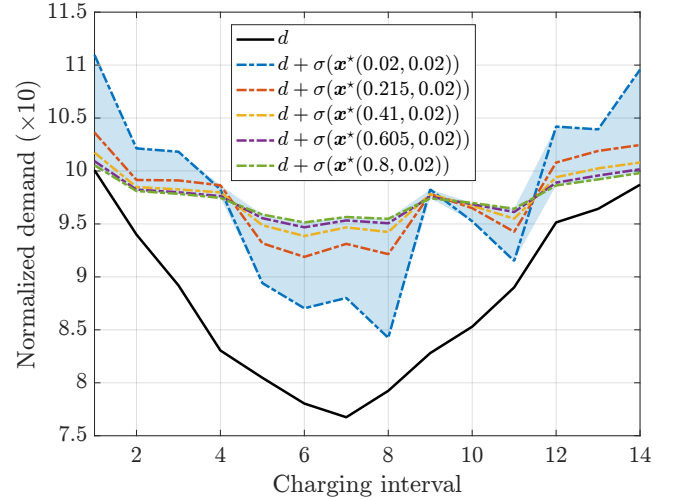

Fig. 4: Sum between the normalized average inflexible demand d and a sample of five charging strategies $(1/N)\sum_{i\in\mathcal{N}} x_i^\star(\bar{a}, \bar{b})$ at the equilibrium, for given price signals $\bar{a}$, $\bar{b} > 0$ (dashed dotted coloured lines). The shaded blue area denotes the union over all the 25 experiments performed.

external observer) wants to make accurate forecasts on the aggregate electricity consumption of end-users in response to price-signals, aimed at enabling flexibility offered by the users themselves. We want to analyze the effect of price signals a, $b > 0$ on the aggregate consumption $\sigma(\boldsymbol{x}^\star(a,b))$ of a fleet of EVs over a certain period T, and the possibility to unlock the so-called "valley filling" phenomenon [10]. We then consider the following collection of optimization problems:

$$\forall i \in \mathcal{N} : \begin{cases} \min_{x_i} & q_i x_i^\top x_i + c_i^\top x_i + (a(\sigma(\boldsymbol{x}) + d) + b\mathbf{1}_T)^\top x_i \\ \text{s.t.} & \mathbf{1}_T^\top x_i \geq \gamma_i, \ x_i \in [0, \bar{x}_i]^T, \\ & \sigma(\boldsymbol{x}) \leq \bar{c}, \end{cases}$$

(16)

where we have used the values reported in Tab. I to run numerical experiments. In particular, we have run 25 instances of (16) obtained from all the possible combinations of five values for both a and b chosen equally spaced within the set $[0.02, 0.8]$. In all the considered cases, we verified ex-post that the minimum eigenvalue of the Hessian matrix H associated with the cost in (8) satisfies $\lambda_{\min}(H) \geq 4.8 \times 10^{-2}$.

Then, Fig. 3 reports the averaged convergent behaviour of the sequence of queries $\{\hat{\boldsymbol{x}}^k\}_{k\in\mathbb{N}}$ produced by Algorithm 1. Note that we usually obtain an equilibrium $\boldsymbol{x}^\star$, computed through a standard extragradient type method [46], in less than the $K = 200$ iterations allocated. In Fig. 4, instead, we show the effect that different price signals have on the aggregate charging strategy of the fleet of EVs. Specifically, we note that acting on the parameters a, b produces a fine control of the aggregate EVs consumption, increasing demand during off-peak periods, thereby producing a desirable valley-filling phenomenon.

Therefore, our active learning procedure represents a key tool for a DSO or an energy retailer in forecasting the aggregate electricity consumption of private customers in response to price-signals, in order to enable for the flexibility offered by the users themselves, or eventually maximize their profits.
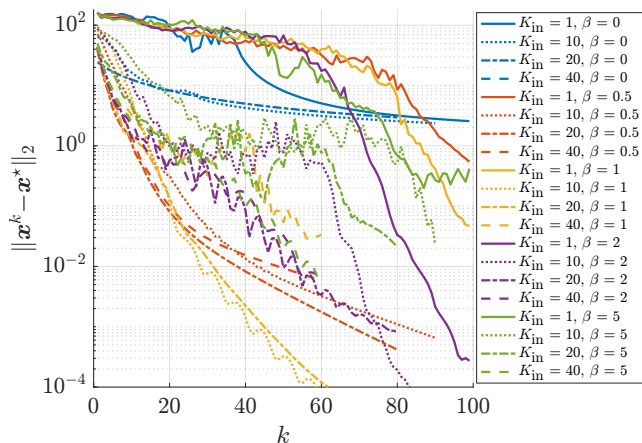
Fig. 5: Hyperparameters analysis on the Nash equilibrium problem formalized in [47, Ex. 1].

B. Generalized Nash games

For this multi-agent decision-making problem we consider examples from [47], [48] to cover several cases of interest.

Specifically, [47, Ex. 1] considers $N = 10$ agents with cost functions $J_i(x_i, \boldsymbol{x}_{-i}) = N(1 + (i-1)/2)x_i - x_i(60N - \mathbf{1}_N^\top \boldsymbol{x})$, while $\Omega = [7, 100]^N$. We first perform a numerical analysis to set the two main hyperparameters affecting the execution of Algorithm 1, namely the learning coefficient β and the number K_{in} of iterations during the data collection phase. The selected values will be adopted also for the remaining numerical instances considered in this subsection. Fig. 5 illustrates the impact those hyperparameters have on the convergence of Algorithm 1 to an equilibrium of the underlying game (computed through a standard extragradient type method [46]), averaged over 20 random initialization procedures for each coefficients combination. In particular, we have considered $\beta \in \{0, 0.5, 1, 2, 5\}$ and $K_{\text{in}} \in \{0.01, 0.1, 0.2, 0.4\} \cdot K$, with $K = 100$. Different rules for selecting K_{in} can be adopted, e.g., choosing K_{in} as a function of the dimension $n_{-i} - 1$ of θ_i.

What we can infer from Fig. 5 is that while, on the one hand, it seems that some long enough random initialization procedure is indeed required (solid lines corresponding to $K = 1$ are associated with the slowest behaviours), on the other hand, an exceedingly large learning rate (green and violet lines) does not appear to bring significant benefit in terms of rate of convergence. We will thus use $\beta = 1$ and $K_{\text{in}} = \lceil K/10 \rceil$.

The performance of the active learning-based approach in predicting an equilibrium solution of the considered quadratic game with no coupling constraints is illustrated in Fig. 6. In particular, the top plot shows the behaviours of the query points $\hat{\boldsymbol{x}}^k$ computed iteratively by the external entity to collect information on the agents' BR mappings. Remarkably, the approximation accuracy of the BR mapping surrogates increases over the iterations, yielding $\lim_{k \to \infty} \|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2 = 0$, and therefore allowing the external entity to practically succeed in its prediction task in less than 20 iterations. These considerations are motivated by the bottom plot in Fig. 6 that reports the convergent behaviour of θ_1^k (the other parameter vectors have a similar evolution), thus supporting Theorem 4.5 numerically.
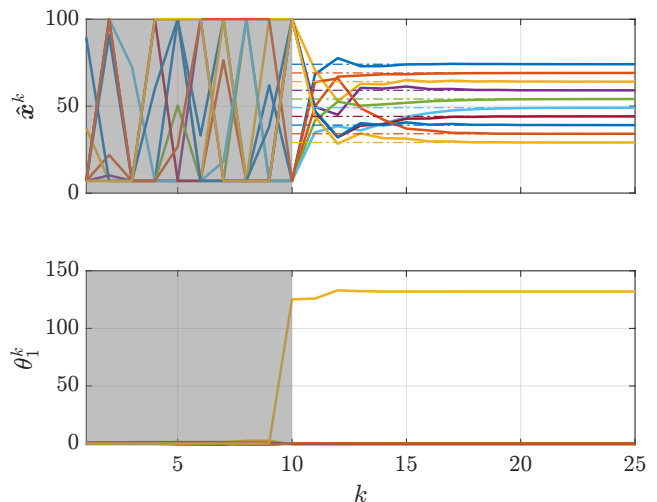

Fig. 6: Nash equilibrium problem formalized in [47, Ex. 1]. The shaded regions correspond to the random data collection phase. Top: sequence $\{\hat{\boldsymbol{x}}^k\}_{k \in \mathbb{N}}$ (solid lines) converging to an equilibrium (dashed lines). Bottom: convergent behaviour of $\{\theta_i^k\}_{k \in \mathbb{N}}$ to reconstruct the BR mapping of agent 1. Only the first 25 samples out of the allocated $K = 100$ are reported.
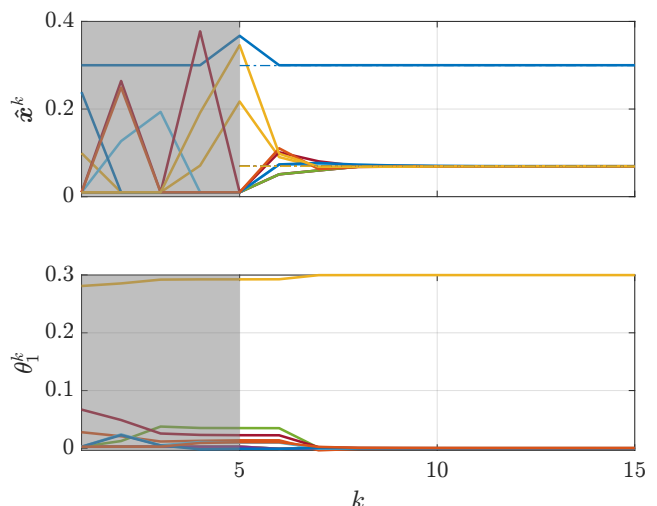

Fig. 7: GNEP formalized in [48, Ex. 1]. The shaded regions correspond to the random data collection phase. Top: sequence $\{\hat{\boldsymbol{x}}^k\}_{k \in \mathbb{N}}$ (solid lines) converging to an equilibrium (dashed lines). Bottom: convergent behaviour of the parameter vector reconstructing the BR mapping of agent 1.

Similar conclusions can be also drawn from Fig. 7 that illustrates the case in which Algorithm 1 is applied to the example described in [48, Ex. 1] (in this case, $K = 50$). Modelling the internet switching behaviour induced by selfish users, in this example each cost function reads as:

$$J_i(x_i, \boldsymbol{x}_{-i}) = -\frac{x_i}{\mathbf{1}_N^\top \boldsymbol{x}} \left(1 - \mathbf{1}_N^\top \boldsymbol{x}\right).$$

Then, while the first agent's strategy is constrained so that $x_1 \in [0.3, 0.5]$, for all $j \neq 1$ we have $x_j \in [0.01, 100]$. In addition, a standard linear constraint $\mathbf{1}_N^\top \boldsymbol{x} \leq 1$ couples the strategies of the whole population of decision-makers, thus actually resulting
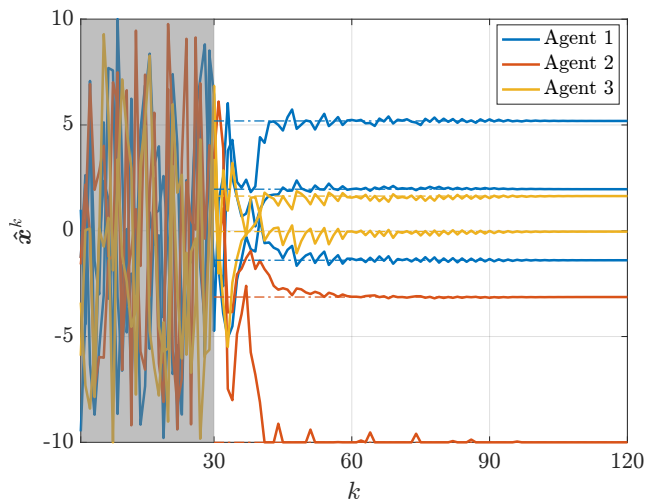
Fig. 8: GNEP described in [48, Ex. 3]. The shaded region corresponds to the data collection phase ($K_{\text{in}} = 30$). Sequence $\{\hat{\boldsymbol{x}}^k\}_{k \in \mathbb{N}}$ (solid lines) converging to a GNE (dashed lines).

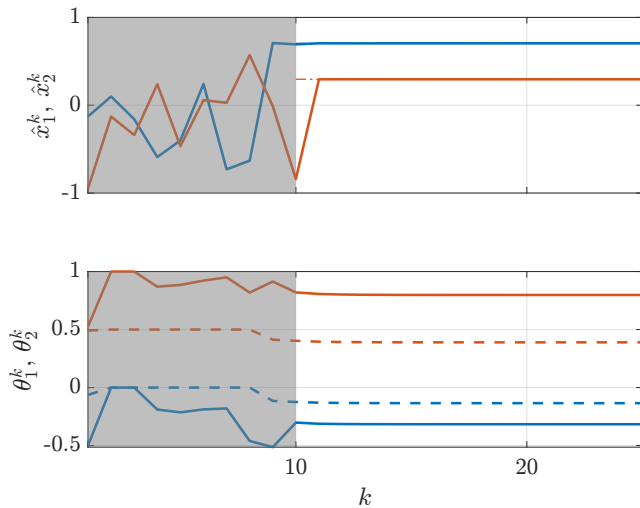
Fig. 9: GNEP formalized in [48, Ex. 11]. The shaded region corresponds to the random data collection phase ($K = 100$). Top: Sequence $\{\hat{\boldsymbol{x}}^k\}_{k \in \mathbb{N}}$ (solid lines) converging to the normalized equilibrium (dashed lines). Bottom: convergent behaviour of the parameter vectors reconstructing the BR mapping for both agents (same colour, solid lines correspond to the first element of each θ_i, while dashed lines to the second).

into a GNEP. An example with multidimensional decision vectors is instead the one illustrated in Fig. 8, which amounts to the GNEP described in [48, Ex. 3] and involving $N = 3$ agents with three, two and two variables, respectively. Even though the point $\{\hat{\boldsymbol{x}}^k\}_{k \in \mathbb{N}}$ converges to is different from the GNE reported in [48, Ex. 3], it still coincides with an equilibrium of the underlying GNEP, being a fixed point of the BR mappings.

We conclude this part by discussing two simple, yet significant, examples involving two agents only. The first one consists in the GNEP in [48, Ex. 11] featuring an infinite number of equilibria $\{\text{col}(\alpha, 1 - \alpha) \mid \alpha \in [1/2, 1]\}$, for which Fig. 9 shows that Algorithm 1 is actually able to return one of those solutions – specifically that coinciding
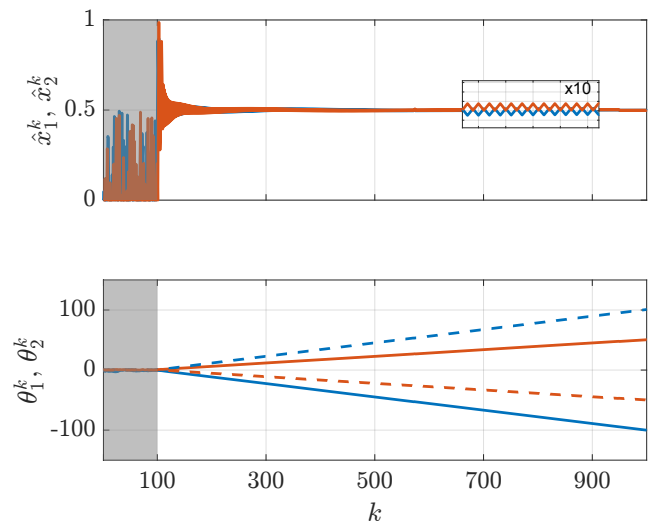

Fig. 10: Nash equilibrium problem with no equilibria. The shaded region corresponds to the random data collection phase ($K_{\text{in}} = 100$, $K = 1000$). Top: Sequence $\{\hat{\boldsymbol{x}}^k\}_{k \in \mathbb{N}}$ obtained by the external entity. Bottom: diverging behaviour of the parameter vectors reconstructing the BR mapping for both agents (same colour, solid lines correspond to the first element of each θ_i, while dashed lines to the second one).

TABLE II: Minimum eigenvalue of the Hessian matrix of (7)

	Fig. 6	Fig. 7	Fig. 8	Fig. 9	Fig. 10
$\lambda_{\min}(H)$	0.11	0.16	0.001	0.65	*

with $\alpha = 0.7$. After executing such an example 1000 times, we have noticed that Algorithm 1 always returns the same equilibrium corresponding to $\alpha = 0.7$ in all those instances in which it has converged. On the contrary, the second example considered does not have any Nash equilibrium due to the lack of quasiconvexity in the agents' cost functions, which are taken as $J_1(x_1, x_2) = -x_1^2 + 2x_1x_2$ and $J_2 = x_2 - 2x_1x_2$, $x_i \in [0, 1]$ for $i \in \{1, 2\}$. Figure 10 reports the numerical results for this example, where it is evident the non-convergent behaviour of the main quantities involved, particularly the parameter vectors θ_i. According to the discussion in §V-A, in this case the sufficient conditions established in Theorem 4.5 do not hold, and therefore no certificate for the existence of Nash equilibria is provided, which is indeed the case. Finally, Table II shows that, in all those numerical instances considered for which convergence of the parametric estimates happens, also the sufficient condition for the uniqueness of the solution in $\mathcal{M}(\tilde{\theta})$ is met (namely, positive definiteness of the Hessian matrix H of the cost function, along with convexity of Ω), thus verifying the requirements in Theorem 4.5 numerically.

C. Multi-agent feedback controller synthesis

The control application described in §II-B is slightly more challenging, mostly because the conditions for the stack of the (highly nonlinear) action-reaction mappings (4) to admit a stationary action profile are not clear. For this reason, we

selected 10 random numerical instances on which Algorithm 1 returned a fixed point of each $f_i(\cdot)$ in (4) after just one execution, thus ensuring the existence of a stationary action profile according to Corollary 4.6 (Fig. 12 reports just one converging instance of the first example for agent 1). In particular, we generated random, discrete-time LTI test models with unstable eigenvalues, satisfying the observability and reachability conditions discussed in §II-B2. Given $N = 3$ agents with $n_{u_i} = 1$ for simplicity, we have $n_z \sim \mathcal{U}(N, 3N)$, $n_y \sim \mathcal{U}(N, 2N)$ and $n_{y_i} \sim \mathcal{U}(2, n_y)$, while weights $r_i \sim \mathcal{U}(1, 10)$, $Q_i = WW^\top$ and $W \sim \mathcal{U}(0, 1)$, $i \in \mathcal{N}$.

We started by analyzing the impact the hyperparameters have on the convergence of Algorithm 1 when applied to the first randomnly generated instance. We replicate precisely the same approach as described in the previous subsection, i.e., $K = 100$, average over 20 executions of Algorithm 1 for coefficients combination $\beta \in \{0, 0.5, 1, 2, 5\}$ and $K_{\text{in}} \in \{0.01, 0.1, 0.2, 0.4\} \cdot K$. The results in Fig. 11 show that, unlike the GNEP setting, for this application the convergence rate is more sensible to the learning rate β rather than the length of the random initialization phase (the blue lines do not even converge indeed). We thus set $\beta = 2$ and $K_{\text{in}} = 0.1K$ to conduct the numerical simulations of this subsection.

The random initialization phase is here performed by presenting to the agents centralized LQR gains obtained with output and input weights $Q = I_{n_y}$ and $R = I_N$, and after perturbing the original dynamical matrix. Moreover, to avoid the emptiness of (4) due to the unboundedness of κ_i, for some $\boldsymbol{\kappa}_{-i}$ the external entity computes points $\hat{\boldsymbol{\kappa}} \in \hat{\Omega} = \Omega = [-20, 20]^{Nn_z}$.

As already spotted out in the comments following Fig. 5, from our numerical experience we have found that, given a numerical instance enjoying the existence of a stationary action profile, Algorithm 1 may not converge at each execution. With the application considered in this subsection, however, this behaviour is even more pronounced, possibly also in view of the highly nonlinear structure of the underlying mappings. These considerations hence motivate us in performing a statistical analysis to shed further light on this fact, and the last column of Table III reports the percentage of convergent runs over 1000 executions of Algorithm 1. In all the numerical simulations in which convergence of the parametric estimates happens – possibly to different points – we observe that: i) Algorithm 1 returns a fixed point of the stack of the action-reaction mappings, thus verifying numerically the theory in §IV; and, quite interestingly, ii) the point it returns is always the same. Finally, to populate the last column of Table III, for each numerical example we have taken the smallest eigenvalue of the Hessian matrix for (8) among the minimum ones $\lambda^i_{\min}(H)$ in all those cases in which the convergence of the parametric estimates happened. Such a column indeed shows that the condition on $\mathcal{M}(\tilde{\theta})$ of being a singleton is always met.

D. Comparison results

We now compare the performance of Algorithm 1 against those offered by `TensCalc` [45], a centralized state-of-the-art method for GNE computation. We remark that, unlike our approach, `TensCalc` requires a fully centralized scenario
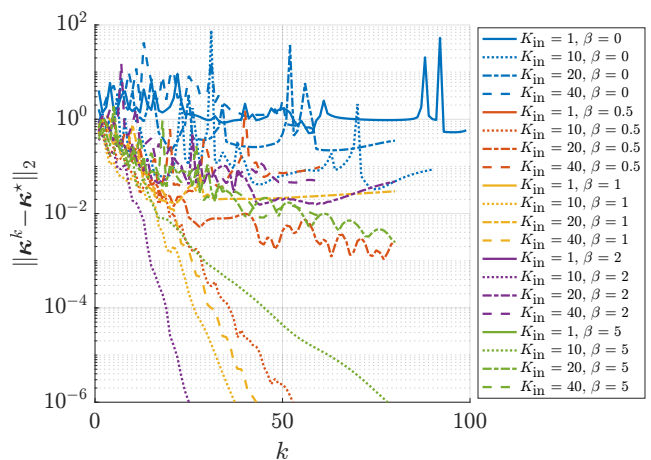

Fig. 11: Hyperparameters analysis on the first randomly drawn multi-agent feedback controller synthesis problem.
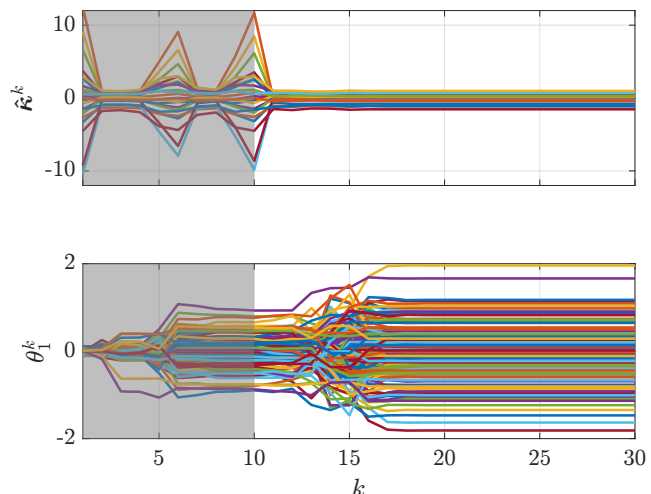

Fig. 12: First randomly drawn multi-agent feedback controller synthesis problem. The shaded region corresponds to the random data collection phase ($K = 100$). Top: Sequence $\{\hat{\boldsymbol{x}}^k\}_{k \in \mathbb{N}}$ converging to a stationary action profile. Bottom: convergent behaviour of the parameter vector reconstructing the action-reaction mapping (4) of agent 1.

where the agents' cost functions J_i and local constraint sets $\mathcal{X}_i$ have to be known by the central entity. We thus consider four examples provided with the toolbox associated to `TensCalc`[2], namely a zero-sum game and three different min-max problems, illustrated and described in Fig. 13.

Specifically, Fig. 13 reports just one numerical instance on top the 50 generated to compare the average computational time of the two approaches. There, one can appreciate how our active learning scheme produces a convergent sequence to the Nash equilibrium (NE) (computed with `TensCalc`). In the first example in Fig. 13.(a), the random initialization phase and the four iterations required to converge take 2.2 [s], as opposed to `TensCalc` that provides a solution in 1.3 [ms]. For the examples in Fig. 13.(b)–(d), instead, Algorithm 1 shows a fast convergence, ranging from 34.2 [ms] to 42.1 [ms], while

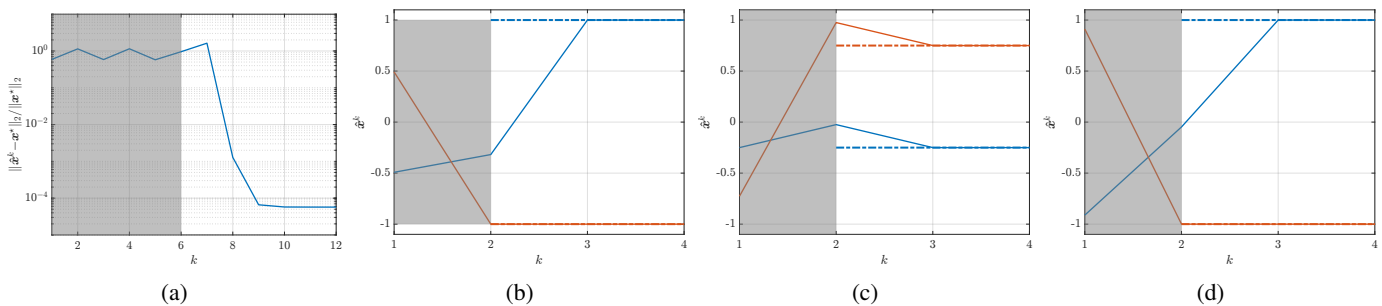[2]Available at `https://github.com/hespanha/tenscalc`.

Fig. 13: Convergence behaviour exhibited by Algorithm 1 in four of the examples provided with `TensCalc` toolbox: (a) Two-player zero-sum game with $n_1 = n_2 = 200$ and quadratic cost functions $J_1(x_1, x_2) = \|x_1\|^2 + 2(x_2 - 2 \cdot \mathbf{1}_{200})^\top x_1$, $J_2(x_2, x_1) = 9\|x_2\|^2 - 2(x_1 - 2 \cdot \mathbf{1}_{200})^\top x_2$; (b) Min-max problem with scalar decision variables, $J_1(x_1, x_2) = (x_1 + x_2)^2$, $J_2(x_2, x_1) = -(x_2 + x_1)^2 + 2(x_2 + 2)^2$, and $|x_2| \leq 1$; (c) Min-max problem with scalar decision variables, $J_1(x_1, x_2) = (x_1 + x_2 + 1)^2$, $J_2(x_2, x_1) = -(x_2 + x_1 + 1)^2 + 2x_2^2$, and $|x_1| \leq 0.25$; (d) Min-max problem with scalar decision variables, $J_1(x_1, x_2) = (x_1 + x_2)^2$, $J_2(x_2, x_1) = -(x_2 + x_1)^2 + 2(x_2 + 2)^2$, $|x_1| \leq 2$, and $|x_2| \leq 1$.

TABLE III: Multi-agent feedback controller synthesis – Numerical results

Example	n_z	n_y	n_{y_i}	$\%_{\text{conv}}$	$\min_i \{\lambda_{\min}^i(H)\}$
1	8	6	$\{5, 2, 5\}$	60	5.39×10^{-4}
2	7	4	$\{3, 2, 4\}$	49.3	1.23×10^{-3}
3	4	6	$\{2, 4, 2\}$	87.6	4.29×10^{-4}
4	5	6	$\{4, 5, 5\}$	82.5	1.76×10^{-4}
5	6	5	$\{2, 5, 3\}$	98.1	1.05×10^{-4}
6	4	6	$\{2, 3, 3\}$	89.5	2.59×10^{-4}
7	8	6	$\{2, 5, 4\}$	94.9	6.80×10^{-3}
8	4	3	$\{3, 3, 2\}$	98.6	1.85×10^{-4}
9	5	6	$\{2, 5, 2\}$	62.9	1.79×10^{-4}
10	8	5	$\{4, 5, 2\}$	94.7	3.54×10^{-3}

`TensCalc` is generally slower (0.3 to 0.34 [s]). In all cases, note that `TensCalc` needs to "generate" a solver, a procedure lasting for around 4 seconds on the laptop employed.

VII. CONCLUSION AND OUTLOOK

We have proposed an active learning scheme allowing an external entity to actively learn faithful local surrogates of action-reaction mappings, privately held by a population of interacting agents, in order to find a stationary action profile of the underlying multi-agent interaction process. We showed in numerical experiments the practical effectiveness and versatility of the procedure on a number of typical competitive multi-agent decision-making and control problems.

The proposed approach paves the way to numerous extensions. Looking at the specific problem of learning a GNE in GNEPs, for instance, given the tight connection between an equilibrium solution and a minimizer of the Nikaido-Isoda function [49], one may design a tailored procedure along the line of Algorithm 1 to locally approximate such a function. Finally, we remark that problem (7) is based completely on *exploiting* the current predictors $\hat{f}_i$. In alternative to the initial random exploration phase, one could also investigate possible solutions for better *exploring* the decision set Ω, as done for example in reinforcement learning [50] and surrogate-based global optimization [51], and analyze its potential benefits.

APPENDIX

A. Technical proofs

Proof of Lemma 3.3: Let $\bar{r} : \mathbb{R}^n \times \mathbb{R}^p \to \bar{\mathbb{R}}$ be defined by $\bar{r}(x, \theta) := r(x, \theta) + \iota_{\mathcal{X}}(x)$. Since $\mathcal{X}$ is a convex set and r is a convex function, then $\bar{r}$ is also a convex function for each given $\theta \in \mathbb{R}^p$. By applying [36, Th. 2.6], it follows that $\mathcal{M}(\theta) = \text{argmin}_{y \in \mathbb{R}^n} \bar{r}(y, \theta)$ is a convex set for all $\theta \in \mathbb{R}^p$.

To show continuity of $r^\star$ at a generic $\tilde{\theta} \in \mathbb{R}^p$, we need to prove that $\lim_{k \to \infty} r^\star(\theta^k) = r^\star(\tilde{\theta})$, for any sequence $\{\theta^k\}_{k \in \mathbb{N}}$ such that $\lim_{k \to \infty} \theta^k = \tilde{\theta}$. Consider such a generic sequence and let $x^k \in \mathcal{M}(\theta^k)$ and $\tilde{x} \in \mathcal{M}(\tilde{\theta})$. In view of Assumption 3.2, $\theta \mapsto r(x^k, \theta)$ is also convex and differentiable w.r.t. θ and $\tilde{x} \in \mathcal{M}(\tilde{\theta})$, i.e., $\tilde{x}$ is a global minimizer of $r(\cdot, \tilde{\theta})$, and therefore we have that

$$
\begin{aligned}
r^\star(\theta^k) = r(x^k, \theta^k) &\geq r(x^k, \tilde{\theta}) + \left.\frac{\partial r(x^k, \theta)}{\partial \theta}\right|_{\theta=\tilde{\theta}}^\top (\theta^k - \tilde{\theta}) \\
&\geq r(\tilde{x}, \tilde{\theta}) + \left.\frac{\partial r(x^k, \tilde{\theta})}{\partial \theta}\right|_{\theta=\tilde{\theta}}^\top (\theta^k - \tilde{\theta}) \\
&\geq r(\tilde{x}, \tilde{\theta}) - D(\tilde{\theta})\|\theta^k - \tilde{\theta}\|,
\end{aligned}
$$

where the last inequality holds since the partial derivative $\partial r(x, \theta)/\partial \theta$ evaluated in $\tilde{\theta}$ is bounded w.r.t. x, with bound $D(\tilde{\theta}) := \sup_{x \in \mathcal{X}} \left| \left.\frac{\partial r(x,\theta)}{\partial \theta}\right|_{\theta=\tilde{\theta}} \right| \in \mathbb{R}$. Therefore, we obtain:

$$
r(\tilde{x}, \tilde{\theta}) - D(\tilde{\theta})\|\theta^k - \tilde{\theta}\| \leq r^\star(\theta^k) = r(x^k, \theta^k) \leq r(\tilde{x}, \theta^k) \quad (17)
$$

where the last inequality follows by the optimality of x^k. Since $\theta \mapsto r(x, \theta)$ is continuous and $\lim_{k \to \infty} \theta^k = \tilde{\theta}$, we have that $\lim_{k \to \infty} r(\tilde{x}, \theta^k) = r(\tilde{x}, \tilde{\theta})$, and the proof concludes by applying the squeeze theorem to (17). $\blacksquare$

Proof of Lemma 3.6: Lower semicontinuity of $\bar{r}(x,\theta)$ readily follows by the continuity of $r(x,\theta)$ at any given point $(\tilde{x},\tilde{\theta}) \in \mathbb{R}^n \times \mathbb{R}^p$, as

$$\lim_{(x,\theta)\to(\tilde{x},\tilde{\theta})} \inf \ \bar{r}(x,\theta) = \lim_{\varepsilon\to 0} \left[\inf_{(y,\xi)\in\mathcal{B}_\varepsilon(\tilde{x},\tilde{\theta})} r(y,\xi) \right]$$

$$= \begin{cases} r(\tilde{x},\tilde{\theta}) & \text{if} \quad \tilde{x} \in \mathcal{X} \\ +\infty & \text{if} \quad \tilde{x} \notin \mathcal{X} \end{cases}$$

which in turn implies that $\lim_{(x,\theta)\to(\tilde{x},\tilde{\theta})} \inf \ \bar{r}(x,\theta) \geq \bar{r}(\tilde{x},\tilde{\theta})$ for all $(\tilde{x},\tilde{\theta}) \in \mathbb{R}^n \times \mathbb{R}^p$. To show level-boundedness, note that $\mathcal{X}$ is bounded and nonempty, and therefore, for all $\alpha \in \mathbb{R}$, $\{x \in \mathbb{R}^n \mid \bar{r}(x,\theta) \leq \alpha\} \subseteq \mathcal{X}$ for all $\theta \in \mathbb{R}^p$. Thus, by setting $\mathcal{S} = \mathcal{X}$ and $g = \bar{r}$ in Definition 3.5, the claim follows. ∎

Proof of Lemma 3.7: By [36, Th. 2.6], each vector x^k in (5) is uniquely defined since $\frac{1}{2}\|\cdot\|_2^2$ is strictly convex, proper because $\mathcal{X}$ is nonempty and therefore, by Lemma 3.3, $\mathcal{M}(\theta^k)$ is a convex set. Moreover, since $x \mapsto r(x,\theta)$ is continuous for all $\theta \in \mathbb{R}^p$ and $\mathcal{X}$ is compact, by Lemma 3.6 the function $\bar{r}(x,\theta)$, defined by $\bar{r}(x,\theta) = r(x,\theta) + \iota_{\mathcal{X}}(x)$, is lower semicontinuous w.r.t. x and level-bounded in x locally uniformly w.r.t. θ. Still in view of Lemma 3.3, we also have that $\lim_{k\to\infty} r^\star(\theta^k) = r^\star(\tilde{\theta})$. Thus, from [36, Th. 1.17(b)] it follows that the sequence $\{x^k\}_{k\in\mathbb{N}}$ generated by (5) is bounded and all its cluster points lie in $\mathcal{M}(\tilde{\theta})$, which is however a singleton and contains $\tilde{x}$ only.

For the sake of contradiction, assume that $x^k \nrightarrow \tilde{x}$ as $k \to \infty$. Then, there exists some $\varepsilon > 0$ and an infinite subsequence $\{x^k\}_{k\in\mathcal{I}}$ so that $\|x^k - \tilde{x}\|_2 > \varepsilon$ for all $k \in \mathcal{I}$. Since $\{x^k\}_{k\in\mathcal{I}}$ is also bounded, by the Bolzano-Weierstrass theorem there exists an infinite subsequence $\{x^k\}_{k\in\mathcal{I}'}$, with $\mathcal{I}' \subset \mathcal{I}$, such that $\lim_{k\to\infty,k\in\mathcal{I}'} x^k = \bar{x}$, with $\bar{x} \neq \tilde{x}$. Since $\bar{x}$ is a cluster point of $\{x^k\}_{k\in\mathbb{N}}$, by relying on [36, Th. 1.17(b)] we have that $\bar{x} \in \mathcal{M}(\tilde{\theta})$. This however is in conflict with the fact that $\mathcal{M}(\tilde{\theta})$ is a singleton, i.e., $\mathcal{M}(\tilde{\theta}) = \{\tilde{x}\}$, and hence the assumption $x^k \nrightarrow \tilde{x}$ for $k \to \infty$ is contradicted, proving the statement. ∎

Proof of Lemma 4.2: Let us consider a single agent $i \in \mathcal{N}$, as the proof for the remaining ones is identical *mutatis mutandis*. By making use of a contradiction argument, we will show that any of the parameters $\bar{\theta}_i \in \mathcal{A}_i := \mathcal{A}_i(\bar{x}_i, \bar{\boldsymbol{x}}_{-i})$ asymptotically belongs to the set of minimizers of $\xi_i \mapsto \frac{1}{k}\sum_{t=1}^k \ell_i(x_i^t, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^t, \xi_i))$, and therefore it shall happen that $\bar{\theta}_i \in \mathcal{A}_i$. Recall, indeed, that the parameter update reads as $\theta_i^{k+1} = \mathrm{argmin}_{\xi_i\in\mathbb{R}^{p_i}} \frac{1}{k}\sum_{t=1}^k \ell_i(x_i^t, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^t, \xi_i))$, where equality follows by virtue of the global optimality assumed and the tie-break rule $\mathcal{T}(\cdot)$ in place. Thus, for the sake of contradiction, let us assume that $\lim_{k\to\infty} \theta_i^k = \tilde{\theta}_i$ with $\tilde{\theta}_i \notin \mathcal{A}_i$.

To start we note that, by combining the definition of limit [52, Def. 5.4] and the properties of the loss function stated in Standing Assumption 4.1, along with those of the affine surrogates $\hat{f}_i$ described in §IV-D, $\lim_{k\to\infty} x_i^k = \bar{x}_i$ and $\lim_{k\to\infty} \hat{\boldsymbol{x}}_{-i}^k = \bar{\boldsymbol{x}}_{-i}$ so that $(\bar{x}_i, \bar{\boldsymbol{x}}_{-i}) \in \Omega$ directly imply that $\lim_{k\to\infty} \ell_i(x_i^k, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^k, \tilde{\theta}_i)) = v > 0$, for any $\tilde{\theta}_i \notin \mathcal{A}_i$, while for all $\tilde{\theta}_i \in \mathcal{A}_i$ instead, $\lim_{k\to\infty} \ell_i(x_i^k, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^k, \tilde{\theta}_i)) = 0$. Since we are assuming that $\lim_{k\to\infty} \theta_i^k = \bar{\theta}_i$, by considering a generic θ_i^k obtained from (6) we know that for any $\tilde{\varepsilon} > 0$ there exists some $\tilde{M} := \tilde{M}(\tilde{\varepsilon}) \geq 0$ so that $|\ell_i(x_i^k, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^k, \theta_i^k)) - v| \leq \tilde{\varepsilon}$, i.e., $-\tilde{\varepsilon} + v \leq \ell_i(x_i^k, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^k, \theta_i^k)) \leq \tilde{\varepsilon} + v$, for all $k \geq \tilde{M}$. On the other hand, we also know from $\lim_{k\to\infty} \ell_i(x_i^k, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^k, \tilde{\theta}_i)) = 0$

that for any $\bar{\varepsilon} > 0$ there exists some $\bar{M} := \bar{M}(\bar{\varepsilon}) \geq 0$ so that $\ell_i(x_i^k, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^k, \tilde{\theta}_i)) \leq \bar{\varepsilon}$ for all $k \geq \bar{M}$.

Without loss of generality, let us then pick $\tilde{\varepsilon} = v/2$ so that $-\ell_i(x_i^k, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^k, \theta_i^k)) \leq -v/2$, and fix $\bar{\varepsilon} < \tilde{\varepsilon} = v/2$, which is always possible by continuity and k sufficiently large. Under this latter condition, precisely $k > \max\{\tilde{M}, \bar{M}\}$, evaluating the summation of the difference of the loss functions in $\bar{\theta}_i$ and θ_i^k at iteration k, with each term $\Delta\ell_{i,k}^t := \ell_i(x_i^t, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^t, \tilde{\theta}_i)) - \ell_i(x_i^t, \hat{f}_i(\hat{\boldsymbol{x}}_{-i}^t, \theta_i^k))$, leads to:

$$\frac{1}{k}\sum_{t=1}^k \Delta\ell_{i,k}^t = \frac{1}{k}\left(\sum_{t=1}^{\min\{\tilde{M},\bar{M}\}-1} \Delta\ell_{i,k}^t + \sum_{t=\min\{\tilde{M},\bar{M}\}}^{\max\{\tilde{M},\bar{M}\}-1} \Delta\ell_{i,k}^t \right.$$
$$\left. + \sum_{t=\max\{\tilde{M},\bar{M}\}}^{k} \Delta\ell_{i,k}^t \right).$$

While the first term in the summation is constant for fixed $\bar{\varepsilon}$ and $\tilde{\varepsilon}$ (and therefore for fixed $\bar{M}$ and $\tilde{M}$), namely $\sum_{t=1}^{\min\{\tilde{M},\bar{M}\}-1} \Delta\ell_{i,k}^t =: \alpha$, the second and third elements can always be upper bounded by $(k-\bar{\rho})\bar{\varepsilon} - (k-\tilde{\rho})v/2 + \beta$, where $\bar{\rho}, \tilde{\rho} \geq 0$ and $\beta \in \mathbb{R}$ are constants depending on the values $\tilde{M}$ and $\bar{M}$ take. Specifically, if $\bar{M} < \tilde{M}$, $\bar{\rho} = \bar{M}$, $\tilde{\rho} = \tilde{M}$ and $\beta < 0$, whereas if $\bar{M} \geq \tilde{M}$, $\bar{\rho} = \tilde{M}$, $\tilde{\rho} = \bar{M}$ and $\beta > 0$. Thus, taking the limit for $k \to \infty$ yields inequality:

$$\lim_{k\to\infty} \frac{1}{k}\sum_{t=1}^k \Delta\ell_{i,k}^t \leq \lim_{k\to\infty} \frac{1}{k}[\alpha + \beta + (k-\bar{\rho})\bar{\varepsilon} - (k-\tilde{\rho})v/2] < 0$$

in view of $\bar{\varepsilon} < \tilde{\varepsilon} = v/2$, meaning that we can always find some iteration index such that the cost obtained with some $\tilde{\theta}_i \in \mathcal{A}_i$ is strictly smaller than the one obtained with θ_i^k updated through the rule in (6), and assumed to be convergent to some $\tilde{\theta}_i \notin \mathcal{A}_i$. This clearly represents a contradiction and hence concludes the proof. ∎

Proof of Proposition 4.4: Let us prove that $r(\boldsymbol{x},\theta) = \sum_{i\in\mathcal{N}} \|x_i - (\nu_i^{k+1})^\top \boldsymbol{x}_{-i} - c_i^{k+1}\|_2^2$ satisfies Assumption 3.1 and 3.2. Since Ω is a nonempty polytope, it is convex and bounded. Moreover, function r is quadratic and positive semidefinite, and therefore continuous and convex w.r.t. $\boldsymbol{x}$, for all $\theta \in \mathbb{R}^p$. In addition, $r(\boldsymbol{x},\theta)$ can be expressed also as:

$$r(\boldsymbol{x},\theta) = \|A(\boldsymbol{x})\theta - b(\boldsymbol{x})\|_2^2$$
$$= \theta^\top A(\boldsymbol{x})^\top A(\boldsymbol{x})\theta + 2b(\boldsymbol{x})^\top A(\boldsymbol{x})\theta + b(\boldsymbol{x})^\top b(\boldsymbol{x})$$

where $A(\boldsymbol{x})$, $b(\boldsymbol{x})$ are suitably defined affine functions of $\boldsymbol{x}$, which is a quadratic positive semidefinite function w.r.t. to θ, and therefore convex and differentiable for all $\boldsymbol{x} \in \Omega$. Thus,

$$\sup_{\boldsymbol{x}\in\Omega} \left| \frac{\partial r(\boldsymbol{x},\theta)}{\partial\theta} \right| = \max_{\boldsymbol{x}\in\Omega} \left| 2A(\boldsymbol{x})^\top A(\boldsymbol{x})\theta + A(\boldsymbol{x})^\top b(\boldsymbol{x}) \right|$$
$$= \max_{i=1,\dots,q} \pm \{2A(v_i)^\top A(v_i)\theta + A(v_i)^\top b(v_i)\}$$

where $v_1,\dots,v_q$ are the vertices of Ω and the last inequality follows since $2A(\boldsymbol{x})^\top A(\boldsymbol{x})\theta + A(\boldsymbol{x})^\top b(\boldsymbol{x})$ is convex and quadratic w.r.t. $\boldsymbol{x}$ and the maximum of a convex quadratic

function over a polytope is attained at one of its vertices. For all $\theta \in \mathbb{R}^p$, we thus obtain that:

$$\max_{\boldsymbol{x} \in \Omega} \left| 2A(\boldsymbol{x})^\top A(\boldsymbol{x})\theta + A(\boldsymbol{x})^\top b(\boldsymbol{x}) \right|$$
$$= \max\{ 2A(v_{i(\theta)})^\top A(v_{i(\theta)})\theta + A(v_{i(\theta)})^\top b(v_{i(\theta)}),$$
$$- 2A(v_{j(\theta)})^\top A(v_{j(\theta)})\theta + A(v_{j(\theta)})^\top b(v_{j(\theta)})\} =: D(\theta)$$

for some indices $i(\theta)$, $j(\theta) \in \{1, \ldots, q\}$. By relying on Lemma 3.7 we hence obtain the desired result, since all its assumptions are satisfied. ∎

Proof of Theorem 4.5: The convergence of each θ_i^k to some $\tilde{\theta}_i$ clearly implies that, in view of Standing Assumption 4.1, also the associated composition mapping estimate converges, i.e., for all $i \in \mathcal{N}$, $\lim_{k \to \infty} \hat{f}_i(\boldsymbol{x}_{-i}, \theta_i^k) = \hat{f}_i(\boldsymbol{x}_{-i}, \tilde{\theta}_i)$ with $\hat{f}_i(\boldsymbol{x}_{-i}, \tilde{\theta}_i)$ possibly different from the true action-reaction mapping $f_i(\boldsymbol{x}_{-i})$. Then, in view of Proposition 4.4, we also have that $\lim_{k \to \infty} \hat{\boldsymbol{x}}^k = \tilde{\boldsymbol{x}}$. This latter relation in turn yields $\lim_{k \to \infty} \boldsymbol{x}^k = \bar{\boldsymbol{x}}$, as the action-reaction mappings $f_i(\cdot)$ are single-valued and continuous in view of Standing Assumption 2.2. Note that both sequences $\{\hat{\boldsymbol{x}}^k\}_{k \in \mathbb{N}}$ and $\{\boldsymbol{x}^k\}_{k \in \mathbb{N}}$ are feasible in view of (7) and the fact that each x_i^k is determined through $f_i(\cdot)$ that implicitly accounts for the common constraints Ω. Since $\hat{\boldsymbol{x}}^k$ is iteratively chosen following the tie-break rule induced by $\frac{1}{2}\|\cdot\|_2^2$, as the set of global minimizers in (7) may not be unique, it may happen that $\tilde{\boldsymbol{x}} \neq \bar{\boldsymbol{x}}$. We now make use of a local exact approximation argument to show that $\lim_{k \to \infty} \|\hat{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2 = 0$, and hence that the two limit points above coincide, and actually yield, by virtue of Lemma 4.2, a stationary action profile $\boldsymbol{x}^\star$ in the sense of Definition 2.1.

In view of the definition of limit we can then always find some $\bar{k} \in \mathbb{N}$ and $\mu_i > 0$ such that a neighbourhood of $\bar{x}_i$, say $\mathcal{B}_{\mu_i}(\bar{x}_i)$, contains infinitely many points [53], meaning that $x_i^k \in \mathcal{B}_{\mu_i}(\bar{x}_i)$ for all $k \geq \bar{k}$, $i \in \mathcal{N}$. By virtue of the consistency property proved in Lemma 4.2 it thus follows that the pointwise approximation shall be exact, namely each $\tilde{\theta}_i$ is so that, for all $i \in \mathcal{N}$, $\|\hat{f}_i(\tilde{\boldsymbol{x}}_{-i}, \tilde{\theta}_i) - f_i(\tilde{\boldsymbol{x}}_{-i})\|_2 = 0 = \|\hat{f}_i(\tilde{\boldsymbol{x}}_{-i}, \tilde{\theta}_i) - \tilde{x}_i\|_2$.

Since (7) is solved at the global optimum at every $k \in \mathbb{N}$, and the single-valuedness of each $f_i(\cdot)$, this latter relation readily yields $\sum_{i \in \mathcal{N}} \|\tilde{x}_i - \hat{f}_i(\tilde{\boldsymbol{x}}_{-i}, \tilde{\theta}_i)\|_2^2 = \sum_{i \in \mathcal{N}} \|\tilde{x}_i - f_i(\tilde{\boldsymbol{x}}_{-i})\|_2^2 = 0 = \sum_{i \in \mathcal{N}} \|\tilde{x}_i - \bar{x}_i\|_2^2$, and hence $\|\tilde{x}_i - \bar{x}_i\|_2 = 0$ for all $i \in \mathcal{N}$. This means that $\bar{\boldsymbol{x}} = \tilde{\boldsymbol{x}}$, and in view of Definition 2.1, they coincide with a fixed point of the action-reaction mappings $f_i(\cdot)$, as $\sum_{i \in \mathcal{N}} \|f_i(\bar{\boldsymbol{x}}_{-i}) - \bar{x}_i\|_2^2 = 0$, i.e., $\|f_i(\bar{\boldsymbol{x}}_{-i}) - \bar{x}_i\|_2 = 0$ for all $i \in \mathcal{N}$, concluding the proof. ∎

B. Derivation of Kalman filter equations in (12)

The updates (12) are obtained by applying linear Kalman filtering to the linear time-varying system (we omit the index i for simplicity of notation): $\theta^{k+1} = \theta^k + \xi^k$, $x^k = (\phi^k)^\top \theta^k + \zeta^k$, where ξ^k, ζ^k are zero-mean white noise terms with covariance βI and 1, respectively. The Kalman filter equations are:

$$\begin{aligned} M^k &= P^k \phi^k / (1 + (\phi^k)^\top P^k \phi^k) \\ \theta^{k+1/2} &= \theta^k + M^k(x^k - (\phi^k)^\top \theta^k) \\ P^{k+1/2} &= (I - M^k(\phi^k)^\top)P^k \\ &= P^k - P^k \phi^k (\phi^k)^\top P^k / (1 + (\phi^k)^\top P^k \phi^k) \\ \theta^{k+1} &= \theta^{k+1/2} \\ P^{k+1} &= P^{k+1/2} + \beta I. \end{aligned}$$

Since $P^{k+1/2}\phi^k = P^k\phi^k - P^k\phi^k(\phi^k)^\top P^k\phi^k/(1 + (\phi^k)^\top P^k\phi^k) = P^k\phi^k/(1 + (\phi^k)^\top P^k\phi^k) = M^k$, we obtain $\theta^{k+1} = \theta^k + P^k\phi^k/(1 + (\phi^k)^\top P^k\phi^k)(x^k - (\phi^k)^\top \theta^k)$.

REFERENCES

[1] K. J. Aström, P. Albertos, M. Blanke, A. Isidori, W. Schaufelberger, and R. Sanz, *Control of complex systems*. Springer Science & Business Media, 2011.

[2] F. Facchinei and C. Kanzow, "Generalized Nash equilibrium problems," *Annals of Operations Research*, vol. 175, no. 1, pp. 177–211, 2010.

[3] B. Settles, "Active learning," in *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool Publishers, 2012.

[4] A. Bemporad, "Active learning for regression by inverse distance weighting," *Information Sciences*, vol. 626, pp. 275–292, 2023.

[5] L. Ljung, "Perspectives on system identification," *Annual Reviews in Control*, vol. 34, no. 1, pp. 1–12, 2010.

[6] F. D'Ettorre, M. Banaei, R. Ebrahimy, S. A. Pourmousavi, E. Blomgren, J. Kowalski, Z. Bohdanowicz, B. Łopaciuk-Gonczaryk, C. Biele, and H. Madsen, "Exploiting demand-side flexibility: State-of-the-art, open issues and social perspective," *Renewable and Sustainable Energy Reviews*, vol. 165, p. 112605, 2022.

[7] F. C. Schweppe, R. D. Tabors, J. L. Kirtley, H. R. Outhred, F. H. Pickel, and A. J. Cox, "Homeostatic utility control," *IEEE Transactions on Power Apparatus and Systems*, no. 3, pp. 1151–1163, 1980.

[8] O. Corradi, H. Ochsenfeld, H. Madsen, and P. Pinson, "Controlling electricity consumption by forecasting its response to varying prices," *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp. 421–429, 2012.

[9] N. Vespermann, S. Delikaraoglou, and P. Pinson, "Offering strategy of a price-maker energy storage system in day-ahead and balancing markets," in *2017 IEEE Manchester PowerTech*. IEEE, 2017, pp. 1–6.

[10] Z. Ma, D. S. Callaway, and I. A. Hiskens, "Decentralized charging control of large populations of plug-in electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 1, pp. 67–78, 2011.

[11] S. Grammatico, "Dynamic control of agents playing aggregative games with coupling constraints," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4537–4548, 2017.

[12] C. Cenedese, F. Fabiani, M. Cucuzzella, J. M. A. Scherpen, M. Cao, and S. Grammatico, "Charging plug-in electric vehicles as a mixed-integer aggregative game," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 4904–4909.

[13] J. Fearnley, M. Gairing, P. Goldberg, and R. Savani, "Learning equilibria of games via payoff queries," in *Proceedings of the fourteenth ACM conference on Electronic commerce*, 2013, pp. 397–414.

[14] Y. Babichenko, "Query complexity of approximate Nash equilibria," *Journal of the ACM (JACM)*, vol. 63, no. 4, pp. 1–24, 2016.

[15] P. W. Goldberg and A. Roth, "Bounds for the query complexity of approximate equilibria," *ACM Transactions on Economics and Computation (TEAC)*, vol. 4, no. 4, pp. 1–25, 2016.

[16] Y. Babichenko and A. Rubinstein, "Communication complexity of approximate Nash equilibria," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, 2017, pp. 878–889.

[17] S. Hart and N. Nisan, "The query complexity of correlated equilibria," *Games and Economic Behavior*, vol. 108, pp. 401–410, 2018.

[18] P. W. Goldberg and F. J. Marmolejo-Cossío, "Learning convex partitions and computing game-theoretic equilibria from best-response queries," *ACM Transactions on Economics and Computation (TEAC)*, vol. 9, no. 1, pp. 1–36, 2021.

[19] E. A. Viqueira and C. Cousins, "Learning simulation-based games from data," in *Proceedings of the 18th International Conference on Autonomous Agents and Multi-Agent Systems*, vol. 2019, 2019.

[20] E. A. Viqueira, C. Cousins, and A. Greenwald, "Improved algorithms for learning equilibria in simulation-based games," in *Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems*, 2020, pp. 79–87.

[21] A. Marchesi, F. Trovò, and N. Gatti, "Learning probably approximately correct maximin strategies in simulation-based games with infinite strategy spaces," in *Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems*, 2020, pp. 834–842.

[22] Y. Vorobeychik and M. P. Wellman, "Stochastic search methods for Nash equilibrium approximation in simulation-based games," in *Proceedings of the 7th International Conference on Autonomous Agents and Multi-Agent Systems*, 2008, pp. 1055–1062.

[23] Y. Vorobeychik, "Probabilistic analysis of simulation-based games," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 20, no. 3, pp. 1–25, 2010.

[24] A. Al-Dujaili, E. Hemberg, and U. M. O'Reilly, "Approximating Nash equilibria for black-box games: A bayesian optimization approach," *arXiv preprint arXiv:1804.10586*, 2018.

[25] V. Picheny, M. Binois, and A. Habbal, "A Bayesian optimization approach to find Nash equilibria," *Journal of Global Optimization*, vol. 73, no. 1, pp. 171–192, 2019.

[26] S. Clarke, G. Dragotto, J. F. Fisac, and B. Stellato, "Learning rationality in potential games," *arXiv preprint arXiv:2303.11188*, 2023.

[27] Y. Vorobeychik, M. P. Wellman, and S. Singh, "Learning payoff functions in infinite games," *Machine Learning*, vol. 67, no. 1, pp. 145–168, 2007.

[28] Y. Huang and J. Hu, "Distributed stochastic Nash equilibrium learning in locally coupled network games with unknown parameters," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 342–354.

[29] F. Fabiani, A. Simonetto, and P. J. Goulart, "Learning equilibria with personalized incentives in a class of nonmonotone games," in *2022 European Control Conference (ECC)*, 2022, pp. 2179–2184.

[30] ——, "Personalized incentives as feedback design in generalized Nash equilibrium problems," *IEEE Transactions on Automatic Control*, vol. 68, no. 12, pp. 7724–7739, 2023.

[31] H. Witsenhausen, "A minimax control problem for sampled linear systems," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 5–21, 1968.

[32] A. Bemporad, F. Borrelli, and M. Morari, "Min-max control of constrained uncertain discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 9, pp. 1600–1606, 2003.

[33] T. Başar and P. Bernhard, *H-infinity optimal control and related minimax design problems: A dynamic game approach*. Springer Science & Business Media, 2008.

[34] C. A. R. Crusius and A. Trofino, "Sufficient LMI conditions for output feedback control problems," *IEEE Transactions on Automatic Control*, vol. 44, no. 5, pp. 1053–1057, 1999.

[35] D. Barcelli, D. Bernardini, and A. Bemporad, "Synthesis of networked switching linear decentralized controllers," in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 2480–2485.

[36] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009, vol. 317.

[37] P. Kumar and A. Gupta, "Active learning query strategies for classification, regression, and clustering: A survey," *Journal of Computer Science and Technology*, vol. 35, no. 4, pp. 913–945, 2020.

[38] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

[39] F. A. C. Viana, "Things you wanted to know about the Latin hypercube design and were afraid to ask," in *10th World Congress on Structural and Multidisciplinary Optimization*, vol. 19, no. 24.05, 2013, pp. 1–9.

[40] L. Ljung and T. Söderström, *Theory and practice of recursive identification*. MIT press, 1983.

[41] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*. Prentice-Hall, 2000.

[42] S. T. Alexander and A. L. Ghirnikar, "A method for recursive least squares filtering based upon an inverse QR decomposition," *IEEE Transactions on Signal Processing*, vol. 41, no. 1, pp. 20–30, 1993.

[43] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[44] A. Bemporad, "Recurrent neural network training with convex loss and regularization functions by extended kalman filtering," *IEEE Transactions on Automatic Control*, vol. 68, no. 9, pp. 5661–5668, 2023.

[45] J. P. Hespanha, "TensCalc: A toolbox to generate fast code to solve nonlinear constrained minimizations and compute Nash equilibria," *Mathematical Programming Computation*, vol. 14, no. 3, pp. 451–496, 2022.

[46] M. V. Solodov and P. Tseng, "Modified projection-type methods for monotone variational inequalities," *SIAM Journal on Control and Optimization*, vol. 34, no. 5, pp. 1814–1830, 1996.

[47] F. Salehisadaghiani, W. Shi, and L. Pavel, "An ADMM approach to the problem of distributed Nash equilibrium seeking." CoRR, 2017.

[48] F. Facchinei and C. Kanzow, "Penalty methods for the solution of generalized Nash equilibrium problems (with complete test problems)," Institute of Mathematics, University of Würzburg, Tech. Rep., 2009.

[49] H. Nikaidô and K. Isoda, "Note on non-cooperative convex games," *Pacific Journal of Mathematics*, vol. 5, no. S1, pp. 807–815, 1955.

[50] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[51] A. Bemporad, "Global optimization via inverse distance weighting and radial basis functions," *Computational Optimization and Applications*, vol. 77, pp. 571–595, 2020, (Code available at http://cse.lab.imtlucca.it/ bemporad/glis).

[52] B. S. Thomson, J. B. Bruckner, and A. M. Bruckner, *Elementary real analysis*. Prentice-Hall, 2001.

[53] G. Bachman and L. Narici, *Functional analysis*. Courier Corporation, 2000.

Filippo Fabiani is an Assistant Professor at the IMT School for Advanced Studies Lucca, Italy. He received the B.Sc. degree in Bio-Engineering, the M.Sc. degree in Automatic Control Engineering, and the Ph.D. degree in Automatic Control, all from the University of Pisa, in 2012, 2015, and 2019 respectively. In 2018-2019 he was post-doctoral Research Fellow in the Delft Center for Systems and Control at TU Delft, the Netherlands, while in 2019-2022 he was a post-doctoral Research Assistant in the Control Group at the Department of Engineering Science, University of Oxford, United Kingdom.

His research interests include game theory, optimization and control of complex uncertain systems, with applications in generation and load side control for power networks and automated driving.

Alberto Bemporad received his Master's degree cum laude in Electrical Engineering in 1993 and his Ph.D. in Control Engineering in 1997 from the University of Florence, Italy. In 1996-97 he was with the Center for Robotics and Automation, Department of Systems Science & Mathematics, Washington University, St. Louis. In 1997-1999 he held a postdoctoral position at the Automatic Control Laboratory, ETH Zurich, Switzerland, where he collaborated as a Senior Researcher until 2002. In 1999-2009 he was with the Department of Information Engineering of the University of Siena, Italy, becoming an Associate Professor in 2005. In 2010-2011 he was with the Department of Mechanical and Structural Engineering of the University of Trento, Italy. Since 2011 he has been a Full Professor at the IMT School for Advanced Studies Lucca, Italy, where he served as the Director of the institute in 2012-2015. He spent visiting periods at Stanford University, University of Michigan, and Zhejiang University. In 2011 he co-founded ODYS S.r.l., a company specialized in developing model predictive control systems for industrial production. He has published more than 400 papers in the areas of model predictive control, hybrid systems, optimization, automotive control, and is the co-inventor of 21 patents. He is the author or coauthor of various software packages for model predictive control design and implementation, including the Model Predictive Control Toolbox (The Mathworks, Inc.) and the Hybrid Toolbox for MATLAB. He was an Associate Editor of the IEEE Transactions on Automatic Control during 2001-2004 and Chair of the Technical Committee on Hybrid Systems of the IEEE Control Systems Society in 2002-2010. He received the IFAC High-Impact Paper Award for the 2011-14 triennial, the IEEE CSS Transition to Practice Award in 2019, and the 2021 SAE Environmental Excellence in Transportation Award. He has been an IEEE Fellow since 2010.