IEEE Control Systems Letters paper presented at
2023 American Control Conference (ACC)
San Diego, CA, USA. May 31 - June 2, 2023

Game Theory for Autonomy: From Min-Max Optimization to Equilibrium and Bounded Rationality Learning

Kyriakos G. Vamvoudakis[1], Filippos Fotiadis[1], João P. Hespanha[2], Raphael Chinchilla[2], Guosong Yang[3], Mushuang Liu[4], Jeff S. Shamma[5], and Lacra Pavel[6]

Abstract— **Finding Nash equilibria in non-cooperative games can be, in general, an exceptionally challenging task. This is owed to various factors, including but not limited to the cost functions of the game being nonconvex/nonconcave, the players of the game having limited information about one another, or even due to issues of computational complexity. The present tutorial draws motivation from this harsh reality and provides methods to approximate Nash or min-max equilibria in non-ideal settings using both optimization- and learning-based techniques. The tutorial acknowledges, however, that such techniques may not always converge, but instead lead to oscillations or even chaos. In that respect, tools from passivity and dissipativity theory are provided, which can offer explanations about these divergent behaviors. Finally, the tutorial highlights that, more frequently than often thought, the search for equilibrium policies is simply vain; instead, bounded rationality and non-equilibrium policies can be more realistic to employ owing to some players' learning imperfectly or being relatively naive – "bounded rational." The efficacy of such plays is demonstrated in the context of autonomous driving systems, where it is explicitly shown that they can guarantee vehicle safety.**

I. Introduction

Game theory is a mathematical and scientific field that investigates the interactions among multiple decision makers with self-interests [1]. Such interactions have long been ubiquitous in civilian and military applications, hence the research interest in game theory has been incessant, continuously advancing it and making it more applicable to real-world systems that operate in multi-agent environments [2]–[4]. At the same time, it is generally acknowledged that game theory is unable to offer a panacea, i.e., a universally effective algorithm that can enable agents (also called players) to adapt or learn the "best" strategies to respond to other players. This is especially true when the other players' strategies are unpredictable and imperfect—"bounded rational."

Central to game theory is the concept of the Nash equilibrium, which is often used to model perfectly rational players, i.e., players that optimally respond to other (perfectly rational) players. It describes a play which, when followed by all players, nobody has an incentive to deviate from it, and can thus be quite attractive to seek. However, computing or searching for Nash equilibria is not always a straightforward task; factors such as the cost function of each player being nonconvex/nonconcave, or the inability of certain players to compute best-response strategies, can significantly hinder the process of finding a Nash equilibrium.

In scenarios where players are unaware of the strategies of one another, particularly increasing attention has been drawn to the question of how players might reach a Nash equilibrium through some sort of iterative process, i.e., through dynamics [5]–[7]. The field investigating this question is known as *learning in games*, and its literature is extensive [8]–[16]. Depending on the specifics of the game, learning algorithms can, in fact, lead to convergence to a Nash equilibrium. However, there are many cases where learning in games does not exhibit such a pleasant convergent behavior, but instead leads to oscillations or even chaos. Motivated by these issues, this tutorial introduces a variety of algorithms in learning in games, answering the question of why these may converge or diverge. Specifically, we study the behavior of better and best reply dynamics, joint strategy fictitious play, log-linear learning, gradient play, and reinforcement learning (RL), in the settings of zero-sum games [17], potential games [18], weakly acyclic games [19], and strictly/strongly monotone games [20]. In addition, insights into why some algorithms/dynamics work smoothly in certain game settings and others do not are also provided from a dissipativity/passivity perspective [20]. However, the tutorial still acknowledges that Nash equilibria are not panaceas and that the search for such equilibria is not always justified, especially when the assumption that each agent plays perfectly is false.

An alternative to the notion of the Nash equilibrium, which is often found to impose less restrictive assumptions regarding the other players' behaviors, is bounded rationality [21]–[24]. Its underlying principle is found in the augmentation of each player with a prespecified cognitive

[1]K. G. Vamvoudakis and F. Fotiadis are with the Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA. e-mail: {kyriakos,ffotiadis}@gatech.edu.

[2]J. P. Hespanha and Raphael Chinchilla are with the Center for Control Dynamical Systems and Computation, University of California, Santa Barbara, CA 93106-9560 e-mail: {hespanha, raphaelchinchilla}@ucsb.edu

[3]G. Yang is with the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854 USA, e-mail: guosong.yang@rutgers.edu

[4]M. Liu is with the Department of Mechanical and Aerospace Engineering, University of Missouri, Columbia, MO 65211 USA, email: ml529@missouri.edu.

[5]J. S. Shamma is with the Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, IL 61801-3080, USA, e-mail: jshamma@illinois.edu.

[6]L. Pavel is with the Department of Electrical and Computer Engineering, University of Toronto, ON, M5S 3G4, Canada, email: pavel@control.toronto.edu.

This work was supported in part, by ARO under grant No. W911NF-19 − 1 − 0270, by ONR Minerva under grant No. N00014 − 18 − 1 − 2160, by NSF under grant Nos. CAREER CPS-1851588, CPS-2038589, CPS- 2227185, and S&AS-1849198, by the NASA University Leadership Initiative (ULI) under grant No. 80NSSC20M0161, and by the Onassis Foundation-Scholarship ID: F ZQ 064 − 1/2020 − 2021.

ability, which dictates their behavior during play. Hence, unlike the equilibrium, bounded rationality does not require all agents to perform a perfect play. This tutorial analyzes three such models of bounded rationality. In the first model, named level-k thinking [21], [24]–[26], each agent assumes that everyone else has a cognitive level immediately lower than theirs, and—given such an assumption—chooses their policy to be the best response to them. In the second model, named cognitive hierarchy [22], [23], [25], [26], each agent conjectures that the rest of the agents have a cognitive level that is lower than theirs, but follows a distribution instead of being deterministic. In the third model, a predictor-corrector structure is employed to correct the agents' expectations of other agents' behaviors [27]. The tutorial showcases that bounded rationality models can in fact be effective in real-world settings, such as in autonomous driving.

The rest of the tutorial paper is organized as follows. Section II studies two-player min-max games. Section III investigates learning algorithms in multi-player general-sum games. Section IV explains the behaviors of these learning algorithms from a dissipativity/passivity perspective. Section V proposes two models to account for agents' possible bounded rationality. Section VI develops a predictor-corrector game and applies it to autonomous driving.

II. Convergent Second-Order Methods for Min-Max Optimizations

In this section, we restrict our attention to two-player min-max games and construct algorithms with local super-linear convergence to local minima and local min-max. Specifically, we address the use of second-order methods to solve optimizations of the form

$$\min_{u\in\mathcal{U}}\max_{d\in\mathcal{D}} f(u,d), \tag{1}$$

for a twice continuously differentiable function $f:\mathcal{U}\times\mathcal{D}\to\mathbb{R}$ and sets $\mathcal{U}\subset\mathbb{R}^{n_u}$, $\mathcal{D}\subset\mathbb{R}^{n_d}$. This type of optimization arises in many applications, including robust machine learning [28], model predictive control [29], and in reformulating stochastic programming as a min-max optimization [30].

When the sets $\mathcal{U}$ and $\mathcal{D}$ are compact and convex and the function $f(u,d)$ is convex with respect to u and concave with respect to d, the min and max in (1) commute [31] and the optimization becomes relatively simple. However, we are especially interested here in problems for which such assumptions do not hold, the min and max do not commute, and for which the optimizations may have local optima that are not global.

Lacking stringent convexity/concavity assumptions, it is generally not possible to construct efficient optimization algorithms that guarantee convergence to global optima of (1) so we will be satisfied with convergence to appropriately defined "local" optimal. However, we will still strive to construct algorithms exhibiting super-linear convergence (i.e., faster than exponential). We start by reviewing the case of a simple unconstrained minimization, which we then use to motivate the algorithm for min-max optimization. The reader is referred to [17] for the proofs of the results presented here as well as for the constrained optimization case.

A. Minimization

Consider the unconstrained minimization $\min_{u\in\mathcal{U}} f(u)$, $\mathcal{U}\coloneqq\mathbb{R}^{n_u}$. A second order iterative method for this optimization can be constructed by using Newton's root-finding algorithm to solve the first-order optimality condition

$$\nabla_u f(u)=0, \tag{2}$$

which leads to the iteration

$$u(k+1)=u(k)-H_{\mathrm{uu}}f\big(u(k)\big)^{-1}\nabla_u f\big(u(k)\big), \tag{3}$$

where $\nabla_u f(u)$ and $H_{\mathrm{uu}}f(u)$ denote the gradient (as a column vector) and Hessian matrix of $f(u)$, respectively, computed at the point u. A few observations are in order:

1) On the positive side, when f is a strictly convex quadratic form, the iteration (3) converges to the (unique) global minimum in a single iteration. Further, if f is not quadratic but still strongly convex, the iteration (3) converges super-linearly as $k\to\infty$ to the (unique) global minimum. We recall that a twice differentiable function is strongly convex if the Hessian matrix satisfies $H_{\mathrm{uu}}f(u)\geqslant\epsilon I$, $\forall u$ for some $\epsilon>0$.
2) On the negative side, for a general twice differentiable function f, any stationary point of f (i.e., any point for which (2) holds) is an equilibrium point of (3). In this case, the iteration (3) may converge either to local minima or local maxima.

This last observation reveals the major weakness of (2): *the Newton root-finding iteration* (3) *ignores whether we are looking for a minimizer or a maximizer* and would remain unchanged if we replaced the optimization criterion from $f(u)$ to $-f(u)$.

Example 1. Consider the optimization,

$$\min_{u\in\mathbb{R}} u^3-3u, \tag{4}$$

for which $\forall u\in\mathbb{R}$,

$$f(u)\coloneqq u^3-3u,\quad \nabla_u f(u)=3u^2-3,\quad H_{\mathrm{uu}}(u)=6u.$$

The corresponding Newton iteration (3) is of the form

$$u(k+1)=u(k)-\frac{3u(k)^2-3}{6u(k)},$$

for which both the local minimum $u^{\min}\coloneqq 1$ and the local maximum $u^{\max}\coloneqq -1$ are locally asymptotically stable equilibria with super-linear convergence. Specifically,

$$\begin{cases} u(0)>0 \quad\Rightarrow\quad u(k)\to u^{\min}\coloneqq 1, \text{(local minimum)},\\ u(0)<0 \quad\Rightarrow\quad u(k)\to u^{\max}\coloneqq -1, \text{(local maximum)},\\ u(0)=0 \quad\Rightarrow\\ \qquad \text{iteration fails since } H_{\mathrm{uu}}(u)=6u \text{ is not invertible.}\end{cases}$$

Moreover, the iteration never actually "converges" to the global "infimum" $u\to-\infty$. □

The example and discussion above motivate the question:

"Can we modify the Newton iteration (3) so that local minima become (locally) asymptotically stable, whereas local maxima become unstable?" We shall see the answer is yes!

To proceed we consider an alternative interpretation of (3), which is based on the second order Taylor expansion of the twice differential function f around a point $u \in \mathcal{U}$:

$$\begin{aligned} f(u+\delta u) =& f(u) + \nabla_u f(u)^\top \delta u \\ &+ \frac{1}{2}\delta u^\top H_{\mathrm{uu}}(u)\delta u + O(\|\delta u\|^3). \end{aligned}$$

For a strongly convex function f, the iteration in (3) can then be written as $u(k+1) = u(k) + \delta u(k)$, where the update $\delta u(k)$ is the optimal increment that minimizes the quadratic approximation at the point $u(k)$, i.e.,

$$\begin{aligned} \delta u(k) \coloneqq & \operatorname*{arg\,min}_{\delta u \in \mathbb{R}^{n_u}} f\big(u(k)\big) \\ & + \nabla_u f\big(u(k)\big)^\top \delta u + \frac{1}{2}\delta u^\top H_{\mathrm{uu}}\big(u(k)\big)\delta u \qquad (5) \\ = & -H_{\mathrm{uu}}\big(u(k)\big)^{-1} \nabla_u f\big(u(k)\big), \end{aligned}$$

where the last equality uses the fact that f is strongly convex and therefore $H_{\mathrm{uu}}\big(u(k)\big)$ is positive definite.

Suppose now that we modify the update $\delta u(k)$ in (5) to:

$$\begin{aligned} \delta u(k) \coloneqq & \operatorname*{arg\,min}_{\delta u \in \mathbb{R}^{n_u}} f\big(u(k)\big) + \nabla_u f\big(u(k)\big)^\top \delta u \\ & + \frac{1}{2}\delta u^\top \Big(H_{\mathrm{uu}}\big(u(k)\big) + \epsilon\big(u(k)\big) I\Big)\delta u \qquad (6) \\ = & -\Big(H_{\mathrm{uu}}\big(u(k)\big) + \epsilon\big(u(k)\big) I\Big)^{-1} \nabla_u f\big(u(k)\big), \qquad (7) \end{aligned}$$

with $\epsilon(u) \geqslant 0$ chosen so that the minimum in (6) is finite and unique, i.e., so that

$$H_{\mathrm{uu}}(u) + \epsilon(u) I > 0. \qquad (8)$$

The modified Newton step in (7) gained an important feature that holds even for non-convex criterion $f(u)$: the computation of (7) never fails since $H_{\mathrm{uu}}\big(u(k)\big) + \epsilon\big(u(k)\big) I$ is always non-singular, and yet the equilibrium points of the modified Newton step remain precisely the stationary points of f since $\delta u(k) = 0 \quad \Leftrightarrow \quad \nabla_u f\big(u(k)\big) = 0$. More importantly, as suggested in [32] and stated formally in the result that follows, all locally stable equilibrium points for the modified Newton step in (7) must be strict local minima. We recall that a strict local minimum $u^\star$ is a unique global minimum in a sufficiently small neighborhood of $u^\star$.

Theorem 1 (Theorem 1 in [17])**.** *Let $f : \mathcal{U} \to \mathbb{R}$, $\mathcal{U} \coloneqq \mathbb{R}^{n_u}$ be a three times differentiable function and $\epsilon : \mathcal{U} \to [0, \infty)$ any differentiable function for which* (8) *holds. For every equilibrium point $u^\star$ of* (7)*, $u^\star$ is locally asymptotically stable if and only if $u^\star$ is a strict local minimum of f.* □

The condition (8) on $\epsilon(u)$ is quite mild and can always be made to hold by choosing $\epsilon(u)$ sufficiently large. However, by selecting large values for $\epsilon(u)$ the convergence speed slows down. In fact, we only get super-linear convergence for $\epsilon(u) = 0$, because only in this case (7) jumps in a single step to the minimum of the quadratic approximation to f. In practice, this means that we should select $\epsilon(u) > 0$ only when $H_{\mathrm{uu}}(u)$ is not positive definite.

Example 2. For the optimization in (4), the modified Newton step in (7) becomes $u(k+1) = u(k) - \frac{3u(k)^2 - 3}{6u(k) + \epsilon\left(u(k)\right)}$ and, for (8) to hold, we need

$$\begin{cases} \epsilon(u) \geqslant 0 & u > 0, \\ \epsilon(u) > -6u & u \leqslant 0. \end{cases} \qquad (9)$$

In this case,

$$\begin{cases} u(0) > u^{\max} \coloneqq -1 & \Rightarrow \quad u(k) \to u^{\min} \coloneqq 1 \\ & \quad \text{(local minimum)}, \\ u(0) < u^{\max} \coloneqq -1 & \Rightarrow \quad u(k) \to -\infty \\ & \quad \text{(global "infimum")}, \\ u(0) = u^{\max} \coloneqq -1 & \Rightarrow \quad u(k) = u^{\max}, \forall k \\ & \quad \text{(unstable equilibrium)}. \end{cases}$$

Selecting the function ϵ with $\epsilon(u) = 0$ around $u^{\min}$ results in super-linear convergence to $u^{\min}$, but if $\epsilon(u^{\min}) \neq 0$, the convergence is only exponential. For example, picking $\epsilon(u) = -6u + \eta$ with $\eta > 0$, (9) holds for all u, but the modified Newton step in (7) becomes $u(k+1) = u(k) - \frac{3u(k)^2 - 3}{\eta}$, which is just a gradient descent. □

B. Min-Max optimization

An intuitive (but surprisingly recent) definition of local min-max for (1) can be stated as follows: a pair $(u^\star, d^\star) \in \mathcal{U} \times \mathcal{D}$ is called a *local min-max* if

1) $d^\star$ is a local maximum of the function $d \mapsto f(d, u^\star)$, and
2) $u^\star$ is a local minimum of the function

$$u \mapsto g_\epsilon(u) \coloneqq \max_{d \in \mathcal{D} \cap B_\epsilon(d^\star)} f(d, u),$$

for every sufficiently small $\epsilon > 0$, where $B_\epsilon(d^\star)$ denotes a closed ball centered at $d^\star$ with radius ϵ [33], [34].

In essence, this means that $u^\star$ achieves the outer minimum in (1) in any sufficiently small neighborhoods of $d^\star$, which can be any point that achieves in the inner maximization (for $u = u^\star$).

As for local minima, this characterization of local min-max admits fairly simple first-order and second-order conditions for optimality:

Theorem 2 (Propositions 18-20 in [33])**.** *Assume that $f : \mathcal{U} \times \mathcal{D} \to \mathbb{R}$, $\mathcal{U} \coloneqq \mathbb{R}^{n_u}$, $\mathcal{D} \coloneqq \mathbb{R}^{n_d}$ is twice differentiable.*

1) Necessity: *If (u, d) is a local min-max point, then*

$$\nabla_u f(u, d) = 0, \quad \nabla_d f(u, d) = 0. \qquad (10)$$

2) Sufficiency: *If (u, d) satisfies* (10) *and*

$$\begin{aligned} & H_{\mathrm{dd}} f(u, d) < 0, \qquad (11) \\ & H_{\mathrm{uu}} f(u, d) \\ & \quad - H_{\mathrm{ud}} f(u, d)\big(H_{\mathrm{dd}} f(u, d)\big)^{-1} H_{\mathrm{du}} f(u, d) > 0 \end{aligned}$$

then (u, d) is a local min-max point. □

In the statement of Theorem 2, $\nabla_u f(u,d)$ and $H_{\text{uu}} f(u,d)$ denote the gradient and Hessian of the function $u \mapsto f(u,d)$; $\nabla_d f(u,d)$ and $H_{\text{dd}} f(u,d)$ the gradient and Hessian of the function $d \mapsto f(u,d)$; and $H_{\text{ud}} f(u,d) \in \mathbb{R}^{n_u \times n_d}$, $H_{\text{du}} f(u,d) \in \mathbb{R}^{n_d \times n_u}$ matrices with second derivatives of f with respect to the entries of u and d.

The construction of a second order method could be based on applying Newton's root finding algorithm to the first order optimality conditions (10) [29]. However, motivated by what we saw in Section II-A, we will instead construct a second order method by solving a (potentially modified) local quadratic approximation to (1). Specifically, we will use the iteration

$$\begin{bmatrix} u(k+1) \\ d(k+1) \end{bmatrix} = \begin{bmatrix} u(k) + \delta u(k) \\ d(k) + \delta d(k) \end{bmatrix}, \tag{12}$$

with updates $\delta u(k)$, $\delta d(k)$ that are local min-max to the following quadratic approximation to (1)

$$\begin{aligned} &\min_{\delta u \in \mathcal{U}} \max_{\delta d \in \mathcal{D}} f + \begin{bmatrix} \nabla_u f^\top \nabla_d f^\top \end{bmatrix} \begin{bmatrix} \delta u \\ \delta d \end{bmatrix} \\ &+ \frac{1}{2} \begin{bmatrix} \delta u^\top & \delta d^\top \end{bmatrix} \begin{bmatrix} H_{\text{uu}} f + \epsilon_u I & H_{\text{ud}} f \\ H_{\text{du}} f & H_{\text{dd}} f - \epsilon_d I \end{bmatrix} \begin{bmatrix} \delta u \\ \delta d \end{bmatrix}, \end{aligned} \tag{13}$$

where all derivatives and the functions $f, \epsilon_u, \epsilon_d : \mathcal{U} \times \mathcal{D} \to \mathbb{R}$ are all computed at the current point $u(k), d(k)$. The quadratic approximation in (13) is accurate up to third order terms when ϵ_u and ϵ_d are both zero, but strictly positive terms may be needed to make sure that (13) has a min-max point. In fact, it is straightforward to show that a unique min-max exists provided that

$$\begin{aligned} &H_{\text{dd}} f - \epsilon_d I < 0, \\ &H_{\text{uu}} f + \epsilon_u I - H_{\text{ud}} f \big(H_{\text{dd}} f - \epsilon_d I \big)^{-1} H_{\text{du}} f > 0, \end{aligned} \tag{14}$$

in which case we have

$$\begin{bmatrix} \delta u(k) \\ \delta d(k) \end{bmatrix} = \begin{bmatrix} H_{\text{uu}} f + \epsilon_u I & H_{\text{ud}} f \\ H_{\text{du}} f & H_{\text{dd}} f - \epsilon_d I \end{bmatrix}^{-1} \begin{bmatrix} \nabla_u f \\ \nabla_d f \end{bmatrix}. \tag{15}$$

In the minimization in Section II-A, selecting the modification ϵI so that the quadratic approximation in (6) had a (finite) unique minimum sufficed to guarantee that all locally asymptotically stable equilibrium points of the modified Newton iteration corresponded to strict local minima. For min-max optimizations, existence of a (finite) unique min-max to (13) turns out not to suffice to create instability for equilibrium points that are not local min-max.

When the function f is three time differentiable and the functions ϵ_u, ϵ_d are differentiable, it is straightforward to show that the local linearization of the dynamical system (12), (15) around an equilibrium point (u,d) has dynamics given by

$$I - \begin{bmatrix} H_{\text{uu}} f + \epsilon_u I & H_{\text{ud}} f \\ H_{\text{du}} f & H_{\text{dd}} f - \epsilon_d I \end{bmatrix}^{-1} \begin{bmatrix} H_{\text{uu}} f & H_{\text{ud}} f \\ H_{\text{du}} f & H_{\text{dd}} f \end{bmatrix}. \tag{16}$$

If we always selected $\epsilon_u = \epsilon_d = 0$, this matrix would be zero, which would be consistent to super-linear convergence at every equilibrium point. Instead, we will select the functions ϵ_u, ϵ_d to satisfy the following three conditions:

C1 *Min-Max sufficiency for quadratic approximation:* The inequalities in (14) always hold.

C2 *Minimal modification:* Whenever the original min-max optimality conditions in (11) hold, we must have $\epsilon_d(u,d) = 0$.

C3 *Instability:* Whenever the original min-max optimality conditions in (11) do not hold, the matrix in (16) must have at least one eigenvalue with absolute value strictly larger than 1.

The condition C1 guarantees that the increments in (15) are indeed min-max points to the quadratic approximation in (13); the condition C2 enforces that we do not modify the quadratic form with some $\epsilon_d \neq 0$ when this is not needed and turns out to also guarantee local exponential stability of min-max points for the original optimizations; and the condition C3 guarantees instability of points that do not satisfy the sufficiency condition (11) for local min-max. While not necessary for local exponential stability of the min-max points, in C2 it would make sense to actually require $\epsilon_d(u,d) = \epsilon_u(u,d) = 0$, as this would lead to super-linear convergence. The result that follows formalizes these observations:

Theorem 3. *Let $f : \mathcal{U} \times \mathcal{D} \to \mathbb{R}$, $\mathcal{U} \coloneqq \mathbb{R}^{n_u}$, $\mathcal{D} \coloneqq \mathbb{R}^{n_d}$ be a three times differentiable function and $\epsilon_u, \epsilon_d : \mathcal{U} \times \mathcal{D} \to [0, \infty)$ be any differentiable functions that satisfy C1–C3. For every equilibrium point $(u^\star, d^\star)$ of* (12)–(13)*, we have that:*

1) *If the 2nd order min-max sufficient conditions* (11) *hold then $(u^\star, d^\star)$ is locally exponentially stable.*
2) *If the 2nd order min-max sufficient conditions* (11) *do not hold, then $(u^\star, d^\star)$ is unstable.* □

To use Theorem 3, it remains to show how to select functions ϵ_u, ϵ_d that satisfy C1–C3. This can be done as follows:

1) For values u, d for which the original min-max optimality conditions in (11) hold, we can simply set $\epsilon_u(u,d) = \epsilon_u(u,d) = 0$. These values guarantee that both C1 and C2 hold.
2) For values u, d for which the original min-max optimality conditions in (11) do not hold, we first pick $\epsilon_d \geqslant 0$ sufficiently large so that $H_{\text{dd}} f - \epsilon_d I < 0$, holds and then $\epsilon_u \geqslant 0$ sufficient large so that $H_{\text{uu}} f + \epsilon_u I - H_{\text{ud}} f \big(H_{\text{dd}} f - \epsilon_d I \big)^{-1} H_{\text{du}} f > 0$ and therefore C1 holds. If C3 does not hold for these values of ϵ_d, ϵ_u, further increase ϵ_u until it does (see Lemma 1 below, which is implicit in [17, proof of Theorem 3]).

Lemma 1 (Theorem 3 in [17])**.** *Suppose that* (11) *does not hold and that $H_{\text{dd}} f - \epsilon_d I < 0$ for some $\epsilon_d \geqslant 0$, then there exists a constant $\epsilon_u \geqslant 0$ sufficiently large so that $H_{\text{uu}} f + \epsilon_u I - H_{\text{ud}} f \big(H_{\text{dd}} f - \epsilon_d I \big)^{-1} H_{\text{du}} f > 0$ and the matrix in* (16) *must have at least one eigenvalue with absolute value strictly larger than 1.* □

We restricted our attention to unconstrained optimization, but the results presented have been extended in [17] to constrained optimizations. An important question that remains mostly unanswered is the construction of similar algorithms applicable to problems for which global min-max are not

local min-max, which is a possibility that cannot arise in the minimization of smooth functions, but can arise in min-max optimizations of smooth functions.

III. Learning in Finite Games

In the preceding section, we investigated min-max optimization problems, and presented procedures that are able to guarantee convergence to a corresponding min-max optimal point. This form of min-max optimization can naturally be interpreted as a two-player game, where the first player wants to maximize the utility function, while the second player wants to minimize it. Therefore, it cannot be of use in games where more than two players are participating, and where the stationary points form a Nash equilibrium instead of a min-max one. Towards this direction, in this section we shift our focus to multi-player general-sum games, and investigate the behaviors of a variety of learning algorithms in such settings, including best and better reply dynamics, fictitious play, log-linear learning, and gradient play.

A. Preliminaries

We consider that there is a set of players, $\mathcal{N} = \{1, \cdots, N\}$, where N can be greater than 2. Each player has a finite set of actions, $\mathcal{A}_i = \{1, \cdots, n_i\}$. The joint action set is $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$. A joint action $a \in \mathcal{A}$ may be written as $a = (a_1, \cdots, a_N)$ with each $a_i \in \mathcal{A}_i$. Alternatively, we may write $a = (a_i, a_{-i})$, which denote the action of player i, a_i, and the actions of all other players, a_{-i}. The utility of player i is a function $u_i : \mathcal{A} \to \mathbb{R}$. The collection of players, $\mathcal{N}$, actions sets, $\mathcal{A}_i$, $i = 1, \cdots, N$, and utility functions, u_i, $i = 1, \cdots, N$, fully specify a finite (normal form) game.

A pure strategy Nash equilibrium is a joint action, $a^\star = (a_i^\star, a_{-i}^\star) \in \mathcal{A}$, such that for all $i \in \mathcal{N}$, $u_i(a_i^\star, a_{-i}^\star) \geqslant u_i(a_i', a_{-i}^\star), \quad \forall a_i' \in \mathcal{A}_i$. Note that, depending on the specific utility functions, there may be a unique Nash equilibrium, multiple Nash equilibria, or none. For both potential games and weakly-acyclic games, a Nash equilibrium is guaranteed to exist. Note that potential games are a subset of weakly-acyclic games in that for potential games, all directed paths terminate at a Nash equilibrium.

The following definitions lead to alternative expressions for a Nash equilibrium and will be useful in defining certain learning dynamics. Particularly, define the *best reply set* as

$$B_i^\star(a_{-i}) = \left\{ a_i' \in \mathcal{A}_i \;\middle|\; u_i(a_i', a_{-i}) \geqslant u_i(a_i, a_{-i}), \forall a_i \in \mathcal{A}_i \right\},$$

i.e., the set of actions that maximize the utility for player i in response to a_{-i}. In terms of the best reply set, $a^\star$ is a Nash equilibrium if for all $i \in \mathcal{N}$, $a_i^\star \in B_i^\star(a_{-i}^\star)$.

Define the *better reply set* as

$$B_i(a_i, a_{-i}) = \left\{ a_i' \in \mathcal{A}_i \;\middle|\; u_i(a_i', a_{-i}) > u_i(a_i, a_{-i}) \right\}$$

i.e., the set of actions that are an improvement to a baseline action. Note that $B^\star(\cdot)$ and $B(\cdot)$ have different domains. Furthermore, the strict inequality in the better reply set definition implies that the better reply set can be empty.

The better reply set induces the directed better reply graph, defined as follows. The nodes are the set of joint actions, i.e., the set $\mathcal{A}$. Given two joint actions, a and a', there exists an edge from a to a' if:

- The two actions deviate by the action of a single player, i.e., $a = (a_i, a_{-i})$ and $a' = (a_i', a_{-i})$ for some $i \in \mathcal{N}$.
- For the deviating player i, $a_i' \in B_i(a_i, a_{-i})$, i.e., the deviating player experienced an increase in utility.

By definition, a Nash equilibrium is a node that will not have any outgoing edges in the better reply graph.

Some discussions will apply to special classes of games:

- **Potential games [18]:** There exists a potential function, $\phi : \mathcal{A} \to \mathbb{R}$, such that for any two joint actions $a = (a_i, a_{-i})$ and $a' = (a_i', a_{-i})$ that deviate by the action of a single player,
$$\phi(a_i, a_{-i}) - \phi(a_i', a_{-i}) = u_i(a_i, a_{-i}) - u_i(a_i', a_{-i}).$$
- **Weakly-acyclic games [19]:** For any joint action $a \in \mathcal{A}$, there exists a directed path that terminates at a Nash equilibrium.
- **Zero-sum games:** There are two players, i.e., $\mathcal{N} = \{1, 2\}$, and $u_1(a_1, a_2) = -u_2(a_1, a_2)$.

We also will be interested in dynamics that utilize randomized actions. Towards this end, let $S = \{s_1, \cdots, s_K\}$, be a finite-set, and define $\Delta[S] \subset \mathbb{R}^m$ to be the probability simplex over S. We will use the following notations to discuss randomized actions:

- $s = \mathbf{rand}[p]$ indicates the element s is randomly selected according to the probability distribution, $p \in \Delta[S]$;
- $s = \mathbf{unif}[S]$ indicates the element s is selected according to a uniform distribution over S;
- $\mathbf{vert}[s_i] \in \Delta[S]$ denotes the simplex vertex vector associated with element, $s_i \in S$. For example, for $S = \{s_1, s_2, s_3\}$,
$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$
are the vertex vectors associated with s_1, s_2, and s_3, respectively.

B. Special Cases of Convergence to Nash Equilibrium

We will consider learning dynamics that evolve over discrete time, $t = 0, 1, 2, \cdots$. Informally, at stage t, the action of player i is selected according to $a_i(t) = \mathcal{L}(\mathcal{I}_i(t))$. Here, $\mathcal{I}_i(t)$ denotes the information available to player i up to stage t, $\mathcal{L} : \mathcal{I}(t) \to \Delta[\mathcal{A}_i]$ is a learning rule that maps player i's information to the probability simplex of its actions, and $a_i(t)$ is randomly selected according to this probability distribution. In some cases of learning dynamics, the above will apply only to players that are activated at stage t, i.e., given the opportunity to revise their action. Non-activated players must repeat their previous action, i.e., $a_i(t) = a_i(t-1)$, if player i is not activated.

- ***Best and Better Reply Dynamics***

A very simple example of a learning rule is the best reply dynamics. Introduce the inertia probability, ρ, with $0 < \rho <$

1. Let $a(0) \in \mathcal{A}$ be an arbitrary initialization. At each stage $t = 1, 2, \cdots$,

$$a_i(t) = a_i(t-1), \quad \text{with probability } \rho, \text{ or}$$
$$a_i(t) = \mathbf{unif}[B_i^\star(a_{-i}(t-1))], \quad \text{with probability } 1-\rho.$$

In words, a player is activated with probability $1-\rho$ and selects the best reply to the previous actions of other players.

A slightly more sophisticated learning rule is better reply dynamics. As before, introduce the inertia probability, ρ, with $0 < \rho < 1$, and let $a(0) \in \mathcal{A}$ be an arbitrary initialization. At each stage $t = 1, 2, \cdots$,

$$a_i(t) = a_i(t-1), \quad \text{with probability } \rho, \text{ or}$$
$$a_i(t) = \mathbf{unif}[B_i(a_i(t-1), a_{-i}(t-1))], \text{ with probability } 1-\rho.$$

In case the better reply set is empty, then $a_i(t) = a_i(t-1)$. Under better reply dynamics, an activated player selects an action that is an improvement, as compared to its previous action, given the previous actions of other players.

Proposition 1. *(i) For potential games, best reply dynamics converge to a Nash equilibrium; (ii) For weakly acyclic games, better reply dynamics converge to a Nash equilibrium.* □

The proof stems from Nash equilibrium being a stationary point of the associated dynamics. Convergence is guaranteed since, from any joint action that is not a Nash equilibrium, there is a positive probability to reach a Nash equilibrium.

These dynamics presume that the actions of other players are observable. Alternatively, one can assume that a player can only observe its realized utility. That is, at stage t, player i observes $u_i(t)$. In this case, it is possible to derive algorithms that have nice properties for weakly acyclic games. These algorithms introduce an experimentation probability, ϵ. The notion of convergence is slightly weakened, however. Rather than the joint action, $a(t)$, converging to a Nash equilibrium, one has that a $t \to \infty$, the joint action $a(t)$ is at a Nash equilibrium with high probability, and this probability approaches unity as the experimentation probability, ϵ, approaches zero [9].

- ***Joint Strategy Fictitious Play***

In joint strategy fictitious play [10], players keep a running tally of the utility associated with each action. For player i at stage t, define

$$U_i(t) = \begin{pmatrix} u_i(1, a_{-i}(t)) \\ \vdots \\ u_i(n_i, a_{-i}(t)) \end{pmatrix} \in \mathbb{R}^{n_i}.$$

This vector indicates the hypothetical utility that would have been received by player i at stage t for each of its actions. Now define $V_i(t) = (1-\gamma(t))V_i(t) + \gamma(t)U_i(t)$. For $\gamma(t) = \frac{1}{t+1}$, $V_i(t)$ is a running average of the utility associated with each action of player i. For constant $0 < \gamma(t) = \gamma^\star < 1$, $V_i(t)$ represents a fading memory weighted average of $U_i(t)$.

Joint strategy fictitious play proceeds as follows. As before, introduce the inertia probability, ρ, with $0 < \rho < 1$. For each i, let $V_i(0)$ be an arbitrary initialization. For $t = 1, 2, \cdots$,

$$a_i(t) = a_i(t-1), \quad \text{with probability } \rho, \text{ or}$$
$$a_i(t) = \arg\max_k \left\{V_i^1(t-1), \cdots, V_i^{n_i}(t-1)\right\}.$$

Proposition 2. *For potential games, joint strategy fictitious play converges to a Nash equilibrium.* □

See [10] for the full presentation, including specific technical assumptions, and proof.

- ***Log-Linear Learning and Equilibrium Selection***

Log-linear learning, introduced in [8], is a modification of best reply dynamics in which players take a noisy best reply. That is, there is a positive probability of playing any action—not just a best reply. One can view not playing a best reply as a form of experimentation, and the probability of experimenting depends on the loss of utility as a result of the experimentation. To proceed, first define the Boltzmann distribution as follows. For $v \in \mathbb{R}^K$ and for $T > 0$, define $\beta(v;T) \in \mathbb{R}^n$ as the vector whose k^{th} component equals $\beta_k(v;T) = \frac{1}{Z}e^{v_k/T}$, where Z is a v-dependent normalization factor so that $\beta(v;T)$ is a probability distribution, i.e., $\sum_{k=1}^K \beta_k(v;T) = 1$.

Now, log-linear learning is defined as follows. Let $a(0) \in \mathcal{A}$ be an arbitrary initialization. At each stage $t = 1, 2, \cdots$, select a single player, say $i^\star(t)$, uniformly at random. Then, $a_i(t) = \beta(U_i(t-1);T), \quad i = i^\star(t)$, otherwise, $a_i(t) = a_i(t-1)$. Here, $U_i(t-1)$ is the hypothetical utility vector defined under joint strategy fictitious play.

The interpretation of log-linear learning being noisy best reply dynamics stems from the properties of the Boltzmann distribution. For large T, the Boltzmann distribution approximates uniform probability. For small T, the Boltzmann distribution favors a maximizer of its argument, i.e., $\beta(v;T)$ places vanishing weight, as $T \downarrow 0$, on all but the maximal elements of v.

Proposition 3 ([8])**.** *Consider a potential game with potential function ϕ. For any $a \in \mathcal{A}$, under log-linear learning,*

$$\lim_{t\to\infty} \mathbf{Pr}\left[a(t) = a\right] = \frac{1}{Z}e^{\phi(a)/T},$$

where Z is a normalizing factor $Z = \sum_{a\in\mathcal{A}} e^{\phi(a)/T}$. □

Log-linear learning induces a finite-state Markov chain where the states are the set of joint actions, i.e., $\mathcal{A}$. Accordingly, the joint actions, $a(t)$, never converge. Nonetheless, one can characterize long-term behavior as follows. This proposition characterizes the stationary distribution of the associated Markov chain. In doing so, it introduces an element of *equilibrium selection* in learning. Thus far, it was stated that certain learning rules (for potential and weakly acyclic games) converge to a Nash equilibrium. Log-linear learning demonstrates a preference among Nash equilibria.

For a potential game, let $a^\star$ denote the maximizer of the potential function, i.e., $a^\star = \arg\max_{a\in\mathcal{A}} \phi(a)$. Note that $a^\star$ is a Nash equilibrium, since by the definition of a potential game, no individual player can increase its utility by

deviating from $a^\star$. Under log-linear learning the probability that $a(t) = a^\star$ in the long run approaches unity as T approaches zero. Since $a^\star$ is one Nash equilibrium out of possibly many, this property reflects favoring a specific Nash equilibrium.

Further details may be found in [35], which considers variations on log-linear learning such as limited observations of other players, simultaneous play, and constrained evolution.

C. Learning over Mixed Strategies

While the learning algorithms discussed thus far involve some form of randomization over action, the discussion has been limited to convergence to pure strategy Nash equilibria where players do not randomize. We now discuss learning over mixed strategies, where the notion of equilibrium itself involves randomization.

- ***Setup***

Without changing notation, we now define utility functions extended to the probability simplex, i.e., $u_i : \Delta[\mathcal{A}_1] \times \cdots \times \Delta[\mathcal{A}_N] \to \mathbb{R}$. Let $p_i \in \Delta[\mathcal{A}_i]$ have components $p_i = \big(p_i(1), \cdots, p_i(n_i)\big)$, and for each $i \in \mathcal{N}$, define

$$u_i(p_1, \cdots, p_N) = \sum_{(a_1, \cdots, a_N) \in \mathcal{A}} p_1(a_1) p_2(a_2) \cdots p_N(a_N) u_i(a_1, \cdots, a_N),$$

i.e, the expected utility when players randomize independently over the mixed strategies $(p_1, \cdots, p_N)$.

We now define mixed strategy Nash equilibrium in a similar manner to pure strategies. A mixed strategy Nash equilibrium is a set of joint probabilities, $p^\star = (p_1^\star, \cdots, p_N^\star) \in \Delta[\mathcal{A}_1] \times \cdots \times \Delta[\mathcal{A}_N]$, such that for all $i \in \mathcal{N}$,

$$u_i(p_i^\star, p_{-i}^\star) \geqslant u_i(p_i', p_{-i}^\star), \quad \forall p_i' \in \Delta[\mathcal{A}_i].$$

In case each p_i lies on a vertex of $\Delta[\mathcal{A}_i]$, the associated equilibrium is a pure strategy equilibrium. Unlike the case of pure strategy Nash equilibria, there will always exist at least one mixed strategy Nash equilibrium (e.g., [36]), which need not be unique.

- ***Smooth Fictitious Play***

To simplify the exposition, we will restrict the discussion to games that have a *pairwise interaction* structure. For such games, there exist matrices, $M_{ij} \in \mathbb{R}^{n_i \times n_j}$, so that $u_i(p_i, p_{-i}) = \sum_{j \in \mathcal{N}, j \neq i} p_i^\top M_{ij} p_j$.

As before, play proceeds over stages $t = 1, 2, \cdots$. At stage t, player i selects an action according to $a_i(t) = \mathbf{rand}[p_i(t)]$, where we will define $p_i(t) \in \Delta[\mathcal{A}_i]$ momentarily. For each player, i, introduce the associated empirical frequency vector $q_i(t) \in \Delta[\mathcal{A}_i]$, which evolves according to

$$q_i(t+1) = (1 - \gamma(t)) q_i(t) + \gamma(t) \mathbf{vert}[a_i(t)],$$

with step size $\gamma(t) = \frac{1}{t+1}$. In case $q_i(0) = 0$, $q_i(t)$ tracks the histogram of actions taken by player i, i.e., the relative frequencies that player i used each of its actions.

In smooth fictitious play [37], players choose their actions as a noisy best response to these empirical frequencies, i.e.,

$$p_i(t) = \beta\Big(\sum_{j \in \mathcal{N}, j \neq i} M_{ij} q_j(t); T \Big),$$

where $\beta(\cdot)$ is the previously defined Boltzmann distribution.

The noisy best response can be viewed as a best response to a perturbed utility function. For $T > 0$, define

$$\tilde{u}_i(p_i, p_{-i}; T) = \sum_{j \in \mathcal{N}, j \neq i} p_i^\top M_{ij} p_j - T \sum_{k=1}^{m_i} p_i(k) \log(p_i(i)),$$

which is the original utility function perturbed by the entropy of strategy p_i. Setting $T = 0$ results in the original utility function. With this modification,

$$\tilde{u}_i\Big(\beta\Big(\sum_{j \in \mathcal{N}, j \neq i} p_i^\top M_{ij} p_j; T \Big), p_{-i}; T\Big) \geqslant \tilde{u}(p_i', p_{-i}; T), \quad \forall p_i' \in \Delta[\mathcal{A}_i],$$

i.e., the best response to the perturbed utility takes the form of the Boltzmann distribution.

Following the terminology in [11], a *Nash distribution* is a Nash equilibrium of the perturbed game, i.e., $p^\star = (p_1^\star, \cdots, p_N^\star) \in \Delta[\mathcal{A}_1] \times \cdots \times \Delta[\mathcal{A}_N]$, such that for all $i \in \mathcal{N}$,

$$\tilde{u}_i(p_i^\star, p_{-i}^\star; T) \geqslant \tilde{u}_i(p_i', p_{-i}^\star; T), \quad \forall p_i' \in \Delta[\mathcal{A}_i].$$

Proposition 4 (See [12], [13])**.** *For any $T > 0$, empirical frequencies converge to a Nash distribution for zero-sum and potential games.* □

The work in [12] also discusses additional classes of games under which smooth fictitious play converges. See also [38].

- ***Gradient play***

Recall the utility function under pairwise interactions is

$$u_i(p_i, p_{-i}) = \sum_{j \in \mathcal{N}, j \neq i} p_i^\top M_{ij} p_j.$$

Taking the gradient from the perspective of player i yields

$$\nabla_{p_i} u_i(p_i, p_{-i}) = \sum_{j \in \mathcal{N}, j \neq i} M_{ij} q_j.$$

In gradient play [39]–[44], a player's action moves its empirical frequency in the direction of the gradient of its utility function. Empirical frequencies are computed as in smooth fictitious play. Instead of a noisy best response at stage t, $a_i(t) = \mathbf{rand}[p_i(t)]$ with

$$p_i(t) = \Pi_\Delta\Big[q_i(t) + \sum_{j \in \mathcal{N}, j \neq i} M_{ij} q_j \Big],$$

where $\Pi : \mathbb{R}^{m_i} \to \Delta[\mathcal{A}_i]$ is the projection to the simplex.

- ***Analysis***

Learning over mixed strategies induces a discrete-time stochastic process. A widely method of analysis is to construct a deterministic continuous time dynamical system derived from the stochastic discrete-time iterations. This approach, known as the ODE (ordinary differential equation) method of stochastic approximation, is detailed in [45], [46] and applied to learning in games in [14]–[16].

In the case of smooth fictitious play, the associated differential equation is

$$\dot{q}_i = -q_i + \beta\Big(\sum_{j\in\mathcal{N},j\neq i} M_{ij}q_j(t);T\Big), \quad i = 1,2,\cdots,N.$$

Similarly, the associated differential equation for gradient play is

$$\dot{q}_i = -q_i + \Pi_\Delta\Big[q_i(t) + \sum_{j\in\mathcal{N},j\neq i} M_{ij}q_j\Big], \quad i = 1,2,\cdots,N.$$

Informally, applying the methods described in [45], [46] leads to the following: the stochastic iterations converge to (i) with strictly positive probability to a local attractor of the ODE; (ii) with probability one to a global attractor of the ODE; and (iii) with zero probability to an exponentially unstable equilibrium of the ODE. Once the analysis falls on the side of continuous dynamical systems, then one can appeal to a variety of analysis techniques [47], including methods derived from feedback control perspectives [48]–[51].

D. Limitations on Learning and Uncoupled Dynamics

The discussion thus far has been about positive examples of convergence. There are many known cases of non-convergence [47, Chapter 9] of learning dynamics as well:

- Best and better response dynamics need not converge in games with a pure equilibrium [52, Chapter 3].
- Fictitious play need not converge in games with a pure equilibrium [53].
- Fictitious play need not converge in games with a unique mixed equilibrium [54].
- Gradient play cannot converge to a mixed strategy equilibrium in matrix games (based on a zero-trace condition in the associated ODE).

It is important to recognize that learning in games is not about the computation of Nash equilibria, which is known to be intractable from the perspective of both computation [55] and communication [56]. Rather, the motivation is to understand, even as a plausibility argument, how a Nash equilibrium may arise out of simple adaptive interactions. Accordingly, non-convergence need not constitute a disqualifying behavior and has been proposed as a possible solution concept in itself [57].

A common restriction of dynamics studied under learning in games is the notion of *uncoupled dynamics*, introduced in [58]. Learning dynamics are considered uncoupled if the evolution of a player's strategy does not depend *explicitly* on the utility functions of other players. There may be implicit dependence through the actions of others. This restriction again stems from the program of learning in games not being one of computation, but rather motivated by modeling.

We see that all the dynamics presented herein—best/better reply, fictitious play, log-linear learning, and gradient play—are uncoupled dynamics. For pure Nash equilibria, reference [59] shows that there are no uncoupled rules with one-stage memory (as in best/better reply dynamics) that converge to Nash equilibria in all games for which there exists a pure Nash equilibrium. Uncoupled dynamics play a role in reference [58], which shows that a broad class of fixed-order learning dynamics cannot converge to Nash equilibrium for a specific 3-player/2-action game [54] (although higher-order learning dynamics [60], [61] are able to overcome this perceived impossibility [42]). These results illustrate the lack of a universal (uncoupled) dynamic that converge to Nash equilibrium for all games. In contrast, there are universal algorithms for stochastic notions of convergence (e.g., meta-stability) [62] or alternative solution concepts (e.g., correlated equilibria) [63].

IV. Passivity and Learning

The preceding discussion renders clear that, in general, it can be very difficult to guarantee that learning dynamics will converge to a Nash equilibrium. This is because each individual learning algorithm's behavior highly depends on the corresponding game setting. The purpose of this section is to thus shed some light on these dependencies, and discuss how they can be explained if one regards them through a passivity point of view.

Passivity is a general input-output system property, a special case of dissipativity, a concept introduced by Willems [64]. These concepts have had numerous applications in control, but in game theory they have started to be used only recently. In [65] passivity was used to analyze gradient-play for a particular CDMA power control application, while in the context of population games, the notion of δ-passivity was used to analyze certain game dynamics in [66]. Here, we use incremental and equilibrium-independent passivity, [67], drawing on results from [61].

We focus on two representative instances in RL: payoff-based play, [5], and Q-learning, [6]. We show how one can exploit geometric features of different classes of games, together with dissipativity/passivity properties of interconnected systems to guarantee global convergence to a Nash equilibrium. Besides simplifying the proof of convergence, one can generate algorithms that work for classes of games with less stringent assumptions, by using passivity and basic properties of interconnected systems.

A. Game Theory and Nash Equilibrium Seeking

In this section we review the framework of learning or seeking a Nash equilibrium in multi-player games. Consider a set of players or agents $\mathcal{N} = \{1,\ldots,N\}$ involved in a game. The game can be a *continuous-action* game, where each player $i \in \mathcal{N}$ has a continuous action set to select its decision from $\Omega_i \subset \mathbb{R}^{n_i}$, or a *finite-action* game, where it has a finite set of actions (pure strategies) $\mathcal{A}_i$, $|\mathcal{A}_i| = n_i$, [7], [68]. Each player $i \in \mathcal{N}$ aims to take a decision x_i, so as to minimize its (expected) cost J_i or maximize its (expected) payoff/utility $\mathcal{U}_i = -J_i$. Its cost/payoff depends on the opponents' strategies x_{-i} and such inherent intertwining between the agents' decisions introduces challenges in solving a game: when optimizing for its own reward, an agent needs to know the others' decisions. A Nash equilibrium is a state where none of the agents has any incentive to

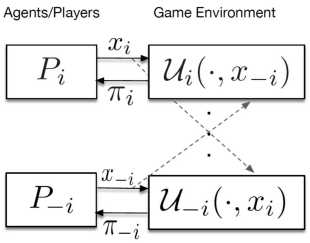

Fig. 1: A generic RL NE seeking dynamics.

change its decision, in that sense called individually optimal. In a classical setting, an introspective calculation of a Nash equilibrium requires complete knowledge: namely, that each player knows the cost/payoff functions and the strategies of all the other players, [7]. This is quite restrictive. An agent has some knowledge, but this is limited/incomplete, so it must be supplemented with whatever information an agent can get by feedback.

We consider a *repeated* game, where players use previous game iterations to gather information about the other agents or the game, and correspondingly, adjust their decisions, that is "learn." Typical learning rules/dynamics proposed in the game theoretic literature are best-response (fictitious-play), projected-gradient (better-response) play, proximal play, RL (payoff-based learning) and so on, [7], [68]. Some are motivated by the (potentially bounded) rationality of the players, [7], others are motivated by biologically inspired learning mechanisms, such as imitation or RL, [68]. These processes can be modeled either in discrete time or in continuous time. Let P_i denote the algorithm or dynamics by which agent $i \in \mathcal{N}$ updates in time its decision x_i (action/strategy). All agents P_is are interconnected one with another, directly or indirectly, depending on the feedback an agent gets. This results in one big interconnected dynamic system denoted as P, which is the overall Nash equilibrium (NE) seeking (or learning dynamics) system. This system can be equivalently written as the feedback interconnected system $P = (P_i, P_{-i})$, where P_{-i} denotes the learning algorithm/dynamics used by everybody else, except player i. A generic RL NE seeking dynamics is shown in Figure 1. Two properties are essential:

(C1) an equilibrium point of P is (related to) a NE of the game denoted $x^\star$, and
(C2) such equilibrium is (globally) asymptotically stable.

It is in this context that passivity can help. Dissipativity and passivity play a critical a role in analyzing interconnections of dynamical systems, but classical notions are defined with respect to the origin as equilibrium point, [64], [69]. On the other hand, *equilibrium independent* dissipativity and passivity (EID/EIP) are defined for an arbitrary equilibrium point, [67], [70]. A dynamical system Σ, given by $\dot{x} = f(x,u), y = h(x,u)$, with $\overline{u}, \overline{x}, \overline{y}$ an equilibrium condition, is *Equilibrium Independent Passive* (EIP) if there exists a differentiable, positive semi-definite storage function $V_{\overline{x}} : \mathbb{R}^n \to \mathbb{R}$ such that $\dot{V}_{\overline{x}}(x) \leqslant (y-\overline{y})^\top (u-\overline{u})$. Σ is *output-strictly* EIP if for some $\beta > 0$ $\dot{V}_{\overline{x}}(x) \leqslant (y-\overline{y})^\top (u-\overline{u}) - \beta \|y-\overline{y}\|^2$.

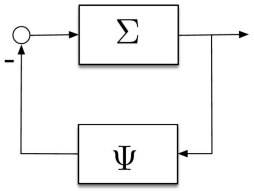

Fig. 2: Feedback configuration

Such individual properties of the component subsystems can help in analyzing stability properties of an interconnected system without precise knowledge of an equilibrium point, but only knowing that it exists. Furthermore, for an interconnected system made up of EID/EIP subsystems, a good candidate for a Lyapunov function is the sum of the individual storage functions, [67]. A static nonlinear EIP/incrementally passive mapping is, equivalently, monotone.

Since the Nash equilibrium $x^\star$ is unknown a-priori, equilibrium independent properties are particularly useful to exploit in NE seeking. For the interconnected system $P = (P_i, P_{-i})$, if each individual P_i has EIP properties, the problem is easily solved. However, because of the coupling between agent objectives, such individual EID/EIP properties do not hold in a game in general. An alternative idea is to see if the *overall NE seeking dynamics/system* P can be recast as an interconnection of some (strictly) EIP systems. In the next section, we show that two popular RL algorithms can be recast as a feedback interconnection $(\Sigma, -\Psi)$, cf. Fig. 2, between some EIP/EID dynamical system Σ and a specific game mapping Ψ. Once (C1) holds, then (C2), that is (global) stability of the equilibria of P, or convergence to a NE of the game, follows easily from passivity properties of interconnected systems.

B. RL and Passivity

In this section, we consider a repeated finite-action game and two instances of RL, namely, payoff-based RL (P-RL) and Q-learning, respectively. In this setting, an agent i does not necessarily know the structure/form of its own payoff/cost function $\mathcal{U}_i/J_i$ but can know its own realized payoff/cost π_i as a result of some action it takes. This represents the reinforcement signal it gets from playing the game, [5], [6]. Update rules build off this setting are called RL algorithms, cf. Fig. 1.

At each iteration k of play, each player $i \in \mathcal{N}$ uses an action $j \in \mathcal{A}_i$ or a pure strategy $\mathbf{e}_i(k)$, selected randomly out of its n_i possible choices, with probability $x_{ij}(k)$. Equivalently, $\mathbb{P}[\mathbf{e}_i(k) = \mathbf{e}_j] = x_{ij}(k)$, where $\mathbf{e}_j$ is the j^{th} unit vector in $\mathbb{R}^{n_i}$. Accordingly, player i receives a payoff value $\pi_i(k) := \mathcal{U}_i(\mathbf{e}_i(k), \mathbf{e}_{-i}(k))$, called its realized payoff at step k, where $\mathbf{e}_{-i}(k)$ denotes the pure strategy profile used by the others, except player i. Player i's mixed strategy $x_i(k) := (x_{ij}(k))_{j \in \mathcal{A}_i} \in \Delta_i$ specifies the probabilities with which actions/pure strategies are selected, with Δ_i denoting the simplex. Because of randomization, each player i is optimizing its expected payoff $\mathcal{U}_i(x_i, x_{-i})$.

In RL, (mixed) strategies are updated based on the re-

ceived payoff π_i and some internal score variables. Consider that each player $i \in \mathcal{N}$ keeps a *score* $z_i \in \mathbb{R}^{n_i}$ of all its actions (updated based on its received payoff π_i) and maps this score into a mixed strategy $x_i \in \Delta_i$ via a static choice map $\sigma_i : \mathbb{R}^{n_i} \to \Delta_i$, $x_i = \sigma_i(z_i)$. A typical choice map is the soft-max function $\sigma_i(z_i) := \frac{1}{\sum_{j\in\mathcal{A}_i}\exp(\frac{1}{\epsilon}z_{ij})}\left[\exp(\frac{1}{\epsilon}z_{i1})\ldots\exp(\frac{1}{\epsilon}z_{in_i})\right]^\top$, where $\epsilon > 0$ is a regularization/temperature parameter, [71]. As $\epsilon \to \infty$, actions are selected with uniform probability ("exploration"), while as $\epsilon \to 0$, the soft-max function selects the action associated with the highest score (best-response/"exploitation").

Consider that at the k-th instance of play, player i updates its score $z_i(k)$ and mixed strategy $x_i(k)$ as

$$z_i(k+1) = z_i(k) + \alpha_i(k)\,\pi_i(k)\,\mathrm{diag}\left(\frac{1}{x_i(k)}\right)\mathbf{e}_i(k)$$
$$x_i(k) = \sigma_i(z_i(k)). \tag{17}$$

This is known as P-RL, [71], related to that proposed by Erev and Roth, [5]. Alternatively, when the update is

$$z_i(k+1) = z_i(k) + \alpha_i(k)\ \mathrm{diag}\left(\frac{\pi_i(k)\mathbf{1} - z_i(k)}{x_i(k)}\right)\mathbf{e}_i(k)$$
$$x_i(k) = \sigma_i(z_i(k)), \tag{18}$$

the scheme is known as the (individual) Q-learning algorithm [6], [72]. In the above, $\alpha_i(k)$ is a diminishing step-size, for example $\frac{1}{k+1}$, $\mathbf{1}$ is the all ones vector and $\mathrm{diag}(\frac{1}{x_i(k)})$ denotes the diagonal matrix with $1/x_{ij}(k)$ on its diagonal.

It is known that P-RL converges in 2x2 games, in 2-player partnership (potential) games, but cycles in 2-player zero-sum games with a unique mixed NE. A prototypical example of the latter class of games is the standard 2-player Rock-Paper-Scissor game. On the other hand, Q-learning converges in all these classes of games, (Proposition 4.2, [6]). What is the reason for such a different behavior? In the following, we show that this can be explained via passivity.

Under standard assumptions, the long-term behavior of stochastic processes (17) and (18) can be analyzed via stochastic approximation, [73], based on the behavior of their deterministic mean dynamics. The mean dynamics can be obtained by taking the expectation of the stochastic iterate increments, using $\mathbb{E}\left(\pi_i(k)\,\mathrm{diag}\left(\frac{1}{x_i(k)}\right)\mathbf{e}_i(k)\right) = U_i(x_{-i}(k))$, where $\mathbb{E}(\cdot)$ denotes expectation and $U_i(x_{-i}) := (\mathcal{U}_i(\mathbf{e}_j, x_{-i}))_{j\in\mathcal{A}_i}$. The mean dynamics of P-RL (17) is

$$P_i : \begin{cases} \dot{z}_i = U_i(x_{-i}), \\ x_i = \sigma_i(z_i), \qquad z_i(0) \in \mathbb{R}^{n_i}. \end{cases} \tag{19}$$

This processing is shown in Fig. 3, where x_{-i} and Δ_{-i} are the mixed strategy and simplex set for everybody else, except i. In particular, the score z_i is the *dual* variable to the *primal* variable x_i. Therefore, (19) describe the evolution of learning in the *dual space* $\mathbb{R}^{n_i}$, whereas the strategy trajectory in Δ_i is induced via the choice map, σ_i.

The mean dynamics of Q-learning (18) are

$$P_i : \begin{cases} \dot{z}_i = U_i(x_{-i}) - z_i, \quad z_i(0) \in \mathbb{R}^{n_i} \\ x_i = \sigma_i(z_i), \end{cases} \tag{20}$$

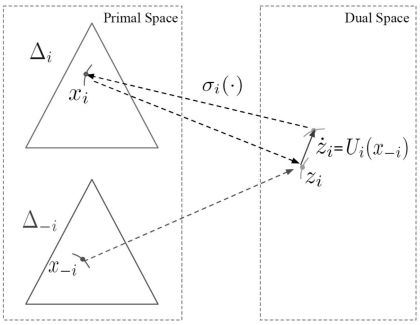

Fig. 3: Payoff-based RL.

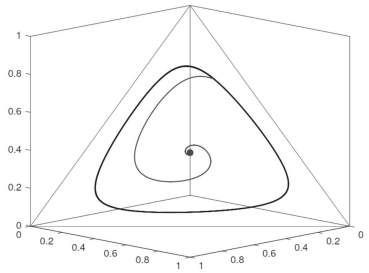

Fig. 4: RL in the Rock-Paper-Scissor game.

which is an "exponentially discounted" score dynamics.

Consider the Rock, Paper, Scissor (RPS) game, a 2-player zero-sum game, a benchmark game where the two payoff matrices are

$$A = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}, B = A^\top.$$

The expected payoffs of the two players are $\mathcal{U}_1(x_1, x_2) = x_1^\top A x_2$ and $\mathcal{U}_2(x_1, x_2) = x_1^\top B x_2$, respectively. The expected payoff vector $U(x) = (U_i(x_{-i}))_{i\in\mathcal{N}} = (\nabla_{x_i}\mathcal{U}_i(x))_{i\in\mathcal{N}}$ is $U(x) = \begin{bmatrix} 0 & A \\ B^\top & 0 \end{bmatrix} x = \Phi x$, where $\Phi + \Phi^\top = 0$. The RPS game is a zero-sum (null) monotone game, with a unique (mixed) NE strategy at $(1/3, 1/3, 1/3)^\top$. Fig. 4 shows strategy trajectories for player 1, under payoff-based (P-RL) dynamics (blue) and under Q-learning dynamics (red), respectively. Q-learning converges to the unique mixed Nash equilibrium $(1/3, 1/3, 1/3)^\top$, while P-RL dynamics cycles.

A passivity approach in the *dual space* can be used to explain this behavior. The overall P-RL learning of all agents (19) is

$$P : \begin{cases} \dot{z} = U(x), \quad z(0) \in \mathbb{R}^n \\ x = \sigma_\epsilon(z), \end{cases} \tag{21}$$

where $\sigma_\epsilon(z) := (\sigma_i(z_i))_{i\in\mathcal{N}}$, with equilibria $\overline{x}^\star = \sigma_\epsilon(\overline{z}^\star)$

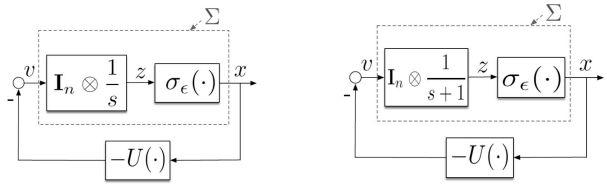

Fig. 5: P-RL and Q-learning

such that $U(\overline{x}^\star) = 0$. Here $z = (z_i)_{i\in\mathcal{N}} \in \mathbb{R}^n$, $x = (x_i)_{i\in\mathcal{N}} \in \Delta$ and $U(x) = (U_i(x_{-i}))_{i\in\mathcal{N}}$ denote the player stacked scores, mixed-strategies and payoff game mapping, respectively.

The overall Q-learning of all agents (20) is

$$P : \begin{cases} \dot{z} = U(x) - z, \quad z(0) \in \mathbb{R}^n \\ x = \sigma_\epsilon(z). \end{cases} \tag{22}$$

Any equilibrium of (22) is characterized by $\overline{x}^\star = \sigma_\epsilon(\overline{z}^\star)$, $U(\overline{x}^\star) = \overline{z}^\star$, or is a fixed-point $\overline{x}^\star = \sigma_\epsilon(U(\overline{x}^\star))$. Any such equilibrium is called a Nash distribution, [6], that is a Nash equilibrium of an ϵ-perturbed game. For small ϵ, $\overline{x}^\star$ approximates the Nash equilibria of the game, [72].

Then, P in (21) and (22) can be represented as a feedback interconnected system (Σ, U) shown in Figure 5 (left) and (right), respectively. On the feedback path, U is the payoff game mapping. On the forward path, Σ is the cascade connection between a bank of integrators (left) or a bank of low-pass filters (right), respectively, and the static map σ_ϵ. The soft-max mapping $\sigma_\epsilon(\cdot)$ is monotone and ϵ-cocoercive (output-strictly EIP) and is the gradient of the lse function [61, Prop. 2]. Based on this, it can be shown that Σ in Figure 5 (left) is EIP, while Σ in Figure 5 (right) is output-strictly EIP (OSEIP), respectively, both with storage function the Bregman divergence of lse, [61, Prop. 3]. Leveraging this output-strictly EIP, (global) asymptotic stability of the closed-loop system for Q-learning (22) in Figure 5 (right) can be shown for any $\epsilon > 0$, in any N-player *monotone* game, that is when the (negative) payoff game mapping $-U$ is EIP (monotone). The standard Rock-Paper-Scissor game belongs to this class of games. On the other hand, for the P-RL system in Figure 5 (left) Σ is only EIP and as a result only stability can be shown when $-U$ is monotone (EIP). We note that passivity techniques can be used to extend Q-learning results to the larger class of *hypomonotone* games, as well as to design higher-order Q-learning dynamics. The main idea is to balance the shortage of passivity on the feedback path in Figure 5 (right) by the $\epsilon-$excess of passivity of Σ on the feedforward path, [61, Thm. 1]. Additionally, generalizations are possible to continuous-kernel games, in the form of mirror-descent dynamics [74] and even to bandit learning and their higher-order variants [75], [76]. The analysis for other instances of such algorithms can be found in [20].

V. Non-Equilibrium Learning in Stochastic Games

Thus far, our discussion has focused on how individual learning algorithms may or may not converge to a stationary point of a game, be it either a NE or a min-max optimal point. At the same time, it has been made clear that there is no universal algorithm guaranteeing that such convergence will be attained; rather, convergence is highly dependent on the corresponding game setting, and is often only local. Convergence to a NE is also further jeopardized when the players of the game are bounded rational, either due to cognitive limitations or because of physical constraints. Hence, we are now motivated to move away from the elusive idea of the NE and study alternative solution concepts, particularly based on bounded rationality theory.

A. Problem Formulation

Consider an N-player (agent) stochastic game, defined by the tuple $(\mathcal{S},\ \mathcal{N},\ \mathcal{A},\ r,\ p,\ \gamma)$, where $\mathcal{S} = \{1, \dots, |\mathcal{S}|\}$ is a finite state space; $\mathcal{N} = \{1, \dots, N\}$ is a set of players; $\mathcal{A} = \mathcal{A}^1 \times \mathcal{A}^2 \dots \times \mathcal{A}^N$ is a joint action space, with $\mathcal{A}^i$ being the individual finite action space of player i, $\forall i \in \mathcal{N}$; $r = \{r^1,\ \dots,\ r^N\}$ is a joint immediate reward function, with $r^i : \mathcal{S} \times \mathcal{A}^i \times \mathcal{A}^{-i} \times \mathcal{S} \to \mathbb{R}^1$ being the individual immediate reward function of each player $i \in \mathcal{N}$; $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,\ 1]$ is the conditional probability transition function, so that $p(s', a^1, \dots, a^N, s)$ is the probability of transitioning from a state $s \in \mathcal{S}$ to a state $s' \in \mathcal{S}$ given actions $a^i \in \mathcal{A}^i$, $\forall i \in \mathcal{N}$; and $\gamma \in (0, 1)$ is a discount factor.

For brevity, we also denote as $p^i(s', a^i, a^{-i}, s)$ the probability of transitioning from a state $s \in \mathcal{S}$ to a state $s' \in \mathcal{S}$ given actions $a^i \in \mathcal{A}^i$ and $a^{-i} \in \mathcal{A}^{-i}$, $\forall i \in \mathcal{N}$. In addition, $\forall i \in \mathcal{N}$, we define as $\mathcal{M}^i,\ \mathcal{M}^{-i}$ the space of the mappings $\mathcal{S} \to \mathcal{A}^i$, $\mathcal{S} \to \mathcal{A}^{-i}$ respectively, and as $\mathcal{J}$ the space of the mappings $\mathcal{S} \to \mathbb{R}$.

Given the stochastic game, each agent $i \in \mathcal{N}$ interacts over time with the environment and the other agents and takes an action $a_t^i \in \mathcal{A}^i$ at every time instant $t \in \mathbb{N}$. Owing to those actions, a sequence $\{s_t\}_{t\in\mathbb{N}}$ of states $s_t \in \mathcal{S}$ will be visited at each time instant $t \in \mathbb{N}$, depending on the conditional transition probabilities given by p and the initial state s_0. In this context, the goal of the player $i \in \mathcal{N}$ is to choose a policy $\pi^i \in \mathcal{M}^i$, i.e., a mapping describing which action $a^i \in \mathcal{A}^i$ is taken at any state $s \in \mathcal{S}$, to maximize their expected discounted cumulative reward, or value, given by

$$J^i_{\pi^i,\pi^{-i}}(s) = \mathbb{E}_p \left[\sum_{t=0}^{\infty} \gamma^t r^i(s_{t+1}, \pi^i(s_t), \pi^{-i}(s_t), s_t) \;\middle|\; s_0 = s \right], \tag{23}$$

where $\pi^{-i} = \{\pi^j\}_{j\in\mathcal{N}\setminus\{i\}} \in \mathcal{M}^{-i}$, and the expected value operator $\mathbb{E}_p$ is taken over the transition probabilities p.

The optimal policy $\pi^{i\star} \in \mathcal{M}^i$ of player i, which maximizes (23), can be obtained from:

$$\pi^{i\star}(s) \in \arg\max_{a^i\in\mathcal{A}^i} \sum_{s'\in\mathcal{S}} p^i\left(s', a^i, \pi^{-i}(s), s\right) \cdot \left(r^i\left(s, a^i, \pi^{-i}(s), s'\right) + \gamma J^i_{\pi^{i\star},\pi^{-i}}(s') \right), \ \forall s \in \mathcal{S},$$

where $J^i_{\pi^{i\star},\pi^{-i}} \in \mathcal{J}$ is the optimal value, which satisfies the Bellman equation

$$J^i_{\pi^{i\star},\pi^{-i}}(s) = \max_{a^i\in\mathcal{A}^i} \sum_{s'\in\mathcal{S}} p^i\left(s', a^i, \pi^{-i}(s), s\right) \cdot \left(r^i\left(s, a^i, \pi^{-i}(s), s'\right) + \gamma J^i_{\pi^{i\star},\pi^{-i}}(s') \right), \ \forall s \in \mathcal{S}. \tag{24}$$

Given (23)-(24), it is evident that the value of agent $i \in \mathcal{N}$ does not depend only on their policy, but also on the other agents' policies π^{-i}. These policies are generally unknown, thus it is not straightforward for agent i to maximize (23); a

[1] We denote $\mathcal{A}^{-i} = \underset{j\in\mathcal{N}\setminus\{i\}}{\times} \mathcal{A}^j$, $\forall i \in \mathcal{N}$.

model of the policies π^{-i} of the players in $\mathcal{N}\backslash\{i\}$ is needed for the maximization to be performed.

A common solution to this problem of lack of knowledge is for each agent $i \in \mathcal{N}$ to assume that all agents in $\mathcal{N}\backslash\{i\}$ will also optimize their own values, and that this assumption is made by all agents. Hence, in this case, finding a policy that maximizes (23) is ultimately equivalent to computing a Nash equilibrium [77].

Definition 1. The tuple $\{\mu^{i\star},\ \mu^{-i\star}\}$, $i \in \mathcal{N}$, with $\mu^{i\star} \in \mathcal{M}^i$ and $\mu^{-i\star} = \{\mu^{j\star}\}_{j\in\mathcal{N}\backslash\{i\}} \in \mathcal{M}^{-i}$, constitutes a Nash equilibrium if $\forall \mu^i \in \mathcal{M}^i$, $\forall i \in \mathcal{N}$, it holds that

$$J^i_{\mu^{i\star},\mu^{-i\star}}(s) \geqslant J^i_{\mu^i,\mu^{-i\star}}(s),\ \forall s \in \mathcal{S}. \quad \square$$

Adopting the approach of the Nash equilibrium to model other agents' behaviors leads to two important issues. First, due to the finite nature of the action set $\mathcal{A}$, finding a Nash equilibrium requires a mostly intractable amount of computations even if all agents share the same reward functions r^i, $\forall i \in \mathcal{N}$ [78]; second, in a realistic scenario, it is not necessary that all agents are perfectly rational[2], hence they may not operate on a Nash equilibrium [21], especially during initial plays of the learning mechanisms. Therefore, instead of relying on the concept of the equilibrium, we will instead seek bounded rationality models to capture other agent behaviors.

B. Recursive Reasoning

Inspired by [21] and [24], we will now model the different levels of rationality for each agent $i \in \mathcal{N}$ participating in the stochastic game, using recursive reasoning.

- ***Level-k Thinking***

Level-k thinking is a model of bounded rationality used to represent a player's strategy, while also relaxing the assumption that every agent seeks a policy that is based solely on the notion of the Nash equilibrium. In particular, level-k thinking defines different levels of rationality, where at each level an agent assumes that the rest of the players follow a policy given by an immediately lower level. Then, the agent proceeds to optimize their cumulative reward (23) given such an assumption. Taking the aforementioned into account, we formulate a level-k thinking model as follows.

Level-0: An agent $i \in \mathcal{N}$ with rationality of level-0, also defined as a level-0 agent, is a player that behaves naively [22], [24]; such an agent neither considers a model of the other agents' behavior, nor tries to maximize their own cumulative reward (23). Hence, the policy $\pi_0^{i\star} \in \mathcal{M}^i$ of a level-0 agent $i \in \mathcal{N}$ can be chosen arbitrarily, so that

$$\pi_0^{i\star}(s) = a^i,\ a^i \in \mathcal{A}^i,\ \forall s \in \mathcal{S}. \tag{25}$$

Apart from choosing it as a constant one, the level-0 policy can also be chosen to be uniformly random [24].

Level-k $\in \mathbb{N}_+$: Unlike a level-0 agent, an agent $i \in \mathcal{N}$ with a rationality of level-k, $k \in \mathbb{N}_+$, reasons about the behavior of the other agents. In particular, for $k \in \mathbb{N}_+$, a level-k agent assumes that the level of rationality of the rest of the agents is $k-1$. Based on this assumption, a level-k agent acts strategically by trying to maximize their expected discounted cumulative reward, and by choosing a level-k policy $\pi_k^{i\star} \in \mathcal{M}^i$ that satisfies, $\forall k \in \mathbb{N}_+$:

$$\pi_k^{i\star} \in \arg\max_{\pi^i\in\mathcal{M}^i} J^i_{\pi^i,\ \pi^{-i\star}_{k-1}}(s),\ \forall s \in \mathcal{S}, \tag{26}$$

where $\pi^{-i\star}_{k-1} = \{\pi^{j\star}_{k-1}\}_{j\in\mathcal{N}\backslash\{i\}} \in \mathcal{M}^{-i}$. Since the action space $\mathcal{A}$ and the state space $\mathcal{S}$ are finite, there exists at least one policy satisfying (26). Hence, for any agent $i \in \mathcal{N}$, it is necessary and sufficient for a level-k policy $\pi_k^{i\star}$ to satisfy

$$J^i_{\pi_k^{i\star},\ \pi^{-i\star}_{k-1}}(s) \geqslant J^i_{\pi^i,\ \pi^{-i\star}_{k-1}}(s),\ \forall s \in \mathcal{S},\ \forall \pi^i \in \mathcal{M}^i.$$

- ***Level-Recursive Computation of Level-k Policies***

We proceed to find the level-k policies described by (26), $\forall k \in \mathbb{N}_+$. To this end, we define the Bellman operator $T^i_{\mu^{-i}}$ and the Q-factor operator $Q^i_{\mu^{-i}}$, $\forall i \in \mathcal{N}$, that map functions of the form $J \in \mathcal{J}$ to functions of the form

$$\left(T^i_{\mu^{-i}}J\right)(s) \triangleq \max_{a^i\in\mathcal{A}^i} \sum_{s'\in\mathcal{S}} p^i\left(s', a^i, \mu^{-i}(s), s\right) \cdot \left(r^i\left(s, a^i, \mu^{-i}(s), s'\right) + \gamma J(s')\right),\ \forall s \in \mathcal{S}, \tag{27}$$

and

$$\left(Q^i_{\mu^{-i}}J\right)(s, a^i) \triangleq \sum_{s'\in\mathcal{S}} p^i\left(s', a^i, \mu^{-i}(s), s\right) \cdot \left(r^i\left(s, a^i, \mu^{-i}(s), s'\right) + \gamma J(s')\right),\ \forall s \in \mathcal{S},\ a^i \in \mathcal{A}^i, \tag{28}$$

for any policies $\mu^{-i} \in \mathcal{M}^{-i}$. Using these operators, a level-recursive procedure for computing the level-k policies can be implemented through Algorithm 1. For a detailed convergence analysis of this Algorithm, see [26].

Algorithm 1 Level-Recursive Computation

Input: Sufficiently small constant $\epsilon > 0$, maximum level $k^\star \in \mathbb{N}_+$.

Output: Level-k policies $\hat{\pi}^i_k = \pi^{i\star}_k$, $\forall i \in \mathcal{N}$, $\forall k \in \{0, \ldots, k^\star\}$.

procedure
  for $i = 1, \ldots, N$ **do** ▷ Initialization
    $\hat{\pi}^i_0(s) \leftarrow \pi^{i\star}_0(s),\ \forall s \in \mathcal{S}$.
    for $k = 1, \ldots, k^\star$ **do**
      Initialize $J^i_k(s)$ randomly, $\forall s \in \mathcal{S}$.
    end for
  end for
  for $k = 1, \ldots, k^\star$ **do** ▷ Level-k policy estimation
    for $i = 1, \ldots, N$ **do**
      repeat
        $v(s) \leftarrow J^i_k(s),\ \forall s \in \mathcal{S}$.
        $J^i_k(s) \leftarrow T^i_{\hat{\pi}^{-i}_{k-1}} v(s),\ \forall s \in \mathcal{S}$.
      until $\left\|J^i_k - v\right\|_\infty < \epsilon$.
      $\hat{\pi}^i_k(s) \leftarrow \arg\max_{a^i\in\mathcal{A}^i} (Q^i_{\hat{\pi}^{-i}_{k-1}} J^i_k)(s, a^i),\ \forall s \in \mathcal{S}$.
    end for
  end for
end procedure

[2]In the sense that not every agent may be able to find the Nash equilibrium; that not every agent assumes that the rest of the agents will optimize their own values; or that not every agent actually seeks a Nash equilibrium.

Algorithm 2 Level-Paralleled Computation

Input: Sufficiently small constant $\epsilon > 0$, maximum level $k^\star \in \mathbb{N}_+$.

Output: Level-k policies $\hat{\pi}_k^i = \pi_k^{i\star},\ \forall i \in \mathcal{N},\ \forall k \in \{0, \dots, k^\star\}$.

1: **procedure**
2: **for** $i = 1, \dots, N$ **do** ▷ Initialization
3: $\hat{\pi}_0^i(s) \leftarrow \pi_0^{i\star}(s),\ \forall s \in \mathcal{S}$.
4: **for** $k = 1, \dots, k^\star$ **do**
5: Initialize $J_k^i(s)$ randomly, $\forall s \in \mathcal{S}$.
6: **end for**
7: **end for**
8: **repeat** ▷ Level-k policy estimation
9: $\Delta \leftarrow 0$.
10: **for** $k = 1, \dots, k^\star$ **do**
11: **for** $i = 1, \dots, N$ **do**
12: $v(s) \leftarrow J_k^i(s),\ \forall s \in \mathcal{S}$.
13: $J_k^i(s) \leftarrow T^i_{\hat{\pi}_{k-1}^{-i}} v(s),\ \forall s \in \mathcal{S}$.
14: $\hat{\pi}_k^i(s) \leftarrow \arg\max_{a^i \in \mathcal{A}^i} (Q^i_{\hat{\pi}_{k-1}^{-i}} J_k^i)(s, a^i),\ \forall s \in \mathcal{S}$.
15: $\Delta \leftarrow \max\left\{\Delta,\ \left\| J_k^i - v \right\|_\infty\right\}$.
16: **end for**
17: **end for**
18: **until** $\Delta < \epsilon$.
19: **end procedure**

- ***Level-Paralleled Computation of Level-k Policies***

The recursion presented in Algorithm 1 has a computational hurdle; for the execution of step $k \in \mathbb{N}_+$ to begin, the previous step $k-1$ needs to have terminated. Hence, one may desire to implement a version of Algorithm 1 that updates the value and policy estimates $J_k^i, \hat{\pi}_k^i$, simultaneously, over all $k \in \mathbb{N}_+$. Such a procedure is described in Algorithm 2.

While Algorithm 2 allows for the parallel estimation of the level-k policies over all the levels $k \in \mathbb{N}$, its convergence – the proof of which can be found in [26] – relies on the following uniqueness assumption, commonly imposed in multi-agent frameworks [78].

Assumption 1. The cost functions of any two distinct policies are distinct, i.e., for any two policies $\mu^i,\ \mu'^i \in \mathcal{M}^i$ and a joint policy $\mu^{-i} \in \mathcal{M}^{-i}$, $i \in \mathcal{N}$, it holds that

$$\mu^i \neq \mu'^i \Longrightarrow J^i_{\mu^i,\ \mu^{-i}} \neq J^i_{\mu'^i,\ \mu^{-i}}. \qquad \square$$

Remark 1. The "for-loop" in lines $10-17$ of Algorithm 2 can be executed in an asynchronous manner (as in [79]), without affecting the convergence of the algorithm. In fact, several iterations of this loop can be omitted at multiple instances of the wider "repeat-while" loop in lines $8-18$. Hence, if the state space $\mathcal{S}$ is large, a significant speed-up can be attained with respect to the level-recursive Algorithm 1. $\square$

Given Assumption 1, Algorithm 2 can be effectively executed, in an asynchronous manner, to offer a computational improvement with respect to Algorithm 1. Nevertheless, if the communication or the memory overhead is high, then the benefits of executing Algorithm 2 asynchronously will be overshadowed; in such cases, Algorithm 1 is preferable.

- ***Cognitive Hierarchy***

According to the level-k thinking model presented previously, a level-k agent assumes that the rest of the agents are level-$(k-1),\ \forall k \in \mathbb{N}_+$. However, such an assumption can be restrictive; if the rest of the agents are at a lower level of rationality, but not exactly at $k-1$, all optimality guarantees are dropped. Therefore, it is of interest to construct a more generalized model of bounded rationality, which will allow for the other agents' levels to vary, and not be deterministically equal to $k-1$. To this end, we construct a bounded rationality model based on *cognitive hierarchy* [21], in order to generalize level-k thinking. According to this model, a level-k agent does not necessarily assume that the rest of the agents are level-$(k-1)$, but that their cognitive level follows a distribution over $\{0, 1, \dots, k-1\}$. If g is a probability mass function over $\mathbb{N}$, then such a distribution $\mathcal{P}_k$ over $\kappa \in \{0, 1, \dots, k-1\}$, $k \in \mathbb{N}_+$, can be defined by the probability mass function:

$$P_k(\kappa) = \frac{g(\kappa)}{\sum_{i=0}^{k-1} g(i)},\ \forall \kappa \in \{0, 1, \dots, k-1\}. \tag{29}$$

It is common to select $g(\cdot)$ to represent a Poisson distribution, since experiments have shown that the proportion of players with a cognitive level of $k-1$ usually decreases as k increases [80]. By adopting the Poisson model, one has

$$g(\kappa) = \frac{\lambda^\kappa e^{-\lambda}}{\kappa!}, \tag{30}$$

where $\lambda > 0$ is the mean and the variance of the model.

Given (29)-(30), the cognitive hierarchy model derives the following policies $\mu_k^{i\star} \in \mathcal{M}^i$ at each level $k \in \mathbb{N}$, $\forall i \in \mathcal{N}$.

Level 0: The level-0 policy in cognitive hierarchy is defined exactly as in the case of level-k thinking. That is,

$$\mu_0^{i\star}(s) = a^i,\ a^i \in \mathcal{A}^i,\ \forall s \in \mathcal{S}.$$

Level $k \in \mathbb{N}_+$: According to the bounded rationality model of cognitive hierarchy, an agent $i \in \mathcal{N}$ of level-k, $k \in \mathbb{N}_+$, assumes that each of the other agents has a level of intelligence κ, given by the distribution (29)-(30). That is, $\kappa \sim \mathcal{P}_k$. Since κ is a random variable, it is in the interest of agent $i \in \mathcal{N}$ to maximize the expectation of their value over $\kappa \sim \mathcal{P}_k$, and pick their policy according to

$$\mu_k^{i\star} \in \arg\max_{\mu^i \in \mathcal{M}^i} \mathbb{E}\left[J^i_{\mu^i,\ \mu_\kappa^{-i\star}}(s) \ \middle|\ \kappa \sim \mathcal{P}_k \right],\ \forall s \in \mathcal{S}, \tag{31}$$

where $\mu_\kappa^{-i\star} = \{\mu_\kappa^{j\star}\}_{j \in \mathcal{N}\setminus\{i\}}$. By slightly modifying the operators (27)-(28), so that they are taken with respect to the expected value of the now random policies of the agents in $\mathcal{N}\setminus\{i\}$, one can extend Algorithms 1 and 2 to solve (31).

C. Limited Communication

So far, only time-triggered policies have been derived. That is, every player updates their action in each time step of the stochastic game, and an infinite amount of communication resources is assumed to be available. As this might be a restrictive assumption, here will present an intermittent version of the bounded rational policies presented previously. Two different approaches will be particularly considered.

- ***Concurrent Estimation***

In the first approach, one can design an intermittent rule that will be incorporated within the models of level-k thinking and cognitive hierarchy. Specifically, at each level k of level-k thinking, an intermittent policy that is the best response to an intermittent level $k-1$ policy is designed. Similarly for cognitive hierarchy, at each level k, an intermittent policy that is the best response to a distribution of lower-level intermittent policies is constructed. In short, when computing their intermittent policies, bounded rational agents of any level consider the fact that the other agents *also* use intermittency. In what follows, we focus on derivations for the level-k thinking model, as the results are similar for the cognitive hierarchy case. Towards designing the aforementioned policies, we define an augmented state space of the form $\tilde{\mathcal{S}} = \mathcal{S} \times \mathcal{A}$ [81]. This set includes the nominal states $s_t \in \mathcal{S}$, along with the actions $a_{t-1} \in \mathcal{A}$ played by each player in the previous time step $t-1 \in \mathbb{N}$, creating a pair $x_t = (s_t,\ a_{t-1}) \in \tilde{\mathcal{S}}$. The tuple of the stochastic game is also redefined into $(\tilde{\mathcal{S}},\ \mathcal{N},\ \mathcal{A},\ \tilde{r},\ \tilde{p},\ \gamma)$, so that the reward $\tilde{r}$ and the transition probabilities are defined over $\tilde{\mathcal{S}}$. We additionally define as $\tilde{\mathcal{M}}^i$ the set of policies of agent $i \in \mathcal{N}$ over the space $\tilde{\mathcal{S}}$.

To optimize communication resources concurrently with the rewards r^i, we construct the augmented reward $\tilde{r}^i$ for agent $i \in \mathcal{N}$ by adjoining a communication reward to r^i:

$$\tilde{r}^i(x_{t+1}, a_t^i, \tilde{\pi}^{-i}(x_t), x_t) = \rho^i r^i(s_{t+1}, a_t^i, \tilde{\pi}^{-i}(x_t), s_t) + (1-\rho^i)\mathbb{1}_{a_t^i = a_{t-1}^i}, \quad (32)$$

where $x_{t+1},\ x_t \in \tilde{\mathcal{S}}$ are the augmented states at time $t+1$ and t, $a_t^i \in \mathcal{A}^i$ is the action of player i at time t, $\tilde{\pi}^{-i} \in \tilde{\mathcal{M}}^{-i}$ are joint policies of players in $\mathcal{N}\backslash\{i\}$, and $\rho^i \in [0,\ 1]$. Evidently, the reward function (32) is a convex combination of the original reward r^i as well as the indicator function $\mathbb{1}_{a_t^i = a_{t-1}^i}$. The latter term forces constant policies to be more favorable, which subsequently reduces the communication burden of player i. The constant ρ^i is a measure of the communication capabilities of player i; if $\rho^i = 1$, then player i has infinite communication resources and is not penalized for updating their action at each time step; whereas if $\rho^i = 0$, then player i has zero bandwidth and will be rewarded only if they do not update their action. In a realistic scenario, ρ^i will take a value between zero and unity.

Over the new state space sequence $\{x_t\}_{t\in\mathbb{N}}$, and considering the reward function (32) that is augmented with a communication penalty, the cumulative reward is:

$$\tilde{J}^i_{\tilde{\pi}^i, \tilde{\pi}^{-i}}(x) = \mathbb{E}_{\tilde{p}}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}^i(x_{t+1}, \tilde{\pi}^i(x_t), \tilde{\pi}^{-i}(x_t), x_t) \,\middle|\, x_0 = x\right],$$

where $x_t \in \tilde{\mathcal{S}}$. Hence, following the reasoning of Section V-B and for all $k \in \mathbb{N}_+$, the level-k policy of agent $i \in \mathcal{N}$ with incorporated intermittency can be obtained as:

$$\tilde{\pi}_k^{i\star} \in \arg\max_{\tilde{\pi}^i \in \tilde{\mathcal{M}}^i} \tilde{J}^i_{\tilde{\pi}^i,\ \tilde{\pi}^{-i\star}_{k-1}}(x),\ \forall x \in \tilde{\mathcal{S}}. \quad (33)$$

Notice that, unlike (26), in (33) the value function is defined over the summation of the rewards (32), which are augmented with a communication reward. Accordingly, the base level-0 policy of the time-triggered case (25) can be generalized to the intermittent case, so that for every agent $i \in \mathcal{N}$ it will be defined as:

$$\tilde{\pi}_0^{i\star}(x_t) = \tilde{\pi}_0^{i\star}(s_t,\ a_{t-1}) = a_{t-1}^i,\ a_{t-1}^i \in \mathcal{A}^i,\ \forall s_t \in \mathcal{S}. \quad (34)$$

In essence, a level-0 agent's policy is constant, as they will always play the action used in a previous time step. Notice that, as (33)-(34) are of the same form as (25)-(26), Algorithms 1 and 2 can be effectively utilized to compute the event triggered level-k policies $\tilde{\pi}_k^{i\star}$, $\forall i \in \mathcal{N}$, $k \in \mathbb{N}$.

While equations (33)-(34) describe an effective way of optimally incorporating communication constraints within the level-k thinking model, they suffer from a drawback; information regarding the actual best-response policies is diluted by the intermittency as the levels of intelligence increase, and depending on the value of ρ^i, $i \in \mathcal{N}$. This is because intermittency "quantizes" a player's policy, making it piece-wise constant to save communication resources.

- ***Sequential Estimation***

In the second approach, one can design the intermittency rule *a posteriori*, so that it is distinct from the models of bounded rationality. In particular, the level-k policies are initially derived as in Section V-B, and they are subsequently "quantized" in order to obtain their intermittent version. As a result, although each level-k agent can follow an intermittency scheme a posteriori, they do not assume that the lower-level agents do so as well. This approach can solve the problem of information dilution described previously, but it has a different drawback; it leads to overall suboptimality, as the decision of whether to communicate or not is taken *after* the level-k policies have been designed.

Given the nominal level-k policies (25)-(26) for player $i \in \mathcal{N}$, a communication aware policy $\hat{\pi}_k^i$ at the level $k \in \mathbb{N}_+$ can be designed in an a posteriori sense, so that:

$$\hat{\pi}_k^i(x_k) = \begin{cases} a_{t-1}^i & \gamma_k^i(x_k) = 0 \\ \pi_k^{i\star}(s_k) & \gamma_k^i(x_k) = 1. \end{cases} \quad (35)$$

In (35), $\gamma_k^i \in \Gamma \triangleq \tilde{\mathcal{S}} \to \{0, 1\}$ is an event that indicates whether communication will take place or not, and which should be optimized. In that sense, its optimal value $\gamma_k^{i\star}$ should be such that

$$\gamma_k^{i\star} \in \arg\max_{\gamma_k^i \in \Gamma} \hat{J}^i_{\hat{\pi}_k^i,\ \pi_{k-1}^{-i\star}}(x),\ \forall x \in \tilde{\mathcal{S}}, \quad (36)$$

where

$$\hat{J}^i_{\hat{\pi}_k^i, \pi_{k-1}^{-i\star}}(x) = \mathbb{E}_{\tilde{p}}\left[\sum_{t=0}^{\infty} \gamma^t \tilde{r}^i(x_{t+1}, \hat{\pi}_k^i(x_t), \pi_{k-1}^{-i\star}(s_t), x_t) \,\middle|\, x_0 = x\right].$$

Hence, given (35) and (36), the optimal intermittent policy, designed in a sequential, a posteriori manner, is given by

$$\hat{\pi}_k^{i\star}(x_k) = \begin{cases} a_{t-1}^i & \gamma_k^{i\star}(x_k) = 0 \\ \pi_k^{i\star}(s_k) & \gamma_k^{i\star}(x_k) = 1. \end{cases}$$

Remark 2. The optimization (36) can be performed using standard policy or value iteration techniques. □

VI. Potential Games in Autonomous Driving

As already pointed out, solution seeking in game-theoretic approaches is generally complex, hence leading to scalability issues for real-time operations in autonomous systems. In addition, the possible lack of information and the bounded rationality of other agents raise concerns regarding the system performance, especially for safety-critical systems like autonomous driving. In this section, we will introduce a predictor-corrector game structure to address these challenges and see how to make game-theoretic approaches more practical and reliable to autonomous driving applications.

A. Problem Formulation

Consider a set of traffic agents $\mathcal{N}$ represented by the following discrete time models: $x_i(t+1) = f_i(x_i(t), a_i(t))$, where $x_i(t) \in \mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ and $a_i(t) \in \mathcal{U}_i \subseteq \mathbb{R}^{m_i}$ are, respectively, the state and action of agent i at the time step t, $i \in \mathcal{N}$, and f_i is the system evolution model. Denote the state of all other agents except for agent i as x_{-i}, i.e., $x_{-i} = \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N\}$. Similarly, let $a_{-i} = \{a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_N\}$ and $f_{-i} = \{f_1, \ldots, f_{i-1}, f_{i+1}, \ldots, f_N\}$. Denote the dimension of a_{-i} as m_{-i}, i.e., $a_{-i}(t) \in \mathbb{R}^{m_{-i}}$. Denote the global state, action, and dynamics as $x = \{x_i, x_{-i}\}$, $a = \{a_i, a_{-i}\}$, and $f = \{f_i, f_{-i}\}$, respectively.

In a driving scenario, every traffic agent has its own driving objective, e.g., tracking a desired speed without collisions and with ride comfort. We use the cost function $J_i : \mathcal{A} \to \mathbb{R}$ to characterize agent i's objective, where $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_N$ with $\mathcal{A}_i$ representing the strategy space of agent i. We notice that human drivers' maneuvers are generally motivated by a foreseen gain or loss within a finite prediction horizon, and therefore, their decision-making process can be modeled as a receding horizon optimal control process. That is, at each t, agent i aims to find its optimal action sequence (or called strategy) $\mathbf{a}_i^\star(t) \in \mathcal{A}_i$ such that

$$\begin{aligned}\mathbf{a}_i^\star(t) &\in \arg\min_{\mathbf{a}_i(t)\in\mathcal{A}_i} J_i(\mathbf{a}_i(t), \mathbf{a}_{-i}(t)) \\ &= \arg\min_{\mathbf{a}_i(t)\in\mathcal{A}_i} \sum_{\tau=t}^{t+T-1} \Psi_i(x_i(\tau), x_{-i}(\tau), a_i(\tau), a_{-i}(\tau)),\end{aligned} \tag{37}$$

where $\mathbf{a}_i(t) = \{a_i(t), a_i(t+1), \ldots, a_i(t+T-1)\} \in \mathcal{A}_i$, $\mathcal{A}_i$ is determined by $\mathcal{U}_i$, $\mathbf{a}_{-i} = \{\mathbf{a}_1, \ldots, \mathbf{a}_{i-1}, \mathbf{a}_{i+1}, \ldots, \mathbf{a}_N\} \in \mathcal{A}_{-i}$ is the set of all other agents' strategies except for agent i, Ψ_i is the instantaneous cost at each time instant, $T \in \mathbb{Z}_{++}$ is the horizon length. After deriving $\mathbf{a}_i^\star(t)$, agent i implements the first element $a_i^\star(t)$ and repeats the same procedure at the next time instant, $t+1$, to timely respond to any change in the environment.

As we can see from (37), agent i's cost J_i depends on not only $\mathbf{a}_i$ but also $\mathbf{a}_{-i}$, indicating that agent i's driving performance is jointly affected by both its own and the surrounding agents' behaviors, characterizing agents' interactions. At each t, if every agent i aims to optimize its own performance J_i, then the multi-agent optimization (37) becomes a multi-player game, and the set of all agents' optimal strategies $\{\mathbf{a}_1^\star(t), \mathbf{a}_2^\star(t), \cdots, \mathbf{a}_N^\star(t)\}$, if nonempty, composes a pure-strategy Nash equilibrium (PSNE).

Although the receding horizon multi-player game formulation (37) is consistent with human driver decision making, it is generally difficult to solve, and the difficulties include: a) ***PSNE existence:*** Given arbitrary J_i, a PSNE that solves (37) may not exist; b) ***PSNE convergence:*** Even if a PSNE exists, a solution seeking algorithm, e.g., best response (BR) dynamics, may not necessarily converge; c) ***Computational scalability:*** To solve (37), multiple and iterative optimizations are generally required at each t, causing significant computational burden; and d) ***Lack of global situation awareness:*** To find PSNE, the ego vehicle needs to know all agents' cost functions, which may not be realistic in a traffic setting.

B. Predictor-Corrector Potential Game

To address the above challenges and to solve (37) in real time, we develop a predictor-corrector potential game (PCPG) approach, featuring a potential game (PG) based Predictor and a best-response based Corrector.

- ***PG Predictor***

In the Predictor, the ego vehicle (denoted as agent i) assumes that the surrounding agents' behaviors are governed by a pre-determined typical cost function $\hat{J}_j : \mathcal{A} \to \mathbb{R}$. Such a $\hat{J}_j$ can be learned offline via inverse RL or imitation learning from realistic traffic datasets by the ego vehicle and does not necessarily equal J_j, which is agent j's actual cost function. Mathematically, the ego vehicle solves the following N-player game at each t.

$$\mathbf{a}_j^+(t) \in \arg\min_{\mathbf{a}_j(t)\in\mathcal{A}_j} \hat{J}_j(\mathbf{a}_j(t), \mathbf{a}_{-j}(t)) \tag{38}$$

where $j \in \mathcal{N}$, and $\hat{J}_i = J_i$.

Assumption 2. $\hat{J}_j$ is everywhere differentiable on an open super-set of $\mathcal{A}$, $\forall j \in \mathcal{N}$, $\mathcal{A}_i \neq \varnothing$, $\forall i \in \mathcal{N}$, and $\mathcal{A}$ is a connected and compact set. □

To address the challenges caused by game complexity, we formulate the game (38) as a continuous exact potential game, which is defined as follows.

Definition 2 (Continuous Exact Potential Game [82])**.** Under Assumption 2, The game (38) is a continuous exact potential game if and only if there exists a function $F : \mathcal{A} \to \mathbb{R}$ such that F is everywhere differentiable on an open super-set of $\mathcal{A}$, and $\frac{\partial \hat{J}_j(\mathbf{a}_j, \mathbf{a}_{-j})}{\partial \mathbf{a}_j} = \frac{\partial F(\mathbf{a}_j, \mathbf{a}_{-j})}{\partial \mathbf{a}_j}$ holds $\forall \mathbf{a}_j \in \mathcal{A}_j$, $\forall \mathbf{a}_{-j} \in \mathcal{A}_{-j}$, and $\forall j \in \mathcal{N}$. The function F is called the potential function. □

In this section, the term PG always refers to the continuous exact PG, and the word "continuous exact" may be omitted when no confusion is caused. A PG has many appealing properties, including guaranteed PSNE existence and best-response dynamics convergence. We refer the readers to [82] for a detailed discussion. The following Theorem shows how to make the game (38) a PG.

Theorem 4 (Theorem 6 in [82]). *Let the cost function $\hat{J}_j$ in* (38) *satisfy*

$$\begin{aligned}&\hat{J}_j(\mathbf{a}_j(t),\mathbf{a}_{-j}(t))\\&=\alpha J_j^{\text{self}}(\mathbf{a}_j(t))+\beta\sum_{k\in\mathcal{N},k\neq j}J_{jk}(\mathbf{a}_j(t),\mathbf{a}_k(t)),\end{aligned}\tag{39}$$

where $J_j^{\text{self}}:\mathcal{A}_j\to\mathbb{R}$ is a function determined solely by agent j's action, $J_{jk}:\mathcal{A}_j\times\mathcal{A}_k\to\mathbb{R}$ satisfies

$$\begin{aligned}&J_{jk}(\mathbf{a}_j(t),\mathbf{a}_k(t))=J_{kj}(\mathbf{a}_k(t),\mathbf{a}_j(t)),\\&\forall j,k\in\mathcal{N},j\neq k,\ \textit{and}\ \forall\mathbf{a}_j\in\mathcal{A}_j,\mathbf{a}_k\in\mathcal{A}_k,\end{aligned}$$

and α and β are two real numbers. Then the game (38) *is a PG with the following potential function,*

$$\begin{aligned}F(\mathbf{a}(t))=&\alpha\sum_{j\in\mathcal{N}}J_j^{\text{self}}(\mathbf{a}_j(t))\\&+\beta\sum_{j\in\mathcal{N}}\sum_{k\in\mathcal{N},k<j}J_{jk}(\mathbf{a}_j(t),\mathbf{a}_k(t)).\end{aligned}\tag{40}$$

□

Theorem 4 states that if $\hat{J}_j$ is designed to be a linear combination of two components: J_j^{self} and $\sum_{k\in\mathcal{N},k\neq j}J_{jk}$, then the resulting game is a PG. Such a cost function design can meet the autonomous driving application needs in general, as the first term J_j^{self} can be used to model self-focused objectives such as tracking a desired speed, minimizing fuel consumption, and maintaining ride comfort, and the second term $\sum_{k\in\mathcal{N},k\neq j}J_{jk}$ can be used to capture symmetric pairwise agent interactions such as a pairwise collision penalty. Please see [2], [3] for examples where AV cost function design follows, or can be slightly revised to follow, the form (39). After formulating the game as a PG, the PSNE seeking problem, which generally requires multiple and iterative optimizations, can be solved by one simple optimization [82], addressing the computational scalability challenge. Specifically, the PSNE $\mathbf{a}^+(t)=\{\mathbf{a}_1^+(t),\mathbf{a}_2^+(t),\cdots,\mathbf{a}_N^+(t)\}$ of the game (38) can be found by $\mathbf{a}^+(t)\in\arg\min_{\mathbf{a}(t)\in\mathcal{A}}F(\mathbf{a}(t))$, where F is determined by (40).

- ***PG Corrector***

The Predictor finds the ego vehicle optimal strategy by assuming pre-determined others' cost functions $\hat{J}_j$. We acknowledge that $\hat{J}_j$ may not equal J_j, which may lead to $\mathbf{a}_{-i}^+(t)\neq\mathbf{a}_{-i}^\star(t)$, where $\mathbf{a}_{-i}^\star(t)=\{a_{-i}^\star(t),a_{-i}^\star(t+1),\cdots,a_{-i}^\star(t+T-1)\}$ is the surrounding agents' actual strategies at t. To account for such a discrepancy, we design a best response-based Corrector.

Define action deviation at t as

$$\omega_{-i}(t)=a_{-i}^\star(t)-a_{-i}^+(t).\tag{41}$$

In the Corrector, the ego vehicle aims to find its best response to a corrected prediction on the surrounding agents' strategies, i.e.,

$$\mathbf{a}_i^\star(t)\in\arg\min_{\mathbf{a}_i(t)\in\mathcal{A}_i}J_i(\mathbf{a}_i(t),\hat{\mathbf{a}}_{-i}^\star(t)),\tag{42}$$

with $\hat{\mathbf{a}}_{-i}^\star(t)=\mathbf{a}_{-i}^+(t)+\hat{\boldsymbol{\omega}}_{-i}(t)$; $\hat{\boldsymbol{\omega}}_{-i}(t)=\mathbf{1}_T\otimes\omega_{-i}(t-1)$, where $\otimes$ represents the Kronecker product, and $\mathbf{1}_T$ is a vector of ones with T elements. Note that J_i in (42) is the cost function of the ego vehicle and hence is supposed to be known. The corrected prediction $\hat{\mathbf{a}}_{-i}^\star(t)$ in (42) takes into account both the prediction from the PG, $\mathbf{a}_{-i}^+(t)$, and the observed action deviation at $t-1$, $\omega_{-i}(t-1)$. By assuming constant deviations over $[t-1,t+T-1]$, we are considering consistent driving styles of the surrounding agents: If agent j performs more aggressively than a typical agent at $t-1$, then a similar aggressiveness level should be expected over the horizon $[t,t+T-1]$. We define the prediction error $e(\tau)$, $\tau\in[t,t+T-1]$, $\tau\in\mathbb{Z}_+$, as the difference between the predicted other agents' actions from (42) and their actual actions, i.e., $e(\tau)=a_{-i}^\star(\tau)-\hat{a}_{-i}^\star(\tau)$.

The following theorem shows that with the PCPG, the prediction error $e(\tau)$ admits a bound.

Theorem 5. *[Bounded prediction error [27]] Assume that $a_{-i}^\star:\mathbb{Z}_+\to\mathbb{R}^{m_{-i}}$ and $a_{-i}^+:\mathbb{Z}_+\to\mathbb{R}^{m_{-i}}$ are Lipschitz continuous functions with*

$$\begin{aligned}&\|a_{-i}^\star(\tau)-a_{-i}^\star(t-1)\|\leqslant K_1\cdot(\tau-t+1)\cdot\Delta t,\\&\|a_{-i}^+(\tau)-a_{-i}^+(t-1)\|\leqslant K_2\cdot(\tau-t+1)\cdot\Delta t,\end{aligned}$$

where $K_1\in\mathbb{R}_+$ and $K_2\in\mathbb{R}_+$ are two constants, and $\Delta t\in\mathbb{R}_{++}$ is the sampling time. Then $\|e(\tau)\|$ is bounded by

$$\|e(\tau)\|\leqslant(K_1+K_2)\cdot(\tau-t+1)\cdot\Delta t,\tag{43}$$

$\forall\tau\in[t,t+T-1]$, $\tau\in\mathbb{Z}_+$. □

Theorem 5 shows that the prediction error $e(\tau)$ remains bounded over a finite prediction horizon, and the bound depends on the sampling time Δt, horizon length T, and the constants K_1 and K_2. In autonomous driving, the value of K_1 can be determined from acceleration, jerk, and angular acceleration limits of the individual vehicles, and the value of K_2 can be determined from offline closed-loop simulations of the multi-agent system operating according to the PG (38) assessed at the worst case with respect to variations in parameters. Given the bound (43), we denote

$$\mathcal{E}(\tau)=\{\hat{e}\in\mathbb{R}^{m_{-i}}|\|\hat{e}\|\leqslant C\cdot(t+\tau-1)\cdot\Delta t\},\tag{44}$$

as the set of all possible prediction errors at $\tau\in[t,t+T-1]$.

C. Performance Analysis

With the PCPG designed above, the ego vehicle safety can be guaranteed under certain conditions, despite the unknown cost functions of others.

Define a safe set $\mathcal{X}_i^{\text{safe}}(x_{-i}(t))$ as the set of all $x_i(t)$ such that if $x_i(t)\in\mathcal{X}_i^{\text{safe}}(x_{-i}(t))$ for a given $x_{-i}(t)$, then the ego vehicle is considered safe at t. An example of such a safe set, if x represents vehicle position, is $\mathcal{X}_i^{\text{safe}}(x_{-i}(t))=\{x_i(t)|\|x_i(t)-x_j(t)\|\geqslant d_{\text{safe}},\forall j\in\mathcal{N}_{-i}\}$, where $d_{\text{safe}}>0$ is a predefined safe distance. Denote $\hat{\mathcal{X}}_{-i}(\tau|t)$, $\tau\in[t+1,t+T]$, as the set of $\hat{x}_{-i}(\tau|t)$ generated by the set of action sequences $\{\hat{a}_{-i}^\star(t)+\hat{e}(t),\cdots,\hat{a}_{-i}^\star(\tau-1)+\hat{e}(\tau-1)\}$, where $\hat{e}(k)\in\mathcal{E}(k),k\in[t,\tau-1]$, and $\mathcal{E}(k)$ is defined in (44), i.e.,

$$\begin{aligned}&\hat{\mathcal{X}}_{-i}(\tau|t)=\{\hat{x}_{-i}(\tau|t)|\hat{x}_{-i}(\tau|t)=\\&f_{-i}\left(x_{-i}(t),\{\hat{a}_{-i}^\star(t)+\hat{e}(t),\cdots,\hat{a}_{-i}^\star(\tau-1)+\hat{e}(\tau-1)\}\right)\\&|\hat{e}(k)\in\mathcal{E}(k),k\in[t,\tau-1]\},\end{aligned}$$

where
$f_{-i}\left(x_{-i}(t), \{\hat{a}^\star_{-i}(t)+\hat{e}(t), \cdots, \hat{a}^\star_{-i}(\tau-1)+\hat{e}(\tau-1)\}\right)$, with a slight abuse of notation, represents the surrounding agents' state at τ if the action sequence $\{\hat{a}^\star_{-i}(t)+\hat{e}(t), \cdots, \hat{a}^\star_{-i}(\tau-1)+\hat{e}(\tau-1)\}$ is implemented.

Denote $\mathcal{A}_i^{\text{safe}}(t,T) \subseteq \mathcal{A}_i$ as the set of $\mathbf{a}_i(t)$ such that $\forall \tau \in [t+1, t+T]$,

$$x_i(\tau) = f_i\left(x_i(t), \{a_i(t), \cdots, a_i(\tau-1)\}\right) \in \mathcal{X}_i^{\text{safe}}(\hat{x}_{-i}(\tau|t))$$

holds $\forall \hat{x}_{-i}(\tau|t) \in \hat{\mathcal{X}}_{-i}(\tau|t)$.

Next, we summarize the PCPG safety performance.

Theorem 6 (Safety [27])**.** *If the ego vehicle cost function J_i is designed such that*

$$\arg\min_{\mathbf{a}_i(t)\in\mathcal{A}_i} J_i(\mathbf{a}_i(t), \hat{\mathbf{a}}^\star_{-i}(t)) \subseteq \mathcal{A}_i^{\text{safe}}(t,T), \tag{45}$$

and

$$\mathcal{A}_i^{\text{safe}}(t,T) \neq \varnothing, \tag{46}$$

then the PCPG guarantees the ego vehicle safety within the horizon $[t+1, t+T]$, i.e., $\mathbf{a}_i^\star(t)$ from (42) *leads to $x_i(\tau) \in \mathcal{X}_i^{\text{safe}}(x^\star_{-i}(\tau)), \forall \tau \in [t+1, t+T]$, where $x^\star_{-i}(\tau)$ denotes the surrounding agents' actual state at τ.* □

The above theorem states that PCPG guarantees the ego vehicle safety under two conditions: 1) The ego vehicle is safety-conscious, i.e., (45) holds; and 2) A safe strategy exists, i.e., (46) holds. A safety-conscious ego vehicle requires that if $\mathcal{A}_i^{\text{safe}}(t,T) \neq \varnothing$, then a global minimizer of J_i should belong to $\mathcal{A}_i^{\text{safe}}(t,T)$. To design such a cost function, one may consider incorporating the safety constraint as a barrier in J_i. Consider the interior-point method [83] as an example.

Next, we consider the optimality of $\mathbf{a}_i^\star(t)$. With a slight abuse of notation, we denote $\left(\mathbf{a}_i^\star(t), \mathbf{a}^\star_{-i}(t)\right)$ (resp., $\left(\mathbf{a}_i^\star(t), \hat{\mathbf{a}}^\star_{-i}(t)\right)$) as the strategy profile that agents $-i$ take $\mathbf{a}^\star_{-i}(t)$ (resp., $\hat{\mathbf{a}}^\star_{-i}(t)$) and the ego vehicle takes $\mathbf{a}_i^\star(t) \in \arg\min_{\mathbf{a}_i(t)\in\mathcal{A}_i} J_i(\mathbf{a}_i(t), \mathbf{a}^\star_{-i}(t))$ (resp., $\mathbf{a}_i^\star(t) \in \arg\min_{\mathbf{a}_i(t)\in\mathcal{A}_i} J_i(\mathbf{a}_i(t), \hat{\mathbf{a}}^\star_{-i}(t))$).

Theorem 7 (Optimality [27])**.** *Consider the PCPG designed in Section VI-B and the action deviation in* (41)*. If $\omega_{-i}(t)$ varies slowly with time, and $\omega_{-i}(t) - \omega_{-i}(t-1) \to \mathbf{0}$, where $\mathbf{0}$ is a vector of zeros with proper dimensions, then $\left(\mathbf{a}_i^\star(t), \hat{\mathbf{a}}^\star_{-i}(t)\right) \to \left(\mathbf{a}_i^\star(t), \mathbf{a}^\star_{-i}(t)\right)$.* □

The above theorem shows that if the action deviation function $\omega_{-i}(t)$ varies slowly with time, then the outcome from the PCPG approximates the actual PSNE of the system nicely, despite the unknown cost functions of others. Note that although a slowly time-varying $\omega_{-i}(t)$ is desirable from the optimality point of view, it is not required in the safety guarantee. We refer the readers to [27] for more details and for performance in specific traffic scenarios.

REFERENCES

[1] Y. Shoham and K. Leyton-Brown, *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[2] C. Hubmann, M. Becker, D. Althoff, D. Lenz, and C. Stiller, "Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles," in *IEEE Intelligent Vehicles Symposium*, 2017, pp. 1671–1678.

[3] R. Tian, N. Li, I. Kolmanovsky, Y. Yildiz, and A. R. Girard, "Game-theoretic modeling of traffic in unsignalized intersection network for autonomous vehicle control verification and validation," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[4] M. Liu, Y. Wan, F. Lewis, S. Nageshrao, and D. Filev, "A three-level game-theoretic decision-making framework for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[5] I. Erev and A. E. Roth, "Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria," *American Economic Review*, vol. 88, pp. 848–881, 1998.

[6] D. Leslie and E. Collins, "Individual Q-learning in normal form games," *SIAM J. Control and Optimiz.*, vol. 44, no. 2, pp. 495–514, 2005.

[7] T. Başar and G. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed., ser. Classics in Applied Mathematics. SIAM, 1999.

[8] L. Blume, "The statistical mechanics of strategic interaction," *Games and Economic Behavior*, vol. 5, pp. 387–424, 1993.

[9] J. R. Marden, H. P. Young, G. Arslan, and J. S. Shamma, "Payoff-based dynamics for multiplayer weakly acyclic games," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 373–396, 2009.

[10] J. R. Marden, G. Arslan, and J. S. Shamma, "Joint strategy fictitious play with inertia for potential games," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 208–220, 2009.

[11] D. Fudenberg and D. K. Levine, *The theory of learning in games*. MIT press, 1998, vol. 2.

[12] J. Hofbauer and W. Sandholm, "On the global convergence of stochastic fictitious play," *Econometrica*, vol. **70**, pp. 2265–2294, 2002.

[13] J. Shamma and G. Arslan, "Unified convergence proofs of continuous-time fictitious play," *IEEE Transactions on Automatic Control*, vol. **49**, no. 7, pp. 1137–1141, 2004.

[14] M. Benaim and M. Hirsch, "Mixed equilibria and dynamical systems arising from fictitious play in perturbed games," *Games and Economic Behavior*, vol. **29**, pp. 36–72, 1999.

[15] M. Benaim, J. Hofbauer, and S. Sorin, "Stochastic approximations and differential inclusions," *SIAM Journal of Control and Optimization*, vol. 44, no. 1, pp. 323–348, 2005.

[16] ——, "Stochastic approximations and differential inclusions, part II: Applications," *Mathematics of Operations Research*, vol. 31, no. 4, pp. 673–695, 2006.

[17] R. Chinchilla, G. Yang, and J. P. Hespanha, "Newton and interior-point methods for (constrained) nonconvex-nonconcave minmax optimization with stability guarantees: Technical report," University of California, Santa Barbara, Santa Barbara, Tech. Rep., May 2022.

[18] D. Monderer and L. Shapley, "Potential games," *Games and Economic Behavior*, vol. 14, pp. 124–143, 1996.

[19] H. P. Young, *Individual Strategy and Social Structure*. Princeton, NJ: Princeton University Press, 1998.

[20] L. Pavel, "Dissipativity Theory in Game Theory: On the role of dissipativity and passivity in NE-seeking," *IEEE Control Systems Magazine, Special Issue: 50 years of dissipativity theory - Part II*, vol. 42, no. 3, pp. 150–164, 2022.

[21] C. F. Camerer, T.-H. Ho, and J.-K. Chong, "A cognitive hierarchy model of games," *The Quarterly Journal of Economics*, vol. 119, no. 3, pp. 861–898, 2004.

[22] C. F. Camerer, "Behavioral game theory: Plausible formal models that predict accurately," *Behavioral and Brain Sciences*, vol. 26, no. 02, pp. 157–158, 2003.

[23] ——, "Behavioral game theory: Psychological limits on strategic cognition," *International Journal of Psychology*, vol. 51, p. 11, 2016.

[24] T. Strzalecki, "Depth of reasoning and higher order beliefs," *Journal of Economic Behavior & Organization*, vol. 108, pp. 108–122, 2014.

[25] F. Fotiadis and K. G. Vamvoudakis, "Recursive reasoning for bounded rationality in multi-agent non-equilibrium play learning systems," in *2021 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2021, pp. 741–746.

[26] ——, "Recursive reasoning with reduced complexity and intermittency for nonequilibrium learning in stochastic games," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[27] M. Liu, H. E. Tseng, D. Filev, A. Girard, and I. Kolmanovsky, "Safe and human-like autonomous driving: A predictor-corrector potential game approach," *arXiv preprint arXiv:2208.02835*, 2022.

[28] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," *arXiv:1706.06083 [cs, stat]*, Sep. 2019, arXiv: 1706.06083.

[29] D. A. Copp and J. P. Hespanha, "Simultaneous nonlinear model predictive control and state estimation," *Automatica*, vol. 77, pp. 143–154, Mar. 2017.

[30] R. Chinchilla and J. P. Hespanha, "Stochastic programming using expected value bounds," *IEEE Transactions on Automatic Control*, June 2022.

[31] J. v. Neumann, "Zur theorie der gesellschaftsspiele," *Mathematische annalen*, vol. 100, no. 1, pp. 295–320, 1928.

[32] R. J. Vanderbei and D. F. Shanno, "An interior-point algorithm for nonconvex nonlinear programming," *Computational Optimization and Applications*, vol. 13, no. 1, pp. 231–252, 1999.

[33] C. Jin, P. Netrapalli, and M. I. Jordan, "What is Local Optimality in Nonconvex-Nonconcave Minimax Optimization?" *arXiv:1902.00618 [cs, math, stat]*, Feb. 2019, arXiv: 1902.00618.

[34] Y.-H. Dai and L. Zhang, "Optimality Conditions for Constrained Minimax Optimization," *CSIAM Transactions on Applied Mathematics*, vol. 1, no. 2, pp. 296–315, Jun. 2020, arXiv: 2004.09730.

[35] J. R. Marden and J. S. Shamma, "Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation," *Games and Economic Behavior*, vol. 75, no. 2, pp. 788–808, 2012.

[36] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, USA: MIT Press, 1991.

[37] D. Fudenberg and D. Levine, "Consistency and cautious fictitious play," *Journal of Economic Dynamics and Control*, vol. **19**, pp. 1065–1089, 1995.

[38] U. Berger, "Fictitious play in $2\times$ n games," *Journal of Economic Theory*, vol. 120, no. 2, pp. 139–154, 2005.

[39] S. D. Flåm, "Equilibrium, evolutionary stability and gradient dynamics," *International Game Theory Review*, vol. 4, no. 04, pp. 357–370, 2002.

[40] L. Pavel, "Nash equilibrium seeking over networks," in *Encyclopedia of Systems and Control*. Springer, 2021, pp. 1411–1418.

[41] S. Singh, M. J. Kearns, and Y. Mansour, "Nash convergence of gradient dynamics in general-sum games." in *UAI*. Citeseer, 2000, pp. 541–548.

[42] J. S. Shamma and G. Arslan, "Dynamic fictitious play, dynamic gradient play, and distributed convergence to Nash equilibria," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 312–327, March 2005.

[43] E. Mazumdar, L. J. Ratliff, and S. S. Sastry, "On gradient-based learning in continuous games," *SIAM Journal on Mathematics of Data Science*, vol. 2, no. 1, pp. 103–131, 2020.

[44] T. Tatarenko, W. Shi, and A. Nedić, "Geometric convergence of gradient play algorithms for distributed nash equilibrium seeking," *IEEE Transactions on Automatic Control*, vol. 66, no. 11, pp. 5342–5353, 2021.

[45] M. Benaim and M. Hirsch, "A dynamical systems approach to stochastic approximation," *SIAM Journal on Control and Optimization*, vol. **34**, pp. 437–472, 1996.

[46] V. Borkar, *Stochastic Approximation A Dynamical Systems Viewpoint*. Hindustan Book Agency (India), 2008.

[47] W. Sandholm, *Population Games and Evolutionary Dynamics*, ser. Economic Learning and Social Evolution. Cambridge, USA: MIT Press, 2010.

[48] M. J. Fox and J. S. Shamma, "Population games, stable games, and passivity," *Games*, vol. 4, no. 4, pp. 561–583, 2013.

[49] S. Park, N. Martins, and J. Shamma, "From population games to payoff dynamics models: A passivity-based approach," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, Dec. 2019.

[50] L. Pavel, "Dissipativity theory in game theory: On the role of dissipativity and passivity in nash equilibrium seeking," *IEEE Control Systems Magazine*, vol. 42, no. 3, pp. 150–164, 2022.

[51] G. Hu, Y. Pang, C. Sun, and Y. Hong, "Distributed nash equilibrium seeking: Continuous-time control-theoretic approaches," *IEEE Control Systems Magazine*, vol. 42, no. 4, pp. 68–86, 2022.

[52] R. Pass, *A Course in Networks and Markets*. MIT Press2019, 2019.

[53] D. Foster and H. Young, "On the nonconvergence of fictitious play in coordination games," *Games and Economic Behavior*, vol. **25**, pp. 79–96, 1998.

[54] J. Jordan, "Three problems in game theory," *Games and Economic Behavior*, vol. **5**, pp. 368–386, 1993.

[55] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a Nash equilibrium," *SIAM Journal on Computing*, vol. 39, no. 1, pp. 195–259, 2009.

[56] Y. Babichenko and A. Rubinstein, "Communication complexity of approximate nash equilibria," *Games and Economic Behavior*, vol. 134, pp. 376–398, 2022.

[57] C. Papadimitriou and G. Piliouras, "From Nash equilibria to chain recurrent sets: An algorithmic solution concept for game theory," *Entropy*, vol. 20, no. 10, 2018.

[58] S. Hart and A. Mas-Colell, "Uncoupled dynamics do not lead to nash equilibrium," *American Economic Review*, vol. 93, no. 5, pp. 1830–1836, 2003.

[59] ——, "Stochastic uncoupled dynamics and nash equilibrium," *Games and economic behavior*, vol. 57, no. 2, pp. 286–303, 2006.

[60] R. Laraki and P. Mertikopoulos, "Higher order game dynamics," *Journal of Economic Theory*, vol. 148, no. 6, pp. 2666–2695, 2013.

[61] B. Gao and L. Pavel, "On passivity, reinforcement learning, and higher order learning in multiagent finite games," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 121–136, 2020.

[62] D. P. Foster and S. Hart, "Smooth calibration, leaky forecasts, finite recall, and nash dynamics," *Games and Economic Behavior*, vol. 109, pp. 271–293, 2018.

[63] S. Hart and A. Mas-Colell, *Simple adaptive strategies: from regret-matching to uncoupled dynamics*. World Scientific, 2013, vol. 4.

[64] J. Willems, "Dissipative dynamical systems part I: General theory," *Archive for Rational Mechanics and Analysis*, vol. 45, no. 5, pp. 321–351, 1972.

[65] X. Fan, T. Alpcan, M. Arcak, T. Wen, and T. Basar, "A passivity approach to game-theoretic CDMA power control," *Automatica*, vol. 42, pp. 1837–1847, 2006.

[66] M. J. Fox and J. S. Shamma, "Population Games, Stable Games, and Passivity," in *51st IEEE Conf. on Decision and Control*, Dec 2012, pp. 7445–7450.

[67] G. Hines, M. Arcak, and A. Packard, "Equilibrium-independent passivity: A new definition and numerical certification," *Automatica*, vol. 47, no. 9, pp. 1949–1956, 2011.

[68] D. Fudenberg and D. K. Levine, *The Theory of Learning in Games*. The MIT Press, Cambridge, 1998.

[69] D. Hill and P. Moylan, "The stability of nonlinear dissipative systems," *IEEE Trans. on Automatic Control*, vol. 21, pp. 708–711, 1976.

[70] A. Pavlov and L. Marconi, "Incremental passivity and output regulation," *System & Control Letters*, vol. 57, pp. 400–409, 2008.

[71] P. Mertikopoulos and W. Sandholm, "Learning in games via reinforcement and regularization," *Mathematics of Operations Research*, vol. 41, no. 4, pp. 1297–1324, 2016.

[72] R. Cominetti, E. Melo, and S. Sorin, "A payoff-based learning procedure and its application to traffic games," *Games and Economic Behavior*, vol. 70, pp. 71–83, 2010.

[73] M. Benaïm, "Dynamics of stochastic approximation algorithms," *in Le Seminaire de Probabilites, Lecture Notes, Graduate Texts in Mathematics*, vol. 1709, pp. 1–68, 1999.

[74] B. Gao and L. Pavel, "Discounted Mirror Descent Dynamics in Concave Games," in *Proc. of the 58th IEEE Conf. on Decision and Control*, Dec 2019, pp. 5942–5947.

[75] ——, "Second-order mirror descent: exact convergence beyond strictly stable equilibria in concave games," in *Proc. of the 60th IEEE Conf. on Decision and Control*, Dec 2021, pp. 948–953.

[76] ——, "Bandit learning with regularized second-order mirror descent," in *Proc. of the 61st IEEE Conf. on Decision and Control*, Dec 2022.

[77] J. P. Hespanha, *Noncooperative game theory: An introduction for engineers and computer scientists*. Princeton University Press, 2017.

[78] D. Bertsekas, "Multiagent value iteration algorithms in dynamic programming and reinforcement learning," *Results in Control and Optimization*, vol. 1, p. 100003, 2020.

[79] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

[80] C. F. Camerer, T.-H. Ho, and J. K. Chong, "Behavioural game theory: thinking, learning and teaching," in *Advances in understanding strategic behaviour*. Springer, 2004, pp. 120–180.

[81] D. Baumann, J.-J. Zhu, G. Martius, and S. Trimpe, "Deep reinforcement learning for event-triggered control," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 943–950.

[82] M. Liu, I. Kolmanovsky, H. E. Tseng, S. Huang, D. Filev, and A. Girard, "Potential game based decision-making frameworks for autonomous driving," *arXiv preprint arXiv:2201.06157*, 2022.

[83] F. A. Potra and S. J. Wright, "Interior-point methods," *Journal of computational and applied mathematics*, vol. 124, no. 1-2, pp. 281–302, 2000.