State Based Potential Games

Jason R. Marden†

November 19, 2011

Abstract

There is a growing interest in the application of game theoretic methods to the design and control of multiagent systems. However, the existing game theoretic framework possesses inherent limitations with regards to these new prescriptive challenges. In this paper we propose a new framework, termed stated based games, which introduces an underlying state space into the game theoretic environment. This state space provides a system designer with an additional degree of freedom to help coordinate group behavior and overcome these limitations. We develop a spectrum of learning algorithms for classes of state based games and demonstrate the applicability of this enhanced framework on a broad spectrum of cooperative control problems.

1 Introduction

Many engineering systems can be characterized by a collection of interacting subsystems each making local decisions in response to local information. Existing wind farms are one example of such systems where individual wind turbines make local control decisions in response to local wind conditions [25]. Alternative systems include autonomous vehicles for surveillance missions [18, 21, 23] or the routing of information through a network [11,27]. Regardless of the specific application domain, the primary goal in such systems is to design local control policies for the individual subsystems to ensure that the emergent collective behavior is desirable with respect to the system level objective.

Developing an underlying theory for the design and control of multiagent systems remains a central goal for the field of distributed control. One of the core challenges in realizing this goal is that each application domain possesses inherent constraints that must be accounted for in the design process. For example, control strategies for individual turbines in a wind farm are constrained by the lack of a suitable communication system and the fact that the aerodynamic interaction between the turbines is poorly understood [25]. Accordingly, most of the existing research in distributed control focuses on specific applications as opposed to an underlying theory for distributed control, e.g., consensus and flocking [24, 31], sensor coverage [21, 23], routing information over networks [27], among many others.

Game theory is beginning to emerge as a valuable paradigm for the design and control of such multiagent systems [2,10,15]. Utilizing game theory for this purpose requires the following two step design process: (i) define the interaction framework of the agents within a game theoretic environment (game design) and (ii) define local decision making rules that specify how each agent processes available information to formulate a decision (learning design). The goal is to complete both steps (i) and (ii) to ensure that the collective behavior converges to a desirable operating point, e.g., a pure Nash equilibrium of the designed game. One of the major appeals of using game theory for multiagent systems is that game theory provides a
hierarchical decomposition between the design of the interaction framework and the design of the learning rules [10]. For example, if the interaction framework is designed as a potential game [22] then any learning algorithm for potential games can be utilized as a distributed control algorithm with provable guarantees on the emergent collective behavior. This decomposition could be instrumental in shifting research attention from application specific designs to an underlying theory for multiagent systems. The key to this realization is the development of a broad set of tools for both game design and learning design that could be used to address these application specific challenges. Hence, a system designer could appeal to available tools in a “plug-and-play” approach to complete the design process while satisfying the inherent constraints of the application at hand.

Several recent papers focus on identifying the viability of the framework of potential games as a mediating layer for this decomposition [18, 19]. With a broad set of existing results for learning in potential games [3,9,14,16,20,29,33–35], the primary focus of this work is on the development of methodologies for designing the interaction framework as a potential games while meeting constraints and objectives relevant to multiagent systems, e.g., locality of agent objective functions, efficiency guarantees for resulting equilibria, among many others. Unfortunately, the framework of potential games is not broad enough to meet this diverse set of challenges as several limitations are beginning to emerge. One such limitation involves multiagent systems where the desired collective behavior must satisfy a given coupled constraint on the agents’ actions, e.g., satisfy a desired signal to noise ratio in a wireless communication system [12]. It turns out that it is theoretically impossible to model such systems as a potential game where all resulting Nash equilibria optimize the system level objective while at the same time satisfying the given constraint [12]. Alternative limitations focus on the derivation of budget balanced agent objective functions in networked cost sharing problems [8, 19] and the derivation of local agent objective functions for multiagent coordination in distributed engineering systems [13]. These limitations provide the analytical justification for moving beyond potential games to games of a broader structure.

The first contribution of this paper is the formalization of a new game structure, termed state based potential games, as a new mediating layer for this decomposition. State based potential games are an extension of potential games where there is an underlying state space introduced into the game theoretic environment. Here, the underlying state can take on a variety of interpretations ranging from the introduction of dummy agents to dynamics for equilibrium selection. Regardless of the interpretation, the state provides a system designer with an additional degree of freedom to help coordinate group behavior. The framework of state based potential games is rich enough to overcome the aforementioned limitations as highlighted in Section 7. Interestingly, state based potential games can be thought of in a complimentary fashion to recent work in distributed learning algorithms [26, 29, 36] where an underlying state space is introduced into the learning environment to help coordinate behavior. For example, in [36] the authors introduce moods for each agent that impacts the agent’s behavior. The authors utilize these moods (or states) to develop a payoff based learning algorithm that converges to a pure Nash equilibrium in any game where one such equilibrium exists.

The second contribution of this paper focuses on the development of learning algorithms for state based potential games. In particular, we focus on extending two specific classes of learning algorithms for potential games to state based potential games. The first algorithm that we consider is a finite memory better reply process which converges to a (pure) Nash equilibrium in any potential game [35]. Here, a finite memory better reply process classifies virtually any decision making rule where agents seek to improve their one-shot

---

1There are existing frameworks which introduce a state space into the game theoretic environment, e.g., Markov games [30] and dynamic games [4]. A state based potential game represents a special class of Markov games where (i) each agent’s discount factor is 0 and (ii) agents’ utility function satisfy a relationship with a global potential function similar to that of potential games (see Definition 3.2). The novelty of the proposed developments is the use of this enhanced framework from a design perspective where the goal is to attain a dynamical process which yields efficient global behavior in distributed engineering systems.
payoff. This result demonstrates that potential games possess an inherent robustness to decision making rules as any form of “reasonable” behavior will converge to a Nash equilibrium. This result suggests that behavior will also converge to a Nash equilibrium even when agent behavior is corrupted by any number of issues pertinent to engineering systems including delays, inconsistent clock rates, or inaccuracies in information. Our first result in Theorem 4.2 demonstrates that such robustness is also present in state based potential game. In particular, any finite memory better reply process converges to a suitably defined equilibrium of the state based potential game.

Our next result focuses on the learning algorithm log-linear learning [1, 5, 6, 17, 28]. In potential games, log-linear learning provides guarantees on the percentage of time that the joint action profile will be at a maximizer of the potential function. One of the appeals of log-linear learning for multiagent systems is the fact that there are existing methodologies for designing agent utility functions that guarantee (i) that the resulting game is a potential game and (ii) that the action profiles that maximize the system level objective coincide with the potential function maximizers [2, 18, 32]. Our second result in Theorem 5.1 demonstrates that a variant of log-linear learning also provides similar analytical guarantees in state based potential games. Hence, both Theorems 4.2 and 5.1 begin the process of establishing a broad set of learning tools for this new decomposition with state based potential games as the mediating layer.

Lastly, we illustrate the applicability of the framework of state based potential games for the design of local agent control policies that leads to efficient behavior in a broad spectrum of multiagent systems. By local, we mean that each agent’s control policy can only depend on limited information regarding the behavior of other agents. In Section 7, we extensively review the results in [13] which develops a systematic methodology for the design of local utility functions that ensures both the existence and efficiency of the resulting equilibria provided that the system level objective is concave. This methodology yields a state based potential; hence, the control design can be completing by appealing to any available learning algorithm for state based potential games that guarantees convergence to an equilibrium. There are several alternative examples of systems successfully utilizing this architecture including networked resource allocation problems [19], systems with coupled constraints [12], and problems of network formation [7]. This growing number of examples demonstrates the viability of state based potential games as a new mediating layer for the design and control of multiagent systems.

2 Preliminaries

2.1 Background: Finite Strategic Form Games

We consider finite strategic form games with $n$ agents denoted by the set $N := \{1, \ldots, n\}$. Each agent $i \in N$ has a finite action set $A_i$ and a utility function $U_i : A \to \mathbb{R}$ where $A = A_1 \times \cdots \times A_n$ denotes the joint action set. We refer to a finite strategic-form game as “a game,” and we sometimes use a single symbol, e.g., $G$, to represent the entire game, i.e., the agent set, $N$, action sets, $A_i$, and cost functions $U_i$. We denote a game $G$ as a tuple $G = (N, \{A_i\}, \{U_i\})$ as we omit the subscript on the actions sets and utility functions for brevity, i.e., $\{A_i\} = \{A_i\}_{i \in N}$. For an action profile $a = (a_1, a_2, \ldots, a_n) \in A$, let $a_{-i}$ denote the profile of agent actions other than agent $i$, i.e., $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$. With this notation, we sometimes write a profile $a$ of actions as $(a_i, a_{-i})$. Similarly, we may write $U_i(a)$ as $U_i(a_i, a_{-i})$. We also use $A_{-i} = \prod_{j \neq i} A_j$ to denote the set of possible collective actions of all agents other than agent $i$.

In this paper we focus on analyzing equilibrium behavior in distributed systems in addition to distributed learning algorithms for attaining such equilibrium. The most well known form of an equilibrium is the Nash equilibrium.

**Definition 2.1 (Pure Nash Equilibrium)** An action profile $a^* \in A$ is called a (pure) Nash equilibrium if
for each agent \( i \in N \),
\[
U_i(a^*_i, a_{-i}^*) = \max_{a_i \in A_i} U_i(a_i, a_{-i}^*).
\]

A (pure) Nash equilibrium represents a scenario for which no agent has an incentive to unilaterally deviate.

One class of games that plays a prominent role in engineering multiagent systems is that of potential games [22]. In a potential game, the change in an agent’s utility that results from a unilateral change in strategy equals the change in a global potential function. Definition 2.2 makes this idea precise.

**Definition 2.2 (Potential Games)** A game \( G = (N, \{A_i\}, \{U_i\}) \) is called an (exact) potential game if there exists a global function \( \phi : A \rightarrow \mathbb{R} \) such that for every agent \( i \in N \), for every \( a_{-i} \in A_{-i} \), and for every \( a'_i, a''_i \in A_i \),
\[
U_i(a'_i, a_{-i}) - U_i(a''_i, a_{-i}) = \phi(a'_i, a_{-i}) - \phi(a''_i, a_{-i}).
\]

The global function \( \phi \) is called the potential function for game \( G \).

In potential games, any action profile maximizing the potential function is a Nash equilibrium, hence every potential game possesses at least one such equilibrium.

### 2.2 Learning in Games

The focus of this paper is on the attainment of dynamical processes which converges to a desirable system-wide behavior in distributed systems. To that end, we consider a repeated one-shot game, at each time \( t \in \{0, 1, 2, \ldots\} \) each agent \( i \in N \) chooses an action \( a_i(t) \in A_i \) and receives the payoff \( U_i(a(t)) \) where \( a(t) := (a_1(t), \ldots, a_n(t)) \). Each agent \( i \) chooses an action \( a_i(t) \) simultaneously according to the agent’s strategy at time \( t \) denoted as \( p_i(t) \in \Delta(A_i) \), where \( \Delta(A_i) \) is the set of probability distributions over the finite set of actions \( A_i \). We use \( p_{i}^{a_i}(t) \) to represent the probability that agent \( i \) selects action \( a_i \) at time \( t \); thus \( \sum_{a_i \in A_i} p_{i}^{a_i}(t) = 1 \). An agent’s strategy at time \( t \) can rely only on observations from the games played at times \( \{0, 1, 2, \ldots, t - 1\} \). Different learning algorithms are specified by both the assumptions on available information and the mechanism by which the strategies are updated as information is gathered. In the most general form, the strategy adjustment mechanism of agent \( i \) can be written as
\[
p_i(t) = F_i(a(0), \ldots, a(t - 1); U_i),
\]
meaning that an agent’s strategy is conditioned on all previous action profiles in addition to the structural form of the agent’s utility function. In potential games, there are a wide array of distributed learning algorithms that converge to a (pure) Nash equilibrium [3, 5, 6, 9, 16, 20, 29, 33–35].

### 3 State Based Games

In this paper we consider an extension to the framework of finite strategic form games, termed state based games, which introduces an underlying state space to the game theoretic framework. State based games represent a simplification of the class of Markov games [30]. In the proposed state based games we focus on myopic agents and static equilibrium concepts similar to that of Nash equilibrium. The state is introduced

\[^{2}\text{We avoid formally defining the framework of state based games within the context of Markov games as the inherent complexity of Markov games is unwarranted in our proposed research direction. The key difference between Markov games and state based games is the discount factor associated with future payoffs. In Markov games, an agent’s utility represents a discounted sum of future payoffs. Alternatively, in state based games, an agent’s utility represents only the current payoff, i.e., the discount factor is 0. This difference greatly simplifies the analysis of such games.}\]
as a coordinating entity used to improve system level behavior and can take on a variety of interpretations ranging from dynamics for equilibrium selection to the addition of dummy agents in a strategic form game that are preprogrammed to behave in a set fashion. The following definitions impose no restrictions on the admissible state space and transition functions. We do this to preserve the generality of the state based game architecture. Different applications will impose such restrictions as demonstrated in Section 7.

### 3.1 Definition of State Based Games

In state based games there exists a set of agents $N$ and an underlying finite state space $X$. Each agent $i \in N$ has a state invariant action set $A_i$ and a state dependent utility function $U_i : A \times X \to \mathbb{R}$ where $A$ is the set of joint actions.\(^3\) Lastly, there is a Markovian state transition function $P : A \times X \to \Delta(X)$ where $\Delta(X)$ denotes the set of probability distributions over the finite state space $X$. We denote a state based games $G$ by the tuple $G = \{N, \{A_i\}, \{U_i\}, X, P\}$.

Repeated play of a state based game produces a sequence of action profiles $a(0), a(1), \ldots$, and a sequence of states $x(0), x(1), \ldots$, where $a(t) \in A$ is referred to as the action profile at time $t$ and $x(t) \in X$ is referred to as the state at time $t$. The sequence of actions and states is generated according to the following process. At any time $t \geq 0$, each agent $i \in N$ myopically selects an $a_i(t) \in A_i$ according to some specified decision rule of the general form

$$ a_i(t) = F_i(a(0), \ldots, a(t - 1), x(0), \ldots, x(t); U_i), $$

meaning that an agent’s action choice is potentially conditioned on all previous action profiles, all previous states, and the structural form of the agent’s utility function. For example, if an agent used a myopic Cournot adjustment process for (4) then

$$ a_i(t) \in \arg\max_{a_i \in A_i} U_i(a_i, a_{-i}(t - 1), x(t)). $$

The state $x(t)$ and the action profile $a(t) := (a_1(t), \ldots, a_n(t))$ together determine each agent’s payoff $U_i(a(t), x(t))$ at time $t$. After all agents select their respective action, the ensuing state $x(t + 1)$ is chosen according to the state transition function $x(t + 1) \sim P(a(t), x(t))$ and the process is repeated. Here we use the symbol “$\sim$” to signify that the state is selected randomly according to the probability distribution $P(a(t), x(t))$.

### 3.2 Equilibrium for State Based Games

We are interested in analyzing the equilibrium points of the Cournot adjustment process in (5) for state based games. Before formally defining this equilibrium concept we introduce the notion of reachable states. For an action state pair $[a^0, x^0]$, the set of reachable states by an action invariant state trajectory $a^0$ is defined as $X(a^0|x^0) \subseteq X$ where a state $x \in X(a^0|x^0)$ if and only if there exists a time $t > 0$ such that

$$ \Pr \left[ x(t) = x | x(0) = x^0, x(k + 1) \sim P(x(k), a^0) \forall k \in \{0, 1, \ldots, t - 1\} \right] > 0 $$

Roughly speaking, the set $X(a^0|x^0)$ represents all states that may eventually emerge along an action invariant state trajectory starting from $[a^0, x^0]$. Accordingly, we now define the notion of recurrent state equilibria\(^4\)

\(^3\)One could also permit state dependent action sets where the set of available actions for agent $i$ given the state $x$ is $A^*_i \subseteq A_i$. However, such genericity is not needed for the results in this paper.

\(^4\)Stationary strategies are commonly studied in Markov or dynamic games. A stationary strategy represents an assignment of an action to each state. A collection of stationary strategies, one for each agent, represents an equilibrium if no single agent has a unilateral incentive to deviate to a different stationary strategy irrespective of the underlying state. In contrast, a recurrent state equilibrium does not require such a comprehensive strategy. Rather, a recurrent state equilibrium focuses on characterizing situations where a single action profile represents a pure Nash equilibrium for all states that may emerge. Definition 3.1 makes this idea precise.
as follows:

**Definition 3.1 (Recurrent State Equilibrium)** The action state pair \([a^*, x^*]\) is a recurrent state equilibrium with respect to the state transition process \(P(\cdot)\) if the following two conditions are satisfied:

(i) The state \(x^*\) is recurrent according to the process \(P(a = a^*, x)\) with initial state \(x(0) = x^*\). In terms of reachable states this implies that \(x^* \in X(a^*|x)\) for every state \(x \in X(a^*|x^*)\).

(ii) For every agent \(i \in N\) and every state \(x \in X(a^*|x^*)\),

\[
U_i(a_i^*, a_{-i}^*, x) = \max_{a_i \in A_i} U_i(a_i, a_{-i}^*, x)
\]

3.3 State Based Potential Games

A recurrent state equilibrium may or may not exist for a given state based game. We next provide a generalization of potential games to state based games which guarantee the existence of such an equilibrium. Much like potential games, the following class of games imposes an equivalence between differences in an agent’s utility and a global potential function resulting from a unilateral deviation.

**Definition 3.2 (State Based Potential Games)** A state based game \(G = \{N, \{A_i\}, \{U_i\}, X, P\}\) is a state based potential game if there exists a potential function \(\phi : \mathcal{A} \times X \rightarrow \mathbb{R}\) that satisfies the following two properties for every action state pair \([a, x]\) \(\in \mathcal{A} \times X\):

(i) For any agent \(i \in N\) and action \(a_i' \in A_i\)

\[
U_i(a_i', a_{-i}, x) - U_i(a, x) = \phi(a_i', a_{-i}, x) - \phi(a, x).
\] (6)

(ii) For any state \(x'\) in the support of \(P(a, x)\)

\[
\phi(a, x') \geq \phi(a, x).
\]

The second condition states that the potential function is non-decreasing along any action invariant state trajectory. The first condition states that each agent’s utility function is directly aligned with the potential function for all possible unilateral deviations from all action state pairs. Relaxing our requirement on equality between the change in utility and the change in the potential function in (6) gives rise to the class of ordinal state based potential games where the definition is equivalent to Definition 3.2 where (6) is relaxed and of the form

\[
U_i(a_i', a_{-i}, x) - U_i(a, x) > 0 \Rightarrow \phi(a_i', a_{-i}, x) - \phi(a, x) > 0.
\]

The following lemma proves the existence of a recurrent state equilibrium in any ordinal state based potential game.

---

5The condition that “for every action state pair \([a, x]\) \(\in \mathcal{A} \times X\)” can be relaxed by exploiting the properties of the state transition function \(P(\cdot)\). For a given state \(x\), the state transition function \(P(\cdot)\) reveals all action state pairs that could have generated \(x\), thereby restricting the class of unilateral deviations to consider. For example, if \(x(t)\) is the state at time \(t\) then the state \(x(t)\) must lie within the support of \(P(a(t - 1), x(t - 1))\). This restricts the set of possible action state pairs that could have been realized at time \(t - 1\) and hence restricts the set of unilateral deviations to consider which are of the form \((a_i, a_{-i}(t - 1))\). To that end, define \(V(a, x) \subseteq X\) as the support of the distribution \(P(a, x)\) and \(X(a) := \cup_{x \in X} V(a, x) \subseteq X\) as the set of states that are reachable with the action profile \(a\). Note that \(X(a|x) \subseteq X(a)\). We can now replace the statement “for every action state pair \([a, x]\) \(\in \mathcal{A} \times X\)” to “for every action profile \(a \in \mathcal{A}\) and state \(x \in X(a)\).”
Lemma 3.1  A recurrent state equilibrium exists in any ordinal state based potential game.\textsuperscript{6}

Proof: Let \([a^*, x^*] \in \arg \max_{(a, x) \in A \times X} \phi(a, x)\) be an action state pair that maximizes the potential function. Let \(X^*\) represent the recurrent states of the process \(P(a = a^*, x)\) starting from the initial state \(x(0) = x^*\) which is by definition nonempty. Choose any state \(x^{**} \in X^*\). The action state pair \([a^*, x^{**}]\) is a recurrent state equilibrium.

\[\square\]

4 Learning in State Based Potential Games

Having established the existence of a recurrent state equilibrium in any (ordinal) state based potential game, we now seek to identify how agents can learn to play such an equilibrium in a distributed fashion. We adopt the traditional framework of learning in games with the sole exception that the underlying game is not static because of the dynamic nature of the state.

4.1 Better Reply with Inertia Dynamics

Learning Nash equilibria in potential games is inherently robust. This theme is encapsulated in [35] which proves that any “finite memory better reply process” converges to an equilibrium in potential games.\textsuperscript{7} Informally, a finite memory better reply process refers to any process in which agents seek to improve their one-shot payoff. This robustness property is of fundamental importance to distributed engineering systems where the agents update rules (or control strategies) are inherently heterogenous stemming from delays or corruption in information, varying degrees of accessible information, and variations in clock rates amongst the agents which implies that the agents do not always update their actions each iteration. The result in [35] demonstrates that such levels of heterogeneity will not impact the asymptotic behavior of such learning algorithms.

In this section we seek to replicate these results for state based potential games thereby demonstrating that state based potential games also possess this robustness property. To that end, we consider the “better reply with inertia” dynamics for state based games. Define an agent’s strict better reply set for any action state pair \([a, x] \in A \times X\) as

\[B_i(a; x) := \{a_i' \in A_i : U_i(a_i', a_{-i}, x) > U_i(a, x)\}.\]

The better reply with inertia dynamics can be described as follows. At each time \(t > 0\), each agent \(i \in N\) selects an action at time \(t\) according to a strategy \(p_i \in \Delta(A_i)\) defined as

\begin{align*}
B_i(a(t-1); x(t)) &= \emptyset \Rightarrow p_i^{a_i(t-1)} = 1, \quad (7) \\
B_i(a(t-1); x(t)) \neq \emptyset \Rightarrow \left\{ 
\begin{array}{ll}
p_i^{a_i(t-1)} = \epsilon, \\
p_i^{a_i'} = \frac{(1-\epsilon)}{|B_i(a(t-1); x(t))|}, & \forall a_i' \in B_i(a(t-1); x(t)), \\
p_i^{a_i''} = 0, & \forall a_i'' \notin B_i(a(t-1); x(t)) \cup \{a_i(t-1)\},
\end{array} \right. \quad (8)
\end{align*}

for some constant \(\epsilon \in (0, 1)\) referred to as the agent’s inertia.

\textsuperscript{6}It is important to note that this lemma does not make any claims regarding the uniqueness of recurrent state equilibrium in ordinal state based potential games. Consequently, in Section 5 we focus on learning algorithms which provide equilibrium selection in ordinal state based potential games.

\textsuperscript{7}This result actually holds for a broader class of games known as weakly acyclic games.
Theorem 4.1 Let $G = \{N, \{A_i\}, \{U_i\}, X, P\}$ be an ordinal state based potential game with potential function $\phi : A \times X \rightarrow R$. If all agents adhere to the better reply with inertia dynamics then the action state pair converges almost surely to an action invariant set of recurrent state equilibria.

We omit the proof as the better reply with inertia dynamics is a special case of the finite memory better reply with inertia dynamics proved in Theorem 4.2 of the ensuing section.

4.2 Finite Memory Better Reply with Inertia Dynamics

We consider a finite memory better reply process with inertia for state based games. Let \[ [a(t-1), x(t-1)] \] be the action state pair at time $t-1$ and let $x(t)$ be the ensuing state selected according to the distribution $P(a(t-1), x(t-1))$. In this setting, each agent selects an action at time $t$ using information from the past $m$ action state pairs, i.e., $[a(t-m), x(t-m)], \ldots, [a(t-1), x(t-1)]$, and the current state $x(t)$, where $m$ is referred to as the memory length.

Define the $m$-step action history at time $t$ as $h^m_a(t) := \{a(t-m), \ldots, a(t-1)\}$, the $m$-step state history $h^m_x(t) := \{x(t-m), \ldots, x(t-1)\}$, and the tuple as $h^m(t) := [h^m_a(t), h^m_x(t)]$. A finite memory better response function is a set valued function of the form $B^m_i : (A \times X)^m \times X \rightarrow 2^{A_i}$, where $2^{A_i}$ denotes the power set of $A_i$. We refer to the range of $B^m_i$ as the (finite memory) better response set. A function $B^m_i$ is a finite memory better response function if it satisfies the following two conditions:

(i) If an action is in agent $i$’s finite memory better response set, then it must be a better response for some admissible action state pair in the agent’s history, i.e.,

\[ a_i \in B^m_i(h^m(t); x(t)) \]

\[ \Rightarrow \]

\[ a_i \in B_i(\tilde{a}; \tilde{x}) \] for some $\tilde{a} \in h^m_a(t)$ and $\tilde{x} \in \{h^m_x(t), x(t)\}$.

(ii) If an action $a_i$ is a better response for every action profile in the history $h^m_a$ given the current state $x(t)$, then the action $a_i$ must be in the finite memory better response set, i.e.,

\[ a_i \in B_i(\tilde{a}; x(t)) \] for all $\tilde{a} \in h^m_a(t)$

\[ \Rightarrow \]

\[ a_i \in B^m_i(h^m(t), x(t)) \].

Notice that the better response function used in the previous section satisfies both of these conditions. The finite memory better reply with inertia dynamics is identical to the better reply with inertia dynamics described in the previous section with the sole exception that the strategy highlighted in (7) and (8) now utilizes the finite memory better response set, $B^m_i(\cdot)$, as opposed to the better response set, $B_i(\cdot)$.

Theorem 4.2 Let $G = \{N, \{A_i\}, \{U_i\}, X, P\}$ be an ordinal state based potential game with potential function $\phi : A \times X \rightarrow R$. If all agents adhere to the finite memory better reply with inertia dynamics then the action state pair converges almost surely to an action invariant set of recurrent state equilibria.

Proof: Before starting the proof, we point out an observation regarding recurrent state equilibria. For any time $t^0 > 0$, if (i) the action state pair $[a(t^0), x(t^0)]$ is a recurrent state equilibrium, (ii) the action $a(t^0)$ is repeated for $m$ timesteps, i.e., $a(t^0) = \ldots = a(t^0 + m)$, and (iii) all agents adhere to any finite memory better reply process, then for any time $t^1 \geq t^0 + m$ the action state pair $[a(t^1), x(t^1)]$ will be a recurrent state equilibrium and furthermore $a(t^1) = a(t^0)$. Therefore, in order to prove the theorem, it suffices to show
that there exists a finite number of timesteps $T > 0$ and a finite constant $\gamma > 0$ such that the following is true: for any time $t^0 > 0$ and action state pair $[a(t^0), x(t^0)]$, if all agents adhere to any finite memory better reply process with inertia, then $[a(t^0 + T - m), x(t^0 + T - m)]$ will be a recurrent state equilibrium and $a(t^0 + T - m) = \ldots = a(t^0 + T)$ with at least probability $\gamma$.

The structure of the proof relies on two types of behavior: (i) saturate the memory and (ii) unilateral optimization.

**Step 1: Saturate the memory.** Let $[a^0, x^0] := [a(t^0), x(t^0)]$ be the action state pair at time $t^0$. Suppose the action profile $a^0$ is repeated $m$ consecutive periods, i.e., $a(t^0) = \ldots = a(t^0 + m)$, and the state $x(k + 1) \sim P(a^0, x(k))$ for all $k \in \{t^0, \ldots, t^0 + m\}$. This event happens with at least probability $\epsilon^{nm}$ because of agents’ inertia. Let $t^1 := t^0 + m$. If $[a^0, x^0]$ is a recurrent state equilibrium then $a(t) = a^0$ for all times $t \geq t^1$ and we are done.

**Step 2: Unilateral optimization.** Suppose $[a^0, x^0]$ is not a recurrent state equilibrium. If $[a^0, x(t^1)]$ is a recurrent state equilibrium, then we can repeat the above argument and we are done. If $[a^0, x(t^1)]$ is not a recurrent state equilibrium, then there exists an agent $i \in N$ with an action $a'_i \in A_i$ for some state $x' \in X(a^0(x(t^1)))$ such that $U_i(a'_i, a^0_{-i}, x') > U_i(a^0, x')$ and consequently $\phi(a'_i, a^0_{-i}, x') > \phi(a^0, x')$. Since $x' \in X(a^0(x(t^1)))$, there exists a time $t^2 \in \{t^1, \ldots, t^1 + |X|\}$ such that

$$\Pr[x(t^2) = x'|x(t^1), a(t^1) = a(t^1 + 1) = \ldots = a(t^2 - 1) = a^0] > \delta > 0.8$$

The event $x(t^2) = x'$ happens with at least probability $\delta \epsilon^n |X|$ because of the agents’ inertia. By Condition (ii) of the finite memory better response function, $a'_i \in B_{\delta}^m(h^m(t^2); x')$. Conditioned on the event $x(t^2) = x'$, the action profile $a^1 := (a'_i, a^0_{-i})$ will be played at time $t^2$ with at least probability $\epsilon^{n-1}(1 - \epsilon)/|A|$, i.e., all agents other than agent $i$ play their current action because of inertia and agent $i$ selects action $a'_i$ from his finite memory better response set. Note that $\phi(a^1, x(t^2)) > \phi(a^0, x(t^2))$ because

$$U_i(a'_i, a^0_{-i}, x(t^2)) - U_i(a^0, x(t^2)) > 0 \iff \phi(a'_i, a^0_{-i}, x(t^2)) - \phi(a^0, x(t^2)) > 0.$$

Furthermore, note that $\phi(a^0, x(t^2)) \geq \phi(a^0, x^0)$ because of Condition (ii) of ordinal state based potential games, i.e., the potential function is nondecreasing along any action invariant state trajectory. Therefore, $\phi(a^1, x(t^2)) > \phi(a^0, x^0)$.

**Step 3: Saturate the memory.** Suppose the action profile $a^1$ is repeated $m$ consecutive periods, i.e., $a(t^2) = \ldots = a(t^2 + m) = a^1$, and the state $x(k + 1) \sim P(a^1, x(k))$ for all $k \in \{t^2, \ldots, t^2 + m\}$. This event happens with at least probability $\epsilon^{nm}$ because of the agents’ inertia. If $[a^1, x(t^2)]$ is a recurrent state equilibrium then $a(t) = a^1$ for all times $t \geq t^2$ and we are done. Otherwise, we can repeat the same argument as before.

One can repeat this argument at most a finite number of times, because the potential function is increasing with each step and both the joint action set $A$ and state space $X$ are finite. As a result, for any time $t_0 > 0$ and any action state pair $[a(t^0), x(t^0)] \in A \times X$, if all agents adhere to any finite memory better reply process with inertia, then $[a(t^0 + T - m), x(t^0 + T - m)]$ will be a recurrent state equilibrium and $a(t_0 + T - m) = \ldots = a(t^0 + T)$ with at least probability $\gamma$ where

$$T := m + (|X| + m) \cdot |A \times X|,$$

$$\gamma := \epsilon^{nm(\delta \epsilon^n |X| + m)} \epsilon^{n-1}(1 - \epsilon)/|A|^{|A \times X|}.$$

\[\square\]

\textsuperscript{8}We choose $\delta$ in the above setting to satisfy the given condition for any action state pair $[a', x']$ and feasible state $x' \in X(a'|x')$. 

9
5 Equilibrium Selection in State Based Potential Games

A second feature of potential games that is desirable in engineering systems is that of equilibrium selection. In particular, there are learning dynamics for potential games that guarantee convergence to the potential function maximizer as opposed to any equilibrium, e.g., log-linear learning [5, 6, 17]. This type of equilibrium selection is significant in many engineering systems as the interaction framework associated with a distributed multiagent system can frequently be represented as a potential game where the action profile that optimizes the system level objective is precisely the potential function maximizer [2, 18]. In this section we demonstrate that such algorithms also extend to the class of state based potential games under some restrictions on the state transition matrix $P$.

5.1 Background: Binary log-linear learning for potential games

Consider an exact potential game $G$ with potential function $\phi : \mathcal{A} \rightarrow \mathbb{R}$. The following learning algorithm for potential games is known as binary log-linear learning [5, 17]. At each time $t > 0$, one agent $i \in \mathcal{N}$ is randomly chosen and allowed to alter his current action. All other agents must repeat their current action at the ensuing time step, i.e. $a_{-i}(t) = a_{-i}(t - 1)$. At time $t$, agent $i$ selects one trial action $a_i$ uniformly from the agent’s action set $\mathcal{A}_i \setminus \{a_i(t - 1)\}$. Then agent $i$ employs the strategy $p_i(t) \in \Delta(\mathcal{A}_i)$ where

$$
p_i^{a_i(t-1)}(t) = \frac{e^{\frac{1}{\tau} U_i(a(t-1))}}{e^{\frac{1}{\tau} U_i(a(t-1))} + e^{\frac{1}{\tau} U_i(a_i, a_{-i}(t-1))}}
$$

$$
p_i^{a_i}(t) = \frac{e^{\frac{1}{\tau} U_i(a(t-1))}}{e^{\frac{1}{\tau} U_i(a(t-1))} + e^{\frac{1}{\tau} U_i(a_i, a_{-i}(t-1))}}
$$

$$
p_i^{a_i}(t) = 0, \; \forall a_i' \in \mathcal{A}_i \setminus \{a_i, a_i(t - 1)\}
$$

for some temperature $\tau > 0$. The temperature $\tau$ determines how likely agent $i$ is to select a suboptimal action. As $\tau \rightarrow \infty$, agent $i$ will select either action $a_i$ or $a_i(t - 1)$ with equal probability. As $\tau \rightarrow 0$, agent $i$ will select the action that yields the higher payoff, i.e., comparing $U_i(a_i, a_{-i}(t - 1))$ and $U_i(a(t - 1))$, with arbitrarily high probability. In the case of a non-unique comparison, i.e., $U_i(a_i, a_{-i}(t - 1)) = U_i(a(t - 1))$, agent $i$ will select either $a_i$ or $a_i(t - 1)$ at random (uniformly).

In the repeated potential game in which all agents adhere to binary log-linear learning, the stationary distribution of the joint action profiles is $\mu \in \Delta(\mathcal{A})$ where [5]

$$\mu(a; \tau) = \frac{e^{-\frac{1}{\tau} \phi(a)}}{\sum_{a \in \mathcal{A}} e^{-\frac{1}{\tau} \phi(a)}}.$$  \hfill (9)

One can interpret the stationary distribution $\mu$ as follows: for sufficiently large times $t > 0$, $\mu(a; \tau)$ equals the probability that $a(t) = a$ for a fixed temperature $\tau > 0$. As one decreases the temperature, $\tau \rightarrow 0$, all the weight of the stationary distribution $\mu$ is on the joint actions that maximize the potential function.

5.2 State based log-linear learning

Consider the following variant of log-linear learning for state based games. At each time $t$, at most one agent $a$ is randomly selected according to a fixed probability distribution $q$ where $q_i > 0$ denotes the probability that agent $i$ is selected. Furthermore, we require that $\sum_{i \in \mathcal{N}} q_i < 1$ meaning that there is a positive probability that no agent is selected, which we define as $q_0 = 1 - \sum_{i \in \mathcal{N}} q_i > 0$. For simplicity, we assume that $q_i = 1/(n + 1)$ for all $i \in \mathcal{N}$. All non-selected agents must repeat their current action at the ensuing
time step, i.e. \( a_{-i}(t) = a_{-i}(t - 1) \). At time \( t \), agent \( i \) selects one trial action \( a_i \) uniformly from the agent’s action set \( A_i \setminus \{a_i(t - 1)\} \). Then agent \( i \) employs the strategy \( p_i(t) \in \Delta(A_i) \) where

\[
p_i^{a_i(t-1)}(t) = e^{\frac{1}{\tau} U_i(a_i(t-1),x(t))} \frac{1}{e^{\frac{1}{\tau} U_i(a_i(t-1),x(t))} + e^{\frac{1}{\tau} U_i(a_i,a_{-i}(t-1),x(t))}}
\]

\[
p_i^{a_i}(t) = e^{\frac{1}{\tau} U_i(a_i(t-1),x(t))} \frac{1}{e^{\frac{1}{\tau} U_i(a_i(t-1),x(t))} + e^{\frac{1}{\tau} U_i(a_i,a_{-i}(t-1),x(t))}}
\]

\[
p_i^{a_i}(t) = 0, \ \forall a_i' \in A_i \setminus \{a_i,a_i(t-1)\}
\]

for some temperature \( \tau > 0 \). Let \( a_i \sim p_i(t) \) be the action selected at time \( t \). The action profile at time \( t \) is \( a(t) = (a_i, a_{-i}(t)) \). The ensuing state \( x(t + 1) \) is then chosen randomly according to the transition probability \( P(a(t), x(t)) \). Here, it is important to note that state transitional probabilities \( P(\cdot) \) and agent selection probabilities \( q \) are fixed for all \( \tau \).

The analysis of log-linear learning for potential games characterizes the precise stationary distribution as a function of the temperature \( \tau \). This analysis relies on the fact that the underlying process is reversible which in general is not satisfied in state based games. However, the importance of the result is not solely the explicit form of the stationary distribution, but rather the fact that as \( \tau \to 0 \) the support of the limiting stationary distribution is precisely the set of action profiles that maximize the potential function. The support of the limiting distribution is referred to as the stochastically stable states. More precisely, an action profile \( a \in A \) is stochastically stable if and only if \( \lim_{\tau \to 0} \mu(a; \tau) > 0 \). We now demonstrate that the process described above guarantees that the stochastically stable states, which are now action state pairs \( [a, x] \), are contained in the set of state based potential function maximizers.

**Theorem 5.1** Let \( G = \{N, \{A_i\}, \{U_i\}, X, P\} \) be an ordinal state based potential game with a state invariant potential function \( \phi : A \to \mathbb{R} \) that satisfies the following three conditions for any action profile \( a \in A \) and state \( x \in X(a) \):\(^9\)

(i) The action invariant state transition process \( P(a, \cdot) \) is aperiodic and irreducible over the states \( X(a) \).

(ii) For any agent \( i \in N \) and action \( a_i' \in A_i \)

\[
U_i(a_i', a_{-i}, x) - U_i(a, x) \leq \phi(a_i', a_{-i}) - \phi(a)
\]

(iii) For any agent \( i \in N \) and action \( a_i' \in A_i \) there exists a state \( x' \in X(a) \) such that

\[
U_i(a_i', a_{-i}, x') - U_i(a, x') = \phi(a_i', a_{-i}) - \phi(a)
\]

For such state based potential games, the process log-linear learning guarantees that an action state pair \([a^*, x^*]\) is stochastically stable if and only if the action profile \( a^* \in \arg\max_{a \in A} \phi(a) \).

Note that Conditions (ii) and (iii) of Theorem 5.1 provide a relaxation to the potential game structure by relaxing the equality constraint. We present the proof of Theorem 5.1 in the Appendix.\(^10\)

\(^9\)Recall that \( V(a, x) \subseteq X \) is the support of the distribution \( P(a, x) \) and \( X(a) = \cup_{x \in X} V(a, x) \subseteq X \) represents the set of states that are reachable with the action profile \( a \).

\(^10\)One application for Theorem 5.1 if from [19] where the authors focus on utility design for a general class of resource allocation problems with submodular and separable objective functions. In such concave games, the authors proved that it is impossible to design budget-balanced utility functions within the framework of strategic form games that ensures that the optimal allocation is a pure Nash equilibrium. The authors were able to accomplish this task using the framework of state based games; however, no learning algorithms were provided which converge to such equilibria. The extension of log-linear learning presented here provides such an algorithm.
6 A Controls Based Formulation of State Based Games

A common assumption in game theory is that an agent can select any action in the agent’s action set at any instance in time. In multiagent systems this assumption is not necessarily true. Rather, each agent has the ability to influence his action through different control strategies. Accordingly, we focus on the situation where each agent $i$ has a set of control strategies $\Pi_i$ that the agent can use to influence the agent’s action choice. Let $\Pi := \prod_i \Pi_i$ be the set of joint control strategies. We represent the action transition function of agent $i$ by a deterministic (or stochastic) transition function $g_i : A_i \times \Pi \rightarrow A_i$. In a repeated state based game, we adopt the convention that $a_i(t + 1) = g_i(a_i(t), \pi(t))$ for any agent $i \in N$ and time $t \geq 1$ where $\pi(t) = (\pi_1(t), \ldots, \pi_n(t))$ is the joint control decision at time $t$. This implies that an agent’s ensuing action is potentially influenced by the control strategies of all agents. We assume throughout that each agent has a null control strategy $\pi_i^0 \in \Pi_i$ such that for any agent $i \in N$ and action $a_i \in A_i$ we have $a_i = g_i(a_i, \pi^0)$ where $\pi^0 = (\pi_1^0, \ldots, \pi_n^0)$.

We re-formulate the state based game described in Section 3 for control based decisions as follows: First, we embed the action profiles $A$ and the original state space $X$ into a new space $Y = A \times X$. Each agent $i \in N$ now has a state invariant control strategy set $\Pi_i$ as described above and a new state dependent utility function of the form $U_i : \Pi \times Y \rightarrow \mathbb{R}$. Lastly, there is a Markovian state transition function $Q : \Pi \times Y \rightarrow \Delta(Y)$ which encompasses both the previous state transition function $P(\cdot)$ and the new action transition functions $\{g_i(\cdot)\}$. Repeated play of a state based game proceeds in the same fashion as before. All previous definitions and results directly carry over to this controls based formulation by integrating the null action appropriately into the definitions and learning algorithms as action invariant trajectories are now encoded by the null control strategy $\pi^0$. For example, Condition (ii) of Definition 3.2 now takes the following form: for any policy state pair $[\pi, y]$ and any state $y'$ in the support of $Q(\pi, y)$ the potential function satisfies $\phi(\pi^0, y') \geq \phi(\pi, y)$. We omit the remaining details for brevity.

7 Illustrations: Designing Local Control Laws

Consider a multiagent system consisting of a set of agents $N$, an action set $A_i = \mathbb{R}$ for each agent $i \in N$, and a system level objective of the form $W : A \rightarrow \mathbb{R}$ that a system designer seeks to maximize. We assume throughout that $W$ is concave, continuously differentiable, and that a solution is guaranteed to exist. Here, we focus on the design of local control laws where the information available to each agent is represented by an undirected and connected graph $G = \{N, E\}$ with nodes $N$ and edges $E$. These local control policies produce a sequence of action profiles $a(0), a(1), a(2), \ldots$ where at each iteration $t \in \{0, 1, \ldots\}$ each agent $i$ makes a decision independently according to a local control law of the form:

$$a_i(t) = F_i \left( \{\text{Information about agent } j \text{ at time } t\}_{j \in N_i} \right) \tag{10}$$

where $N_i := \{j \in N : (i, j) \in E\}$ denotes the neighbors of agent $i$. The goal in this setting is to design the local controllers $\{F_i(\cdot)\}_{i \in N}$ within the given informational constraints such that the collective behavior converges to a joint decision $a^* \in \arg \max_{a \in A} W(a)$.

Consider a distributed gradient ascent algorithm where each agent’s control policy represents a gradient ascent process with respect to the system level objective $W$. This approach ensures that the action profile will converge to the optimizer of the system level objective $W$. However, it is important to highlight that the structure of the agents’ control policies may not satisfy our desired local constraints in (10). An alternative approach is to have each agent’s control policy represent a gradient ascent process with respect to a given local utility function $U_i$ as opposed to the system level objective $W$. Is it possible to design local utility functions for the individual agents such that (i) all resulting equilibria optimize the system level objective and (ii) the game possesses a structure that can be exploited in distributed learning?
We now review the methodology developed in [13] which accomplishes this task. The underlying design fits into the controls based formulation of state based games as depicted in Section 6. Accordingly, the structure of the agents’ local utility functions is of the form

\[ U_i : \prod_{j \in N_i} (\Pi_j \times Y_j) \to R. \]

(11)

where \( \Pi_j \) is the set of control policies for agent \( j \) and \( Y_j \) is the set of local state variables for agent \( j \). The forthcoming design embodies continuous action sets (or control policies) as opposed to finite action sets as considered in the rest of this paper. While the learning results contained in this paper do not readily apply to this setting, the designed game is a state based potential game. Accordingly, results in [13] prove that gradient ascent dynamics guarantee convergence to recurrent state equilibria for such continuous state based potential games. The details of the design are as follows:

**Agents:** The agent set is \( N = \{1, 2, \ldots, n\} \).

**States:** The starting point of the design is an underlying state space \( Y \) where each state \( y \in Y \) is defined as a tuple \( y = (a, e) \), where \( a = (a_1, \ldots, a_n) \in \mathbb{R}^n \) is the action profile and \( e = (e_1, \ldots, e_n) \) is a profile of agent based estimation terms where \( e_i = (e_{i1}^1, \ldots, e_{in}^n) \in \mathbb{R}^n \) is agent \( i \)'s estimation for the joint action profile \( a \). The term \( e_i^k \) captures agent \( i \)'s estimate of agent \( k \)'s action \( a_k \).

**Action Sets:** Each agent \( i \) is assigned a set of control policies \( \Pi_i \) that permits the agents to change their value and change their estimation through communication with neighboring agents. Specifically, a control for agent \( i \) is defined as a tuple \( \pi_i = (\hat{a}_i, \hat{e}_i) \) where \( \hat{a}_i \in \mathbb{R} \) indicates a change in the agent’s action and \( \hat{e}_i := (\hat{e}_i^1, \ldots, \hat{e}_i^n) \) indicates a change in the agent’s estimation terms \( e_i \). We represent the changes in estimation terms for an agent \( k \) by the tuple \( \hat{e}_i^k := \{\hat{e}_{i-j}^k\}_{j \in N_i} \) where \( \hat{e}_{i-j}^k \in \mathbb{R} \) represents the estimation value that agent \( i \) “passes” to agent \( j \) regarding the action of agent \( k \).

**State Dynamics:** We now describe how the state evolves as a function of the control choices \( \pi(0), \pi(1), \ldots, \pi(k) \) is the control profile at stage \( k \). Define the initial state as \( y(0) = [a(0), e(0)] \) where \( a(0) = (a_1(0), \ldots, a_n(0)) \) is the initial action profile and \( e(0) \) is an initial estimation profile that satisfies

\[ \sum_{i \in N} e_i^k(0) = n \cdot a_k(0) \]

(12)

for each agent \( k \in N \); hence, the initial estimation values are contingent on the initial action profile.\(^\text{11}\) We represent the state transition function \( Q(\pi, y) \) by a set of local state transition functions \( \{Q_i^e(\pi, y)\}_{i \in N} \) and \( \{Q_{i,k}(\pi, y)\}_{i,k \in N} \). For any distinct agents \( i, k \in N \), state \( y = (a, e) \), and control choice \( \pi = (\hat{a}, \hat{e}) \), the local state transition functions take on the form

\[
Q_i^e(\pi, y) = a_i + \hat{a}_i \\
Q_{i,i}(\pi, y) = e_i^i + n \cdot \hat{a}_i + \sum_{j \in N; i \in N_j} \hat{e}_{j-i}^j - \sum_{j \in N_i} \hat{e}_{i-j}^i \\
Q_{i,k}(\pi, y) = e_i^k + \sum_{j \in N; i \in N_j} \hat{e}_{j-i}^k - \sum_{j \in N_i} \hat{e}_{i-j}^k.
\]

(13)

It is straightforward to show that for any sequence of control choices \( \pi(0), \pi(1), \ldots \), the resulting state trajectory \( y(t) = (a(t), e(t)) = Q(\pi(t-1), y(t-1)) \) satisfies for all times \( t \geq 1 \) and agents \( k \in N \)

\[ \sum_{i=1}^n e_i^k(t) = n \cdot a_k(t). \]

(14)

\(^\text{11}\)Note that satisfying condition (12) is trivial as we can set \( e_i^i(0) = n \cdot a_i(0) \) and \( e_i^j(0) = 0 \) for all agents \( i, j \in N \) where \( i \neq j \).
**Agent Utility Functions:** The last part of our design is the agents’ utility functions. For any state $y \in Y$ and admissible control $\pi \in \Pi$ the utility function of agent $i$ is defined as

$$U_i(\pi, y) = \sum_{j \in N_i} W(\tilde{e}_1^j, \tilde{e}_2^j, ..., \tilde{e}_n^j) + \sum_{j \in N_i} \sum_{k \in N} [\tilde{e}_k^i - \tilde{e}_k^j]^2$$

where $(\tilde{v}, \tilde{e}) = Q(\pi, y)$ represents the ensuing state. Note that the agents’ utility functions are local and of the form (11).

In [13] the authors prove two properties regarding the designed state based game. First, the designed game is a state based potential game with potential function

$$\phi(\pi, y) = \sum_{i \in N} W(\tilde{e}_1^i, \tilde{e}_2^i, ..., \tilde{e}_n^i) + \frac{1}{2} \sum_{i \in N} \sum_{j \in N} \sum_{k \in N} [\tilde{e}_k^i - \tilde{e}_k^j]^2$$

where $(\tilde{a}, \tilde{e}) = Q(\pi, y)$ represents the ensuing state. This ensures (i) the existence of a recurrent state equilibrium and (ii) that the game possesses an underlying structure that can be exploited by distributed learning algorithms, e.g., gradient play [13]. The second property is that all equilibria of the designed game are optimal with respect to $W$ provided that the designed interaction graph satisfies some relatively weak connectivity conditions.$^{12}$ More specifically, a control state pair $[(\pi, y)] = [(\hat{a}, \hat{e}), (a, e)]$ is a recurrent state equilibrium if and only if the following conditions are satisfied:

- (i) The action profile $a$ is optimal, i.e., $a \in \arg\max_{a' \in A} W(a')$.
- (ii) The estimation profile $e$ is consistent with $a$, i.e., for any $\forall i, k \in N$ we have $e_k^i = a_k$.
- (iii) The change in action profile satisfies $\hat{a} = 0$.
- (vi) The change in estimation profile satisfies the following for all agents $i, k \in N$,

$$\sum_{j \in N_i} \hat{e}_k^{i \rightarrow j} = \sum_{j \in N : i \in N_j} \hat{e}_k^{j \rightarrow i}.$$  

Hence, this design provides a systematic methodology for distributing an optimization problem under virtually any desired degree of locality in agent objective functions.

### 8 Conclusions

The beauty of game theory centers on its generality which makes its tools and structure accessible to a broad class of distributed systems encompassing both the social and engineering sciences. Unfortunately, as game theory evolves from a descriptive tool for social systems to a prescriptive tool for engineering systems, the existing framework is not suitable for meeting this new set of design challenges. Our primary goal in this paper is to provide a new game theoretic framework that is better suited for handling this prescriptive agenda – state based potential games. We demonstrate that this framework is broad enough to meet the challenges inherent to a wide variety of multiagent systems possessing unique objectives and constraints. Having a common architecture for multiagent systems is crucial as it paves the way for a collective effort in developing an underlying theory for the design and control of multiagent systems.

$^{12}$One such condition is that the interaction graph is connected, undirected, and contains at least one odd-length cycle (non-bipartite). We direct the reader to [13] for alternative sufficient conditions on the interaction graph.
References


9 Appendix

We start with an outline of the proof of Theorem 5.1. The learning algorithm state based log-linear learning induces a finite Markov process over the state space \( \mathcal{A} \times X \). The analysis of this Markov process is challenging stemming from the fact that the process is not reversible. However, in Lemma 9.1 we prove that this Markov process fits into the class of regular perturbed processes as introduced in [33]. This realization allows for this process to be analyzed in a tractable fashion using graph theoretic techniques termed resistance trees [33]. Lemmas 9.2 and 9.3 establish two key analytical properties of resistance trees for state based log-linear learning which simplify the analysis of the stationary distribution for the given process. Lastly, Lemma 9.4 complete the proof by exploiting these properties to provide the desired characterization.

Before proceeding with the proof, we first introduce a brief background on the analysis of regular perturbed processes using the theory of resistance trees [33].

9.1 Background on Regular Perturbed Processes

For a detailed review of the theory of resistance trees, please see [33]. Let \( P^0 \) denote the probability transition matrix for a finite state Markov chain over the state space \( Z \). Consider a “perturbed” process such that the size of the perturbation can be parameterized by a scalar \( \epsilon > 0 \), and let \( P^\epsilon \) be the associated transition probability matrix. The process \( P^\epsilon \) is called a regular perturbed Markov process if \( P^\epsilon \) is ergodic for all sufficiently small \( \epsilon > 0 \) and \( P^\epsilon \) approaches \( P^0 \) at an exponentially smooth rate [33]. Specifically, the latter condition means that for all sufficiently small \( \epsilon \), the resistance

\[
R(z \rightarrow z') = \frac{1}{\lim_{\epsilon \rightarrow 0^+} P_{zz'}^\epsilon} = \frac{1}{P_{zz'}^0},
\]

and

\[
P_{zz'}^\epsilon > 0 \text{ for some } \epsilon > 0 \Rightarrow 0 < \lim_{\epsilon \rightarrow 0^+} \epsilon R(z \rightarrow z') < \infty,
\]

for some nonnegative real number \( r(z \rightarrow z') \), which is called the resistance of the transition \( z \rightarrow z' \). (Note in particular that if \( P_{zz'}^0 > 0 \) then \( r(z \rightarrow z') = 0 \).) We will adopt the convention that if \( P_{zz'}^\epsilon = 0 \) for all sufficiently small \( \epsilon \), then the resistance \( R(z \rightarrow z') = \infty \).

Now construct a complete directed graph with \( |Z| \) vertices, one for each state. The weight on the directed edge \( z_i \rightarrow z_j \) is \( R(z_i \rightarrow z_j) \). A tree, \( T \), rooted at vertex \( z_j \), or \( z_j \)-tree, is a set of \( |Z| - 1 \) directed edges such that, from every vertex different from \( z_j \), there is a unique directed path in the tree to \( z_j \). The resistance of a rooted tree, \( T \), is the sum of the resistances \( R(z_i \rightarrow z_j) \) on the \( |Z| - 1 \) edges that compose it. The stochastic potential, \( \gamma(z_j) \), of state \( z_j \) is defined to be the minimum resistance over all trees rooted at \( z_j \). The following theorem gives a simple criterion for determining the stochastically stable states (Lemma 1, [33]).

**Theorem 9.1** Let \( P^\epsilon \) be a regular perturbed Markov process, and for each \( \epsilon > 0 \) let \( \mu^\epsilon \) be the unique stationary distribution of \( P^\epsilon \). Then \( \lim_{\epsilon \rightarrow 0} \mu^\epsilon \) exists and the limiting distribution \( \mu^0 \) is a stationary distribution of \( P^0 \). The stochastically stable states (i.e., the support of \( \mu^0 \)) are precisely those states with minimum stochastic potential.

9.2 Proof of Theorem 5.1

Log-linear learning induces a finite and aperiodic process over the state space \( \mathcal{A} \times X \). We will now show that the trees rooted at recurrent state equilibria that optimize the potential \( \phi(\cdot) \) will have minimum stochastic potential over all possible states. Therefore, according to Theorem 9.1, the probability that the state will be at a recurrent state equilibrium that optimizes the potential \( \phi(\cdot) \) can be made arbitrarily close to 1.

---

\(^\dagger\)We adopt the convention that \( P_{zz'} \) denotes the probability of transitioning from state \( z \) to state \( z' \).
Accordingly, of the transition is where the associated transition probability matrix. The probability of transitioning from \(x \rightarrow x'\) when the action profile changes by a unilateral deviation, i.e., \(a \rightarrow a' = (a_i', a_{-i})\) where \(a_i' \neq a_i\) and is of the form

\[
[a, x] \rightarrow [a', x']
\]

where \(P(x'|a', x) > 0\). The second type of transition is when the action profile stays constant and is of the form

\[
[a, x] \rightarrow [a, x']
\]

where \(P(x'|a, x) > 0\). We will now prove Theorem 5.1 using the following sequence of lemmas. The first lemma states that the considered process is in fact a regular perturbed process.

**Lemma 9.1** **Log-linear learning for state based potential games induces a regular perturbed Markov process.** Let \([a, x]\) be any action state pair. The resistance of the feasible transitions is of the form

\[
R([a, x] \rightarrow ([a_i, a_{-i}, x']) = \max_{a_i' \in \{a_i, a_i'\}} U_i(a_i^*, a_{-i}, x) - U_i(a_i', a_{-i}, x)
\]

(15)

\[
R([a, x] \rightarrow [a, x']) = 0
\]

(16)

**Proof:** We analyze this process with respect to \(\epsilon := e^{-\frac{1}{\tau}}\) rather than the temperature \(\tau\). Let \(P^\epsilon\) denote the associated transition probability matrix. The probability of transitioning from \([a^0, x^0]\) to \([a^1, x^1]\) where \(a^1 := (a^1_i, a^0_{-i})\) for some agent \(i\) and action \(a^1_i \neq a^0_i\) is

\[
P^\epsilon_{[a^0, x^0] \rightarrow [a^1, x^1]} = \frac{q_i}{|A_i| - 1} \cdot \frac{\epsilon^{-U_i(a^1_i, a^0_{-i}, x^0)}}{\epsilon^{-U_i(a_i, a^0_{-i}, x^0)}} \cdot P(x^1|x^0, a^1).
\]

(17)

The probability is divided into three components: (i) the probability of selecting agent \(i\) and trial action \(a^1_i\), (ii) the probability of agent \(i\) selecting action \(a^1_i\), and (iii) the probability of the state traversing to \(x^1\). Let \(a^* = \arg \max_{a_i \in \{a^0_i, a^1_i\}} U_i(a^0_i, a^0_{-i}, x^0)\). Multiplying the numerator and denominator of (17) by \(\epsilon^{-U_i(a^1_i, a^0_{-i}, x^0)}\), we obtain

\[
P^\epsilon_{[a^0, x^0] \rightarrow [a^1, x^1]} = \frac{q_i}{|A_i| - 1} \cdot \sum_{a_i \in \{a^0_i, a^1_i\}} \frac{\epsilon^{-U_i(a^1_i, a^0_{-i}, x^0)} - U_i(a_i^0, a_{-i}, x^0)}{\epsilon^{-U_i(a_i, a^0_{-i}, x^0)} - U_i(a^0_i, a^0_{-i}, x^0)} \cdot P(x^1|x^0, a^1).
\]

Accordingly,

\[
\lim_{\epsilon \to 0^+} P^\epsilon_{[a^0, x^0] \rightarrow [a^1, x^1]} = \frac{q_i}{|A_i| - 1} \cdot \frac{1}{z_i} \cdot P(x^1|x^0, a^1) > 0.
\]

where \(z_i = \{a_i \in \{a^0_i, a^1_i\} : U_i(a_i, a^0_{-i}) = \max_{a_i \in \{a^0_i, a^1_i\}} U_i(a_i, a^0_{-i}, x^0)\}\) denotes agent \(i\)’s best response set. This implies that the transitional probability decays at an exponentially smooth rate and the resistance of the transition is

\[
R([a^0, x^0] \rightarrow [a^1, x^1]) = U_i(a^*_i, a^0_{-i}, x^0) - U_i(a^1_i, a^0_{-i}, x^0) \geq 0.
\]

(18)

Similarly, the probability of transitioning from \([a^0, x^0]\) to \([a^0, x^1]\) is

\[
P^\epsilon_{[a^0, x^0] \rightarrow [a^0, x^1]} \geq q_0 \cdot P(x^1|x^0, a^0) > 0.
\]
The resistance of this transition is
\[ R([a^0, x^0] \to [a^1, x^1]) = 0. \] (19)

\[ \square \]

The second lemma proves a property about resistance trees with minimum stochastic potential. Let \([a^0, x^0] \to [a^1, x^1] \) represent a feasible transition where \(a^1 := (a_i^1, a_{-i}^0)\) for some agent \(i\) and action \(a_i^1 \neq a_i^0\). Suppose \(U_i(a_i^0, a_{-i}^0, x^0) > U_i(a_i^1, a_{-i}^0, x^0)\). By Lemma 9.1 and Condition (ii) of Theorem 5.1, we know that
\[ R([a^0, x^0] \to [a^1, x^1]) = U_i(a^0, x^0) - U_i(a^1, a_{-i}^0, x^0) \geq \phi(a^0) - \phi(a^1). \]

Note that if \(U_i(a_i^0, a_{-i}^0, x^0) \leq U_i(a_i^1, a_{-i}^0, x^0)\) the associated resistance would be 0. We call such an edge “easy” if
\[ R([a^0, x^0] \to [a^1, x^1]) = \max\{\phi(a^0) - \phi(a^1), 0\}. \]

Furthermore, we also say that all edges of the form \([a^0, x^0] \to [a^0, x^1]\) are easy.

**Lemma 9.2** All edges of a tree with minimum stochastic potential must be easy.

**Proof:** We prove this lemma by contradiction. Suppose there exists a tree \(T\) rooted at a state \([a, x]\) of minimum stochastic potential which does not consist of only easy edges. Then there exist an edge in the tree \(T\) of the form \([a^0, x^0] \to [a^1, x^1]\), where \(a^1 = (a_i^1, a_{-i}^0)\) for some agent \(i \in N\) and action \(a_i^1 \neq a_i^0\), such that
\[ R([a^0, x^0] \to [a^1, x^1]) > \max\{\phi(a^0) - \phi(a^1), 0\}. \]

By Condition (iii) we know that there exists a state \(x' \in X(a^0)\) such that
\[ U_i(a_i^1, a_{-i}^0, x') - U_i(a_i^0, x') = \max\{\phi(a^0) - \phi(a^1), 0\}. \] (20)

Consider a sequence of feasible transitions, or path, of the form
\[ P := \{[a^0, x^0] \to a_i^0 \to [a^0, x'] \to [a^1, x''] \to a_i^1 \to [a^1, x^1]\}. \]

Note that both sequences of action invariant transitions, represented by “\(\to a_i^0\)” and “\(\to a_i^1\)”, are feasible because of Condition (i) and have a total resistance of 0. The intermediate transition \([a^0, x'] \to [a^1, x'']\) has a resistance \(= \max\{\phi(a^0) - \phi(a^1), 0\}\).

Let \(P' = \{[a^1, x^1] \to ... \to [a, x]\}\), be the unique sequence of transitions from \([a^1, x^1]\) to \([a, x]\) in the original tree \(T\). We now demonstrate two different approaches for augmenting the graph structure depending on whether there are any common edges between \(P\) and \(P'\).

**Case #1:** We start first with the easy case where there are no commonalities between the intermediate states in \(P\) and \(P'\); this is, if an edge \([\tilde{a}, \tilde{x}] \to [\cdot, \cdot]\) is in the path \(P\) then there is no edge leaving the state \([\tilde{a}, \tilde{x}]\), i.e., an edge of the form \([\tilde{a}, \tilde{x}] \to [\cdot, \cdot]\), in the path \(P'\). Consider a new tree \(T'\) rooted still at \([a, x]\) constructed from the original tree \(T\) as follows:

(i) Add the edges \(P\) to the tree \(T\). The new edges have a combined resistance \(= \max\{\phi(a^0) - \phi(a^1), 0\}\).

(ii) Remove the edge \([a^0, x^0] \to [a^1, x^1]\) in the original tree \(T\). This edge has a resistance \(> \max\{\phi(a^0) - \phi(a^1), 0\}\).
(iii) Remove the redundant edges from tree $T$. That is, if we added the edge $[\tilde{a}, \tilde{x}] \rightarrow [...]$ in step (i) then remove the edge exiting the state $[\tilde{a}, \tilde{x}]$ in the original tree $T$. The total resistance of these edges have a resistance $\geq 0$.

Note that $T'$ is also a valid tree rooted at $[a, x]$. For example, any state $[\tilde{a}, \tilde{x}] \in \mathcal{P}$ has a unique path to $[a^1, x^1]$ along $\mathcal{P}$ and then a unique path from $[a^1, x^1]$ to $[a, x]$ along $\mathcal{P}'$. This is true because there are no commonalities between $\mathcal{P}$ and $\mathcal{P}'$; hence, $\mathcal{P}'$ still exists in the new tree $T'$. Since all edges leaving states not in $\mathcal{P}$ are unchanged given the graph augmentation above, each state still possesses a unique path either to $\mathcal{P}$ or directly to $[a, x]$. Hence, $T'$ is a valid tree rooted at $[a, x]$. The resistance of the tree $T'$, denoted by $R(T')$, is

$$R(T') \leq R(T) + R([a^0, x^0] \rightarrow [a^1, x^1]) - R([a^0, x^0] \rightarrow [a^1, x^1]) < R(T).$$

Hence, the original tree $T$ does not have minimum stochastic potential.

**Case #2:** Now, consider the case where there are commonalities between the intermediate states in $\mathcal{P}$ and $\mathcal{P}'$, i.e., there exists a state $[\tilde{a}, \tilde{x}]$ such that $[\tilde{a}, \tilde{x}] \rightarrow [\tilde{a}', \tilde{x}']$ is an edge in $\mathcal{P}$ and $[\tilde{a}, \tilde{x}] \rightarrow [\tilde{a}'', \tilde{x}'']$ is an edge in $\mathcal{P}'$. In this case, the tree construction depicted above need not lead to a valid tree rooted at $[a, x]$ because of the removal of these “common” edges. To rectify this situation, let $\mathcal{P}$ be the complete set of transitions in $\mathcal{P}$ such that there are no conflicts of this sort. Construct the new tree $T''$ in the same fashion depicted above with the sole difference being in step (i) where we now add the edges $\bar{\mathcal{P}}$, as opposed to $\mathcal{P}$, to the tree $T$. Note that $T''$ is also a valid tree rooted at $[a, x]$. For example, any state $[\tilde{a}, \tilde{x}] \in \mathcal{P}$ has a unique path to the root $[a, x]$ either through $[a^1, x^1]$ and the path $\mathcal{P}'$ or a truncated version of the path $\mathcal{P}$ and the original tree $T$. Since all edges leaving states not in $\mathcal{P}$ are unchanged given the graph augmentation above, each state still possesses a unique path either to $\mathcal{P}$ or directly to $[a, x]$. Hence, $T''$ is a valid tree rooted at $[a, x]$. The resistance of the tree $T''$ also satisfies

$$R(T'') < R(T).$$

Hence, the original tree $T$ does not have minimum stochastic potential providing the contradiction. It is straightforward to repeat this argument to show that the edges of the form $[a^0, x^0] \rightarrow [a^0, x^1]$ must also be easy, i.e., have resistance 0.

We next utilize the fact that all edges in a tree with minimum stochastic potential must be easy to derive a relationship between the stochastic potential of different states.

**Lemma 9.3** Let $T_0$ be a tree of minimum stochastic potential rooted at the state $[a^0, x^0]$ and $T_1$ be a tree of minimum stochastic potential rooted at the state $[a^1, x^1]$ where $a^i = (a^i_1, a^0_{-i})$ for some agent $i \in N$ and action $a^0_i \neq a^1_i$. Suppose there exists a set of edges in tree $T_0$ of the form

$$[a^1, x^1] \rightarrow a^i_1 \rightarrow [a^1, x'] \rightarrow [a^0, x''] \rightarrow a^0_0 \rightarrow [a^0, x^0]$$

and a set of edges in tree $T_1$ of the form

$$[a^0, x^0] \rightarrow a^0_0 \rightarrow [a^0, x'''] \rightarrow [a^1, x'''] \rightarrow a^1_1 \rightarrow [a^1, x^1].$$

for some states $x', x'', x''' \in X$. Then, the stochastic potential of the two states is related as follows

$$\gamma([a^0, x^0]) - \gamma([a^1, x^1]) = \phi(a^1) - \phi(a^0).$$
**Proof:** Consider a new tree $T_{0\rightarrow 1}$ rooted at $[a^1, x^1]$ constructed from the original tree $T_0$ as follows:

(i) Consider an easy path $P$ of the form

$$[a^0, x^0] \xrightarrow{a^0} [a^0, \tilde{x}] \xrightarrow{a^1} [a^1, \tilde{x}'] \xrightarrow{a^1} [a^1, x^1]$$

Add the edges of $P$ to the tree $T_0$. These new edges have a combined resistance equal to $\max\{\phi(a^0) - \phi(a^1), 0\}$.

(ii) Remove the edge $[a^1, x'] \xrightarrow{a^1} [a^0, x'']$ in the original tree $T_0$. This edge has a resistance $\max\{\phi(a^1) - \phi(a^0), 0\}$.

(iii) Consider an easy path $P'$ of the form

$$[a^1, x'] \xrightarrow{a^1} [a^1, x^1]$$

Add the edges of $P'$ to the tree $T_0$. These new edges have a combined resistance equal to 0.

(iv) Remove the edge leaving $[a^1, x^1]$ in the original tree $T_0$. This edge has a resistance $\geq 0$.

(v) Remove the other redundant edges in the original tree $T_0$. These edges have a total resistance $\geq 0$.

Note that $T_{0\rightarrow 1}$ is a tree rooted at $[a^1, x^1]$ since there is a unique path from any state to this new root. To see this consider any state $[\tilde{a}, \tilde{x}]$ in the first set of action invariant transitions $\xrightarrow{a^1}$ in the path (21). Then, there is a unique path to the root $[a^1, x^1]$ through the action state pair $[a^1, x^1]$ and the path $P'$. Consider any state $[\tilde{a}, \tilde{x}]$ in the second set of action invariant transitions $\xrightarrow{a^0}$ in the path (21). Then, there is a unique path to the root $[a^1, x^1]$ through the action state pair $[a^0, x^0]$ and the path $P$. This property trivially holds for all alternative states since all edges leaving states not in $P$ or $P'$ are unchanged given the graph augmentation above.

The stochastic potential of $[a^1, x^1]$ is bounded above by

$$\gamma([a^1, x^1]) \leq R(T_{0\rightarrow 1}) = \gamma([a^0, x^0]) + \phi(a^0) - \phi(a^1)$$

Repeating the above analysis starting from the tree $T_1$ yields

$$\gamma([a^0, x^0]) \leq \gamma([a^1, x^1]) + \phi(a^1) - \phi(a^0).$$

Which gives us

$$\gamma([a^0, x^0]) - \gamma([a^1, x^1]) = \phi(a^1) - \phi(a^0).$$

\[\square\]

Before completing the proof of Theorem 5.1, we introduce the following notation. We denote a path of easy transition as highlighted in Lemma 9.3 compactly as

$$[a^0, x^0] \xrightarrow{\text{easy}} [a^1, x^1].$$

We will now use the previous lemma to finish the proof.

**Lemma 9.4** An action state pair $[a^*, x^*]$ has minimum stochastic potential over all states if and only if the action profile $a^*$ maximizes the potential function $\phi$.
Proof: Suppose that an action state pair \([a^0, x^0]\) has minimum stochastic potential over all states and \(\phi(a^0) < \phi(a^*)\) for some action profile \(a^* \in A\). Let \(T_0\) be a minimum resistance tree rooted at \([a^0, x^0]\). In the tree \(T_0\), there exists a series of easy transition paths of the form
\[
[a^*, x^*] = [a^m, x^m] \xrightarrow{\text{easy}} [a^{m-1}, x^{m-1}] \xrightarrow{\text{easy}} ... \xrightarrow{\text{easy}} [a^1, x^1] \xrightarrow{\text{easy}} [a^0, x^0]
\]
for some states \(x^1, x^2, ..., x^m \in X\) where each transition is highlighted by a unilateral deviation, i.e., for each \(k \in \{1, 2, ..., m\}\) the action profile \(a^k = (a^k_i, a^k_{i-1})\) for some agent \(i \in N\) with action \(a^k_i \neq a^{k-1}_i\).

Construct a new tree \(T_1\), from the original tree \(T_0\), rooted at \([a^1, x^1]\) according to the procedure highlighted in Lemma 9.3. The new tree has resistance
\[
R(T_1) = R(T_0) + \phi(a^0) - \phi(a^1).
\]

Construct a new tree \(T_2\), from the tree \(T_1\), rooted at \([a^2, x^2]\) according to the procedure highlighted in Lemma 9.3. The new tree has resistance
\[
R(T_2) = R(T_1) + \phi(a^1) - \phi(a^2) = R(T_1) + \phi(a^0) - \phi(a^2).
\]

Repeat this process \(m\) times to construct a tree \(T_m\) rooted at \([a^m, x^m]\). The new tree has resistance
\[
R(T_m) = R(T_0) + \phi(a^0) - \phi(a^*) < R(T_0).
\]

Therefore, the action state pair \([a^0, x^0]\) does not have minimum stochastic potential over all states and hence the contradiction. The same analysis can be repeated to show that all action state pairs \([a^*, x]\) where \(a^* \in \arg \max_{a \in A} \phi(a)\) and \(x \in X(a^*)\) are stochastically stable.

\(\square\)