Who to Blame? Learning and Control Strategies with Information Asymmetry

Changliu Liu\textsuperscript{1}, Wenlong Zhang\textsuperscript{2} and Masayoshi Tomizuka\textsuperscript{1}

Abstract—The rise of robot-robot interactions (RRI) is pushing for novel controller design techniques. Instead of using fixed control laws, robots should choose actions to minimize some cost functions specified by the designer. However, since the cost function of one robot may not be known to other robots (information asymmetry), special reasoning strategies are needed for multiple robots to learn to cooperate. Analysis shows that conventional learning and control strategies can lead to instability in a multi-agent system since the imperfection of other agents is not considered. In this paper, a new learning and control strategy that deals with interactions among imperfect agents is proposed. Analysis and simulation results show that the proposed strategy improves the performance of the system.

I. INTRODUCTION

In the robotics community, the idea of multi-robot teams has received much attention due to its wide applications in search and rescue, intelligent manufacturing, and unknown environment exploration. As a consequence, more and more robot-robot interactions (RRI) and human-robot interactions (HRI) are anticipated to take place, which requires advanced control of the robots.

This problem can be analyzed in the framework of multi-agent systems (MAS), where all intelligent entities (humans and robots) are viewed as agents. Conventionally, distributed control laws are designed for each robot agent \cite{Wang2020}. This strategy usually works well on problems with predefined environment and interaction patterns, but has limited extendability, as it is hard to design a universal control law that leads to desired results in many cases, especially when the system topology is time-varying (interactions are happening among different agents at different time).

Another popular approach is to design a cost function and a corresponding reasoning strategy (how to minimize cost under uncertainties) for each agent and let the behavior evolve during interactions \cite{Shamma2017}, so that the multi-agent system can be self-organized \cite{Zhang2018}. The cost function determines the behavior of each agent, which can be understood as the agent’s character. In certain cases, the cost function of an agent may not be clear to other agents, which is natural during HRI. This scenario is called information asymmetry \cite{Zhang2019}. The problem of interest is how to design the reasoning strategy so that the desired behavior will evolve even in the case of information asymmetry.

The reasoning strategy for an agent can be divided into the learning strategy (estimating the behavior of other agents) and the control strategy (minimizing its own cost based on the estimates). From the agent perspective, most learning and control strategies follow a centralized design philosophy, such as reinforcement learning \cite{Sutton2018} and adaptive control \cite{Liu2019}, which adjust agent’s estimation if its observation deviates from its prediction. However, in MAS, where each agent acts on its estimation of the unknowns, the gap between an agent’s observation and prediction can be caused by the following two factors:

1) the agent’s wrong estimation (wrong prediction);
2) other agents’ wrong estimations and subsequent improper actions (deviated observations).

Since the centralized design strategies (called the ‘Blame-Me’ strategy) do not address the second factor, they are likely to produce instability in MAS as will be shown in this paper.

To deal with this problem, a new ‘Blame-All’ strategy that considers both factors will be proposed.

Given the reasoning strategies, it is important to determine whether the agents can cooperate. To answer the question, the closed loop response of the MAS needs to be analyzed, especially the response in the equilibrium where single agent cannot change its control law for lower cost. For example, in the Bayesian game theory, a Bayesian Nash Equilibrium (BNE) \cite{Shamma2018} is defined as a profile of control laws that minimizes the expected cost for each agent given its belief about others’ costs. However, since the learning process is not considered, BNE is not suitable to analyze dynamic systems with information asymmetry. To deal with this problem, the Trapped Equilibrium (TE) is introduced in this paper to characterize the control laws (as well as the resulting time-invariant closed loop system) when all learning processes have converged (but may or may not converge to the true values). The aforementioned question will be answered regarding the TE, e.g. whether the cooperative goal can be achieved in the TE.

This paper discusses the closed loop performance of MAS in the TE under different learning and control strategies in a dynamic simultaneous game, and proposes the Blame-All strategy to improve the closed loop performance. The remainder of this paper is organized as follows: the mathematical problem is formulated in section III; section IV sets up an analytical framework for quadratic games; section V shows the drawbacks of the Blame-Me strategy; the Blame-All strategy is discussed in section VI; section VII gives an illustrating example; and section VIII concludes the paper.

\textsuperscript{1}C. Liu and M. Tomizuka are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA (e-mail: changliu1iu@berkeley.edu).

\textsuperscript{2}W. Zhang is with the Polytechnic School, Ira A. Fulton Schools of Engineering, Arizona State University, Mesa, AZ, 85212 USA (email: Wenlong.Zhang@asu.edu).
The observed state and control inputs; 
(b) Time step

\[ \hat{\theta}_i, \tilde{\theta}_i \]

denotes the inputs of all other agents except the agent 
affine with respect to 
\[ u \]
is the Nash Equilibrium. The resulting control laws 
The intersection of the response curves among all agents 
is the solution of the optimization problem in (2) is indeed 
B. The Simultaneous Dynamic Game

\[ x(k + 1) = f(x(k)) + \sum_{i=1}^N h_i(x(k))u_i(k) \]  
(1)

Assume there is no direct communication among the agents. Then for any agent \( i \), it can only choose its control \( u_i \) based on the observations (the current state and the previous inputs of other agents) and its reference goal \( G_i \subset \mathbb{R}^n \) by minimizing a cost function \( J_i(x, u_i, G_i) \), e.g. 
\[ u_i = \arg \min J_i(x, u_i, G_i) \]  
(2)

For simplicity, it is assumed that \( G_i = G \) for all \( i \).

B. The Simultaneous Dynamic Game

Since all agents’ inputs can affect the system state \( x \), the solution of the optimization problem in (2) is indeed a response curve, e.g. \( u_i = g_i(x, G_i, u_{\neg i}) \) where \( u_{\neg i} \) denotes the inputs of all other agents except the agent \( i \).

The intersection of the response curves among all agents is the Nash Equilibrium. The resulting control laws \( \{u_i^o\}_i \) are optimal in the sense that an agent cannot get a lower cost by deviating from this control law alone. Figure 1 illustrates the NE in the \( (u_1, u_2) \) function space in a two-agent case. However, the NE is only defined for systems with complete information, e.g. the cost functions \( J_i \)’s or the response curves \( g_i \)’s are globally known. When those are not globally known, the system is with incomplete information. The scenario that the cost functions known to different agents are not identical is called information asymmetry. In this paper, the case that an agent only knows its own cost function is considered, in which case the agents can only act on their estimates of other’s response curves.

Figure 2 illustrates the reasoning process under the adaptive algorithm (or the Blame-Me strategy). In Fig.2a, agent \( i \) \((i = 1, 2)\) chooses its action \( u_i \) in the intersection of its response curve (the solid curve) and the estimated response curve of the other agent (the dashed curve of the same color). However, the action pair \( u_1, u_2 \) in the yellow dot deviates from the agents’ predictions of what the other would do, which provides incentives for the agents to update their estimations. From agent 1’s perspective, since the deviation is only attributed to its wrong estimation, the updated estimate of agent 2’s response curve should go through the observed action pair \( u_1, u_2 \) in Fig.2a. A new control law can be chosen based on the new estimation as shown in Fig.2b. Agent 2 goes through the same reasoning. The implicit assumption under this strategy is that other agents are optimal given the observed data, which is a common assumption in data-driven estimation of cost functions [8]. However, this strategy is not ideal as it results in ‘overreaction’ as the new action pair moves farther away from the NE in Fig.2b.
In the Blame-All strategy, it is assumed that other agents are only optimal given their estimates. To predict what others would do, how they are estimating ‘me’ should also be considered. For example, in Fig.3a, the response curve of agent 1 is estimated by both agents (shown as the red and black dashed lines and denoted as Agent 1’s ideal response curve). If the two agents follow the same initialization rule, they will have consensus on the estimates of the response curves. The ideal response curves define a virtual NE (the light blue dot). An agent’s best action is to choose the corresponding action on its response curve, assuming others choosing actions in the virtual NE. Since the virtual NE is not the true NE, the observed action pair is still deviating from the predicted. From agent 1’s perspective, the deviation is caused both by its wrong estimation and the agent 2’s improper action. Hence the estimate of agent 2’s response curve is updated only to compensate the gap (between the virtual NE and the action pair) in the u2 direction, while the gap in the u1 direction is compensated by updating agent 1’s ideal response curve as shown in Fig.3b. In this way, the true response curves are found and the NE is achieved. 

C. Evaluation of the Cooperation: the Trapped Equilibrium

Following the above procedure, the update of the control law can be continued. The problems of interest are: under different strategies, whether the parameter estimates converge to true values and what is the resulting closed loop system. 

The Trapped Equilibrium is defined as the time-invariant closed loop system where all agents do not have incentives to change their control laws, as their learning processes converge. In Fig.2 and Fig.3, being in the TE means that the yellow dot remains in the same location under the specific reasoning strategy. Under the TE, the closed loop stability around the goal point will be analyzed to answer the question whether the agents can cooperate.

IV. QUADRATIC GAMES

In this section, the scope is narrowed down to quadratic games, which rise naturally in multi-robot cooperations [9].

A. The Model and Assumptions

Let the system goal G be driving the state x to the target state in the origin. Consider a quadratic game, i.e.

$$ J_i(x(k)) = x^T(k+1)Px(k+1) + u_i^T(k)u_i(k)\theta_i, \forall i $$

where $\theta_i \in \mathbb{R}$ encodes the preference of agent $i$ which is not known to others, and $P \in \mathbb{R}^{n \times n}$ is positive definite, which is chosen such that the closed loop system in the Nash Equilibrium is globally asymptotically stable around the origin. Define $R = \text{diag}(\theta_1I_{r_1}, ..., \theta_NI_{r_N}) \in \mathbb{R}^{r \times r}$. Let $B_i = h_i(x(k))$ be constant and $B = [B_1 \cdots B_N] \in \mathbb{R}^{n \times r}$. Denote the matrices that contain the diagonal and the off-diagonal entries of $R + B^TBP$ as $H_1$ and $H_2$ respectively. It is assumed that $H_1, H_2$ are invertible. If $H_1$ is singular, then some $J_i$ does not depend on some entries in $u_i$; if $H_2$ is singular, then there are no cross coupling terms among some control pairs.

B. The Optimal Control Law

For any agent $i$, the response curve $\partial J_i/\partial u_i = 0$ with respect to the cost function (3) is

$$ \theta_i u_i(k) = -B_i^T P x(k+1) $$

where others’ inputs are implicitly contained in $x(k+1)$. By (1) and using estimates for other agents’ inputs,

$$ u_i(k) = -[\theta_iI_r + B_i^T P B_i]^{-1} B_i^T P f(x(k)) + \sum_{j \neq i} B_j \hat{u}_j(k) $$

(5)

Hence the optimal control law is a linear combination of a state feedback control law and a predictive control law (where the actions of other agents are predicted). The predictive control law varies under different strategies, which will be discussed in the following two sections.

C. The Benchmark System in the Nash Equilibrium

In the Nash Equilibrium, agents have correct predictions, i.e. $\hat{u}_j(k) = u_j(k)$. Stacking the control law (5) for all agents and moving the inverse terms to the left, then $H_1U(k) = -B^T P f(x(k)) - H_2U(k)$. Hence the saddle solution with respect to (3) is:

$$ U^n(k) = [u_1^n(k) \cdots u_N^n(k)] = -[R + B^T P B]^{-1} B^T P f(x(k)) $$

(6)

which coincides with the solution to a centralized cost function, e.g. $J(x(k)) = x^T(k+1)Px(k+1) + U^T(k)RU(k)$.

Denote the system’s feedback gain as $K = [R + B^T P B]^{-1} B^T P$, the agent i’s feedback gain as $K_i = T_i K$ where $T_i = [0 \cdots 0, I_r, 0 \cdots 0] \in \mathbb{R}^{r \times n}$. Hence the optimal control law for agent $i$ under complete information is $u_i^c(k) = -K_i f(x(k))$. Let $y_{ij} \in \mathbb{R}^{T_i \times r_j}$ be the block entries of $[R + B^T P B]^{-1} B^T P$, which depends on $\theta_1, \cdots, \theta_N$. $y_{ij}$ encodes interactions among agents, as the control strategy of agent $i$ depends on parameters of agent $j$, i.e. $K_i = \sum_j y_{ij} B_j^T P$.

The closed loop system in the Nash Equilibrium is

$$ x(k+1) = [I - BK] f(x(k)) $$

(7)

V. THE PROBLEM WITH THE ADAPTIVE CONTROLLER OR THE BLAME-ME STRATEGY

A. The Adaptive Control Algorithm

In the case that $\theta_j$ is only known to agent $j$, according to the control law (6), the adaptive control law of agent $i$ can be written as

$$ u_i(k) = -T_i[\hat{R}^{(i)} + B^T P B]^{-1} B^T P f(x(k)) $$

(8)

where $\hat{R}^{(i)} = \text{diag}(\hat{\theta}_1^{(i)} I_{r_1}, \cdots, \hat{\theta}_N^{(i)} I_{r_N})$ contains the estimated parameters. Agent i’s learning objectives are other agents’ response curves (4), e.g. the solid curves in Fig.1. Since $\theta_j$ is a scalar, it is intuitive to do the parameter estimation using the following equation with reduced order:

$$ \theta_j \hat{u}_j(k) = -u_j^T(k) B_j^T P x(k+1) $$

(9)
The parameter \( \theta_j \) can be learned by agent \( i \) using the recursive least square (RLS) method [6], i.e.

\[
\begin{align*}
\hat{\theta}_j^{(i)}(k + 1) &= \hat{\theta}_j^{(i)}(k) + c_j^{(i)}(k + 1)u_j^{(i)}(k)F(k) \\
\hat{e}_j^{(i)}(k + 1) &= -u_j^{(i)}(k)B_j^TP\hat{x}(k + 1) - \hat{\theta}_j^{(i)}(k)u_j^{(i)}(k)
\end{align*}
\]

where \( F(k) \in \mathbb{R}^+ \) is the learning gain. For simplicity, \( F(k) \) is set to be \( (\sum u_j^{(i)}(k)^2)^{-\frac{1}{2}} \) if \( u_j^{(i)}(k) \neq 0 \) and 0 otherwise. Note that \( u_j^{(i)}(k) \) is known to agent \( i \) at \( k + 1 \).

B. The Closed Loop System

**Proposition 1 (Multiplicative uncertainty).** The closed loop dynamics of the system when all agents are using the control law (8) follow from

\[
\begin{align*}
\hat{U}(k) &= (I + \Delta(k))U^o(k) \\
\hat{x}(k + 1) &= [I - B[I + \Delta(k)]K]f(\hat{x}(k))
\end{align*}
\]

where \( \Delta(k) \in \mathbb{R}^{r \times r} \), whose diagonal entries are all zero.

**Proof.** Notice that

\[
\begin{align*}
\hat{x}(k) - x(k) &= -(\hat{\theta}(k) - \theta(k)) + \hat{e}(k)
\end{align*}
\]

Stuck the equation for all agents by

\[
\begin{align*}
\Delta(k) = - \begin{bmatrix}
0 & \hat{e}_1(k)^2 & \cdots & \hat{e}_1(k)^2 \\
\hat{e}_2(k)^2 & 0 & \cdots & \hat{e}_2(k)^2 \\
\vdots & \vdots & \ddots & \vdots \\
\hat{e}_N(k)^2 & \hat{e}_N(k)^2 & \cdots & 0
\end{bmatrix}
\end{align*}
\]

Hence (11) and (12) hold.

**Remark 1.** (12) shows that the uncertainty introduced by information asymmetry is multiplicative and (14) indicates the structure of the uncertainty in the Blame-All strategy.

C. The Trapped Equilibrium

Note that \( \hat{\theta}_j^{(i)}(k) = \hat{\theta}_j^{(m)}(k), \forall i, m \neq j \) if both agent \( i \) and agent \( m \) are using the same learning algorithm with the same initial conditions. Let \( \hat{\theta}_j(k) = \hat{\theta}_j^{(i)}(k), \forall i \neq j \). Define \( \hat{R}(k) = \text{diag}(\hat{\theta}_1(k)I_{n_1}, \ldots, \hat{\theta}_N(k)I_{n_N}) \).

**Proposition 2 (Convergence of the learning algorithm).** In system (12) with uncertainty (14), for all initial conditions, the learning algorithm (10) converges, i.e. \( \lim_{k \to \infty} \hat{R}(k) = R^e \) and consequently \( \lim_{k \to \infty} \Delta(k) = \Delta^e \), if

\[
(\Delta^e + I)^T[\hat{R} + B^TP\hat{B}]{\Delta^e} + \hat{R}^e] = 0
\]

**Proof.** When \( \hat{u}_j(k) = 0, \hat{\theta}_j(k + 1) = \hat{\theta}_j(k) \) is trivial. Consider the case \( \hat{u}_j(k) \neq 0 \). Note that (4) holds only for \( x = x^e \) and \( \hat{u}_j = u_j^e \), e.g. \( \hat{\theta}_j(\hat{u}_j(k) - \delta u_j(k)) = -B_j^TP(\hat{x}(k + 1) - \hat{B}u(k)) \). Hence

\[
\hat{\theta}_j \hat{u}_j(k) + B_j^TP\hat{x}(k + 1) = (\hat{\theta}_j T_j + B_j^TPB)\delta u(k)
\]

Thus the estimate of the reaction curve is biased, so is \( \hat{\theta}_j \). Multiply \( \hat{u}_j^e(k) \) on both sides, then \( \hat{\theta}_j(k + 1)\hat{u}_j^e(k)u_j^{(i)}(k) = \hat{u}_j^e(k)T_j[I + \Delta(k)]U^o(k) \). Since \( \hat{u}_j(k) = T_j[I + \Delta(k)]U^o(k) \), the relationship between the estimation error \( \hat{\theta}_j(k + 1) \) and the uncertainty \( \Delta(k) \) follows from

\[
\begin{align*}
-\hat{\theta}_j(k + 1) &= \frac{(\Delta^e + I)^T[\hat{R} + B^TP\hat{B}]{\Delta^e} + \hat{R}^e}{(U^o(k))^T(I + \Delta(k))U^o(k)}
\end{align*}
\]

For all initial conditions, the learning algorithm converges if \( (I + \Delta^e)^T[\hat{R} + B^TP\hat{B}]{\Delta^e} = -\hat{\theta}_j(I + \Delta^e)^T[\hat{R} + B^TP\hat{B}] \), which can be rearranged as:

\[
(\Delta^e + I)^T[\hat{R} + B^TP\hat{B}]^{-1}[\hat{R}^e] = 0
\]

Stacking the equations for all \( j, \) (15) holds.

**Remark 2.** (14) and (15) determine the TE of the MAS, e.g. \( \Delta^e = \hat{R}^e = 0 \), or \( \Delta^e = [R^e + B^TP\hat{B}]^{-1}\hat{R}^e \). The first TE is efficient as it is identical with the NE. When \( \Delta^e \neq 0 \), the cooperation under the second TE is inefficient.

**Remark 3 (Unstable modes).** When \( N = 2 \), suppose \( B_1^TPB_2 = c_jI \) for some scalar \( c_j \). Then \( \hat{R}^e = -H_1 + R, \Delta^e = H_2^{-1}H_1 \) satisfies (14) and (15), hence defines a TE. Moreover, \( (I + \Delta^e)K = H_2^{-1}(H_1 + H_2)[H_1 + H_2]^{-1}B^TP = H_2^{-1}B^TP \). By (12), the closed loop dynamics is

\[
\hat{x}(k + 1) = [I - BH_2^{-1}B^TP]f(\hat{x}(k))
\]

which can be unstable when \( ||H_2|| \to 0 \).

VI. THE BLAME-ALL STRATEGY

A. The Algorithm for the Agent

To overcome the instability in the Blame-Me strategy, the Blame-All strategy is proposed. Putting (8) in the form of (5), it is clear that the estimate of \( u_i(k) \) in the Blame-All strategy is equivalent to \( \hat{u}_j^{(i)}(k) = -[\hat{\theta}_j^{(i)} + B_j^TPB_j]^{-1}[f(\hat{x}(k)) + \sum_{m \neq j} B_m \hat{u}_m^{(i)}(k) + B_i u_i(k)] \), which is biased since agent \( j \) does not know \( u_i(k) \) in advance. The term \( u_i(k) \) should also be estimated, e.g.

\[
\hat{u}_j^{(i)}(k) = -[\hat{\theta}_j^{(i)} + B_j^TPB_j]^{-1}[f(\hat{x}(k)) + \sum_{m \neq j} B_m \hat{u}_m^{(i)}(k)]
\]

where the estimate \( \hat{u}_i^{(i)}(k) \) should be calculated using the same algorithm in estimating other agents’ behavior as in (20). Following the procedure in obtaining (6) from (5), (20) can be solved for all \( j = 1, \ldots, N \), e.g.

\[
\hat{u}_j^{(i)}(k) = -T_j[\hat{R}(k) + B^TP\hat{B}]^{-1}B^TPf(\hat{x}(k))
\]

where \( \hat{R}(i) = \text{diag}(\hat{\theta}_1^{(i)}I_{r_1}, \ldots, \hat{\theta}_i^{(i)}I_{r_i}, \ldots, \hat{\theta}_N^{(i)}I_{r_N}) \). Specifically, \( \hat{\theta}_i^{(i)} \) is the estimate of \( \theta_i^{(i)} \) for \( j \neq i \) made by agent \( i \) to compensate for others’ wrong estimations of \( \theta_i \). When all agents are using the same algorithm with the same initial condition, they should agree on their estimates, e.g.

\[
\hat{\theta}_i^{(i)}(k) = \hat{R}(j) = \hat{R}_i
\]

Then by (21), \( \hat{u}_j^{(i)}(k) \) does not depend on \( i \), e.g. \( \hat{u}_j^{(i)}(k) = \hat{u}_j(k), \forall i \). Stack all \( \hat{u}_j(k) \).

\[
\hat{U}(k) = -[\hat{R}(k) + B^TP\hat{B}]^{-1}B^TPf(\hat{x}(k))
\]
\( \hat{U}(k) \) is the virtual NE (the light blue dot in Fig.3) that is consensual among all agents. Plugging (22) into (5), agent \( i \)'s control law becomes
\[
\text{Proposition 3 (Multiplicative uncertainty). The closed loop dynamics of the system when all agents are using control law (23) follow from (11) and (12) where}
\[
\Delta(k) = H_1^{-1}H_2\hat{R}(k) + B^TPB^{-1}\hat{R}(k)
\]
\[
\text{Proof. Subtracting } \hat{u}(k) \text{ from (23), } u(k) - u_i(k) =
- \left[ \hat{\theta}_i + B_i^TPB_i \right]^{-1}B_i^TP\left[ (x(k)) + (B - B_iT_i)\hat{U}(k) \right] =
- T_iH_1^{-1}H_2\hat{U}(k), \text{ where } \hat{U}(k) = \hat{U}(k) - U^{(0)(k)} =
- \left[ \hat{R}(k) + B^TPB^{-1}\hat{R}(k)U^0(k) \right]. \text{ Stack the equations for all agents, then } \hat{U}(k) - U^{(0)(k)} = \Delta(k)U^0(k) \text{ where } \Delta(k) \text{ follows from (26). Equations (11) and (12) hold.}
\]
\]

\[
\text{Proposition 4 (Convergence of the learning algorithm). In system (12) with uncertainty (26) and learning algorithm (23), } \hat{\theta}(k) \text{ is bounded } \forall j = 1,..N. \text{ It converges to } \theta_j \text{ in finite time steps if and only if the set } \{ u_j(k) = 0 \} \text{ does not contain any trajectory of the closed loop system.}
\]
\[
\text{Proof. When } u_j(k) \neq 0, \hat{e}_j(k+1) = -\hat{\theta}_j(k + 1) = -\hat{\theta}_j(k)\hat{u}_j(k)B_i^TP\left[ (x(k)) + (B - B_iT_i)\hat{U}(k) \right] =
- \left[ \hat{\theta}_j + B_i^TPB_i \right]^{-1}B_i^TP\left[ (x(k)) + (B - B_iT_i)\hat{U}(k) \right] =
- \left[ \hat{R}(k) + B^TPB^{-1}\hat{R}(k)U^0(k) \right]. \text{ Stack the equations for all agents, then } \hat{U}(k) - U^{0(k)} = \Delta(k)U^{(0)}(k) \text{ where } \Delta(k) \text{ follows from (26). Equations (11) and (12) hold.}
\]

Proposition 4 implies two TEs under the Blame-All strategy: 1) the parameter estimation converges in finite steps and the system converges to the benchmark system in (7); 2) the system is trapped in the set \( \{ u_j(k) = 0 \} \). To get rid of the second TE, agent \( j \) is allowed to choose a random control input when the optimal input in (23) is always zero, which is called a perturbation to the system.

\[
\text{Theorem 5 (System performance under the Blame-All strategy). For any } N, \text{ the closed loop system under the Blame-All strategy will converge to the benchmark system (7) in the Nash Equilibrium, if the agent } j \text{ is allowed to perturb the system when the system is trapped in } \{ u_j(k) = 0 \}, \text{ but } B_i^TP\hat{x}(k) \neq 0.
\]
\[
\text{Proof. The proof is in two steps. First, we will show that under certain initial conditions, the system will converge to the benchmark system in two time steps. Then we show that a perturbation to the system is equivalent to starting over with a new set of initial conditions. Hence the system will always converge to the benchmark system.}
\]

By Prop.4, if \( u_j(0) = -T_j(\Delta(0) + I)Kf(x(0)) = 0, \forall j \), the parameters converge at \( k = 1 \), i.e. \( \hat{R}(1) = R \). By (26), \( R(1) = R \implies \Delta(1) = 0 \), which further implies that \( \hat{U}(1) = U^{(0)}(1) \) and \( \hat{x}(2) = x^{(0)}(2) \) and \( \hat{R}(k) = R, \Delta(k) = 0 \) for all \( k \geq 2 \). Hence the system converges to the benchmark system in two time steps if \( T_j(\Delta(0) + I)Kf(x(0)) \neq 0, \forall j \).

If the system is trapped in the set \( \{ u_j(k) = 0 \} \), i.e. \( T_j(\Delta(k) + I) \equiv 0 \) for some \( j \). As \( B_i^TP\hat{x}(k) = 0 \), the pair \( \hat{u}_j(k + 1) = 0 \) and \( \hat{x}(k) \) is not optimal with respect to \( J_j \). Suppose agent \( j \) chooses a random input \( \hat{u}_j(k) \neq 0 \) at \( k \). At \( k + 1 \), the system starts over again with the new initial conditions: \( R(k + 1) = \hat{R}(k) \), and \( \hat{x}(k + 1) \). When the perturbation goes to the right direction, \( T_j(\Delta(k + 1) + I)Kf(x(k + 1)) \neq 0 \). Then the system will converge to the benchmark system in two steps.

Remark 4. The perturbation only works in the Blame-All case, since the convergence is only affected by the initial conditions (in contrast, in the Blame-Me case, the convergence highly depends on the parameter \( \theta_i \)'s, according to Prop.2). The improved performance in the Blame-All case is due to the introduction of the consensus \( \hat{U}(k) \) among all agents, which decouples learning and control strategies.
TE under the Blame-Me Strategy: When $\alpha \approx 0$, $B_i = h_i(x)$ is almost constant and $\theta_1^{(2)} = \theta_2^{(1)} = -2$ is a TE by Remark 3. The resulting closed loop matrix $I - B H_2^{-1} B^T P$ in (19) has an eigenvalue at 10 (the maximum singular value is 10.8), which causes instability as shown in Fig.5(b). Due to the instability, the target position was not reached. Figure 6 shows the simulation profile under this strategy and illustrates how the oscillation starts.

TE under the Blame-All Strategy: Figure 7 shows the system performance under the Blame-All strategy. The Blame-All strategy outperforms the Blame-Me strategy as the parameter estimation converges in two time steps and the state goes to zero asymptotically, which verifies Thm.5.

VIII. CONCLUSIONS

This paper investigated algorithm design techniques when information is asymmetric among agents. The Trapped Equilibrium was introduced to analyze the closed loop performance of the MAS under different strategies.

Quadratic simultaneous games were analyzed in the paper as they rise naturally from multi-robot cooperations. The conclusions are: 1) In a quadratic simultaneous game, the uncertainty introduced by asymmetric information is multiplicative (Prop.1 and Prop.3). 2) The Blame-Me strategy can cause instability in the quadratic simultaneous game (Remark.3). 3) Under the proposed Blame-All strategy, the only Trapped Equilibrium assures parameter convergence to true values and system convergence to the benchmark system in the Nash Equilibrium (Thm.5).

This paper can serve as a building block in analyzing interactions with even lesser shared information. In the future, the method will be tested in human robot cooperations.

REFERENCES


Fig. 6: The simulation profile under the Blame-Me strategy

Fig. 7: The simulation profile under the Blame-All strategy

VII. AN EXAMPLE

Consider the case where two mobile robots cooperate in moving a large object as shown in Fig.4a. Suppose there are rigid connections among the robots and the object. Define $x = [p^T, v^T, \alpha, \omega]^T \in \mathbb{R}^6$ where $p$ and $v$ are the position and velocity of the center of mass (which is assumed to be at the middle of the object), $\alpha$ is the orientation of the object and $\omega$ is the angular velocity around the center of mass. The inputs of the agents are the forces, e.g. $u_i = F_i \in \mathbb{R}^2$. Then the system dynamics is in the form of (1),

$$x(k+1) = \begin{bmatrix} p(k) + v(k)dt \\ v(k) - \frac{dt}{m} f r 1(v(k)) \\ \alpha(k) + \omega(k)dt \\ \omega(k) - \frac{dt}{l_2} f r 2(\omega(k)) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{dt}{m} I_2 & \frac{dt}{m} l_2 \\ 0 & \frac{dt}{M} I_1 \\ \frac{dt}{M} l_1 & \frac{dt}{M} l_2 \end{bmatrix} U(k)$$

(27)

where $m$ is the mass, $M$ is the inertia, $dt$ is the sampling time, $f r 1(\cdot)$ and $f r 2(\cdot)$ are the frictions, and $l_1$ and $l_2$ are the vectors pointing from the center of mass to the robot 1 and 2 respectively. The task for the robots is to move the object from $(4, 10, 0, 0, -\pi/2, 0)$ to $(0, 0, 0, 0, 0, 0)$ as shown in Fig.4b. The cost function is given by (3).

In the simulation, $dt = m = 0.2, M = 0.067, |l_1| = |l_2| = 1, f r 1(v) = 0.02v$ and $f r 2(\omega) = 0.0067\omega$. The parameters in the cost function are $\theta_1 = 0.3, \theta_2 = 0.9$ and $P = \text{diag} \begin{bmatrix} 4 & 0 & 0.3 & 0.1 \\ 0 & 4 & 0.1 & 0.3 \\ 0.3 & 0.1 & 1 & 0.1 \\ 0.1 & 0.3 & 0.1 & 1 \end{bmatrix}$

The agents initiate the learning process by setting the estimates to be 1. The simulated trajectories are shown in Fig.5 where (a) shows the trajectory under complete information, (b) and (c) show the trajectories under the two strategies when information is asymmetric.