Selective Negotiations for Scaling Stochastic Dynamic Games

Kamran Vakil and Alyssa Pierson

***Abstract*— This paper presents asocial agents for selective negotiations in stochastic dynamic games. Game-theoretic frameworks in the belief-space show promise in modeling complex interactions in scenarios such as surveillance, herding, and racing. Stochastic dynamic games can be solved as a continuous POMDP to find a local Nash Equilibrium solution of all agents using a game-theoretic belief-space variant of iLQG. However, the scalability of this method suffers due to the large dimensionality of beliefs which the iLQG must propagate, and fails to consider environmental features such as dynamic obstacles. We introduce asocial agents models, which follow fixed input trajectories in the planning horizon. Ego agents can selectively toggle which agents it considers asocial, thereby reducing computational overhead. Simulations demonstrate that our approach to selective negotiations can help scale stochastic dynamic games with faster computation times and minimal performance decline.**

I. INTRODUCTION

Trajectory planning for multiagent systems under uncertainty is a complex problem with extensive research. Recent work in multiagent systems focuses on developing planners which take into account both the interactivity between agents and the quality of information each agent possesses when planning trajectories. These planners have application in surveillance, pursuer-evader games, racing, and traffic negotiation. A recent approach combines game-theoretic decision making with belief-space planning [1]. Game-theoretic planning provides an easy way to model interactions and decision making between autonomous agents. Belief-space planning allows agents to address their quality of information. This method allows agents to take actions to gain information and use it to their advantage while fulfilling tasks. By formulating the problem as a game-theoretic Partially Observable Markov Decision Process (POMDP), a local Nash Equilibrium can be found using an iterative Linear Quadratic Gaussian (iLQG).

However, the large quantity of beliefs and interactivity of game-theoretic planning means scalability suffers as the number of agents increase. Depending on the scenario, modeling interactions between certain agents can yield non-influence, resulting in wasted computations. For example, a driver may only need to consider the behavior of cars which are nearby, as distant cars are not impacted by the driver's behavior. Ideally, the driver would selectively model the policies of agent's whose interactivity is important while partially ignoring others. We seek to extend prior work in [1] to allow for this selective planning and increase the scalability and complexity of games. Figure 1 illustrates a selective negotiation that changes over time.

Kamran Vakil and Alyssa Pierson are with the Department of Mechanical Engineering at Boston University, Boston, MA 02215, USA. [kamranv, pierson]@bu.edu

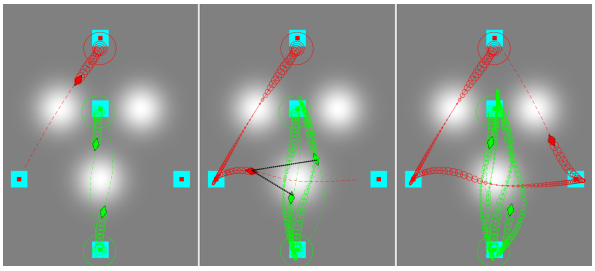

Fig. 1. Selective negotiations allow an agent to ignore asocial agents that have little impact on their actions. Our framework allows the ego agent to choose its most relevant neighbors, which may evolve over time. Running this specific game with selective negotiations for $N = 50$ trials results in a 38% decrease in computation time compared to full negotiation planning.

The main contributions of this paper are:

1) The introduction of "asocial" agents within games;
2) A selective negotiation framework for toggling when agents are social or asocial; and
3) Simulations that demonstrate reductions in computation effort with minimal performance impact.

A. Background and Related Work

Game-theoretic models excel in modeling social dilemmas, where agents must make decisions when their objectives are at odds. Common applications include modeling human interactions in driving and racing scenarios [2]–[4]. Approaches to solving the Nash Equilibria include Best Iterated Response [5], iterative quadratic approximations [6], and solving the necessary conditions [4]. [1] solves the Nash Equilibrium's necessary condition of a static quadratic game at each stage of a backwards pass in an iLQG.

Belief-Space Planning [7] uses distributions of the robot's state estimate to represent the robot's uncertainties. These distributions are called beliefs, and computing policies over the belief space can be described by a POMDP [8]. POMDPs are a commonly used framework for modeling real world processes under uncertainty. While solving POMDPs to global optimality is NP-Hard [9], optimization based approaches [10] [11] such as the iterative Linear Quadratic Gaussian(iLQG) [12] scale linearly in the planning horizon l, making it feasible for real world implementation.

To address complexity across large teams, some distributed algorithms will break a centralized problem into smaller sub-problems, reducing which agents coordinate with an ego agent [13] [14] [15]. These methods only consider the existence of agents in close proximity, ignoring all other agents. This limits the complexity of games which can be played and can lead to difficulty tracking agents that enter or exit the ego's proximity. Thus, we seek a method which can track all agents without modeling all of their interactions.

The remainder of this paper is organized as follows: Section II summarizes mathematical preliminaries, belief dynamics, and iLQG algorithm from [1]. Section III introduces the asocial agent, derives a linear feedback policy for mixed social/asocial agent modeling, and introduces the modified iLQG with selective negotiation. We show the utility of asocial agents and selective negotiation in Section IV.

II. Mathematical Preliminaries

In this section, we formulate the POMDP and stochastic system in the belief space. We then approximate the general Bayesian Filter update with an EKF to propagate our Gaussian beliefs. We then briefly reintroduce the iLQG algorithm for dynamic games with belief space planning from [1]. We first take inspiration from [16] and introduce a formal definition for all agents in [1] which contain the social ability to negotiate with other agents.

Definition 1 (Social Agent)**.** *An agent is considered social if it considers the interactions between itself and other agents when planning its inputs.*

Assumption 1 (Common Knowledge)**.** *Social agents have models for cost, dynamics, and observations of other agents.*

Assumption 2 (First Order Beliefs)**.** *All agents share the same beliefs about each other. An agent i's belief about an agent j and that agent j's belief about itself are the same.*

Later, Section III-B introduces our Asocial Agents, which do not consider the interactions of others and have a different Common Knowledge model.

A. POMDP Formulation

We start by defining POMDPs in their most general form following notation from [10]. The expected return of each individual agent under a control trajectory of all agents defined as u subject to uncertainty on the observed measurements z over the horizon l is determined by the action value function $Q^i(\mathrm{b}_0, \mathrm{u}) = \mathbb{E}\left[c^i_l(\mathrm{b}_l) + \sum_{k=0}^{l-1} c^i_k(\mathrm{b}_k, \mathrm{u}_k)\right]$, where c^i_k and c^i_l are the cost at time k and terminal cost of agent a^i and b is the belief about the expected state x of the system, in this case the state and covariance of the state. Since there exists an action value function for each agent, there are N distinct action-value functions for $i \in \{1, ..., N\}$.

Given an initial belief b_0 for agents a^i, $i = \{1, ..., N\}$, we seek to solve the stochastic optimal control problem

$$\begin{aligned} \pi^i = &\underset{\mathrm{u}^i}{\operatorname{argmin}}\, Q^i(\mathrm{b}_0, \mathrm{u}) \quad \forall i \in \{1, ..., N\}, \\ &s.t. \quad \mathrm{b}_{k+1} = \beta(\mathrm{b}_k, \mathrm{u}_k, \mathrm{z}_{k+1}). \end{aligned} \tag{1}$$

where β denotes the stochastic belief dynamics of b_k and π^i denotes the optimal policy of agent a^i. A general solution to (1) can be defined recursively by the Bellman equation [1]:

$$\begin{aligned} Q^i_k(\mathrm{b}_k, \mathrm{u}_k) &= c^i_k(\mathrm{b}_k, \mathrm{u}_k) + \mathop{\mathbb{E}}_{\mathrm{z}_{k+1}}\left[V^i_{k+1}(\beta(\mathrm{b}_k, \mathrm{u}_k, \mathrm{z}_{k+1},))\right], \\ V^i_k(\mathrm{b}_k) &= \min_{\mathrm{u}^i_k} Q^i_k(\mathrm{b}_k, \mathrm{u}_k), \quad V^i_l = c^i_l(\mathrm{b}_l), \\ \pi^i_k(\mathrm{b}_k) &= \underset{\mathrm{u}^i_k}{\operatorname{argmin}}\, Q^i_k(\mathrm{b}_k, \mathrm{u}_k), \end{aligned} \tag{2}$$

where $V^i_k(\mathrm{b}_k)$ is the value function and $\pi^i_k(\mathrm{b}_k)$ is the optimal policy at time step k.

B. Problem Formulation and Belief Dynamics

In order to find a solution to the continuous POMDP, we follow [1] and consider beliefs as Gaussian distributions. We approximate the belief dynamics through an EKF, and use a quadratic approximation of the value function about a nominal trajectory in the belief space. We use these to formulate an algorithm which iteratively computes a local Nash equilibrium over all agents through a belief space variant of iLQG which performs a Bellman backwards recursion.

We assume nonlinear stochastic dynamics and observation models for any single agent a^i as:

$$\mathrm{x}^i_{k+1} = f(\mathrm{x}^i_k, \mathrm{u}^i_k, \mathrm{m}^i_k), \quad \mathrm{m}^i_k \sim \mathcal{N}(0,\, I), \tag{3}$$

$$\mathrm{z}^i_k = h(\mathrm{x}^i_k, \mathrm{x}^{\neg i}_k, \mathrm{n}^i_k), \qquad \mathrm{n}^i_k \sim \mathcal{N}(0,\, I), \tag{4}$$

where m_k and n_k denote process and measurement noise whose distributions can be arbitrarily transformed inside the equations, and $\mathrm{x}^{\neg i}_k$ refers to the state of all other agents. We formulate the joint process and measurement functions of all agents a^i, $i = \{1, ..., N\}$ [1]

$$\begin{aligned} f(\mathrm{x}_k, \mathrm{u}_k, \mathrm{m}_k) = [&f^1(\mathrm{x}^1_k, \mathrm{u}^1_k, \mathrm{m}^1_k)^\top, \ldots, \\ &f^N(\mathrm{x}^N_k, \mathrm{u}^N_k, \mathrm{m}^N_k)^\top]^\top, \end{aligned} \tag{5}$$

$$h(\mathrm{x}_k, \mathrm{n}_k) = [h^1(\mathrm{x}^1_k, \mathrm{x}^{\neg 1}_k, \mathrm{n}^1_k)^\top, \ldots, h^N(\mathrm{x}^N_k, \mathrm{x}^{\neg N}_k, \mathrm{n}^N_k)^\top]^\top.$$

We define $\mathrm{b}_k = (\hat{\mathrm{x}}^\top_k, \Sigma_k)$ as the Gaussian belief, where mean state $\hat{\mathrm{x}}^\top_k$ and variance Σ_k describes the stochastic state $\mathrm{x}_k \sim \mathcal{N}(\hat{\mathrm{x}}^\top_k, \Sigma_k)$. We must now propagate these beliefs through a Bayesian Filter. We follow [10] and approximate the Bayesian filter as an EKF with standard EKF update equations to make the belief propagation tractable [1], $\hat{\mathrm{x}}_{k+1} = f(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0) + K_k(\hat{\mathrm{z}}_{k+1} - h(f(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0), 0))$, $\Sigma_{k+1} = \Gamma_{k+1} - K_k H_k \Gamma_{k+1}$, with corresponding matrices defined by $\Gamma_{k+1} = A_k \Sigma_k A^\top_k + M_k M^\top_k$, $K_k = \Gamma_{k+1} H^\top_k (H_k \Gamma_{k+1} H^\top_k + N_k N^\top_k)^{-1}$, $A_k = \frac{\partial f}{\partial \mathrm{x}}(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0)$, $M_k = \frac{\partial f}{\partial \mathrm{m}}(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0)$, $H_k = \frac{\partial h}{\partial \mathrm{x}}(f(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0), 0)$, $N_k = \frac{\partial f}{\partial \mathrm{n}}(f(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0), 0)$. We redefine $\mathrm{b}_k = [\hat{\mathrm{x}}^\top_k, \mathrm{vec}(\Sigma_k)^\top]^\top$, where $\mathrm{vec}(\Sigma_k)^\top$ is the matrix Σ_k reshaped into vector form. We define $\mathrm{s} = [\mathrm{b}^\top, \mathrm{u}^\top]^\top$ as shorthand for belief and controls. We formulate the stochastic belief dynamics as $\mathrm{b}_{k+1} = g(\mathrm{b}_k, \mathrm{u}_k) + W(\mathrm{b}_k, \mathrm{u}_k)\xi_k, \quad \xi_k \sim \mathcal{N}(0, I)$, where

$$g(\mathrm{b}_k, \mathrm{u}_k) = \begin{bmatrix} f(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0) \\ \mathrm{vec}(\Gamma_{k+1} - K_k H_k \Gamma_{k+1}) \end{bmatrix},$$

$$W(\mathrm{b}_k, \mathrm{u}_k) = \begin{bmatrix} \sqrt{K_k H_k \Gamma_{k+1}} \\ 0 \end{bmatrix},$$

where ξ_k is a Gaussian with dimension of state x that is applied to the stochastic part of b_k. ξ_k represents both process and measurement noise in the belief transition.

C. Algorithm for Dynamic Game Belief Space Planning

We formulate the iLQG with the Bellman equations and a quadratic approximation to obtain backwards pass equations from [1]. We start by defining the action value functions [1]

$$Q_k^i = c_k^i + V_{k+1}^i + \frac{1}{2}\sum_{j=1}^{n_x} W_k^{(j),\top} V_{\mathrm{bb},k+1}^i W_k^{(j)}, \tag{6}$$

$$Q_{\mathrm{s},k}^i = c_{\mathrm{s},k}^i + g_{\mathrm{s},k}^\top V_{\mathrm{b},k+1}^i + \sum_{j=1}^{n_x} W_{\mathrm{s},k}^{(j),\top} V_{\mathrm{bb},k+1}^i W_k^{(j)},$$

$$Q_{\mathrm{ss},k}^i = c_{\mathrm{ss},k}^i + g_{\mathrm{s},k}^\top V_{\mathrm{bb},k+1}^i g_{\mathrm{s},k} + \sum_{j=1}^{n_x} W_{\mathrm{s},k}^{(j),\top} V_{\mathrm{bb},k+1}^i W_{\mathrm{s},k}^{(j)},$$

where the subscripts b, s, bb, and ss denote gradients and Hessians, except for g_k and W_k where they denote Jacobians. Dropping the k for notation convenience, we recover other partial derivatives from (6),

$$Q_{\mathrm{s}}^i = \begin{bmatrix} Q_{\mathrm{b}}^i \\ Q_{\mathrm{u}^1}^i \\ \vdots \\ Q_{\mathrm{u}^N}^i \end{bmatrix} \quad Q_{\mathrm{ss}}^i = \begin{bmatrix} Q_{\mathrm{bb}}^i & Q_{\mathrm{bu}^1}^i & \cdots & Q_{\mathrm{bu}^N}^i \\ Q_{\mathrm{u}^1\mathrm{b}}^i & Q_{\mathrm{u}^1\mathrm{u}^1}^i & \cdots & Q_{\mathrm{u}^1\mathrm{u}^N}^i \\ \vdots & \vdots & \ddots & \vdots \\ Q_{\mathrm{u}^N\mathrm{b}}^i & Q_{\mathrm{u}^N\mathrm{u}^1}^i & \cdots & Q_{\mathrm{u}^N\mathrm{u}^N}^i \end{bmatrix}, \tag{7}$$

$$\hat{Q}_{\mathrm{uu}} = \begin{bmatrix} Q_{\mathrm{u}^1\mathrm{u}}^1 \\ Q_{\mathrm{u}^2\mathrm{u}}^2 \\ \vdots \\ Q_{\mathrm{u}^N\mathrm{u}}^N \end{bmatrix}, \quad \hat{Q}_{\mathrm{ub}} = \begin{bmatrix} Q_{\mathrm{u}^1\mathrm{b}}^1 \\ Q_{\mathrm{u}^2\mathrm{b}}^2 \\ \vdots \\ Q_{\mathrm{u}^N\mathrm{b}}^N \end{bmatrix}, \quad \hat{Q}_u = \begin{bmatrix} Q_{\mathrm{u}^1}^1 \\ Q_{\mathrm{u}^2}^2 \\ \vdots \\ Q_{\mathrm{u}^N}^N \end{bmatrix}, \tag{8}$$

and define our linear feedback policy to be

$$\pi_k = \bar{\mathrm{u}}_k + j_k + K_k \delta \mathrm{b}_k, \tag{9}$$

where $\bar{\mathrm{u}}$ is the nominal input of the agent, $j_k = -\hat{Q}_{\mathrm{uu}}^{-1}\hat{Q}_{\mathrm{u}}$ is the feedforward term, and $K_k = -\hat{Q}_{\mathrm{uu}}^{-1}\hat{Q}_{\mathrm{ub}}$ is the feedback term. This linear feedback policy depends on changes in the joint belief $\delta \mathrm{b}_k$, meaning the predicted inputs will change if any agents deviate from their predicted behavior.

We formulate the backwards equations to propagate the value functions V^i backwards as quadratic approximations

$$V_k^i = Q^i + Q_{\mathrm{u}}^{i,\top} j_k + \frac{1}{2} j_k^\top Q_{\mathrm{uu}}^i j_k, \tag{10}$$

$$V_{\mathrm{b},k}^i = Q_{\mathrm{b}}^i + K_k^\top Q_{\mathrm{uu}}^i j_k + K_k^\top Q_{\mathrm{u}}^i + Q_{\mathrm{ub}}^{i,\top} j_k, \tag{11}$$

$$V_{\mathrm{bb},k}^i = Q_{\mathrm{bb}}^i + K_k^\top Q_{\mathrm{uu}}^i K_k + K_k^\top Q_{\mathrm{ub}}^i + Q_{\mathrm{ub}}^{i,\top} K_k, \tag{12}$$

$$V_l^i = c_l^i(\bar{\mathrm{b}}_l), \quad V_{\mathrm{b},l}^i = \left.\frac{\partial c_l^i(\mathrm{b})}{\partial \mathrm{b}}\right|_{\mathrm{b}=\bar{\mathrm{b}}_l}, V_{\mathrm{bb},l}^i = \left.\frac{\partial^2 c_l^i(\mathrm{b})}{\partial \mathrm{b}^2}\right|_{\mathrm{b}=\bar{\mathrm{b}}_l}. \tag{13}$$

D. Control and Belief Regularization

Following [1] and [17], we implement a Levenberg-Marquardt style regularization [18] to ensure convergence to a policy in two parts: control and belief regularization.

$$\tilde{Q}_{\mathrm{uu}}^i = \hat{Q}_{\mathrm{uu}}^i + \mu_{\mathrm{u}} I, \tag{14}$$

$$Q_{\mathrm{ss},k}^i = c_{\mathrm{ss},k}^i + g_{\mathrm{s},k}^\top (V_{\mathrm{bb},k+1}^i + \mu_b I) g_{\mathrm{s},k} + \sum_{j=1}^{n_x} W_{\mathrm{s},k}^{(j),\top} (V_{\mathrm{bb},k+1}^i + \mu_b I) W_{\mathrm{s},k}^{(j)}, \tag{15}$$

where μ_{u} and μ_{b} are positive scalar values. This adds a quadratic cost to the current control sequence and previous belief trajectory, causing new sequences to deviate less as μ_{u} and μ_{b} increase, respectively.

III. Technical Approach

The prior section outlined the mathematical preliminaries for stochastic dynamic games [1]. Our goal is to scale these methods to greater numbers of agents through selective negotiations, enabled by "asocial" agents. We first modify our regularization to account for more agents in the iLQG with a new conditional line search. Next, we define asocial agents. Finally, we present a selective negotiation framework that utilizes asocial agents in a modified iLQG algorithm.

A. Conditional Line Search

As the number of agents increase, using automatic differentiation to obtain the Jacobians g_s and W_s becomes infeasible due to memory limitations. Instead, we utilize finite differences for these terms which can result in slight errors propagated through our policy. As such, the control and belief regularization will sometimes only converge near the nominal policy. To ensure convergence to a minimum, we implement a conditional line search similar to [10]

$$\pi_k = \bar{\mathrm{u}}_k + \alpha j_k + K_k \delta \mathrm{b}_k, \tag{16}$$

where $\alpha = 1$ when unregularized. Whenever a new proposed trajectory matches the previous proposed trajectory within some percent tolerance, α is decreased. When a new policy is accepted, α is reset to 1. This ensures linear convergence to a policy. Intuitively, our control and belief regularization quadratically converge when far from a minimum. Then, the conditional line search linearly converges if regularization is detected to be ineffective near the minimum.

B. Asocial Agents

We wish to model agents that do not consider the interactivity of its actions nor the interactions of other agents. Ideally, a social agent should be able to reactively plan around the decisions of these agents, but should not be able to influence their decisions. One simple example of this is a dynamic obstacle on a fixed path. A social agent can plan to avoid the obstacle, but it cannot affect the obstacle's trajectory. We define these agents as asocial.

Definition 2 (Asocial Agents)**.** *An agent is considered asocial if the planned actions of all agents do not affect its planned actions, such that* $\pi_k^{as} = \bar{\mathrm{u}}_k^{as}, \forall k = 0, \ldots, l-1$.

In the context of the iLQG, an asocial agent's policy should equal their nominal input, and social agents should assume that they cannot influence the decisions of asocial agents. To allow social agents to plan around asocial agents, we introduce Assumption 3.

Assumption 3 (Asocial Agent Common Knowledge)**.** *Social agents have models for nominal input sequence* $\bar{\mathrm{u}}_k^{as}$*, dynamics, and observations(but not cost) for all asocial agents.*

One method to model asocial agents to satisfy Definition 2 is to complete the iLQG and set the relevant feedforward and feedback terms of the policy to 0, similar to enforcing input constraint violations [12]. However, asocial agents without cost functions (i.e. dynamic obstacles with set inputs) would become impossible to model. Instead, we model asocial agents by assigning their backwards pass action value gradients and Hessians. We define the dynamics and measurement equation of asocial agents as seen in (3,4). We fix the cost gradient and Hessian for asocial agents in our system as

$$Q^{as}_{\mathrm{s},k} = 0_{\mathrm{s}x1}, \quad Q^{as}_{\mathrm{ss},k} = I_{\mathrm{sxs}}, \tag{17}$$

for all steps k in the planning horizon. This is equivalent to setting the action value of the agent to a minimum. This results in $Q^{as}_{\mathrm{uu}} = I_{\mathrm{uxu}}, Q^{as}_{\mathrm{ub}} = 0_{\mathrm{u}x\mathrm{b}}, Q^{as}_{\mathrm{u}} = 0_{\mathrm{u}x1}$ for all asocial agents. Plugging into (9), the linear feedback policy for a fully asocial agent system becomes $\pi^{as}_k = \bar{\mathrm{u}}^{as}_k$.

We now seek to combine social and asocial agents and derive a new linear feedback policy. We first present Lemma 1, which states the properties of a 2x2 Block Matrix Inverse. We use this result in our proof of Theorem 1, which presents our linear feedback policy for our game.

Lemma 1 (2x2 Block Matrix Inverse [19])**.** *Let* $S = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ *where A and D are square blocks of arbitrary size and B and C are conformable with them for partitioning. Let D be invertible. Then S is invertible if and only if the Schur complement* $D' = A - BD^{-1}C$ *of D is invertible, and*

$$S^{-1} = \begin{bmatrix} (D')^{-1} & -(D')^{-1}BD^{-1} \\ -D^{-1}C(D')^{-1} & D^{-1} + D^{-1}C(D')^{-1}BD^{-1} \end{bmatrix}.$$

Proof. We refer to [19] for our proof.

Theorem 1. *Given a system containing social and asocial agents, the joint policies* π_k *of all agents results in linear feedback policies for social agents and the nominal input for asocial agents such that*

$$\pi_k = \begin{bmatrix} \bar{\mathrm{u}}^{so} \\ \bar{\mathrm{u}}^{as} \end{bmatrix} - \begin{bmatrix} (\mu_{\mathrm{u}} I_{u^{so}} + Q^{so}_{\mathrm{u}^{so}\mathrm{u}^{so}})^{-1} Q^{so}_{\mathrm{u}^{so}} \\ 0 \end{bmatrix} - \begin{bmatrix} (\mu_{\mathrm{u}} I_{\mathrm{u}^{so}} + Q^{so}_{\mathrm{u}^{so}\mathrm{u}^{so}})^{-1} Q^{so}_{\mathrm{u}^{so}\mathrm{b}} \\ 0 \end{bmatrix} \delta \mathrm{b}_k, \tag{18}$$

where $Q^{so}_{\mathrm{u}^{so}\mathrm{u}^{so}}$, $Q^{so}_{\mathrm{u}^{so}\mathrm{b}}$, $Q^{so}_{\mathrm{u}^{so}}$ *are obtained from the stacked partial derivatives* (8) *of all social agents.*

Proof. Consider a multiagent system with social agents $a^{so,m}, \{m = 1, ..., M\}$ and p asocial agents $a^{as,p}, \{p = 1, ..., P\}$. We refer to a^{so} and a^{as} as all social and asocial agents, respectively. We now solve the quadratic game as defined by (9). We drop k for notational convenience and obtain the following regularized matrices at timestep k

$$\tilde{Q}_{\mathrm{uu}} = \begin{bmatrix} \mu_{\mathrm{u}} I_{\mathrm{u}^{so}} + Q^{so}_{\mathrm{u}^{so}\mathrm{u}^{so}} & Q^{so}_{\mathrm{u}^{so}\mathrm{u}^{as}} \\ 0 & \mu_{\mathrm{u}} I_{\mathrm{u}^{as}} + I_{\mathrm{u}^{as}} \end{bmatrix}, \tag{19}$$

$$\hat{Q}_{\mathrm{ub}} = \begin{bmatrix} Q^{so}_{\mathrm{u}^{so}\mathrm{b}} \\ 0 \end{bmatrix}, \quad \hat{Q}_{\mathrm{u}} = \begin{bmatrix} Q^{so}_{\mathrm{u}^{so}} \\ 0 \end{bmatrix}, \tag{20}$$

where I are conforming identities and $Q^{so}_{\mathrm{u}^{so}\mathrm{u}^{so}}, Q^{so}_{\mathrm{u}^{so}\mathrm{u}^{as}} \neq 0$.

Now consider the inverse of $\tilde{Q}_{\mathrm{uu}}$ from (19). $\tilde{Q}_{\mathrm{uu}}$ is positive definite due to the regularization from (14), and thus $\tilde{Q}^{-1}_{\mathrm{uu}}$ exists [20]. The sub-matrix $D = \mu_{\mathrm{u}} I_{\mathrm{u}^{as}} + I_{\mathrm{u}^{as}}$ is trivially invertible. If $\tilde{Q}_{\mathrm{uu}}$ and D are invertible and $C = 0$, then the Schur complement $D' = A - BD^{-1}C = A$ is also invertible. From Lemma 1 we find $\tilde{Q}^{-1}_{\mathrm{uu}}$, then use (20) and (9) to find (18) and thus prove Theorem 1.

Intuitively, Theorem 1 claims for asocial agents, the action value is set to a minimum, meaning the nominal input will never change. For social agents, the partial derivatives are only taken with respect to social agent inputs and beliefs. As such, social agents do not attempt to negotiate with asocial agents, they instead reactively plan around asocial inputs.

C. Selective Negotiation Requirements

A key advantage of asocial agents is that they can exist alongside social agents. By design, an agent can also change its designation over the course of a game. We now present our selective negotiation framework, which allows our ego agent to assign social and asocial agents. We define the following requirements for selective negotiation:

1) The ego agent must consider the existence all agents in the problem, even if not all agents are social;
2) The property of being social does not need to be intrinsic to any agent;
3) The method must fit into the previous iLQG as defined in Section II, and provide faster runtime complexity.

Requirement 1 means that the ego agent cannot ignore other agents. Otherwise, certain games may become impossible to formulate without the existence of all agents being known. Requirement 2 means that the ego agent can switch its classification of other agents to be social or asocial. For example, a driver should be able to model a faraway car as asocial, and then model it as social when it gets closer. We fulfill these requirements by assigning agents as social or asocial before the iLQG begins. We do this by using heuristics based on an agent's beliefs, such as nearest neighbor. We can also permanently assign agents, for example, a dynamic obstacle is always asocial.

D. Algorithm for Solving the Nash Equilibrium

We now present our algorithm to solve for the Nash Equilibrium with asocial agents with Algorithm 1. We first assign each agent to be social or asocial using a heuristic. We then assign nominal inputs for each agent, which contains unchanging inputs $\bar{\mathrm{u}}^{as}$ and changing inputs $\bar{\mathrm{u}}^{so}$.

We begin the iLQG to find a local Nash Equilibrium of the social ego agent. The current belief estimate b_0 and nominal controls $\bar{\mathrm{u}}$ are used to find an initial belief trajectory. Then, the backwards pass finds the policy π for all agents, with only the social agent action value functions being propagated. We note that $\bar{\mathrm{u}}^{as}$ is encoded within π and does not change per Theorem 1. The forward pass updates the belief trajectory based on the belief dynamics model and the updated feedback policy π. When all the action value functions of social agents improve, we assign the new belief and control

trajectories as nominal and reduce regularization. Otherwise, we reject the trajectories and increase regularization. This iteration of backwards and forwards pass continues until all action value functions of social agents converge with a relative change less than the threshold ϵ.

Algorithm 1 Nash Equ. Solution with Asocial Agents

Input: Initial belief b_0, models c_k^i, c_l^i, f, h, agents a^i
Output: Predicted trajectories $\bar{\mathrm{b}}$, $\bar{\mathrm{u}}$, feedback law π

$Flag_{a^i} \leftarrow$ isSocialAgentHeuristic$(a^i, \mathrm{b}_0, c_k^i, c_l^i)$
$\bar{\mathrm{u}}^i \leftarrow$ getAgentInput$(a^i, \mathrm{b}_0, c_k^i, c_l^i, Flag_{a^i})$
$\bar{\mathrm{u}} \leftarrow \bar{\mathrm{u}}^{1,\dots,N}$ ▷ Contains $\bar{\mathrm{u}}^{so}, \bar{\mathrm{u}}^{as}$
$\bar{\mathrm{b}} \leftarrow$ Propagate b_0 with g and $\bar{\mathrm{u}}$
while any$(\frac{\|Q^{so}(\bar{\mathrm{b}}_{\mathrm{new}}, \bar{\mathrm{u}}_{\mathrm{new}}) - Q^{so}(\bar{\mathrm{b}}, \bar{\mathrm{u}})\|}{Q^{so}(\bar{\mathrm{b}}, \bar{\mathrm{u}})} > \epsilon)$ **do**
  Backwards Pass:
  $V_{\mathrm{b},l}^{so}, V_{\mathrm{bb},l}^{so} \leftarrow$From terminal bound. conditions (13)
  for k from $l-1$ to 0 **do**
    $Q_{\mathrm{s},k}^{so}, Q_{\mathrm{ss},k}^{so} \leftarrow$ Prop. action value funcs. (6)
    $Q_{\mathrm{s},k}^{as}, Q_{\mathrm{ss},k}^{as} \leftarrow 0_{\mathrm{s}x1}, I_{\mathrm{s}x\mathrm{s}}$ (17)
    $\pi_k, j_k, K_k \leftarrow$ Solve quadratic game (9)
    $V_{\mathrm{b},k}^{so}, V_{\mathrm{bb},k}^{so} \leftarrow$ Prop. value funcs. (10)(11)(12)
  end for
  Forward Pass:
  $\bar{\mathrm{b}}_{\mathrm{new}}, \bar{\mathrm{u}}_{\mathrm{new}} \leftarrow$ Propagate b_0 with g and π
  if any$(Q^{so}(\bar{\mathrm{b}}_{\mathrm{new}}, \bar{\mathrm{u}}_{\mathrm{new}}) \leq Q^{so}(\bar{\mathrm{b}}, \bar{\mathrm{u}}))$ **then**
    $\bar{\mathrm{b}}, \bar{\mathrm{u}} \leftarrow \bar{\mathrm{b}}_{\mathrm{new}}, \bar{\mathrm{u}}_{\mathrm{new}}$ ▷ Only $\bar{\mathrm{u}}^{so}$ changes in $\bar{\mathrm{u}}$
    Lower Regularization (14)(15)(16)
  else Increase Regularization
  end if
end while

E. Social and Asocial Agent Heuristic Selection

The social/asocial heuristic from Algorithm 1 is problem specific, though some ground rules exist for proper heuristic selection. First, the ego agent must always be considered social to derive new trajectories. Likewise, agents without cost functions such as dynamic obstacles must always be considered asocial. More complex selection can be done through analyzing an ego agent's cost function. For example, agents avoiding collisions should model nearby agents as social in order to obtain more reliable collision avoidant trajectory predictions. Further selection can be done by analyzing agents which have coupled dynamics and measurement equations to the ego agent, such as an agent connected to the ego by a spring. Essentially, agents whose interactions greatly influence the reward or actions of the ego should be modeled as social, while other agents can be modeled as asocial. Proper heuristic selection is necessary for large improvements in computation times. For example, for $N = 50$ trials of the game in Figure 1 we observed a 38% decrease in computation times using selective negotiations, with 28% less iterations to converge on average.

F. Dominant Runtime Analysis

We define N and N_{so} as the number of total agents and social agents in the system, respectively. We define the joint state dimension as $\mathcal{O}(n_{\mathrm{x}})$ and assume all agent's contain the same number of states such that $\mathcal{O}(n_{\mathrm{x}}) = \mathcal{O}(Nn_{\mathrm{x}}^i)$. We also assume $n_{\mathrm{x}} = n_{\mathrm{u}} = n_{\mathrm{z}}$. The covariance of the joint state contains $n_{\mathrm{x}}^2/2$ unique elements. The joint belief b thus contains $n_{\mathrm{x}} + n_{\mathrm{x}}^2/2$ elements, or $\mathcal{O}(n_{\mathrm{x}}^2)$ elements. Similar to analysis in [1], we find a computational bottleneck when evaluating the action-value function $Q_{\mathrm{ss},k}^i$ in (6). The term $g_{\mathrm{s},k}^\top V_{\mathrm{bb},k+1}^i g_{\mathrm{s},k}$ requires a matrix multiplication of dimension $\mathcal{O}(n_{\mathrm{x}}^2) \times \mathcal{O}(n_{\mathrm{x}}^2) = \mathcal{O}(n_{\mathrm{x}}^6) = \mathcal{O}(N^6 n_{\mathrm{x}}^{i,6})$ complexity. This operation is only completed for social agents, meaning a full iteration of this algorithm has a dominant runtime complexity of $\mathcal{O}(lN_{so}N^6 n_{\mathrm{x}}^{i,6})$, where l is the planning horizon. The original algorithm's runtime complexity is $\mathcal{O}(lN^7 n_{\mathrm{x}}^{i,6})$. Thus, replacing social agents with asocial agents can linearly lower the dominant runtime since $N_{so} \leq N$.

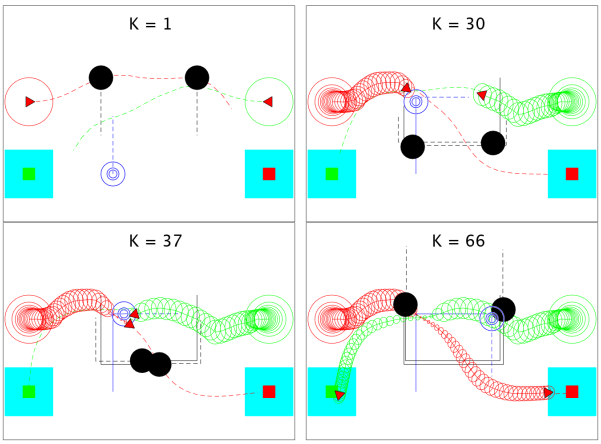

Fig. 2. The social agents (red, green) traverse past dynamic obstacles (black), localize themselves within a moving light source (blue), and proceed towards their goal positions (cyan) while avoiding collisions. The light source is modeled with exponential decay to encourage negotiations. Initially, both social agents wait for obstacles to pass before proceeding towards the light source. They then negotiate so that both can localize themselves without colliding. After localization, the red agent waits for obstacles to pass while the green agent cuts past the obstacle.

IV. Simulations

In this section, we examine the utility of asocial agents and the modified iLQG in two stochastic dynamic games: an obstacle course and a narrow traffic scenario. We implement our algorithm in Matlab simulations with CasADi [21] to leverage automatic differentiation, static C-code/compute graph generation, and sparse operations. We ran our simulations on an Intel Core i7-13700KF at 3.4 GHz. Agents for all simulations were run with a nominal control input sequence of all zeros unless otherwise noted.

A. Obstacle Course

We first demonstrate our algorithm allows social agents with car-like dynamics to plan around social and asocial agents. Agents must avoid dynamic obstacles and another social agent, localize within a moving light source, and move towards their goal location. The states of a^{so}, $\mathrm{x} = [x^{so}, y^{so}, v^{so}, \theta^{so}]$ denote the position (x, y), speed v, and orientation θ. The control inputs $\mathrm{u}^{so} = [\mathrm{u}_{acc}^{so}, \mathrm{u}_{ste}^{so}]$ denote acceleration u_{acc}^{so} and steering angle

TABLE I

AVERAGED COMPETITIVE PERFORMANCE: SOCIAL VS. ASOCIAL AGENT MODELING

Mean $\pm 1\sigma$ for $N = 50$ Trials	Ego (3 a^{so})	Blue (3 a^{so})	Ego (2 a^{so})	Blue (3 a^{so})	Ego (1 a^{so})	Blue (3 a^{so})
Iteration Time (s)	$.2046 \pm .0209$	$.2051 \pm .0211$	$.1843 \pm .0222$	$.2034 \pm .0209$	$.1626 \pm .0233$	$.2042 \pm .0213$
Yield from Shortest Path (m)	1.799 ± 1.273	1.580 ± 1.143	1.605 ± 1.164	1.681 ± 1.211	1.948 ± 1.510	1.640 ± 1.062
Time to Goal (s)	$4.970 \pm .4709$	$4.922 \pm .5369$	$4.808 \pm .4985$	$4.966 \pm .6180$	$4.884 \pm .7713$	$4.936 \pm .6016$
Total $\mathrm{u}_{acc}/\mathrm{u}_{ste}$ $(\mathrm{m/s}^2)$/(rad)	13.725 / 1.112	14.047 / 1.052	14.100 / 1.079	14.039 / 1.123	14.753 / 1.098	14.010 / 1.112

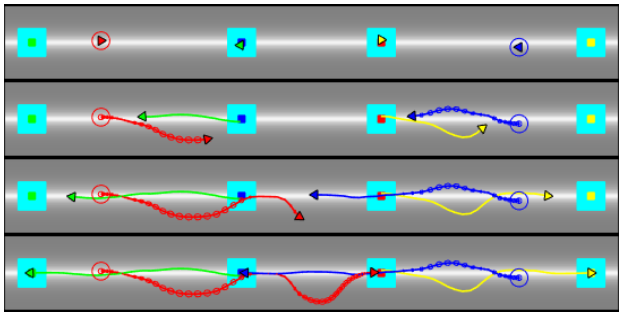

Fig. 3. The ego agent (red) reaches its target without collisions. It minimizes its uncertainty by travelling on the light path, which incentivizes interactions with other agents. The symmetric agent (blue) completes an identical task, as both navigate around other social agents (green,yellow) before negotiating with each other. This example shows a case where all agents model all other agents as social. Covariances for green and yellow agents are omitted for clarity.

u^{so}_{ste}. We encode the dynamics of social agents as $\dot{\mathrm{x}}_k = \begin{bmatrix} v_k \cos\theta_k & v_k \sin\theta_k & \mathrm{u}_{acc,k} & \frac{v_k \tan \mathrm{u}_{ste,k}}{L} \end{bmatrix}^\top$, where L is agent length. The discrete time dynamics are $\mathrm{x}_{k+1} = \mathrm{x}_k + \dot{\mathrm{x}}_k \tau + M(\mathrm{u}_k) \cdot \mathrm{m}_k$, where τ is the timestep and $M(\mathrm{u}_k)$ scales the process noise multiplicative to the control input. We encode the agent's objectives by defining its cost functions

$$c_k(\mathrm{b}_k, \mathrm{u}_k) = \mathrm{u}_k^\top R \mathrm{u}_k + \beta_k \det(\Sigma_{xy,k}) + \mathrm{c}_{coll}(\mathrm{x}_k),$$
$$c_l(\mathrm{b}_l) = \beta_l \det(\Sigma_{xy,l}) + \gamma_l \|d_{go}\|^2, \tag{21}$$

where $\|d_{go}\|$ is the Euclidean distance of (x, y) from the desired position, γ_l, β, R are tuning parameters, Σ_{xy} denotes the ego agent's positional uncertainty, and $\mathrm{c}_{coll}(\mathrm{x})$ denotes an exponential collision barrier as in [1].

We restrict the social agents to noisy position measurements, with more precise measurements near the moving light source. The observation model becomes $\mathrm{z}^{so}_k = [x^{so}, y^{so}]^\top + N(\mathrm{x}^{so}_k, \mathrm{x}^{\neg so}_k) \cdot \mathrm{n}^{so}_k$. We model the asocial agents with states $\mathrm{x} = [x^{as}, y^{as}]$ and inputs $\mathrm{u} = [u_x, u_y]$. The dynamics become $\dot{\mathrm{x}}_k = \begin{bmatrix} \mathrm{u}_{x,k} & \mathrm{u}_{y,k} \end{bmatrix}^\top$, with similar discrete time dynamics and observation model for a^{so}. The asocial agents in Figure 2 are the dynamic obstacles and light source.

We make predefined rectangular trajectories for the asocial agents, with social agents only knowing their inputs within the planning horizon. The resulting behavior in Figure 2 shows the social agents can traverse the obstacle course while avoiding collisions and localizing themselves. As shown in our video, we can find alternative emergent behaviors such as aggressively cutting past the obstacles or taking more conservative trajectories by tuning the cost function.

B. Narrow traffic negotiation

To demonstrate the utility of modeling social agents as asocial, we devise a traffic problem for our ego agent to traverse. As seen in Figure 3, agents must avoid collisions, reduce their uncertainty, and move towards their goals. A goal is considered reached when an agent is within 1 meter of it. We note that an agent which is closer to its goal has less incentive to deviate. For the ego and its symmetric agent, this means whichever traverses the first negotiation faster will gain an advantage in its second negotiation.

We benchmark performance by fixing the number of other agents the ego agent can model as social to $3a^{so}$, $2a^{so}$ and $1a^{so}$. We use a nearest neighbor heuristic which updates at each timestep. In theory, our asocial modeling heuristics($2a^{so}$, $1a^{so}$) should run faster and perform approximately the same as the $3a^{so}$ planner. All other agents model all agents as social. All agents are homogeneous with car-like dynamics, noisy position measurements, and cost functions similar to the social agents in the obstacle course game.

We run $N = 50$ trials and show our results in Table I. We see that when the ego and symmetric agent model $3a^{so}$, their metrics are approximately equal with some advantage to the symmetric agent. As the ego agent considers more agents asocial, its iteration time decreases 10% for $2a^{so}$ and 21% for $1a^{so}$. The overall computation time per timestep for each heuristic follows these trends.

Compared to $3a^{so}$, the $2a^{so}$ ego yields $\sim 11\%$ less at the expense of $0.375m/s^2$ more u_{acc} control effort and reaches the goal in around the same amount of time. This indicates that it converges to slightly more aggressive policies. One explanation for this is that the difference in social agents change when regularization occurs, potentially moving the ego's policies into different local Nash Equilibria under certain circumstances. In the $1a^{so}$ case, the nearest neighbor heuristic does not model the blue agent as social quickly enough, leading to the ego yielding $\sim 8\%$ more with $1.028m/s^2$ extra control effort. This worse performance verifies that only irrelevant agents should be assigned as asocial. Thus, heuristic choices like $2a^{so}$ which take the structure of the game into account will perform better.

V. CONCLUSION AND FUTURE WORK

In this paper, we introduce the asocial agent and examine its utility through two stochastic dynamic games: an obstacle course and a traffic negotiation. We show that asocial agents allow us to encode environmental objects into the iLQG from [1]. We utilize asocial agents for selective negotiations, which simplifies the game-theoretic modeling in our game. We show selective negotiations can result in 10-38% faster computation times with minimal performance decline using various heuristics. Future work will focus on belief space simplifications and a scalable hardware implementation.

REFERENCES

[1] W. Schwarting, A. Pierson, S. Karaman, and D. Rus, "Stochastic Dynamic Games in Belief Space," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2157–2172, Dec. 2021, conference Name: IEEE Transactions on Robotics.

[2] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state," *Autonomous Robots*, vol. 42, no. 7, pp. 1405–1426, Oct. 2018. [Online]. Available: http://link.springer.com/10.1007/s10514-018-9746-1

[3] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game-Theoretic Planning for Self-Driving Cars in Multivehicle Competitive Scenarios," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1313–1325, Aug. 2021. [Online]. Available: https://doi.org/10.1109/TRO.2020.3047521

[4] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 50, pp. 24 972–24 978, Dec. 2019. [Online]. Available: https://pnas.org/doi/full/10.1073/pnas.1820676116

[5] G. Williams, B. Goldfain, P. Drews, J. M. Rehg, and E. A. Theodorou, "Best Response Model Predictive Control for Agile Interactions Between Autonomous Ground Vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 2403–2410, iSSN: 2577-087X.

[6] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, "Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 1475–1481, iSSN: 2577-087X.

[7] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, May 1998. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S000437029800023X

[8] R. D. Smallwood and E. J. Sondik, "The Optimal Control of Partially Observable Markov Processes over a Finite Horizon," *Operations Research*, vol. 21, no. 5, pp. 1071–1088, Oct. 1973, publisher: INFORMS. [Online]. Available: https://pubsonline.informs.org/doi/10.1287/opre.21.5.1071

[9] C. H. Papadimitriou and J. N. Tsitsiklis, "The Complexity of Markov Decision Processes," *Mathematics of Operations Research*, vol. 12, no. 3, pp. 441–450, Aug. 1987. [Online]. Available: https://pubsonline.informs.org/doi/10.1287/moor.12.3.441

[10] J. van den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, Sep. 2012. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0278364912456319

[11] S. Patil, G. Kahn, M. Laskey, J. Schulman, K. Goldberg, and P. Abbeel, "Scaling up Gaussian Belief Space Planning Through Covariance-Free Trajectory Optimization and Automatic Differentiation," in *Algorithmic Foundations of Robotics XI*, H. L. Akin, N. M. Amato, V. Isler, and A. F. van der Stappen, Eds. Cham: Springer International Publishing, 2015, vol. 107, pp. 515–533, series Title: Springer Tracts in Advanced Robotics. [Online]. Available: https://link.springer.com/10.1007/978-3-319-16595-0_30

[12] E. Todorov and W. Li, "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *Proceedings of the 2005, American Control Conference, 2005.*, Jun. 2005, pp. 300–306 vol. 1, iSSN: 2378-5861.

[13] Z. Williams, J. Chen, and N. Mehr, "Distributed Potential iLQR: Scalable Game-Theoretic Trajectory Planning for Multi-Agent Interactions," 2023, publisher: arXiv Version Number: 1. [Online]. Available: https://arxiv.org/abs/2303.04842

[14] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, no. 12, pp. 2105–2115, Dec. 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109806003049

[15] E. Camponogara, D. Jia, B. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44–52, Feb. 2002, conference Name: IEEE Control Systems Magazine.

[16] M. Wooldridge, *An Introduction to Multi-Agent Systems*, 1st ed. England: John Wiley, 2002.

[17] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 4906–4913, iSSN: 2153-0866.

[18] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944. [Online]. Available: https://www.ams.org/qam/1944-02-02/S0033-569X-1944-10666-0/

[19] T.-T. Lu and S.-H. Shiou, "Inverses of 2 × 2 block matrices," *Computers & Mathematics with Applications*, vol. 43, no. 1, pp. 119–129, Jan. 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0898122101002784

[20] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge University Press, 2012.

[21] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, Mar. 2019. [Online]. Available: http://link.springer.com/10.1007/s12532-018-0139-4