Partial Belief Space Planning for Scaling Stochastic Dynamic Games

Kamran Vakil, Mela Coffey, and Alyssa Pierson

***Abstract*— This paper presents a method to reduce computations for stochastic dynamic games with game-theoretic belief space planning through partially propagating beliefs. Complex interactions in scenarios such as surveillance, herding, and racing can be modeled using game-theoretic frameworks in the belief space. Stochastic dynamic games can be solved to a local Nash Equilibrium using a game-theoretic belief space variant of an iterative Linear Quadratic Gaussian (iLQG). However, the scalability of this method suffers due to the large dimensionality of beliefs which the iLQG must propagate. We examine the utility of partial belief space propagation, which allows polynomial runtime to decrease. We validate our findings through simulations and hardware implementation.**

I. INTRODUCTION

Multiagent system trajectory planning under uncertainty is an extensively researched field. Recent work focuses on planners which can take the interactivity between agents and quality of information into account when planning trajectories. Such planners have application in surveillance, pursuer-evader games, and driving scenarios. One recent approach combines game-theoretic decision making with belief space planning [1]. The interactivity between agents can be modeled by game-theoretic planning. Meanwhile, belief space planning allows for the quality of information to be estimated along a planned trajectory, allowing for the ability to gain and leverage information in addition to fulfilling other tasks. Combining these two methods can be done by formulating the problem as a game-theoretic Partially Observable Markov Decision Process (POMDP). Then, a game theoretic variant of an iterative Linear Quadratic Gaussian (iLQG) can be used to find a local Nash Equilibrium between all agents.

However, this method propagates a large quantity of beliefs for a predicted trajectory. In the full belief space, these beliefs quadratically increase as the number of agents linearly increase. This makes scaling the iLQG intractable for larger numbers of agents. Yet, many scenarios may not require full belief propagation in order to complete tasks. For example, consider two agents which can localize themselves strongly with a static beacon but not with each other. In such a case, not propagating the cross covariance between the positions of both agents would decrease computation times with minimal performance degradation. Ideally, an agent would propagate only the most important beliefs for a predicted trajectory. We seek to extend prior work in [1] to allow for this partial belief space planning in order to obtain faster computation times with minimal performance decrease. Figure 1 illustrates an example of how these beliefs can be reduced with minimal performance degradation.

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

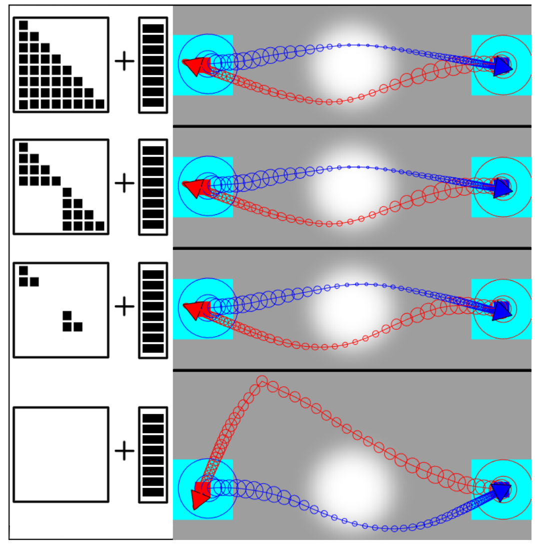

Fig. 1. An example of agents completing an informative circle swap with identical initial conditions using various belief space planning methods (from top to bottom, propagation of full belief, nonzero belief, states plus positional uncertainty belief, and only states belief). The graphics on the left show the beliefs propagated with the planner in terms of a vector of states and matrix of uncertainty for the states. The right shows the trajectories agents take to reach their goal. The partial belief space planners can localize their agents before proceeding to the goal similar to the full belief space planner. Unlike a standard iLQG (bottom), these partial belief space planners attempt to gain and leverage information to their benefit.

The main contributions of this paper are:

1) A proposal of the partial belief space planning framework and its benefits;
2) Insights into selecting which beliefs require propagation and potential adverse effects when done incorrectly;
3) Simulation and hardware results showing the method can allow for real time speeds compared to full belief space planning.

Background and Related Work

Game-theoretic planners [2]–[4] are useful for modeling problems where the objectives of individual agents are at odds. The interactivity between agents is commonly seen in applications such as driving and racing [5]–[7]. One common implementation is to find the Nash Equilibrium between all agents. This can be done using Best Iterated Response [8], iterative quadratic approximations [9], and solving the necessary conditions [7]. The belief space variant iLQG presented in [1] solves the Nash Equilibrium's necessary condition through a static quadratic game at each timestep.

Belief Space Planning [10] represents a robot's uncertainties using distributions of the robot's state estimate. These distributions are called beliefs, and computing policies over the space of beliefs can be described by a POMDP [11]. POMDPs allow for modeling real world processes under uncertainty, though solving a POMDP to global optimality is undecidable [12]. Optimization based approaches [13] [14] including iterative Linear Quadratic Gaussian(iLQG) [15] scale linearly in the planning horizon l, making it attractive for real world applications.

The iLQG from [1] can be abstracted as an iterative gradient descent towards an optimal policy. Following this abstraction, we seek to reduce our computations by fixing the number of beliefs we consider in our descent. We thus take inspiration from the concept of block coordinate descent [16]–[18], which minimizes an objective along some variables while fixing the remainder. While our method does not iterate between different blocks of partial beliefs to reach a minimum policy, we consider it a first step towards faster computation times with minimal performance decreases.

The remainder of this paper is organized as follows: Section II summarizes POMDP formulation and belief dynamics. Section III introduces the partial belief space and how its implementation impacts the iLQG algorithm in [1]. Section IV discusses the performance benefits of using the partial belief space versus full belief space during planning. Section V shows the utility of partial belief space planning empirically through simulation and experimental results.

II. Problem Formulation

In this section, we formulate the stochastic system in the belief space as a POMDP. We then approximate the general Bayesian Filter update as an Extended Kalman filter (EKF) to propagate Gaussian beliefs through belief dynamics.

We follow the same assumptions in [1] of common knowledge and first order beliefs. Common knowledge means agents have models for cost, dynamics, and observations of other agents. First order beliefs means all agents share the same beliefs about each other, i.e. an agent i's belief about agent j and agent j's belief about itself are the same. In our simulations and experiments, we run separate planners for each agent with separate measurements and actuations corrupted by independent noise. This violates the first order beliefs assumption, yet our planner performs well since the difference in beliefs between agents is not drastic.

A. POMDP Formulation

We follow notation from [1], [13] and define POMDPs in a general form. The expected return of each individual agent is determined by the action value function Q^i

$$Q^i(\mathrm{b}_0, \mathrm{u}) = \mathbb{E}\left[c_l^i(\mathrm{b}_l) + \sum_{k=0}^{l-1} c_k^i(\mathrm{b}_k, \mathrm{u}_k)\right], \tag{1}$$

where u is the control trajectory of all agents subject to uncertainty on observed measurements z over the horizon l, c_k^i and c_l^i are the cost at time k and terminal cost of agent i, and b is the belief about the state x of the system. We note that belief in this context refers to the agent's estimated state and uncertainty about its state. There are N distinct action value functions, one for each agent $i \in \{1, ..., N\}$. We seek to solve the stochastic optimal control problem [1]

$$\begin{aligned} \pi^i = &\operatorname*{argmin}_{\mathrm{u}^i} Q^i(\mathrm{b}_0, \mathrm{u}) \quad \forall i \in \{1, ..., N\}, \\ &s.t. \quad \mathrm{b}_{k+1} = \beta(\mathrm{b}_k, \mathrm{u}_k, \mathrm{z}_{k+1}). \end{aligned} \tag{2}$$

where b_0 is the initial belief, β is the stochastic belief dynamics of b_k, and π^i is the optimal policy of agent i. A general solution to (2) can be defined recursively by the Bellman equation [1]:

$$\begin{aligned} Q_k^i(\mathrm{b}_k, \mathrm{u}_k) &= c_k^i(\mathrm{b}_k, \mathrm{u}_k) + \mathop{\mathbb{E}}_{\mathrm{z}_{k+1}}\left[V_{k+1}^i(\beta(\mathrm{b}_k, \mathrm{u}_k, \mathrm{z}_{k+1},))\right], \\ V_k^i(\mathrm{b}_k) &= \min_{\mathrm{u}_k^i} Q_k^i(\mathrm{b}_k, \mathrm{u}_k), \quad V_l^i = c_l^i(\mathrm{b}_l), \\ \pi_k^i(\mathrm{b}_k) &= \operatorname*{argmin}_{\mathrm{u}_k^i} Q_k^i(\mathrm{b}_k, \mathrm{u}_k), \end{aligned} \tag{3}$$

where $V_k^i(\mathrm{b}_k)$ is the value function and $\pi_k^i(\mathrm{b}_k)$ is the optimal policy at time step k.

B. Belief Dynamics

Here, we summarize belief dynamics first presented in [1], which then allows us to present our approach to partially-propogated beliefs. This section provides a primer on belief dynamics. As in prior work, we define our beliefs with Gaussian distributions that are approximated through an EKF. We then use a quadratic approximation of the value function about a nominal trajectory through the belief space, and the iLQG iteratively computes a local Nash equilibrium over all agents in the belief space using a Bellman backwards recursion [1]. We assume nonlinear stochastic dynamics and observation models for any single agent a^i as $\mathrm{x}_{k+1}^i = f(\mathrm{x}_k, \mathrm{u}_k, \mathrm{m}_k), \mathrm{m}_k \sim \mathcal{N}(0, I)$, and $\mathrm{z}_k^i = h(\mathrm{x}_k, \mathrm{n}_k), \mathrm{n}_k \sim \mathcal{N}(0, I)$, where m_k and n_k denote process and measurement noise whose distributions can be arbitrarily transformed inside the equations. We formulate the joint process and measurement functions of all agents i, $i = \{1, ..., N\}$ independently as [1]

$$\begin{aligned} f(\mathrm{x}_k, \mathrm{u}_k, \mathrm{m}_k) = [&f^1(\mathrm{x}_k^1, \mathrm{u}_k^1, \mathrm{m}_k^1)^\top, \dots, \\ &f^N(\mathrm{x}_k^N, \mathrm{u}_k^N, \mathrm{m}_k^N)^\top]^\top, \\ h(\mathrm{x}_k, \mathrm{n}_k) = [h^1(&\mathrm{x}_k^1, \mathrm{n}_k^1)^\top, \dots, h^N(\mathrm{x}_k^N, \mathrm{n}_k^N)^\top]^\top, \end{aligned} \tag{4}$$

though we note that our algorithm applies to the general case of dynamics and measurement functions. We define $\mathrm{b}_k = (\hat{\mathrm{x}}_k^\top, \Sigma_k)$ as the Gaussian belief, where mean state $\hat{\mathrm{x}}_k^\top$ and variance Σ_k describes the stochastic state $\mathrm{x}_k \sim \mathcal{N}(\hat{\mathrm{x}}_k^\top, \Sigma_k)$.

We follow [13] and approximate the Bayesian filter as an EKF with standard EKF update equations to make the belief propagation tractable [1],

$$\begin{aligned} \hat{\mathrm{x}}_{k+1} &= f(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0) + K_k(\hat{\mathrm{z}}_{k+1} - h(f(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0), 0)), \\ \Sigma_{k+1} &= \Gamma_{k+1} - K_k H_k \Gamma_{k+1}, \end{aligned} \tag{5}$$

with corresponding matrices defined by [1] $\Gamma_{k+1} = A_k\Sigma_k A_k^\top + M_k M_k^\top$, $K_k = \Gamma_{k+1}H_k^\top(H_k\Gamma_{k+1}H_k^\top + N_k N_k^\top)^{-1}$, $A_k = \frac{\partial f}{\partial \mathrm{x}}(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0)$, $M_k = \frac{\partial f}{\partial \mathrm{m}}(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0)$, $H_k = \frac{\partial h}{\partial \mathrm{x}}(f(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0), 0)$, $N_k = \frac{\partial f}{\partial \mathrm{n}}(f(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0), 0)$. We define $\mathrm{b}_k = [\hat{\mathrm{x}}_k^\top, \mathrm{vec}(\Sigma_k)^\top]^\top$, where $\mathrm{vec}(\Sigma_k)^\top$ is the matrix Σ_k reshaped into vector form, taking symmetry into account. We denote $\mathrm{s} = [\mathrm{b}^\top, \mathrm{u}^\top]^\top$ for belief and controls. We formulate the stochastic belief dynamics [1]

$$
\begin{aligned}
\mathrm{b}_{k+1} &= g(\mathrm{b}_k, \mathrm{u}_k) + W(\mathrm{b}_k, \mathrm{u}_k)\xi_k, \quad \xi_k \sim \mathcal{N}(0, I), \\
g(\mathrm{b}_k, \mathrm{u}_k) &= \begin{bmatrix} f(\hat{\mathrm{x}}_k, \mathrm{u}_k, 0) \\ \mathrm{vec}(\Gamma_{k+1} - K_k H_k \Gamma_{k+1}) \end{bmatrix}, \\
W(\mathrm{b}_k, \mathrm{u}_k) &= \begin{bmatrix} \sqrt{K_k H_k \Gamma_{k+1}} \\ 0 \end{bmatrix},
\end{aligned}
$$

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.

III. Planning with Partially-Propagated Beliefs

While propagating a full belief space allows for trajectories with complex emergent behaviors, the quadratic increase in beliefs due to tracking uncertainty between states makes the method unscalable for larger numbers of agents. We seek to find similar trajectories while propagating only part of an agent's beliefs about itself and other agents. This would allow for a more scalable approach with real time capabilities. In order to do so, we focus on only propagating some of the uncertainty terms in our stochastic belief dynamics.

A. Partial Belief Space

We start by defining our partially propagated and non-propagated beliefs in the following. We maintain the same definition of full beliefs $\mathrm{b} \in \mathbb{R}^{nx1}$from Section II-B where n denotes the number of beliefs.

Definition 1 (Partially Propagated Belief Space). *The propagated partial beliefs* $\mathrm{b}_p \subseteq \mathrm{b}$ *such that* $\mathrm{b}_p \in \mathbb{R}^{(n-m)\times 1}$*, where* $n - m$ *denotes the number of beliefs we propagate.*

Definition 2 (Nonpropagated Belief Space). *The nonpropagated partial beliefs* $\mathrm{b}_{np} = \mathrm{b}_p^\complement$ *such that* $\mathrm{b}_{np} \in \mathbb{R}^{m\times 1}$.

We only consider propagated beliefs in our belief dynamics, lowering the dimensions of g_k and W_k from Section II-B. This means nonpropagated beliefs are fixed in our belief dynamics and thus do not change. Next, we discuss how partially-propagated beliefs modify the prior work on dynamic game belief space planning, then discuss performance with partially propagated beliefs.

B. Modified Dynamic Game Belief Space Planning

We briefly introduce the iLQG from [1] and discuss differences when using the partial belief space. We define $\mathrm{s}_p = [\mathrm{b}_p^\top, \mathrm{u}^\top]^\top$, which represents the propagated beliefs and control input. We formulate the iLQG with a quadratic approximation and use the Bellman equations to obtain backwards pass equations from [1]. We start by defining the action value functions from [1] in the partial belief space

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

$$
\begin{aligned}
Q_{\mathrm{s}_p,k}^i &= c_{\mathrm{s}_p,k}^i + g_{\mathrm{s}_p,k}^\top V_{\mathrm{b}_p,k+1}^i \\
&\quad + \sum_{j=1}^{n_x} W_{\mathrm{s}_p,k}^{(j),\top} V_{\mathrm{b}_p\mathrm{b}_p,k+1}^i W_k^{(j)}, \qquad (7) \\
Q_{\mathrm{s}_p\mathrm{s}_p,k}^i &= c_{\mathrm{s}_p\mathrm{s}_p,k}^i + g_{\mathrm{s}_p,k}^\top V_{\mathrm{b}_p\mathrm{b}_p,k+1}^i g_{\mathrm{s}_p,k} \qquad (8) \\
&\quad + \sum_{j=1}^{n_x} W_{\mathrm{s}_p,k}^{(j),\top} V_{\mathrm{b}_p\mathrm{b}_p,k+1}^i W_{\mathrm{s}_p,k}^{(j)}, \qquad (9)
\end{aligned}
$$

where the subscripts b_p, s_p, $\mathrm{b}_p\mathrm{b}_p$, and $\mathrm{s}_p\mathrm{s}_p$ denote gradients and Hessians, except for g_k and W_k where they denote Jacobians. In the partial belief space, the partial derivatives are only taken with respect to b_p and s_p. This creates smaller matrices which match dimensions with the partial belief dynamics g_k and W_k. Dropping the k for notation convenience, we recover partial derivatives from (6), (7), (9)

$$
Q_{\mathrm{s}_p}^i = \begin{bmatrix} Q_{\mathrm{b}_p}^i \\ Q_{\mathrm{u}^1}^i \\ \vdots \\ Q_{\mathrm{u}^N}^i \end{bmatrix} Q_{\mathrm{s}_p\mathrm{s}_p}^i = \begin{bmatrix} Q_{\mathrm{b}_p\mathrm{b}_p}^i & Q_{\mathrm{b}_p\mathrm{u}^1}^i & \cdots & Q_{\mathrm{b}_p\mathrm{u}^N}^i \\ Q_{\mathrm{u}^1\mathrm{b}_p}^i & Q_{\mathrm{u}^1\mathrm{u}^1}^i & \cdots & Q_{\mathrm{u}^1\mathrm{u}^N}^i \\ \vdots & \vdots & \ddots & \vdots \\ Q_{\mathrm{u}^N\mathrm{b}_p}^i & Q_{\mathrm{u}^N\mathrm{u}^1}^i & \cdots & Q_{\mathrm{u}^N\mathrm{u}^N}^i \end{bmatrix}, \tag{10}
$$

$$
\hat{Q}_{\mathrm{uu}} = \begin{bmatrix} Q_{\mathrm{u}^1\mathrm{u}}^1 \\ Q_{\mathrm{u}^2\mathrm{u}}^2 \\ \vdots \\ Q_{\mathrm{u}^N\mathrm{u}}^N \end{bmatrix}, \hat{Q}_{\mathrm{ub}_p} = \begin{bmatrix} Q_{\mathrm{u}^1\mathrm{b}_p}^1 \\ Q_{\mathrm{u}^2\mathrm{b}_p}^2 \\ \vdots \\ Q_{\mathrm{u}^N\mathrm{b}_p}^N \end{bmatrix}, \hat{Q}_u = \begin{bmatrix} Q_{\mathrm{u}^1}^1 \\ Q_{\mathrm{u}^2}^2 \\ \vdots \\ Q_{\mathrm{u}^N}^N \end{bmatrix}, \tag{11}
$$

and define our linear feedback policy from [1] to be

$$
\pi_k = \bar{\mathrm{u}}_k + j_k + K_k \delta\mathrm{b}_{p,k}, \tag{12}
$$

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, $K_k = -\hat{Q}_{\mathrm{uu}}^{-1}\hat{Q}_{\mathrm{ub}_p}$ is the feedback term, and $\delta\mathrm{b}_{p,k}$ is the difference between the predicted and current partial belief at timestep k. In the full belief space, this linear feedback policy is proven to return a local Nash Equilibrium control trajectory for all agents [1]. In addition, deviations of any agent from the predicted beliefs will change the policy, allowing for more robust trajectory control of the ego agent. We note that the linear feedback policy for partial belief space only accounts for differences in the partially propagated beliefs. This means that non-propagated beliefs are not taken into account in the updated linear feedback policy. We use quadratic approximations from [1] to formulate the backwards equations. The value functions

V^i are now partially propagated backwards as

$$V_k^i = Q^i + Q_{\mathrm{u}}^{i,\top} j_k + \frac{1}{2} j_k^\top Q_{\mathrm{uu}}^i j_k, \tag{13}$$

$$V_{\mathrm{b}_p,k}^i = Q_{\mathrm{b}_p}^i + K_k^\top Q_{\mathrm{uu}}^i j_k + K_k^\top Q_{\mathrm{u}}^i + Q_{\mathrm{ub}_p}^{i,\top} j_k, \tag{14}$$

$$V_{\mathrm{b}_p\mathrm{b}_p,k}^i = Q_{\mathrm{b}_p\mathrm{b}_p}^i + K_k^\top Q_{\mathrm{uu}}^i K_k + K_k^\top Q_{\mathrm{ub}_p}^i + Q_{\mathrm{ub}_p}^{i,\top} K_k, \tag{15}$$

$$V_l^i = c_l^i(\bar{\mathrm{b}}_{p,l}), \quad V_{\mathrm{b}_p,l}^i = \left.\frac{\partial c_l^i(\mathrm{b}_p)}{\partial \mathrm{b}_p}\right|_{\mathrm{b}_p=\bar{\mathrm{b}}_{p,l}},$$
$$V_{\mathrm{b}_p\mathrm{b}_p,l}^i = \left.\frac{\partial^2 c_l^i(\mathrm{b}_p)}{\partial \mathrm{b}_p^2}\right|_{\mathrm{b}_p=\bar{\mathrm{b}}_{p,l}}. \tag{16}$$

C. Policy Regularization

We implement a Levenberg-Marquardt style regularization [19] similar to [1] and [20] in order to ensure convergence to a policy in two parts: control and belief regularization.

$$\tilde{Q}_{\mathrm{uu}}^i = \hat{Q}_{\mathrm{uu}}^i + \mu_{\mathrm{u}} I, \tag{17}$$

$$Q_{\mathrm{s}_p\mathrm{s}_p,k}^i = c_{\mathrm{s}_p\mathrm{s}_p,k}^i + g_{\mathrm{s}_p,k}^\top (V_{\mathrm{b}_p\mathrm{b}_p,k+1}^i + \mu_b I) g_{\mathrm{s}_p,k} + \sum_{j=1}^{n_x} W_{\mathrm{s}_p,k}^{(j),\top} (V_{\mathrm{b}_p\mathrm{b}_p,k+1}^i + \mu_b I) W_{\mathrm{s}_p,k}^{(j)}, \tag{18}$$

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.

As the number of agents increase, using automatic differentiation to obtain g_s and W_s becomes infeasible due to memory limitations. As such, we utilize finite differences for these terms. This can result in slight errors propagated through our policy, causing the regularization from (17), (18) to sometimes converge near the nominal policy. To ensure convergence to the nominal policy and thus a minimum, we implement a conditional line search similar to [13]

$$\pi_k = \bar{\mathrm{u}}_k + \alpha j_k + K_k \delta \mathrm{b}_{p,k}, \tag{19}$$

where $\alpha = 1$ when unregularized. When 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. Per [20], our control and belief regularization quadratically converge when far from a minimum. Then per [13], the conditional line search linearly converges if regularization is detected to be ineffective near the minimum.

IV. Performance of Partially Propagated Beliefs

The prior section outlined partial belief space planning for stochastic dynamic games. In this section, we explain how the partial belief space can be leveraged to improve runtime performance in the iLQG from [1] while proposing that under certain conditions, it is possible to maintain consistent performance as compared to the full belief space planning.

A. Partial Belief Planner versus Full Belief Planner

We seek to compare the partial belief space iLQG to the full belief space iLQG. We show that the partial belief space planner can return equivalent trajectories to the full belief space planner when non-propagated values are unchanging.

Proposition 1. *If b_p contains all beliefs which change, then the partial belief space planner yields the same predicted trajectories as the iLQG planner in [1].*

Proof. Without loss of generality, $\mathrm{b} = [\mathrm{b}_p^\top, \mathrm{b}_{np}^\top]^\top$. If b_p contains all changing values, then b_{np} is unchanging. Trivially, if all beliefs changed, then $\mathrm{b}_p = \mathrm{b}$ and the algorithm is identical to [1]. Now consider if there exist some beliefs that do not change, such that b_{np} is non-empty. Then, the Jacobians and Hessians of b_{np} are 0. Since the iLQG utilizes the Jacobians and Hessians of each belief in the backwards pass to derive a policy through multiplication and addition, these values do not influence the linear feedback policy. It can be shown the resulting policy is the same from equations in Section III-B. □

Remark 1. *By only propagating b_p, the dimensionality of the backward pass equations is reduced. In the case where b_{np} is unchanging, this results in equal performance to the full belief space planner with improved runtime.*

Thus, using the partial belief propagation means assuming all nonpropagating beliefs are constant. Constant uncertainty values exist in many multiagent systems, such as the special case where all agents have independent process and measurement equations as in (4). Such systems will have covariance values of 0 for the states of any two different agents.

B. Partial Belief Space Planning iLQG implementation

Our modified algorithm to solve for the Nash Equilibrium is very similar to the algorithm presented in [1], except that we compute these terms over the partially propagated belief space. For space constraints, we omit this algorithm and refer the reader to [1]. We utilize CasADi [21] to formulate b, then only select the elements b_p for propagation in our belief dynamics. CasADi allows for automatic differentiation and compute graph/static C code generation, allowing us to easily compute individual elements. The belief dynamics can thus take $\mathrm{b}_{p,k}$ and b_{np} as inputs, and return $\mathrm{b}_{p,k+1}$ as an output. b_p is propagated through the forward passes of the algorithm, and the backward pass equations use gradients, Jacobians, and Hessians with reduced dimensions. The iterations between backwards and forwards passes continues until convergence to a policy. As such, an agent can predict a trajectory in the partial belief space and actuate, then update its full state and uncertainty through a Bayesian Filter.

C. Dominant Runtime Analysis

While the dominant runtime complexity of partial belief space planning is bounded to that of the full belief space planner as $\mathcal{O}(lN^7 n_{\mathrm{x}}^{i,6})$ [1], we examine the case where each agent has independent dynamics and measurement equations to show the potential benefits. We define N as the number

TABLE I

AVERAGED PERFORMANCE FOR 4 AGENT INFO CIRCLE SWAP: FULL VS. PARTIAL BELIEF SPACE PLANNING

Mean $\pm 1\sigma$ for $N = 50$ Trials	Full Beliefs Propagated	Nonzero Uncertainty Beliefs Propagated	Positional Uncertainty Beliefs Propagated	No Uncertainty Beliefs Propagated
Iteration Time (s)	$.1854 \pm .0032$	$.0572 \pm .0016$	$.0309 \pm .0044$	$.0233 \pm .0002$
Distance traveled to Goal (m)	10.1615 ± 1.1126	$10.1525 \pm .9127$	10.5166 ± 1.3228	11.5317 ± 1.6441
Minimum Distance to Center (m)	$1.2485 \pm .6916$	$1.2759 \pm .6168$	$1.4135 \pm .6518$	$1.7867 \pm .6623$
Time to Goal(s)	$4.4295 \pm .6715$	$4.4295 \pm .5767$	$4.4224 \pm .6782$	$4.5065 \pm .7567$
Total u_{acc} (m/s^2)	$6.0300 \pm .9213$	$5.9643 \pm .7806$	6.2206 ± 1.4474	6.7937 ± 1.3531
Total u_{ste} (rad)	$0.7174 \pm .3298$	$.7545 \pm .3485$	$.7578 \pm .3485$	$.8147 \pm .3046$

of total agents in the system. 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 full 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. The reduced belief state for independent agents contains $N(n_x^i + n_x^{i,2}/2)$ elements, or $\mathcal{O}(Nn_x^{i,2})$ elements. Following similar analysis to [1], we find a computational bottleneck when evaluating the action-value function $Q_{\mathrm{ss},k}^i$ in (9). The term $g_{\mathrm{s},k}^\top V_{\mathrm{bb},k+1}^i g_{\mathrm{s},k}$ requires a matrix multiplication of dimension $\mathcal{O}(Nn_x^{i,2}) \times \mathcal{O}(Nn_x^{i,2}) = \mathcal{O}(N^3 n_{\mathrm{x}}^{i,6})$ complexity. This operation is completed for all agents, meaning a full iteration of this algorithm has a dominant runtime complexity of $\mathcal{O}(lN^4 n_{\mathrm{x}}^{i,6})$, where l is the planning horizon. The original algorithm's runtime complexity is $\mathcal{O}(lN^7 n_{\mathrm{x}}^{i,6})$, making partial belief space planning a far more attractive approach.

V. SIMULATIONS AND EXPERIMENTS

In this section, we examine the utility of partial belief space planning through simulations and experiments. Our computations are done with Matlab and CasADi [21]. Simulations were run on an Intel Core i7-13700KF at 3.4 GHz while experiments were run on an Intel i9-9900K at 3.6GHz. Agents for all simulations were run from a cold start nominal input sequence of all zeros while experiments were hot started with the previous input trajectory to speed up convergence rate. Calculations, measurements, and actuations for each agent were parallelized and taken separately with independent Gaussian noise to mimic real life conditions.

We choose to use an informative circle swap for our comparisons as seen in Figure 2. This game incentivizes information seeking agents to take riskier trajectories to localize themselves before proceeding to their goal. When agents are not information seeking, they take safer trajectories like the standard iLQG in Figure 1.

A. Simulation Results: Informative Circle Swap

We compare partial belief space planner performance through a circle swap with an information source placed in between all agents. We give all agents car-like dynamics with states $\mathrm{x}^i = [x^i, y^i, v^i, \theta^i]$, where (x, y) is position, v is velocity, and θ is orientation. The control inputs $\mathrm{u}^i = [\mathrm{u}_{acc}^i, \mathrm{u}_{ste}^i]$ denote acceleration u_{acc}^i and steering angle u_{ste}^i. We encode the dynamics of 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

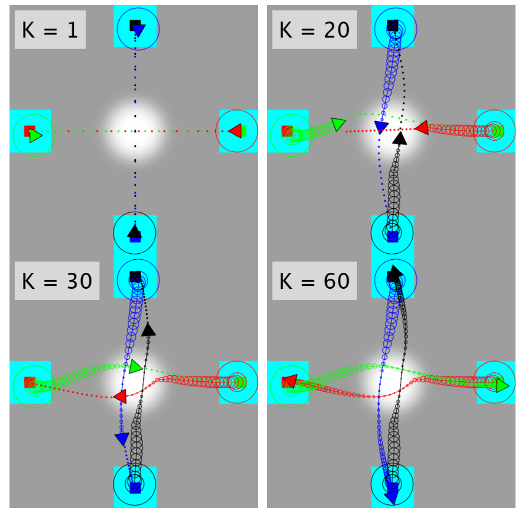

Fig. 2. An informative circle swap with an information source in the center solved using partial belief space planning. All agents attempt to localize within the source rather than take safer trajectories towards their goals.

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

$$\begin{aligned} c_k(\mathrm{b}_k, \mathrm{u}_k) &= \mathrm{u}_k^\top R \mathrm{u}_k + \beta_k \det(\Sigma_{xy,k}) \\ &\quad + \mathrm{c}_{coll}(\mathrm{x}_k) + \gamma_k \|d_{go}\|^2, \\ c_l(\mathrm{b}_l) &= \beta_l \det(\Sigma_{xy,l}) + \gamma_l \|d_{go}\|^2, \end{aligned} \tag{20}$$

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 agents to noisy position measurements, with more precise measurments when near the information source. The observation model becomes $\mathrm{z}_k^i = [x^i, y^i]^\top + N(\mathrm{x}_k^i)\cdot \mathrm{n}_k^i$. We note that this system model of dynamics and measurements means the states of any two agents i and j are independent, and thus the covariance between their states(i.e. $\Sigma_{x^i x^j}$) are always 0.

We start all agents approximately equal distances from their goal and place the information source inbetween them. The resulting behaviors in Figure 2 show agents complete their tasks while localizing themselves near the information source. We compare the behavior of agents through 4 different controllers: full belief space planning (FBS), partial belief space planning where only non-zero/changing uncertainties

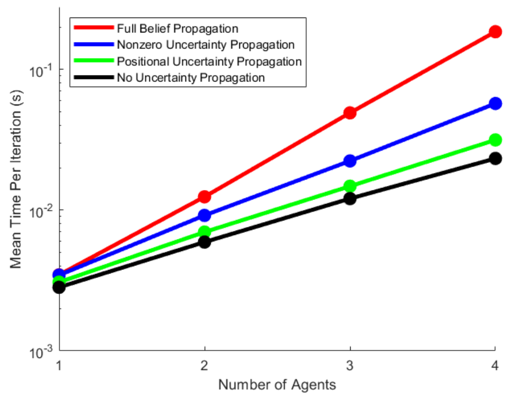

Fig. 3. Log plot of mean time per iteration for agents in a simulated informative circle swap. We compare full belief space propagation, partial belief space planners, and a regular iLQG without uncertainty propagation. Data points were obtained from the mean iteration times over 50 runs.

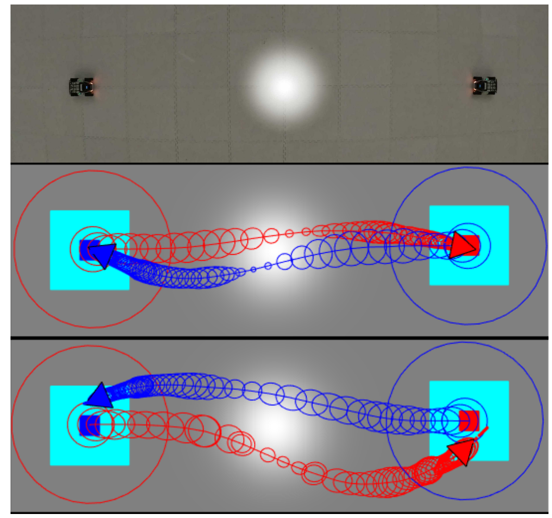

Fig. 4. Experimental results of a two agent informative circle swap solved in realtime using partial belief space planning. Our method (middle) allowed both agents to avoid collisions and localize themselves at the information source before reaching to their goal. If information gain is ignored (bottom), agents do not localize as much and therefore make worse decisions when trying to accurately arrive to their goals.

are propagated(NUB), partial belief space planning where only positional uncertainties are propagated(PUB), and a regular iLQG with no uncertainties propagated. The mean results of 50 trials for each planner in a 4 agent informative circle swap are presented in Table I.

The FBS planner performs effectively the same as the NUB planner, as the partial derivatives of all zero uncertainty values are 0. The deviations are caused by noise in the system. However, the NUB planner runs much faster due to not attempting to propagate the zero beliefs. This could also be achieved by preallocating uncertainty values as 0, however this method would not allow changing beliefs to be fixed. In cases where computation speed is essential, the PUB planner performs faster while still attempting to localize near the information source. While it travels a further distance with more effort and less localization, it does so at $\approx 54\%$ of the iteration time of the NUB planner. Furthermore, the PUB planner performs better than the standard iLQG with no belief space planning by localizing near the center more. This shows the tradeoff between using a partial belief space iLQG and the necessity of choosing to propagate important beliefs that impact the an agent's performance.

We compare the mean time per iteration for our various planners and present the results in Figure 3. These results are consistent with our runtime analysis in Section IV-C and show polynomial decreases in iteration time using partial planners. Furthermore, they are bounded by an iLQG without uncertainty propagation and the full belief space planner. We also note that the number of iterations to converge was very similar between all planners throughout the simulation, meaning the computation time saved per iteration was consistent with the overall computation time saved per timestep.

B. Hardware Experiments: Informative Circle Swap

We ran our partial belief space algorithm on two AgileX LIMO[1] robots, with robot positions known via Optitrack[2] motion capture. Received measurements and actuations were artificially corrupted with noise. Computations were parallelized on a central computer (8-core, 32GB RAM, Windows 10), which received continuous updates about robot poses and sent control inputs to each robot at ∼2.2Hz. In the case where an agent did not converge to an optimal policy in time, the best current calculated policy was used. We utilized dynamics, measurement equations, and cost functions identical to our simulations with differently tuned parameters.

Our mean iteration time for a 2 agent system with NUB planning was 0.0228 seconds. We attribute this increase in iteration time from our simulations to the computer needing to stream each robots pose continuously in parallel while computing new policies. As seen in our video, agents were capable of localizing themselves within the information source while avoiding collisions. When agents do not consider information gain in their cost function, they take safe routes which do not result in localization. This lack of localization means they do not reach their final goal as accurately. We note that using the full belief space would result in larger iteration times and thus less iterations per time step before the best attempt policy is picked. In practice, this can lead to poor policies which are unable to accomplish tasks and prevent collisions.

VI. Conclusion and Future Work

In this paper, we introduce partial belief space planning and show its utility through simulations and hardware experiments. Our findings show partial belief space planning can lead to polynomial runtime improvements, and move the iLQG planner from [1] further into real time applications. Future work will focus on varying which beliefs are propagated at each timestep. This would allow for planners which are capable of reducing computation times online depending on the beliefs about other agents and the environment.

[1] https://global.agilex.ai/products/limo
[2] https://optitrack.com/

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] F. Laine, D. Fridovich-Keil, C.-Y. Chiu, and C. Tomlin, "Multi-Hypothesis Interactions in Game-Theoretic Motion Planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 8016–8023, iSSN: 2577-087X. [Online]. Available: https://ieeexplore.ieee.org/document/9561695

[3] S. Amatya, M. Ghimire, Y. Ren, Z. Xu, and W. Zhang, "When Shall I Estimate Your Intent? Costs and Benefits of Intent Inference in Multi-Agent Interactions," in *2022 American Control Conference (ACC)*, Jun. 2022, pp. 586–592, iSSN: 2378-5861. [Online]. Available: https://ieeexplore.ieee.org/document/9867155

[4] O. So, P. Drews, T. Balch, V. Dimitrov, G. Rosman, and E. A. Theodorou, "MPOGames: Efficient Multimodal Partially Observable Dynamic Games," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 3189–3196. [Online]. Available: https://ieeexplore.ieee.org/document/10160342

[5] 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

[6] 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

[7] 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

[8] 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.

[9] 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.

[10] 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

[11] 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

[12] U. o. W. S. H. Omid Madani, "On the Undecidability of Probabilistic Planning and Infinite-Horizon Partially Observable Markov Decision Problems." [Online]. Available: https://aaai.org/papers/077-aaai99-077-on-the-undecidability-of-probabilistic-planning-and-infinite-horizon-partially-observable-markov-decision-problems/

[13] 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

[14] 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

[15] 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.

[16] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, Jun. 2015. [Online]. Available: https://doi.org/10.1007/s10107-015-0892-3

[17] D. Bertsekas, *Nonlinear Programming*, ser. Athena scientific optimization and computation series. Athena Scientific, 2016. [Online]. Available: https://books.google.com/books?id=rC1EEAAAQBAJ

[18] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, ser. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, Jan. 2000. [Online]. Available: https://epubs.siam.org/doi/book/10.1137/1.9780898719468

[19] 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/

[20] 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.

[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