IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Leadership Inference for Multi-Agent Interactions

Hamzah I. Khan and David Fridovich-Keil

***Abstract*—Effectively predicting intent and behavior requires inferring leadership in multi-agent interactions. Dynamic games provide an expressive theoretical framework for modeling these interactions. Employing this framework, we propose a novel method to infer the leader in a two-agent game by observing the agents' behavior in complex, long-horizon interactions. We make two contributions. First, we introduce an iterative algorithm that solves dynamic two-agent Stackelberg games *with nonlinear dynamics and nonquadratic costs*, and demonstrate that it consistently converges in repeated trials. Second, we propose the Stackelberg Leadership Filter (SLF), an online method for identifying the leading agent in interactive scenarios based on observations of the game interactions. We validate the leadership filter's efficacy on simulated driving scenarios to demonstrate that the SLF can draw conclusions about leadership that match right-of-way expectations.**

***Index Terms*—Leadership Inference, Stackelberg Games, Optimization and Optimal Control, Probabilistic Inference**

I. Introduction

DURING daily commutes, drivers assert themselves in running negotiations with other road users in order to reach their destinations quickly and safely. Right-of-way expectations inform these assertions between road users. Consider the passing lane shown in Fig. 1. Agent $\mathcal{A}_2$ (blue) initially follows behind agent $\mathcal{A}_1$ (red), and we may intuitively perceive $\mathcal{A}_1$ as the leader. If instead $\mathcal{A}_2$ overtakes $\mathcal{A}_1$, the scenario seems to imply a reversal of leadership, with $\mathcal{A}_2$ in front and $\mathcal{A}_1$ behind, as in the inset of Fig. 1. However, this intuition is vague and premature. If $\mathcal{A}_2$ tailgates $\mathcal{A}_1$ or otherwise behaves aggressively, $\mathcal{A}_1$ might speed up or yield to $\mathcal{A}_2$ out of caution. However, aggressive behavior does not necessarily indicate leadership, as $\mathcal{A}_1$ could also react to $\mathcal{A}_2$ tailgating by slowing down and relying on the knowledge that $\mathcal{A}_2$ will not risk a collision. Here, any simple intuition of the leadership dynamics falls short. Depending on each driver's safety and comfort tolerances, either $\mathcal{A}_1$ or $\mathcal{A}_2$ may be the leader. Hence, deciphering leadership dynamics requires understanding common expectations, agent incentives, and other agents' actions. Successfully doing so can improve autonomous intent and behavior prediction for motion planning, as shown by [1].

We turn to optimal decision making and game theory for tools to analyze interactive scenarios. Stackelberg games [2], also known as leader-follower games, stand out because they model interactions with clear leadership hierarchies. In a Stackelberg game, a leader selects its strategy to influence the follower's response. Each strategy in a Stackelberg solution satisfies leadership conditions that describe how the leader's behavior induces the follower to act. Additionally, solving Stackelberg games results in trajectories that we can use for model-predictive control. Using these attractive properties, we propose a leadership inference technique for multi-agent scenarios like that of Fig. 1.

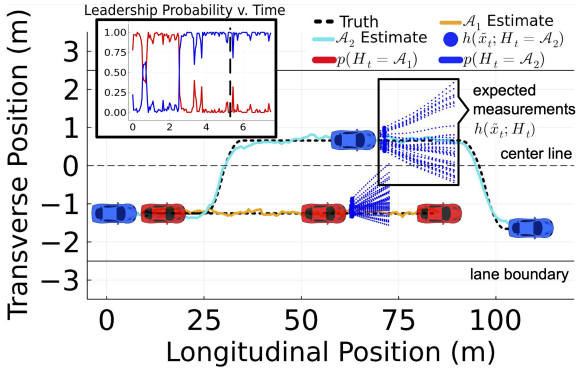


Fig. 1. Agents $\mathcal{A}_1$ (red) and $\mathcal{A}_2$ (blue) initially proceed along the same lane of a two-way road at similar speeds. While $\mathcal{A}_2$ is behind $\mathcal{A}_1$, the SLF infers that $\mathcal{A}_1$ is the leader. During $\mathcal{A}_2$'s passing maneuver, the SLF captures the leadership probability shifting to $\mathcal{A}_2$. The dashed line in the inset indicates the probabilities at the current time. We display the current expected measurements generated by the measurement model h. The blue coloring indicates that most particles in the SLF believe $\mathcal{A}_2$ is the leader.

To this end, we first contribute Stackelberg Iterative Linear-Quadratic Games (SILQGames), an algorithm for solving dynamic Stackelberg games, and we empirically show that it converges for games with nonlinear dynamics and general costs. Second, we propose the Stackelberg Leadership Filter (SLF) to infer leadership over time in interactions based on observations of the agents. We validate that it infers the correct Stackelberg leader in two-agent games and report results on simulations of driving scenarios.

II. Related Work

Leadership Inference. Many prior works develop leadership inference techniques, particularly for robotic swarms and animal sociology. As an abstract concept, leadership is challenging to measure [3], [4]. Leadership models prespecify particular agent(s) as leaders that influence group motion. Swarm applications [5], [6] often assume the Reynolds flocking model [7]. Animal sociology applications define leadership models based on principal component analysis [8] or stochastic inference [9] with hand-selected domain-specific features. By contrast, we explicitly frame these interactions in terms of optimal decision making and game theory and therefore utilize the Stackelberg leadership model. Defining a Stackelberg game requires prespecifying a leader and solving one produces

Manuscript received: October 26, 2023; Revised: January 26, 2024; Accepted: February 19, 2024. This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Science Foundation under Grant No. 2211548. The authors ({hamzah, dfk}@utexas.edu) are with the *Department of Aerospace Engineering and Engineering Mechanics*, University of Texas at Austin.

Digital Object Identifier (DOI): 10.1109/LRA.2024.3381469

equilibrium trajectories for each agent. Hence, by associating a particular leader with solution trajectories in a principled manner, Stackelberg leadership allows for modeling leadership over long-horizon interactions without hand-crafted heuristics.

Stackelberg Games for Motion Planning. Recent advances [10], [11] investigate Stackelberg models of leadership for interactive scenarios involving self-driving vehicles. In particular, Tian *et al.* [1] incorporate leadership as a latent variable by solving open-loop Stackelberg games and comparing expected leader and follower behaviors with observed agent behaviors. Our method generalizes this underlying approach to Stackelberg leadership by modeling a joint distribution over game state and leadership. We solve feedback Stackelberg games for richer access to leadership information.

Solving Dynamic Games. Identifying computationally efficient game-solving techniques with theoretical guarantees of finding equilibria remains an open area of research. Most existing game-solving algorithms consider Nash games, which find equilibria for which each actor is unilaterally optimal given fixed opponent strategies. These algorithms [12]–[17] generally use Newton-based schemes based on iterative and dynamic programming algorithms that have been widespread for decades [18], [19]. We note two axes on which such approaches differ: first, these approaches solve either open-loop games [12]–[15] or feedback games [12], [16], [17]. Second, these algorithms either reduce the game to a simpler problem [15] or directly solve the game [12]–[14], [16], [17]. In particular, Fridovich-Keil *et al.* [16] introduce Iterative Linear-Quadratic Games (ILQGames), an iterative method that approximates solutions to nonlinear dynamic, nonquadratic cost feedback Nash games by repeatedly solving linear-quadratic (LQ) approximations until convergence. Convergence analysis of these methods is subtle, as described in depth by Laine *et al.* [20]. Our work is closely related to [17], which uses ILQ schemes to solve for feedback Stackelberg equilibria. Both our work and [17] utilize similar approaches as [16] to solve for feedback Stackelberg equilibria, a different solution concept than (1) single-agent optima found by Iterative Linear Quadratic Regulation (ILQR) or DDP and (2) feedback Nash equilibria found by [16].

III. Problem Formulation

Let $N = 2$ agents, $\mathcal{A}_1$ and $\mathcal{A}_2$ (e.g., vehicles), operate in a shared n-dimensional state space with state x_t at each time $t \in \mathbb{T} \equiv \{1, 2, \dots, T\}$ and sampling period Δt. Agent $\mathcal{A}_i$ has controls $u_t^{(i)} \in \mathbb{R}^{m^{(i)}}$. The state evolves according to

$$x_{t+1} = f_t\left(x_t, u_t^{(1)}, u_t^{(2)}\right). \tag{1}$$

We denote sequence of states $\boldsymbol{x}_{1:T} = (x_1, x_2, \dots, x_T)$ and $\boldsymbol{u}_{1:T}^{(i)} = (u_1^{(i)}, u_2^{(i)}, \dots, u_T^{(i)})$ as the sequence of $\mathcal{A}_i$'s controls. We assume that f_t is continuous and continuously differentiable in $x_t, u_t^{(1)}, u_t^{(2)}$. $\mathcal{A}_i$'s objective,

$$J^{(i)}\left(\boldsymbol{x}_{1:T}, \boldsymbol{u}_{1:T}^{(1)}, \boldsymbol{u}_{1:T}^{(2)}\right) \equiv \sum_{t=1}^{T} g_t^{(i)}\left(x_t, u_t^{(1)}, u_t^{(2)}\right), \tag{2}$$

describes its preferences in a given scenario. We model the objective (2) as the sum of stage costs $g_t^{(i)}$, assumed to be twice differentiable in $x_t, u_t^{(1)}, u_t^{(2)}$. Each agent $\mathcal{A}_i$ minimizes its objective with respect to its controls $\boldsymbol{u}_{1:T}^{(i)}$.

A. Background: Feedback Stackelberg Games

Stackelberg games model leadership as a mismatch of information. Intuitively, the leader $\mathcal{A}_L$ commits to a strategy and communicates it to the follower $\mathcal{A}_F$. Given this relationship, the leader carefully selects its strategy in order to influence the follower.

Formally, a Stackelberg equilibrium $\{\boldsymbol{u}_{1:T}^{(L*)}, \boldsymbol{u}_{1:T}^{(F*)}(\boldsymbol{u}_{1:T}^{(L*)})\}$ is a tuple of optimal control trajectories for both agents. The function $\boldsymbol{u}_{1:T}^{(F*)}(\boldsymbol{u}_{1:T}^{(L)})$ highlights that $\mathcal{A}_F$'s optimal strategy depends on the leader's (possibly non-optimal) chosen strategy. Using an abuse of notation, we omit the state argument of $\mathcal{A}_i$'s objective $J^{(i)}$, and define $\boldsymbol{\gamma}(u_t^{(i)}) \equiv [\boldsymbol{u}_{1:t-1}^{(i)}, u_t^{(i)}, \boldsymbol{u}_{t+1:T}^{(i*)}]$, containing arbitrary controls from time 1 to $t-1$, control $u_t^{(i)}$ passed as a parameter, and an equilibrium strategy $\boldsymbol{u}_{t+1:T}^{(i*)}$ from time $t+1$ to T. We note that the game dynamics ensure that $\boldsymbol{u}_{t+1:T}^{(i*)}$ is implicitly a function of the state. We define the set of all optimal follower responses at time t, $U_t^{(F*)}\left(u_t^{(L)}\right) \subset \mathbb{R}^{m^{(F)}}$, as

$$U_t^{(F*)}\left(u_t^{(L)}\right) \equiv \underset{u_t^{(F)}}{\operatorname{argmin}}\, J^{(F)}\left(\boldsymbol{\gamma}\left(u_t^{(L)}\right), \boldsymbol{\gamma}\left(u_t^{(F)}\right)\right). \tag{3}$$

We assume $|U_t^{(F*)}(u_t^{(L*)})| = 1$, i.e., that an *optimal* leader strategy results in a unique optimal follower response at each time t. Under this assumption, the set of control trajectories for all agents forms a *feedback Stackelberg equilibrium* if, at every time $t \in \mathbb{T}$, the optimal trajectories satisfy

$$J^{(L)}\left(\boldsymbol{\gamma}\left(u_t^{(L*)}\right), \boldsymbol{\gamma}\left(u_t^{(F*)}\right)\right) = \min_{u_t^{(L)}} \max_{u_t^{(F)} \in U_t^{(F*)}\left(u_t^{(L)}\right)} J^{(L)}\left(\boldsymbol{\gamma}\left(u_t^{(L)}\right), \boldsymbol{\gamma}\left(u_t^{(F)}\right)\right). \tag{4}$$

Since the follower knows the leader's controls at time t, (3) ensures that the follower produces a best response at time t. Next, (4) ensures that the leader's strategy guides the follower towards its least bad option for the leader at time t.

Stackelberg games are generally *non-cooperative*, meaning that agents do not coordinate but plan based on observations of the game state. Agents in open-loop games observe only the initial game state, whereas in *feedback* games, agents adjust their control inputs after observing the state at each time step, producing complex, temporally-nested game constraints (3) and (4). LQ Stackelberg games have analytic solutions given strictly convex costs [21, Eq. 7.14-15].

We denote $\mathcal{S}_T^i(x_t)$ as the T-horizon Stackelberg game solved from state x_t with leader $\mathcal{A}_i$. For a more detailed treatment of Stackelberg equilibria and solving LQ Stackelberg games, refer to Başar and Olsder [21, Ch. 3, 7].

B. Stackelberg Leadership Filtering

We seek to describe a filter that identifies a leadership belief for $\mathcal{A}_i$ based on observations. To this end, we define $H_t \in \{1, 2\}$ to be a binary random variable (RV) indicating the leader at time t. Next, we state our assumptions about the game's observability. We assume state x_t is observable via noisy measurement $z_t \sim \mathcal{N}(h(x_t; H_t), \Sigma_t)$ with known covariance matrix $\Sigma_t \succ 0$ and measurement model h. We also assume that control inputs $u_t^{(i)}$ for each agent $\mathcal{A}_i$ are directly

observable. Next, recall that each agent has an objective that describes its preferences. For this work, we assume all agent objectives $\{J^{(i)}\}$ are known a priori. In general settings, we note that techniques exist [22]–[24] to infer agent objectives from noisy observations, though further work may be required to confirm the computational tractability of simultaneously inferring leadership and objectives. We define the leadership belief for H_t as $b(H_t) = p\{H_t | \boldsymbol{z}_{1:t}\}$.

IV. Inferring Leadership

We propose Stackelberg Iterative Linear-Quadratic Games (SILQGames), which iteratively solves nonlinear dynamic, general cost (non-LQ) Stackelberg games with continuous and differentiable dynamics and costs. We use SILQGames in the Stackelberg Leadership Filter (SLF, Fig. 2) as part of the Stackelberg leadership model. Our method infers the leading agent of a two-agent interaction from observations.

A. Iteratively Solving Stackelberg Games

At a high level, SILQGames (Alg. 1) iteratively solves LQ approximations of Stackelberg games (lines 4 to 8), updates the control trajectories using the solutions to these approximated games (line 9), and terminates if the updated trajectory satisfies a convergence condition (lines 10 to 12). Upon successful convergence, the resulting trajectory constitutes an approximate Stackelberg equilibrium. This type of approach also yields approximate equilibrium solutions in the Nash case, although establishing precise error bounds remains an open problem [20]. We expect a similar result for SILQGames, though it is beyond the scope of this work.

Algorithm 1 Stackelberg Iterative Linear-Quadratic Games

Input: leader $\mathcal{A}_L$, initial state x_1, nominal strategies $\left\{\boldsymbol{u}_{1:T}^{(i),0}\right\}$

Output: converged strategies $\left\{\boldsymbol{u}_{1:T}^{(i),s-1}\right\}$

1: $\boldsymbol{x}_{1:T}^0 \leftarrow$ applyGameDynamics $\left(x_1, \left\{\boldsymbol{u}_{1:T}^{(i),0}\right\}\right)$
2: $\alpha_s \leftarrow \alpha_1$
3: **for** *iteration* $s = 1, 2, \ldots, M_{\text{iter}}$ **do**
4: $\quad \boldsymbol{F}_{1:T} \equiv \{A, B^{(i)}\}_{t=1:T}$ (5a, 5d)
5: $\quad\quad \leftarrow$ linearizeDynamics $\left(\boldsymbol{x}_{1:T}^{s-1}, \left\{\boldsymbol{u}_{1:T}^{(i),s-1}\right\}\right)$
6: $\quad \boldsymbol{G}_{1:T} \equiv \{Q^{(i)}, q^{(i)}, R^{ij}, r^{ij}\}_{t=1:T}$ (5b, 5c, 5e, 5f)
7: $\quad\quad \leftarrow$ quadraticizeCosts $\left(\boldsymbol{x}_{1:T}^{s-1}, \left\{\boldsymbol{u}_{1:T}^{(i),s-1}\right\}\right)$
8: $\quad \boldsymbol{P}_{1:T}^{(i),s}, \boldsymbol{p}_{1:T}^{(i),s} \leftarrow$ solveLQStackelberg$(\{\boldsymbol{F}_{1:T}, \boldsymbol{G}_{1:T}\})$
9: $\quad \boldsymbol{x}_{1:T}^s, \left\{\boldsymbol{u}_{1:T}^{(i),s}\right\} \leftarrow$ stepToward$(\boldsymbol{P}_{1:T}^{(i),s}, \boldsymbol{p}_{1:T}^{(i),s}, \alpha_s)$ (7)
10: $\quad$ **if** $\|\boldsymbol{x}_{1:T}^s - \boldsymbol{x}_{1:T}^{s-1}\|_\infty \leq \tau$ **then**
11: $\quad\quad$ **return** $\boldsymbol{x}_{1:T}^{s-1}, \left\{\boldsymbol{u}_{1:T}^{(i),s-1}\right\}$
12: $\quad$ **end if**
13: $\quad \alpha_{s+1} \leftarrow \max(\alpha_{\min}, \beta\alpha_s)$
14: **end for**

Inputs. SILQGames accepts an initial state x_1 and a leader $\mathcal{A}_L$. It accepts a set of all agents' nominal control trajectories $\{\boldsymbol{u}_{1:T}^{(i),s=0}\}$. We produce a nominal state trajectory $\boldsymbol{x}_{1:T}^0$ by applying the nominal controls from x_1 (line 1).

LQ Game Approximation. At each iteration s, we first linearize the dynamics (lines 4 and 5) and take second-order Taylor series approximations of the costs (lines 6 and 7) about the previous iteration's state and control trajectories, $\boldsymbol{x}_{1:T}^{s-1}, \boldsymbol{u}_{1:T}^{(1),s-1}, \boldsymbol{u}_{1:T}^{(2),s-1}$:

$$A_t = \nabla_x f_t, \quad (5a) \qquad B_t^{(i)} = \nabla_{u^{(i)}} f_t, \quad (5d)$$

$$Q_t^{(i)} = \nabla_{xx}^2 g_t^{(i)}, \quad (5b) \qquad q_t^{(i)} = \nabla_x g_t^{(i)}, \quad (5e)$$

$$R_t^{ij} = \nabla_{u^{(j)}u^{(j)}}^2 g_t^{(i)}, \quad (5c) \qquad r_t^{ij} = \nabla_{u^{(j)}} g_t^{(i)}. \quad (5f)$$

We define the state and control variables for our LQ game approximation as deviations from the previous state and control trajectories: $\delta\boldsymbol{x}_{1:T}^s = \boldsymbol{x}_{1:T}^s - \boldsymbol{x}_{1:T}^{s-1}$ and $\delta\boldsymbol{u}_{1:T}^{(i),s} = \boldsymbol{u}_{1:T}^{(i),s} - \boldsymbol{u}_{1:T}^{(i),s-1}$. We then approximate the game as an LQ problem with linear dynamics and quadratic costs

$$\delta x_{t+1}^s \approx A_t \delta x_t^s + \sum_{i \in \{1,2\}} B_t^{(i)} \delta u_t^{(i),s}, \quad (6a)$$

$$g_t^{(i)}(\cdot,\cdot,\cdot) \approx g_t^{(i)}\left(x_t^{s-1}, u_t^{(1),s-1}, u_t^{(2),s-1}\right) + \frac{1}{2}\|\delta x_t^s\|_{Q_t^{(i)}}^2 + q_t^{(i)\intercal}\delta x_t^s + \sum_{j=1}^{N}\left(\frac{1}{2}\|\delta u_t^{(j),s}\|_{R_t^{ij}}^2 + r_t^{ij\intercal}\delta u_t^{(j),s}\right), \quad (6b)$$

where $\|\cdot\|_M$ is an induced matrix norm. We exclude mixed partials $\nabla_{xu^{(i)}}, \nabla_{u^{(i)}u^{(j)}}$ due to their rarity in cost structures of relevant applications, but they can be included if needed.

In practice, $Q_t^{(i)}$ and R_t^{ij} may not be positive definite. Recall that LQ Stackelberg games have unique global solutions given strictly convex costs. Thus, we enforce positive definiteness, and thus convexity, in the quadratic cost estimates by adding a scaled identity matrix νI to all $Q_t^{(i)}$ and R_t^{ij} terms. This addition increases each eigenvalue by $\nu \in \mathbb{R}_+$ [26, Ch. 3], so a sufficiently large choice of ν guarantees convexity. Finally, we solve the LQ game analytically (line 8) [21, Eq. 7.14-15].

Strategy Update. After approximating the game as LQ and solving it, we update the control strategy (line 9). The analytic solution to the LQ game consists of gain and feedforward terms $\boldsymbol{P}_{1:T}^{(i),s}, \boldsymbol{p}_{1:T}^{(i),s}$ constituting an affine feedback control law that produces strategy $\delta\hat{u}_t^{(i),s} = -P_t^{(i),s}\delta x_t^s - p_t^{(i),s}$. Following standard procedures in ILQR [18], we define update rule

$$u_t^{(i),s} = u_t^{(i),s-1} - P_t^{(i),s}\delta x_t^s - \alpha_s p_t^{(i),s}, \quad (7)$$

where $\alpha_s \in (0, 1]$ is an iteration-varying step size parameter. As α_s approaches 0, the new iterate $u_t^{(i),s}$ approaches the previous iterate $u_t^{(i),s-1}$. Likewise, as α_s approaches 1, we adjust our previous iterate by the full step $\delta\hat{u}_t^{(i),s}$. In single-agent settings, methods like ILQR commonly apply a line search for step size selection. However, this approach requires a detailed description of complex, temporally-nested feedback game constraints (3) and (4). Instead, SILQGames decays the step size (line 13) with configurable decay factor $\beta \in (0, 1)$ and minimum step size $\alpha_{\min}$. Initial step size $\alpha_1 = 1$ unless otherwise specified (line 2).

Convergence Criterion. Optimization algorithms commonly use first-order optimality conditions [26, Ch. 12] to test for convergence, and incorporating a line search guarantees monotone improvement in such a convergence metric. As with a line search, however, using first-order optimality conditions becomes unwieldy due to the feedback game constraints (3)

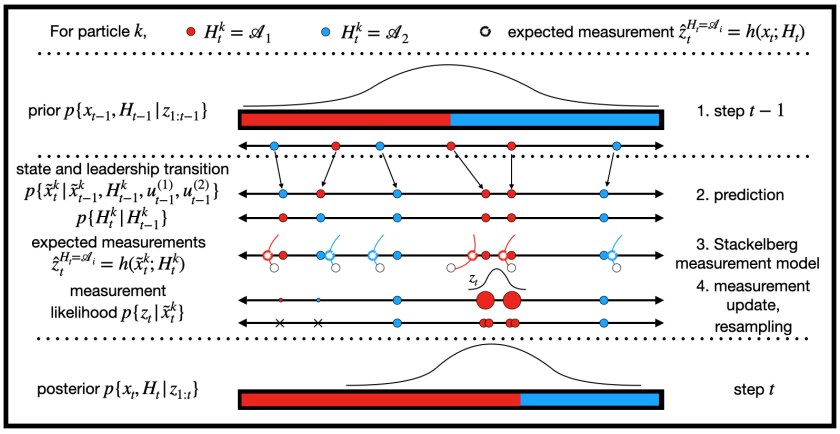


Fig. 2. Each particle in the Stackelberg leadership filter has context $c_t^k = [\tilde{x}_t^k, H_t^k]^\intercal$, where continuous RV $x_t \in \mathbb{R}^n$ describes the state and discrete RV $H_t \in \{1, 2\}$ indicates the leader. 1. At $t-1$, we have a prior distribution over the filter context. For H_{t-1}, the prior is Bernoulli distributed. 2. The continuous state transitions according to game dynamics f_{t-1}. Leadership state evolves stochastically based on a two-state Markov chain. 3. We play a Stackelberg game from each particle's previous state and extract the game state at the current time t as the expected measurement. 4. The algorithm uses a standard particle filter measurement update [25, Ch. 4]. Resampling eliminates unlikely particles and reweights the particle set towards those that are similar to the measurement. Finally, we marginalize over the continuous state and produce a probability of leadership.

and (4). In practice, we define a convergence criterion as a function of the current and next iterate's states:

$$\text{Conv}\left(\boldsymbol{x}_{1:T}^s, \boldsymbol{x}_{1:T}^{s-1}\right) = \left\|\boldsymbol{x}_{1:T}^s - \boldsymbol{x}_{1:T}^{s-1}\right\|_\infty. \tag{8}$$

We compute $\boldsymbol{x}_{1:T}^s$ based on the proposed controls resulting from update step (7). We say SILQGames converges if the metric value falls below a threshold τ. SILQGames stops after a maximum number of iterations M_{iter}, irrespective of convergence. We expect SILQGames to converge, though we do not expect monotone decrease in the convergence criterion as a large step size may occasionally overshoot the Stackelberg equilibrium. Oscillations in the convergence metric can occur when step sizes are consistently too large and may indicate that $\alpha_{\min}$ or β should be reduced. Please refer to our results in Section V-A for further details.

Computational Complexity. The complexity analysis of [16] holds almost identically for SILQGames. For a size-n state, linearizing the dynamics and computing quadratic cost approximations both require taking $O(n^2)$ partial derivatives. Solving the coupled Ricatti equations for the approximate LQ game has complexity $O(n^3)$ for a constant $(N = 2)$ number of agents, so the per-iteration runtime of SILQGames is cubic in n. The entire algorithm runs in $O(sn^3)$, where $s \leq M_{\text{iter}}$ is the number of iterations to convergence.

B. Leadership Filtering

The Stackelberg Leadership Filter (SLF) estimates the likelihood that each agent is the leader of a two-agent interaction given noisy measurements $\boldsymbol{z}_{1:T}$. Let filter context $c_t = [x_t, H_t]^\intercal$ consist of continuous game state x_t and leader H_t. Following conventional Bayesian filtering practices and denoting all agent controls $w_t = \{u_t^{(1)}, u_t^{(2)}\}$ for brevity, the SLF refines prior context belief $b(c_{t-1})$ with update rule

$$b(c_t) \propto p\{z_t | x_t\} \int_{c_{t-1}} p\left\{c_t | c_{t-1}, w_{t-1}\right\} b(c_{t-1}) dc_{t-1}, \tag{9}$$

In (9), the context transition probability term $p\{c_t | c_{t-1}, w_{t-1}\} = p\{x_t, H_t | x_{t-1}, H_{t-1}, w_{t-1}\}$ describes the likelihood of context c_t given the previous context c_{t-1} and each agent's controls. Furthermore, the measurement likelihood $p\{z_t | x_t\}$ quantifies an expected measurement based on how well the new state x_t matches the observation z_t. Thus, we compute the leadership belief at time t by marginalizing $b(c_t) = b(x_t, H_t)$ over x_t:

$$b(H_t) = \int_{x_t} b(x_t, H_t) dx_t. \tag{10}$$

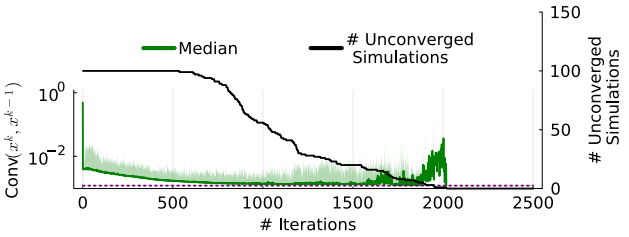
To simplify the context transition probability, we assume conditional independence of x_t and H_t given c_{t-1} and w_{t-1}. While these values often evolve together, we can make this assumption if the state responds slowly to changes in leadership. In particular, if we select a sufficiently small sampling period Δt, then changes in state x_t when $H_t \neq H_{t-1}$ require multiple time steps to observe. After this simplification,

$$p\left\{c_t | c_{t-1}, w_{t-1}\right\} = p\left\{x_t | c_{t-1}, w_{t-1}\right\} p\left\{H_t | c_{t-1}, w_{t-1}\right\}. \tag{11}$$

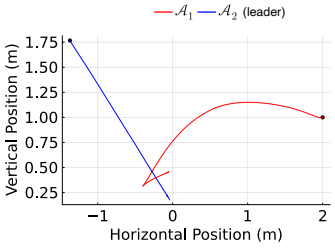
The term $p\{x_t | c_{t-1}, w_{t-1}\}$ indicates that x_t depends on the previous leader and the previous state and controls through the dynamics f_{t-1}. The second term $p\{H_t | c_{t-1}, w_{t-1}\}$ describes how H_t depends on the previous state and controls. In the passing scenario, for example, analyzing this term might allow us to test for a relationship between H_t and whichever vehicle was in front at time $t-1$.

Constructing the SLF as a Bayesian filter first requires a leadership transition process. However, establishing a form for the $p\{H_t | c_{t-1}, w_{t-1}\}$ term is difficult [3], [4], so we leave it to user discretion if such knowledge is available. In the case that a form does not exist, we treat the leadership transition process as a two-state Markov Chain with transition likelihood $p\{H_t \neq H_{t-1} | H_{t-1}\} = p_{\text{trans}}$. In this chain, H_t evolves independently of state x_{t-1} and agent controls w_{t-1}. One example in which this treatment appropriately models leadership is when leadership is correlated with distraction and can thus be modeled as only dependent on time. Despite this simplification in construction, our experiments show the SLF still accounts for the statistical dependence between leadership, state, and controls.

Selecting a Filter. Due to the computational intractability of exactly evaluating Bayesian update rule (9), we use a particle filtering approach. Particle k has context $c_t^k = [\tilde{x}_t^k, H_t^k]^\intercal$. Particle filters use a measurement model to compute the expected observation for a state x_t [25, Ch. 4]. Our measurement model $h(\tilde{x}_t^k; H_t^k)$ solves a Stackelberg game to generate simulated solution trajectories conditioned on the particle's leader. In the measurement update, we compare a subset of the solution to the ground truth observations and update the likelihood of



(a) Median ℓ_∞ convergence metric, with 10th and 90th percentiles.



(b) Stackelberg solution positions.

Fig. 3. We run 100 SILQGames simulations on the non-LQ shepherd and sheep game with leader $\mathcal{A}_2$. The simulations converge in 1133 ± 367 iterations. (a) shows the number of unconverged simulations and (b) shows the solution for one instance.

leadership using (9) and (10). We resample with replacement to eliminate unlikely particles when the effective number of particles, a metric that measures how well the particles represent the distribution, becomes low. We infer the leading agent based on the similarity of expected measurements, generated from Stackelberg games, to observations of the ground truth. Since Stackelberg equilibria satisfy leadership condition (4), converged solutions let the filter observe leadership indirectly via the measurement model.

The Stackelberg Measurement Model. We construct a measurement model that relates the leader H^k_{t-1} in particle k at time $t-1$ with the expected state measurement at time t; in particular, we model the expected measurement from each particle as an equilibrium strategy of game $\mathcal{S}^{H^k_{t-1}}_{T_s}(\tilde{x}^k_{t-1})$. We solve this game with SILQGames over horizon T_s, taking the initial state and leader from previous particle context c^k_{t-1}.

For the third input, we require the user to provide an application-specific function to specify nominal strategies $\boldsymbol{u}^{(i),s}_{t-1:t+T_s-1}$ using previous particles, a heuristic, etc. The SLF calls this function within the measurement model to produce nominal strategies as input to SILQGames. We describe one such heuristic in the appendix. We call the solutions to these games *Stackelberg measurement trajectories* and select the state at time t as the expected measurement.

Next, we clarify a few practical details. First, experiments determine that we must configure T_s carefully, neither too short to provide relevant leadership information nor too long as to cause excessive latency. Second, playing a Stackelberg game from previous state x_{t-1} requires each particle to maintain x_{t-1} as additional context. Third, after producing a measurement trajectory, we attach measurement uncertainty Σ_t to each state in it. Depending on the application, this step may incorporate uncertainty from sensors, processing, etc.

V. Experiments & Results

We first introduce the two-agent LQ shepherd and sheep game [27] and a nonlinear, nonquadratic variant, which we use to validate SILQGames and the SLF. Finally, we run the SLF on realistic driving scenarios.

In the LQ shepherd and sheep game, each agent's state $x^{(i)}_t$ evolves according to planar double-integrator dynamics [28, Eq. 75] discretized at Δt. The game state combines the agent states $x_t = [x^{(1)}_t, x^{(2)}_t]^\intercal$, and each agent controls its horizontal and vertical accelerations. Agents' costs

$$g^{(1)}_t\left(x_t, u^{(1)}_t, u^{(2)}_t\right) = \left(p^{(2)}_{x,t}\right)^2 + \left(p^{(2)}_{y,t}\right)^2 + \|u^{(1)}_t\|^2_2, \quad (12)$$

$$g^{(2)}_t(\ldots) = \left(p^{(1)}_{x,t} - p^{(2)}_{x,t}\right)^2 + \left(p^{(1)}_{y,t} - p^{(2)}_{y,t}\right)^2 + \|u^{(2)}_t\|^2_2, \quad (13)$$

are quadratic in state and controls and incentivize "shepherd" $\mathcal{A}_1$ to minimize "sheep" $\mathcal{A}_2$'s distance to the origin (i.e., the barn) and $\mathcal{A}_2$ to minimize its distance to $\mathcal{A}_1$. We denote the planar positions for $\mathcal{A}_i$ as $p^{(i)}_{x,t}, p^{(i)}_{y,t}$. An analytic Stackelberg solution exists since the game is LQ. When framing the shepherd and sheep game as a Stackelberg game, we note that either agent can be selected as the leader.

We form a nonlinear, nonquadratic variant of (12), (13) by using planar unicycle dynamics [28, Eq. 77] with a velocity state that evolves according to $\dot{v} = \alpha$. We discretize the dynamics at Δt. Each agent $\mathcal{A}_i$ controls yaw rate $\omega^{(i)}_t \in \mathbb{R}$ and longitudinal acceleration $\alpha^{(i)}_t \in \mathbb{R}$. The nonquadratic cost

$$g^{(1')}_t\left(x_t, u^{(1)}_t, u^{(2)}_t\right) = g^{(1)}_t(\cdot,\cdot,\cdot) - \log\left(\ell - p^{(2)}_{x,t}\right) - \log\left(p^{(2)}_{x,t} - \ell\right) - \log\left(\ell - p^{(2)}_{y,t}\right) - \log\left(p^{(2)}_{y,t} - \ell\right) \quad (14)$$

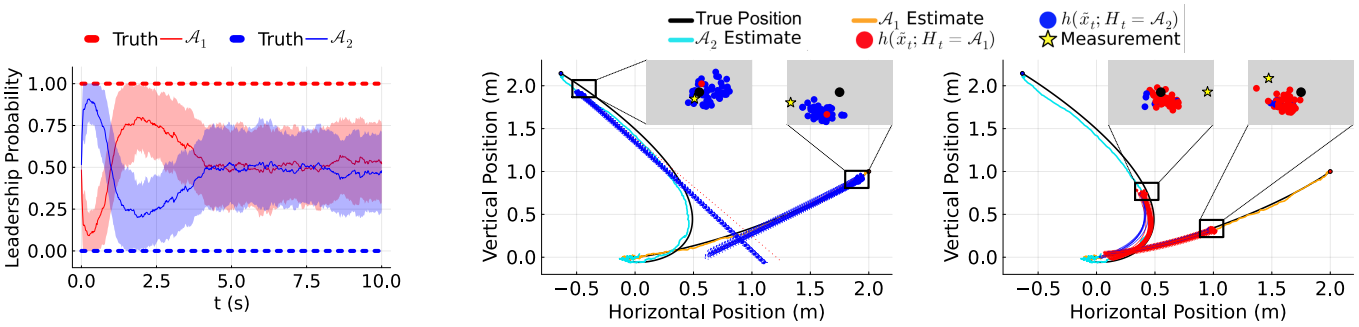
adds log barrier terms to (12) which force $\mathcal{A}_1$ to keep $\mathcal{A}_2$'s position $(p^{(2)}_{x,t}, p^{(2)}_{y,t})$ bounded within an origin-centered square of side length 2ℓ. The cost remains convex.

A. SILQGames Validation

To test convergence for non-LQ games, we run 100 simulations of SILQGames on the non-LQ shepherd and sheep game with $\mathcal{A}_2$ as leader. We fix $\mathcal{A}_1$'s initial position at $(2\,\mathrm{m}, 1\,\mathrm{m})$ and vary $\mathcal{A}_2$'s initial position along the perimeter of a radius-$\sqrt{5}\,\mathrm{m}$ circle. Both agents begin stationary and face toward the origin. The nominal strategies apply zero input. We specify additional parameters in the appendix.

Analysis. The results in Fig. 3a indicate that all simulations converge. The median value of the convergence metric, shown with 10% and 90% percentile bounds, exhibits a generally decreasing trend. These results are consistent with our previous discussion on convergence, as SILQGames converges in every simulation, though without monotone decrease in the convergence criterion. SILQGames behaves similarly when we vary the initial position of the follower.

In Fig. 3b, we report the solution for a particular (arbitrarily chosen) simulation. Both agents' motion follows the incentive structure of the game: the distance between the two agents decreases, as does the distance from $\mathcal{A}_2$ to the origin. As expected, $\mathcal{A}_1$ exerts more control effort than $\mathcal{A}_2$ due to $\mathcal{A}_2$'s leadership role and $\mathcal{A}_1$'s incentive to constrain $\mathcal{A}_2$'s position. Finally, we note that $\mathcal{A}_1$'s motion changes sharply towards the end of its trajectory. Here, the unicycle comes to a stop and moves in reverse. These results demonstrate that, for a



(a) Mean leader probability with standard deviation.

(b) Measurement trajectories at $t = 0.54\,\mathrm{s}$.

(c) Measurement trajectories at $t = 2.04\,\mathrm{s}$.

Fig. 4. We run 100 SLF simulations on analytic solutions to the LQ shepherd and sheep game. (a) indicates that the SLF initially misidentifies the leader but then identifies the leader correctly as $\mathcal{A}_1$ before becoming uncertain due to noise. (b) and (c) are associated with a particular simulation and show the Stackelberg measurement trajectories at $t = 0.54\,\mathrm{s}$ and $t = 2.04\,\mathrm{s}$, respectively. The color of a particle's measurement trajectory indicates leading agent H^k_{t-1}, and the insets show the expected measurements for each particle, the actual measurement, and the ground truth.

game with nonlinear dynamics and convex, nonquadratic costs, SILQGames converges to a solution that appears consistent with the dynamics and costs.

Timing. We collect elapsed times for each iteration of 100 SILQGames simulations on AMD Ryzen 9 5900x 12-core processors. The per-iteration runtime (with standard deviation) of SILQGames is $0.49 \pm 0.29\,\mathrm{s}$. We note that straightforward but nontrivial optimizations (i.e., more principled step size selection, numerical optimization techniques, etc.) such as those used in other iterative game solvers [29], [30] have been shown to improve computational efficiency.

B. Leadership Filter Validation

We validate the leadership filter on analytic solution trajectories of horizon T_{sim} for the LQ shepherd and sheep game played with leader $L_{\text{GT}} = \mathcal{A}_1$. Since we generate the ground truth $\boldsymbol{x}^{\text{GT}}_{1:T}$ with a known leader, a perfect filter should infer the true leader with consistently high confidence. Our results suggest that the SLF produces an observable signal for Stackelberg leadership, but (as one can expect) noise and measurement model configuration significantly affect performance. We simulate noisy state measurements $z_t \sim \mathcal{N}(x^{\text{GT}}_t, \Sigma)$ and pass Σ to the SLF. We list parameter values in the appendix.

Analysis. In our results, the SLF produces the expected leadership probability for part of the simulation horizon. From $1.5-3.5\,\mathrm{s}$ in Fig. 4a, the SLF correctly infers $\mathcal{A}_1$ as the leader with high likelihood. Examining the expected measurements in Fig. 4c at $2.04\,\mathrm{s}$, we note that the observations in this time range more closely match the measurement models generated with $\mathcal{A}_1$ as leader, which the SLF interprets as indicating leadership by $\mathcal{A}_1$. In the left inset of Fig. 4c, the particles that believe $\mathcal{A}_2$ to be the leader have positions further to the left of those that consider $\mathcal{A}_1$ to be the leader. From this effect, we see that the SLF accounts for the statistical dependence of state and leadership despite our simplified treatment of the leadership transition process.

However, we also see complex behavior in Fig. 4a. First, the SLF initially misidentifies the leader as $\mathcal{A}_2$, as shown by Fig. 4b, because the Stackelberg measurement trajectories do not capture leadership information over the whole simulation horizon. Specifically, the measurement trajectories $\{h(\tilde{x}^k_{t-1}, H^k_{t-1})\}$ are straight lines that roughly reduce the state costs of the shepherd and sheep, but do not capture the granularity of motion from the ground truth due to higher control costs over the short horizon $T_s \ll T_{\text{sim}}$.

Second, the SLF is completely uncertain after $4.5\,\mathrm{s}$. Near the origin, the contribution of process noise to the motion outweighs the contribution of the dynamics, and together with measurement noise obfuscate the dynamics.

From these results, we see that the SLF requires parameter T_s to be of sufficient length to capture the influence of leadership on the measurement trajectories. We note that the SLF is sensitive to noise as it infers leadership indirectly by comparing the observed motion with the expected motion of a Stackelberg leader. Thus, too little process noise may lead particles to converge to an incorrect trajectory, and too much reduces the signal-to-noise ratio.

Timing. The mean overall runtime for 100 simulations of an LQ game with 501 steps is $10.91 \pm 1.64\,\mathrm{s}$. For 50 particles and a 75-step measurement horizon, each step of the SLF runs in $0.82 \pm 0.35\,\mathrm{s}$. Self-driving vehicle applications require sub-$100\,\mathrm{ms}$ perception cycle computation time [31], so our implementation is not real-time. To meet real-time computational efficiency requirements, we must parallelize particle computation and optimize SILQGames; the latter is the most expensive step for each particle. These changes have been shown to reduce computation time below $100\,\mathrm{ms}$, as demonstrated by [29], [30], which use fast particle filters with measurement models that solve dynamic games.

C. Realistic Driving Scenarios

We formulate passing and merging scenarios using realistic ground truth trajectories without a clear leader. We demonstrate that the SLF responds to changes in leadership, handles objectives that imperfectly model agent behavior, and that the results match right-of-way expectations. The dynamics and cost terms demonstrate that the SLF does not require LQ assumptions and works for nonconvex costs. Our results further indicate that SILQGames, used within the SLF, converges under these conditions.

Each agent's state evolves according to unicycle dynamics. The simulation runs for T steps at period $\Delta t = 0.05\,\mathrm{s}$. We model stage cost $g^{(i)}_t$ as a weighted sum of incentives $g^{(i)}_{j,t}$,

$$g_t^{(i)} = \sum_{j=1}^{M^{(i)}} w_j^{(i)} g_{j,t}^{(i)}. \tag{15}$$

Weights $\{w_j^{(i)}\} \subset \mathbb{R}^+$ specify the relative priorities of sub-objectives. We define $M^{(i)} = 6$ terms to incentivize driving behaviors corresponding to legal or safety considerations.

$$g_{1,t}^{(i)} = \mathrm{d}(x_t^{(i)}, x_t^{(i),\mathrm{goal}}) \tag{16a}$$
$$g_{2,t}^{(i)} = -\log(\|p_t^{(i)} - p_t^{(j)}\|_2^2 - d_c) \quad \forall i \neq j \tag{16b}$$
$$g_{3,t}^{(i)} = -\log(v_m - |v_t^{(i)}|) - \log(\Delta\psi_m - |\psi_t^{(i)} - \psi_\mathrm{r}|) \tag{16c}$$
$$g_{4,t}^{(i)} = (\omega_t^{(i)})^2 + (\alpha_t^{(i)})^2 \tag{16d}$$
$$g_{5,t}^{(i)} = -\log(\|p_{t,\mathrm{llb}}^{(i)} - p_t^{(i)}\|_2^2) - \log(\|p_{t,\mathrm{rlb}}^{(i)} - p_t^{(i)}\|_2^2) \tag{16e}$$
$$g_{6,t}^{(i)} = \exp(-(1/2)(p_{t,\mathrm{cl}}^{(i)} - p_t^{(i)})^\intercal C^{-1}(p_{t,\mathrm{cl}}^{(i)} - p_t^{(i)})) \tag{16f}$$

Eq. (16a) requires a small distance between vehicle state $x_t^{(i)}$ and goal state $x_t^{(i),\mathrm{goal}}$. For this scenario, $\mathrm{d}(\cdot,\cdot)$ is a weighted Euclidean distance. Eq. (16b) requires a minimum safety radius d_c between the vehicles. Eq. (16c) requires obeying speed limit v_m and avoiding excessive heading deviation $\Delta\psi_m$ from road direction ψ_r. Eq. (16d) incentivizes low control effort. Eq. (16e) enforces left and right lane boundaries $p_{t,\mathrm{llb}}^{(i)}, p_{t,\mathrm{rlb}}^{(i)}$, based on lane width ℓ_w. Eq. (16f) uses a (nonconvex) Gaussian function with covariance C to discourage crossing the center line $p_{t,\mathrm{cl}}^{(i)}$. We specify these parameter values in the appendix. Lastly, we define the direction of motion as the y-direction and the transverse direction as the $-x$-direction to maintain a righthand coordinate frame.

Passing Scenario. The passing scenario begins with $\mathcal{A}_2$ behind $\mathcal{A}_1$ and runs for 7.5 s. In the ground truth trajectories (Fig. 1), $\mathcal{A}_2$ initially follows $\mathcal{A}_1$ for 2.5 s, then passes in the other lane, and ends ahead of $\mathcal{A}_1$ in the initial lane. $\mathcal{A}_1$ drives along the lane at a constant velocity, applying no controls.

We simulate the leadership filter on the passing maneuver. We expect $\mathcal{A}_1$ to start with a high leadership probability and for that probability to decrease once the passing maneuver begins, and vice versa for $\mathcal{A}_2$. In Fig. 1, the state estimate tracks the ground truth, indicating that the leadership filter captures the game dynamics. Since the SLF produces the expected trends in the state estimates and agents' probabilities, our results show that Stackelberg leadership can match right-of-way expectations for scenarios without a ground truth leader. Moreover, the SLF responds appropriately to changing leadership dynamics over time.

Lastly, we analyze the computation time of the SLF (0.027 ± 0.03 s per particle, per step). We note that calls to SILQGames converge in 4.2 ± 13.6 iterations. During the straight portions of the passing maneuver, the nominal trajectories produce better LQ approximations and thus SILQGames converges faster. During the turns, poor nominal strategies lead to slower convergence and result in variability in the computation time of the SLF. Overall, our results show that SILQGames can handle nonconvex cost terms.

Merging Scenario. The merging scenario involves three sections of road (see Fig. 5): two 30 m-long lanes separated by a barrier at $x = 0$ m, a merging segment that decreases from width $2\ell_w$ to ℓ_w over 30 m of length, and a one lane road centered along $x = 0$ m. Both agents start in their own lanes, though $\mathcal{A}_1$ starts behind $\mathcal{A}_2$. In the ground truth, $\mathcal{A}_2$ merges before $\mathcal{A}_1$, which slows down to yield before merging. $\mathcal{A}_2$ delays its merge once it enters the merging segment. We construct the game played within the measurement model to incentivize each agent to merge quickly after entering the merging segment, so the cost we define for $\mathcal{A}_2$ does not exactly reflect its actual behavior.

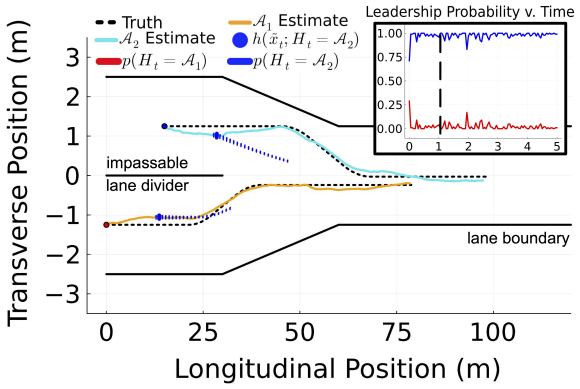


Fig. 5. In this merge, $\mathcal{A}_2$ starts ahead in its lane and $\mathcal{A}_1$ yields to $\mathcal{A}_2$. We see a high leadership likelihood for $\mathcal{A}_2$, as expected because it merges first. The inset indicates the current probabilities with a vertical dashed line.

In Fig. 5, we simulate this merge with the leadership filter. We expect $\mathcal{A}_2$ to lead the interaction as it begins ahead and merges first. Given their objectives, we expect the agents' measurement trajectories to merge quickly, and we see these trajectories quickly move toward the center of the merging segment. Nevertheless, the leadership filter's state estimate tracks the ground truth, including $\mathcal{A}_2$'s delayed merge, and the SLF infers $\mathcal{A}_2$ as the leader. Thus, the results match our right-of-way expectations despite agent objectives that do not exactly describe the observed ground truth behavior. This mismatch results in poor initialization, which affects the computation time (1.02 ± 0.99 s per particle, per step) as calls to SILQGames take longer to converge (69.2 ± 38.3 iterations). As with the passing scenario, the variance in computation time reflects a degradation in the quality of the LQ approximation. We further re-iterate that parallelization is critical for the SLF to run in real-time.

VI. Discussion & Limitations

We contribute SILQGames, an iterative algorithm to solve Stackelberg games with nonlinear dynamics and nonquadratic costs. Through empirical validation on non-LQ game scenarios, we show it reliably converges. We also introduce the Stackelberg Leadership Filter and apply it to noisy scenarios with known leaders and realistic driving situations. Results highlight the SLF's ability to estimate leadership in long-horizon interactions with changing leadership and with objectives that do not exactly reflect observed agent behavior. Furthermore, we discuss the robustness of our method to the measurement horizon and noise.

Future directions include extending SILQGames to $N > 2$ agents and overcoming combinatorial scaling challenges aris-

ing from the pairwise definition of Stackelberg leadership. The number of possible N-agent Stackelberg hierarchies grows exponentially and resolving the dependencies in any such hierarchy is nontrivial. Another critical direction involves establishing theoretical bounds on the number of SILQGames iterations. For the SLF, future work includes enabling real-time application using more efficient estimators and algorithmically adjusting the measurement horizon T_s to observe leadership dynamics over different horizons.

ACKNOWLEDGMENT

We thank Professor Todd Humphreys and members of the CLeAR and SWARM Labs at UT Austin for feedback.

REFERENCES

[1] R. Tian, L. Sun, *et al.*, "Safety assurances for human-robot interaction via confidence-aware game-theoretic human models," in *International Conference on Robotics and Automation*, 2022.
[2] H. v. Stackelberg, *Marktform und gleichgewicht*. 1934.
[3] J. Garland, A. M. Berdahl, *et al.*, "Anatomy of leadership in collective behaviour," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 7, 2018.
[4] A. Strandburg-Peshkin, D. Papageorgiou, *et al.*, "Inferring influence and leadership in moving animal groups," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 373, no. 1746, 2018.
[5] A. Deka, K. Sycara, *et al.*, "Human vs. deep neural network performance at a leader identification task," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 2021.
[6] A. Singh and P. Artemiadis, "Automatic identification of the leader in a swarm using an optimized clustering and probabilistic approach," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2021.
[7] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, pp. 25–34.
[8] A. Y. Carmi, L. Mihaylova, *et al.*, "MCMC-based tracking and identification of leaders in groups," in *2011 IEEE International Conference on Computer Vision Workshops*, 2011, pp. 112–119.
[9] Q. Li, S. J. Godsill, *et al.*, "Inferring dynamic group leadership using sequential Bayesian methods," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 8911–8915.
[10] C. Dextreit and I. V. Kolmanovsky, "Game theory controller for hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 652–663, 2013.
[11] J. Geary, S. Ramamoorthy, and H. Gouk, "Resolving conflict in decision-making for autonomous driving," in *Robotics: Science and Systems XVII*, 2021.
[12] B. Di and A. Lamperski, "Local first-order algorithms for constrained nonlinear dynamic games," in *American Control Conference (ACC)*, IEEE, 2020, pp. 5358–5363.
[13] E. L. Zhu and F. Borrelli, "A sequential quadratic programming approach to the solution of open-loop generalized Nash equilibria," 2023.
[14] L. Cleac'h, M. Schwager, Z. Manchester, *et al.*, "ALGAMES: A fast augmented Lagrangian solver for constrained dynamic games," *Autonomous Robots*, vol. 46, no. 1, pp. 201–215, 2022.
[15] M. Bhatt, A. Yaraneri, and N. Mehr, "Efficient constrained multi-agent interactive planning using constrained dynamic potential games," *arXiv preprint arXiv:2206.08963*, 2022.
[16] D. Fridovich-Keil, E. Ratner, *et al.*, "Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1475–1481.
[17] K. Nakamura, "Opinion-guided games: Strategic coordination through gradient-based opinion dynamics," Master's thesis, Princeton University, 2023.
[18] D. Q. Mayne, "Differential dynamic programming–a unified approach to the optimization of dynamic systems," in *Control and dynamic systems*, vol. 10, 1973, pp. 179–254.
[19] D. Bertsekas, *Dynamic programming and optimal control: Volume I*. 2012, vol. 1.
[20] F. Laine, D. Fridovich-Keil, *et al.*, "The computation of approximate generalized feedback Nash equilibria," *SIAM Journal on Optimization*, vol. 33, no. 1, pp. 294–318, 2023.
[21] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.
[22] L. Peters, D. Fridovich-Keil, *et al.*, "Inferring objectives in continuous dynamic games from noise-corrupted partial state observations," 2021.
[23] N. Mehr, M. Wang, *et al.*, "Maximum-entropy multi-agent dynamic games: Forward and inverse solutions," *IEEE Transactions on Robotics*, 2023.
[24] X. Liu, L. Peters, and J. Alonso-Mora, "Learning to play trajectory games against opponents with unknown objectives," *IEEE Robotics and Automation Letters (RA-L)*, 2023.
[25] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.
[26] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.
[27] F. Laine, *The computation of approximate generalized feedback Nash equilibria*, ASE 389 Guest Lecture, Instructor: David Fridovich-Keil, University of Texas at Austin, Presented: 2021-05-11, 2021.
[28] A. J. Taylor, P. Ong, *et al.*, "Safe backstepping with control barrier functions," *arXiv preprint arXiv:2204.00653*, 2022.
[29] L. Peters, D. Fridovich-Keil, *et al.*, "Inference-based strategy alignment for general-sum differential games," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, 2020, pp. 1037–1045.
[30] L. Peters, "Accommodating intention uncertainty in general-sum games for human-robot interaction," Master's thesis, Hamburg University of Technology, 2020.
[31] S.-C. Lin, Y. Zhang, *et al.*, "The architectural implications of autonomous driving: Constraints and acceleration," in *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, 2018.

APPENDIX

SILQGames Parameters. We vary the initial position of $\mathcal{A}_2$ about $(-1\,\mathrm{m}, 2\,\mathrm{m})$ along a $0.4\,\mathrm{rad}$ arc of a circle. We set convergence threshold $\tau = 1.2{\cdot}10^{-3}$, the maximum number of iterations to 3500, and minimum step size $\alpha_{\min} = 10^{-2}$. We play the game for $10\,\mathrm{s}$ with period $\Delta t = 0.02\,\mathrm{s}$ (501 steps). The nominal controls apply zero input.

SLF Parameters. In our examples, we select nominal strategies with a simple heuristic that returns T_s-length control trajectories for each agent, i.e. at time $t-1$, the nominal strategy for $\mathcal{A}_i$ is $[u_{t-1}^{(i)} \cdots u_{t-1}^{(i)}]$. We configure the number of particles $N_s = 50$. The Stackelberg measurement horizon $T_s = 75$ steps ($1.5\,\mathrm{s}$). Let $p_{\text{trans}} = 0.02$, so transitioning is thus likely enough that particles can switch leadership state and model dynamic leadership transitions without injecting excessive uncertainty into the inference. For the process noise uncertainty W, we set position and heading variances on the order of magnitude of 10^{-3} and velocity variances to 10^{-4}. SLF measurement uncertainty $\Sigma = 5{\cdot}10^{-3} I$. The convergence threshold $\tau = 1.5 \cdot 10^{-2}$, the max iteration count $M_{\text{iter}} = 50$, and step size $\alpha_{\min} = 10^{-2}$.

Driving Scenario Parameters. Let speed limit $v_m = 35\,\mathrm{m\,s^{-1}}$ with initial headings aligned with the road direction ψ_r. Lanes are $\ell_w = 2.5\,\mathrm{m}$ wide. A safety violation occurs if the vehicles come within $d_c = 0.2\,\mathrm{m}$ of one another. We constrain acceleration and rotational velocity magnitudes to $9\,\mathrm{m\,s^{-2}}$ and $2\,\mathrm{rad\,s^{-1}}$. The measurement horizon $T_s = 1.0\,\mathrm{s}$, with sampling periods of $0.05\,\mathrm{s}$ ($20\,\mathrm{Hz}$). We use 100 particles with equal initial chance of $\mathcal{A}_1$ and $\mathcal{A}_2$ as leader. The center line is at $x = 0\,\mathrm{m}$. Each agent begins with velocity $10\,\mathrm{m\,s^{-1}}$. Other parameters are identical to the SLF parameters.