Adversarial Information Acquisition

Brent Schlotfeldt  
GRASP Laboratory  
University of Pennsylvania  
Philadelphia, PA 19104  
Email: brentse@seas.upenn.edu

Nikolay Atanasov  
Dept. of Electrical and Computer Engineering  
University of California San Diego  
La Jolla, CA 92093  
Email: natanasov@ucsd.edu

George J. Pappas  
GRASP Laboratory  
University of Pennsylvania  
Philadelphia, PA 19104  
Email: pappasg@seas.upenn.edu

Abstract—This paper considers a non-cooperative two-player game modeling the problem of adversarial information acquisition in robotics. Each robot is equipped with a sensor, and is tasked with choosing control inputs that maximize an information (e.g. entropy) about the other player, while keeping its own state as uncertain to the other player as possible, subject to the available sensors. This adversarial information gathering problem has applications in surveillance, or search-and-rescue missions where the agent whose state is to be estimated may try to actively avoid the sensing robot. We formulate the problem of adversarial information acquisition, and provide an initial solution based on a variant of Monte Carlo Tree Search for simultaneous action games.

I. INTRODUCTION

In this paper, we consider the problem of two robots interacting in an adversarial game where each robot attempts to estimate the state of its adversary, while keeping its own state hidden. This problem can have applications in search-and-rescue applications, where the agent to be found is mobile, and actively evades the sensing robot. In these problems, it is important to both accurately localize the target agent, while keeping one’s own state hidden so that the target’s ability to actively evade is reduced.

There is much prior work in the literature concerning the dynamics of pursuit-evasion in mobile robotic settings[3]. Approaches to the pursuit-evasion problem are split between considering the worst-case evader (adversary), and using probabilistic frameworks which consider the expected case. A common theme in the pursuit-evasion literature is the objective of reducing the distance to the evader to zero, or forcing the evader into a sensing footprint. In contrast, our problem is formulated using a probabilistic approach which optimizes an information theoretic quantity, namely entropy, about the distribution of the target to be tracked. Rather than closing distance to the target, our approach aims to produce the best estimate of the target’s state, subject to the sensors available. Our previous work considers the information acquisition problem for target tracking [1], however this work assumes that the target being tracked moves independently of the sensing robot, and crucially is not trying to actively evade the sensing robot. In this work, the problem formulation is symmetric in the sense that the adversary is trying to maximize information gained about us, and also minimize the information we can gain about it.

We also draw attention to some biological inspiration for this problem. Motion camouflage [9] is a strategy utilized by dragonflies, which enables them to capture their prey by minimizing the optical flow of their motion. Mischia and Krishnaprasad [8] consider the problem of mutual motion camouflage, where two agents each pursue each other, but attempt to maintain a constant bearing to avoid detection by the other agent. Our problem is related, but rather than considering pursuit-evasion, we consider the dynamics of an adversarial information gathering game.

In the most general case, the information acquisition game proposed is a stochastic game and is difficult to solve. McEneaney [7] discusses a class of stochastic games with finite-dimensional solutions and dynamic programming algorithms to solve them. With some assumptions on the motion and observation models of the agents in our game, the problem can be simplified to a deterministic game. McEneaney [6] introduces a curse-of-dimensionality free max-plus method for deterministic game problems, which is likely to be very applicable to the linear Gaussian version of the information-theoretic game introduced here. Additionally, Grünwald and Dawid [4] present a game-theoretic argument that maximizing entropy and minimizing worst-case expected loss are duals of each other. A comprehensive treatment on adversarial reasoning, is provided in the book by Kott and McEneaney [5]. The approach taken in this work is a variant of Monte Carlo Tree Search [11], for simultaneous action games. We present the details of this approach in Sec. III.

II. PROBLEM FORMULATION

Consider a two-player partial information game with simultaneous moves. Each player $i \in \{1, 2\}$ has a state $x_{i,t}$ that evolves according to the following motion model:

$$x_{i,t+1} = f_i(x_{i,t}, u_{i,t}, w_{i,t})$$

where $u_{i,t} \in \mathcal{U}_i$ is a finite space of admissible moves (control inputs) and $w_{i,t}$ is a random variable specifying the motion noise. Player $i$ can observe its own state $x_{i,t}$ and chooses its moves with the objective of tracking the evolution of the state of the other player. Each player is equipped with a sensor used to collect information about the other player according to the following observation model:

$$z_{i,t} = h_i(x_{i,t}, x_{j,t}, v_{i,t})$$

(2)
where \(v_{i,t}\) is a random variable specifying the observation noise. We assume that the players know their own and each others motion and observation modes but cannot directly observe the other player’s state and control inputs, resulting in a partial information game. The information available to player \(i\) at time \(t\) to choose its move \(u_{i,t}\) is:

\[
\mathcal{I}_{i,t} = z_{i,t}, \quad \mathcal{I}_{i,0} := (z_{i,0:t}, u_{i,0:t-1}), \quad t > 0, \quad i = 1, \ldots, n
\]

**Problem (Information Acquisition Game).** Given its initial state \(x_{i,0}\) and a prior probability density function \(p_{i,0}\) over the state of the other player, the objective of player \(i\) observe the other player’s state and control inputs, resulting in a partial information game. The information available to player \(i\) at time \(t\) to choose its move \(u_{i,t}\) is:

\[
\mathcal{I}_{i,t} = (z_{i,0:t}, u_{i,0:t-1}), \quad t > 0, \quad i = 1, \ldots, n
\]

To evaluate the proposed MCTS algorithm, we consider an information acquisition game between two robots 1 and 2, one equipped with a range-only sensor and the second robot equipped with a bearing sensor, respectively. Suppose that the robots have trivial deterministic dynamics \(x_{i,t+1} = x_{i,t} + u_{i,t}\), \(i \in \{1, 2\}\), but their observation models differ:

\[
\begin{align*}
z_{1,t} &= \|x_{2,t} - x_{1,t}\| + v_{1,t} \\
z_{2,t} &= \arctan\left(\frac{\|x_{1,t} - x_{2,t}\|}{\|x_{1,t} - x_{2,t}\|}\right) + v_{2,t}
\end{align*}
\]

where \(v_{1,t}\) and \(v_{2,t}\) are standard basis vectors and \(v_{i,t} \sim N(0, V_i)\) represent observation noise. In the game playouts generated by the MCTS algorithm 1, since robot 1 has access to the simulated control sequence \(u_{2,0:T-1}\) of player 2, player 1 can linearize its observation model (4) around an estimated trajectory of \(x_{2,t}\) and propagate a Gaussian distribution over \(x_{2,t}\) via the Extended Kalman Filter (EKF).

Thus, for the control additive motion model and observation models (4, 5), we can reduce the general formulation in (3) to a linear Gaussian information acquisition game:

\[
\begin{align*}
\min_{u_{i,0:T-1}} \max_{p_{i,0:T-1}} & \quad \log \det \Sigma_{j,T} - \log \det \Sigma_{i,T} \\
\text{subject to} & \quad x_{i,t+1} = x_{i,t} + u_{i,t}, \quad i \in \{1, 2\}, \\
& \quad \mu_{i,t+1}, \Sigma_{i,t+1} = \rho(\mu_{i,t}, \Sigma_{i,t}, x_{j,t}), \quad i \in \{1, 2\}
\end{align*}
\]

where \(\rho\) specifies the evolution of the mean and covariance of the distribution over \(x_{i,t}\) that player \(j\) maintains.

To investigate the behavior generated by the SM-MCTS algorithm, we choose control inputs for the range-only sensing robot, while holding the bearing-only robot in a fixed position, i.e. \(U_2 = \emptyset\). The sensing robot moves on an 8-connected grid. We use a planning horizon \(T = 8\), with range sensor noise variance \(v_1 = .15 m\), and bearing variance \(v_2 = 5^\circ\). We compare with uniformly random selected actions, in order to visually interpret the behavior. Example trajectory realizations are illustrated in Fig. 1. We note that the motion produced by the MCTS algorithm attempts to keep a far distance from the bearing sensor. Only once it achieves a sufficient distance, it exploits its own sensor to localize the bearing-sensor. This
Algorithm 1 Simultaneous Move Monte Carlo Tree Search

1: function SM-MCTS(x₁,₀, p₁,₀, x₂,₀, p₂,₀, T, κ)
2:   i ← random player in {1, 2}, j ← opposite player of i
3:   root ← NODE(i, x₁,₀, p₁,₀, x₂,₀, p₂,₀, 0)
4:   while within computational budget do
5:       SIMULATE(root)
6:   return arg min \( \frac{\text{root.casts}(u)}{\text{root.visits}(u)} \)
7: end
8: function SIMULATE(node)
9:   if END SIMULATE(node) then
10:      cost = ROLLOUT POLICY(node)
11:   else
12:      u ← TREE POLICY(node)
13:      new_node ← GET SUCCESSOR(node, u)
14:      cost ← \text{SIMULATE(new_node)}
15:      node.visits(u) ← 1
16:      costs(u) ← cost
17:      return cost
18: end
19: function END SIMULATE(node)
20:   if node ∈ Tree then
21:       add node to Tree
22:      return true
23:   else if node.t = (2T − 1) then
24:      return true
25:   else
26:      return false
27: end
28: function TREE POLICY(node)
29:   \( \mathcal{V} \leftarrow \{ u \mid \text{node.visits}(u) = 0 \} \)
30:   if \( \mathcal{V} \) not empty then
31:      return uniform sample from \( \mathcal{V} \)
32:   else
33:      return \( \arg \min_{u \in \mathcal{U}(\text{node}, \epsilon)} \frac{\text{node.casts}(u)}{\text{node.visits}(u)} - \kappa \log \frac{\sum_{u \in \mathcal{U}(\text{node})} \text{node.visits}(u)}{\text{node.visits}(u)} \)
34: end
35: function ROLLOUT POLICY(node)
36:   for \( t = \text{node.t} \ldots (2T − 1) \) do
37:      u ← uniform sample from \( \mathcal{U}(\text{node}, \epsilon) \)
38:      node ← GET SUCCESSOR(node, u)
39:   return ENTROPY(node.\( q \))−ENTROPY(node.\( p \))
40: end
41: function NODE(i, x, p, y, q, t)
42:   node.i ← i, node.x ← x, node.p ← p
43:   node.y ← y, node.q ← q, node.t ← t
44:   node.visits ← zeros(\( |\mathcal{U}(x)|, 1 \)) \( \Rightarrow \) Visitation counts for each child
45:   node.casts ← zeros(\( |\mathcal{U}(x)|, 1 \)) \( \Rightarrow \) Long-term costs for each child
46:   return node
47: end
48: function GET SUCCESSOR(node, u)
49:   i ← node.i
50:   x ← \( f_i(nodex, u, w) \)
51:   p ← BAYES INFERENCE(node.p, x)
52:   if node.t is odd then
53:      j ← random player in \{1, 2\}
54:   else
55:      j ← opposite player of i
56:   return NODE(j, node.y, node.q, x, p, node.t + 1)

is because when the robot is far from the bearing sensor, the bearing angle difference is minimized between time-steps, and this reduces the ability for the bearing-sensor to detect our robot. The cost function is visualized in a heatmap in 2.

V. CONCLUSION AND FUTURE WORK

In this paper, we have posed the problem of adversarial information acquisition, and given some initial insight into its solution with methods based on a variant of Monte Carlo Tree Search (MCTS). We demonstrated some behaviors that arise in a two-robot simulation, where the robots are equipped with range-only, and bearing-only sensors respectively. Future work will aim to prove some properties of the linear-Gaussian information acquisition game described, and in particular investigate the circumstances necessary to achieve a Nash Equilibrium in the game. Additionally we would like to investigate strategies that yield better performance than the uniform rollout policy in the MCTS algorithm described. We also plan to evaluate more extensively the behaviors that arise in environments with obstacles and occlusions.

ACKNOWLEDGMENTS

We gratefully acknowledge support from ARL DCIST CRA W911NF-17-2-0181.

REFERENCES


