Adversarial Information Acquisition

Brent Schlotfeldt
GRASP Laboratory
University of Pennsylvania
Philadelphia, PA 19104
Email: brentsc@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. Mischiati 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:

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

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:

$$z_{i,t} = h_i(x_{i,t}, x_{j,t}, v_{i,t}) \tag{2}$$

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:

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

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 is to choose a sequence of functions $\mu_{i,t}(\mathcal{I}_{i,t})$ for a given finite planning horizon T in order to optimize:

$$\begin{aligned} \min_{\mu_{i,0:(T-1)}} \max_{\mu_{j,0:(T-1)}} & \mathbb{H}(x_{j,T} \mid z_{i,1:T}) - \mathbb{H}(x_{i,T} \mid z_{j,1:T}) \\ \text{subject to } & u_{i,t} = \mu_{i,t}(\mathcal{I}_{i,t}), \qquad i \in \{1,2\}, \\ & x_{i,t+1} = f_i(x_{i,t}, u_{i,t}, w_{i,t}), \quad i \in \{1,2\}, \\ & z_{i,t} = h_i(x_{i,t}, x_{j,t}, v_{i,t}), \quad i \in \{1,2\}. \end{aligned} \tag{3}$$

where $\mathbb{H}(x_{j,T} \mid z_{i,1:T})$ denotes the entropy of the state of player j conditioned on sequence of measurements $z_{i,1:T}$ obtained by player i.

III. Technical Approach

Assuming that the players know each other's prior densities $p_{i,0}$, $p_{j,0}$, the information acquisition game in (3) is a zero-sum two-player stochastic game. Since the control spaces $\mathcal{U}_i$, $i \in \{1,2\}$ are finite and the playing horizon T is finite, the results of Shapley [10] show that the minimax theorem applies to the information acquisition game and its value is finite. The value of the game and the optimal (potentially mixed) strategies for the two players can be determined via value iteration. However, since the state space of the game includes the sets of all density functions over the players' states, the curse of dimensionality limits the use of value iteration only to very small instances.

Instead, we propose a Monte Carlo Tree Search (MCTS) algorithm specifically designed for the simultaneous-move formulation. MCTS and its variants are a class of algorithms for searching game trees and collecting statistics on the performance of various moves. Various approaches for simultaneous-move games with MCTS are suggested in Tak et al. [11], including the Sequential Upper Confidence Trees algorithm (SUCT). This algorithm applies the classic UCT algorithm, which is a selection algorithm for MCTS to balance exploration and exploitation based on optimizing regret [2]. The SUCT algorithm applies UCT on a sequential relaxation of the simultaneous game, to approximate a solution. Our idea follows from this algorithm, but also introduces randomization of the starting player on the serialized game tree.

Given a computational budget, simulations are generated from a root node containing the current robot state $x_{i,0}$, and distribution on the adversary state $p_{j,0}$. Each iteration consists of four stages, namely selection, expansion, rollout, and backpropagation [2]. In the selection step, the algorithm moves down the search tree through nodes that have already been visited, with a selection criteria based on the UCT approach for balancing exploration and exploitation (see line 32 in Alg. 1). The expansion step adds a single new leaf node to the tree (line 20). Once a single new node has been added in the expansion, a random rollout of the game through the rest of the horizon is executed via uniform random actions (line 34), returning the terminal cost. Finally, the cost is recursively backpropagated up to the root of the tree.

Once a sufficient number of simulations have been played out, the best action can be selected by considering the cost of the children of the root node, and selecting the node that has the lowest aggregated cost (line 6).

IV. Evaluation

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:

$$z_{1,t} = \|x_{2,t} - x_{1,t}\| + v_{1,t} \tag{4}$$

$$z_{2,t} = \arctan\left(\frac{e_2^T(x_{1,t} - x_{2,t})}{e_1^T(x_{1,t} - x_{2,t})}\right) + v_{2,t} \tag{5}$$

where e_1, e_2 are standard basis vectors and $v_{i,t} \sim \mathcal{N}(0, V_i)$ 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[1], 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:

$$\begin{aligned} \min_{u_{i,0:(T-1)}} \max_{u_{j,0:(T-1)}} & \log\det\Sigma_{j,T} - \log\det\Sigma_{i,T} \\ \text{subject to } & x_{i,t+1} = x_{i,t} + u_{i,t}, \qquad i \in \{1,2\}, \\ & \mu_{i,t+1}, \Sigma_{i,t+1} = \rho(\mu_{i,t}, \Sigma_{i,t}, x_{j,t}), \ \ i \in \{1,2\}. \end{aligned} \tag{6}$$

where ρ 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. $\mathcal{U}_2 = \emptyset$. The sensing robot moves on an 8-connected grid. We use a planning horizon $T = 8$, with range sensor noise variance $v_1 = .15m$, and bearing variance v_2 = 5°. 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

[1] We emphasize here that each robot only knows the adversary's control inputs in the planning phase, i.e. when game playouts are simulated.

Algorithm 1 Simultaneous Move Monte Carlo Tree Search

function SM-MCTS($x_{1,0}$, $p_{1,0}$, $x_{2,0}$, $p_{2,0}$, T, κ)
  $i \leftarrow$ random player in $\{1, 2\}$, $j \leftarrow$ opposite player of i
  root $\leftarrow$ NODE(i, $x_{i,0}$, $p_{i,0}$, $x_{j,0}$, $p_{j,0}$, 0)
  while within computational budget **do**
    SIMULATE(root)
  return $\underset{u \in \mathcal{U}(x_{i,0})}{\arg\min} \frac{root.costs(u)}{root.vists(u)}$

function SIMULATE(node)
  if ENDSIMULATE(node) **then**
    cost = ROLLOUTPOLICY(node)
  else
    $u \leftarrow$ TREEPOLICY(node)
    new_node $\leftarrow$ GETSUCCESSOR(node, u)
    cost $\leftarrow$ SIMULATE(new_node)
    node.visits(u) += 1
    node.costs(u) += cost
  return cost

function ENDSIMULATE(node)
  if node $\notin$ Tree **then**
    add node to Tree
    return true
  else if node.t $= (2T - 1)$ **then**
    return true
  else
    return false

function TREEPOLICY(node)
  $\mathcal{V} \leftarrow \{u \mid node.visits(u) = 0\}$
  if $\mathcal{V}$ **not** empty **then**
    return uniform sample from $\mathcal{V}$
  else
    return $\underset{u \in \mathcal{U}(node.x)}{\arg\min} \frac{node.costs(u)}{node.vists(u)} - \kappa\sqrt{\frac{\log \sum_a node.visits(a)}{node.visits(u)}}$

function ROLLOUTPOLICY(node)
  for $t =$ node.t $\ldots (2T - 1)$ **do**
    $u \leftarrow$ uniform sample from $\mathcal{U}(node.x)$
    node $\leftarrow$ GETSUCCESSOR(node, u)
  return ENTROPY(node.q)$-$ENTROPY(node.p)

function NODE(i, x, p, y, q, t)
  node.i $\leftarrow i$, node.x $\leftarrow x$, node.p $\leftarrow p$
  node.y $\leftarrow y$, node.q $\leftarrow q$, node.t $\leftarrow t$
  node.visits $\leftarrow$ zeros($|\mathcal{U}_i(x)|$,1) ▷ Visitation counts for each child
  node.costs $\leftarrow$ zeros($|\mathcal{U}_i(x)|$,1) ▷ Long-term costs for each child
  return node

function GETSUCCESSOR(node, u)
  $i \leftarrow$ node.i
  $x \leftarrow f_i$(node.x, u, w)
  $p \leftarrow$ BAYESINFERENCE(node.p, x)
  if node.t is odd **then**
    $j \leftarrow$ random player in $\{1, 2\}$
  else
    $j \leftarrow$ opposite player of i
  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 timesteps, and this reduces the ability for the bearing-sensor to detect our robot. The cost function, averaged over ten trials is included in Fig. 2.

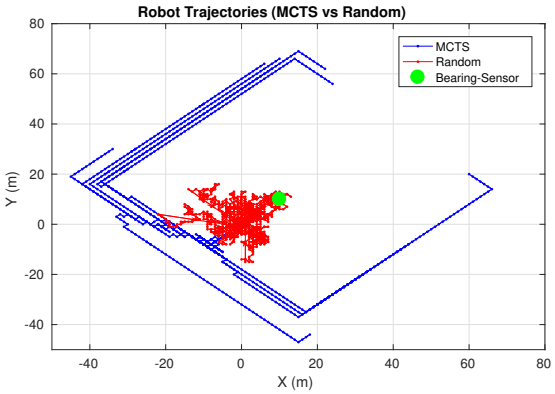

Fig. 1: Range-only sensor trajectories generated by MCTS (Blue), and by Random actions (Red). The bearing sensor is the green dot located in the center.

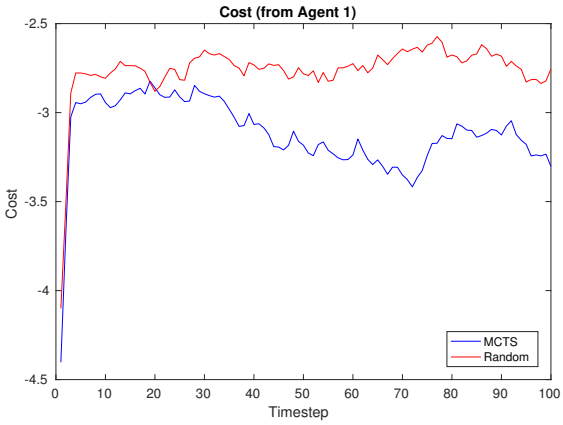

Fig. 2: Value of cost function attained by the range-only sensing robot. In blue, the MCTS algorithm was used, and in red are randomly chosen actions. The plots are averaged over ten trials of 100 timesteps.

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

[1] N. Atanasov, J. Le Ny, K. Daniilidis, and G. Pappas. Information Acquisition with Sensing Robots: Algorithms and Error Bounds. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 6447–6454, 2014.

[2] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[3] T. H. Chung, G. A. Hollinger, and V. Isler. Search and pursuit-evasion in mobile robotics. *Autonomous robots*, 31(4):299, 2011.

[4] P. D. Grünwald and A. P. Dawid. Game Theory, Maximum Entropy, Minimum Discrepancy and Robust Bayesian Decision Theory. *The Annals of Statistics*, 32 (4):1367–1433, 2004.

[5] A. Kott and W. M. McEneaney. *Adversarial reasoning: computational approaches to reading the opponents mind*. CRC Press, 2006.

[6] W. M. McEneaney. Idempotent method for dynamic games and complexity reduction in min-max expansions. In *IEEE Conference on Decision and Control (CDC)*, pages 163–168, 2009.

[7] W. M. McEneaney. Some classes of imperfect information finite state-space stochasticgames with finite-dimensional solutions. *Applied Mathematics and Optimization*, 50(2):87–118, 2004.

[8] M. Mischiati and P. Krishnaprasad. The dynamics of mutual motion camouflage. *Systems & Control Letters*, 61(9):894–903, 2012.

[9] A. Mizutani, J. S. Chahl, and M. V. Srinivasan. Insect behaviour: Motion camouflage in dragonflies. *Nature*, 423(6940):604, 2003.

[10] L. S. Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

[11] M. J. W. Tak, M. Lanctot, and M. H. M. Winands. Monte Carlo Tree Search variants for simultaneous move games. In *IEEE Conference on Computational Intelligence and Games*, pages 1–8, 2014.