Control Synthesis in Partially Observable Environments for Complex Perception-related Objectives.

Zetong Xuan and Yu Wang

Abstract— **Perception-related tasks are frequently encountered in the operation of autonomous systems in partially observable environments. This work studies the control synthesis problem for optimal policy for complex perception-related objectives in environments modeled by partially observable Markov decision processes (POMDPs). To formally express these objectives, we introduce co-safe linear inequality temporal logic (sc-iLTL), which can define complex tasks that are formed by logical concatenation of atomic propositions as linear inequalities on belief space of the POMDPs. Our solution to the control synthesis problem is to transform the sc-iLTL objectives into reachability objectives by constructing the product of the belief MDP and a deterministic finite automaton (DFA) built from the sc-iLTL objective. Then, to address the scalability issue due to the product, we introduce a Monte Carlo Tree Search (MCTS) methods that are guaranteed to find the optimal policy. Finally, we demonstrated the applicability of our method on a drone-probing case study.**

I. Introduction

Perception-related tasks are commonly encountered in the operation of autonomous systems in partially observable environments. For example, consider a drone that operates with a limited field of view and imperfect onboard sensors. Tasks such as confidently detecting a target or avoiding obstacles are perception-related because they rely on estimation of the environment rather than on direct observation. Beyond that, many real-world applications require complex rule-based tasks that consist of sequentially ordered perception-related subtasks. For instance, a surveillance drone might be required to continuously monitor a moving target (ensuring liveness) while simultaneously avoiding obstacles (ensuring safety). These sequential subtasks necessitate a precise formulation of each individual perception-related objective as well as their correct ordering to achieve the overall mission.

In practice, the perception in such partially observable environments is typically captured by *beliefs* in the partially observable Markov decision process (POMDP). A POMDP generalizes a Markov decision process (MDP) by assuming that the true MDP states, referred to as hidden states now, are partially observable through a probabilistic relation to a set of observations. In a POMDP, the perception of the environment is represented by the beliefs, which is the optimal estimation of the probability distribution over hidden states based on all previous and current observations. Thus, perception-related tasks can be expressed by reaching or avoiding certain regions over the space of all beliefs.

On the POMDPs, defining perception-related rule-based tasks requires formal logic over the beliefs. In this work, we introduce co-safe linear inequality temporal logic (sc-iLTL) [1] to describe a perception-related objective. When the complex rule-based task is built from the sequential ordering of perception-related events, sc-iLTL directly specifies such a task by specifying the sequential ordering of different beliefs. By introducing a labeling function to map beliefs into atomic propositions, the sequential ordering of perception-related events determines the satisfaction status of a sc-iLTL objective. Sc-iLTL modifies co-safe linear temporal logic [2] by expressing each atomic proposition as a linear inequality over the space of all beliefs. Here we consider the co-safe fragment of the standard linear temporal logic (LTL) [3] as this fragment is better suited for tasks that are naturally specified in a finite-horizon setting.

Synthesizing an optimal policy for a sc-iLTL objective is challenging due to scalability issues. Control synthesis for a sc-iLTL objective is as hard as solving a reachability problem in the continuous space of beliefs, which is known to be difficult because of the large state space [4]. The reachability problem is formulated by augmenting the original model with an automaton that tracks the satisfaction status of the sc-iLTL objective, so that an optimal policy for the reachability objective in the extended model is equivalent to an optimal policy for the sc-iLTL objective in the original model. However, when sc-iLTL objectives are defined directly in the space of all beliefs, the resulting reachability problem cannot be solved efficiently using point-based methods [5], [6]. Point-based methods rely on the α-vector representation [7] of the value function, such representation assumes the value function is piecewise linear or convex in the belief space. The key assumption is violated by our reachability problem.

To address the challenge in scalability, we propose using Monte Carlo tree search (MCTS) [8]. Methods like MCTS or real-time dynamic programming (RTDP) [9], [10] help mitigate the scalability issue by leveraging the structured dynamics of belief updates. Specifically, in a POMDP with finite hidden states, the number of possible next beliefs given the current belief remains finite even though the space of all beliefs is continuous. MCTS applies best-first search to a finite number of beliefs that is only reachable from the initial beliefs, thus achieving scalability similar to an efficient point-based method. We prefer MCTS over RTDP because MCTS avoids explicit discretization, making its complexity independent of the dimension of the space of all beliefs.

Our main contribution is to propose an MCTS method

Zetong Xuan is with Department of Mechanical & Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA

Yu Wang is with Department of Mechanical & Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA

to synthesize an optimal policy.[1] By constructing the belief MDP which captures the transitions of beliefs, we first lift the control synthesis problem to the belief space. Then, we build a product belief MDP by augmenting the belief MDP with a deterministic finite automaton (DFA) encoding the satisfaction status of the sc-iLTL specification. This product construction transforms the history-dependent iLTL objective into a reachability objective on the product belief MDP. Consequently, a memoryless optimal policy for the reachability objective on this product belief MDP is equivalent to an optimal memory-dependent policy satisfying the iLTL objective on the original POMDP. Finally, we propose an MCTS method to find the optimal memoryless policy, which can then be translated back to the optimal memory-dependent policy on the POMDP.

Related works

Our work falls within the broader domain of control synthesis for LTL objectives in POMDPs. Most existing approaches define LTL objectives over the space of hidden states, making them unsuitable for representing perception-related objectives. Some studies have explored perception-related objectives defined as distribution temporal logic (DTL) [11]. In DTL, a labeling function maps beliefs into atomic propositions using nonlinear inequalities. However, these works focus on linear Gaussian POMDPs, a subclass of POMDPs where beliefs represented as a Gaussian distribution characterized by its mean and covariance, rendering them unsuitable for our problem.

Most existing control synthesis methods for LTL objectives in POMDPs do not explicitly address perception-related tasks, as they primarily use LTL to specify hidden-state behaviors. These approaches typically employ a labeling function that maps hidden states to atomic propositions. Some works [10], [12]–[15] assume that atomic propositions labeled by hidden states are directly observable. However, this assumption is often impractical in real-world applications. Bouton et al. [6] relax this assumption, providing a more realistic framework.

Control synthesis for DTL has been studied in the context of linear Gaussian POMDPs [16]–[18], which, however, are unsuitable for the problem considered here. Linear Gaussian POMDPs form a special subclass of POMDPs where the belief state can be represented as a Gaussian distribution characterized by its mean and covariance. In Linear Gaussian POMDPs, state transitions and observations follow linear functions of the current state and action, with additive Gaussian noise. In contrast, we consider a general POMDP where the belief cannot be represented by a Gaussian distribution. Control synthesis in these approaches relies on discrete approximations of Linear Gaussian POMDPs, meaning the quality of the computed policy relies on the qualities of these approximations.

II. Preliminaries

This section introduces POMDP, belief MDP, and history MDP [19]. POMDP models system dynamics that is, given an action, the probabilistic transition of the hidden states, and the probabilistic outcome of receiving an observation. Belief MDPs with continuous belief state space, capture the transition of optimal estimations of POMDP hidden states. The history MDP refines the belief MDP by restricting to a finite set of beliefs that are reachable within a finite number of steps from the initial belief. Each history state encodes all past actions and observations used to construct a belief. While the number of history states grows over time, this scalability issue can be addressed through best-first search in MCTS.

A. POMDP

We model an autonomous system's dynamics in the unknown and partially observable environment by a POMDP, where the underlying dynamics is an MDP, but the control can only depend on observations probabilistically related to the hidden states.

Definition 1: A POMDP is a tuple $\mathcal{M}_\mathcal{P} = (S, A, T_\mathcal{P}, s_{\text{init}}, \Omega, O)$, where i) S is a finite set of hidden states, ii) A is a finite set of actions, iii) Ω is a finite set of observations, where $\Omega(s)$ denotes the set of possible observations in the state $s \in S$, iv) $T_\mathcal{P} : S \times A \times S \to [0, 1]$ is the transition probability function such that for all $s \in S$, we have $\sum_{s' \in S} T_\mathcal{P}(s, a, s') = 1$, if $a \in A(s)$ and $\sum_{s' \in S} T_\mathcal{P}(s, a, s') = 0$, if $a \notin A(s)$, v) $O : S \times \Omega \to [0, 1]$ is the observation probability function such that for all $s \in S$, we have $\sum_{o \in \Omega} O(s, o) = 1$, vi) s_{init} is the initial distribution of the hidden state.
We call a sequence of states $\sigma_\mathcal{P} : \mathbb{N} \to S$ a *path* of the POMDP if for any $t \in \mathbb{N}$, there exists $a \in A$ such that $T_\mathcal{P}(\sigma_{\mathcal{P}t}, a, \sigma_{\mathcal{P}t+1}) > 0$.

B. Belief MDP and history MDP

The hidden states are not directly observable in a POMDP. But, from the history of observations and actions, one can derive a probabilistic estimation of the hidden states called a belief $b \in B := \Delta(S)$, here we denote the set of probability distributions on S by $\Delta(S)$. A *history* is a sequence of actions and observations, $h_t = \{a_0, o_0, ...a_{t-1}, o_{t-1}\}$ and $h_0 = \emptyset$. We denote the mapping from history h_t to the belief state b_t as $\mathcal{B}(h)$. It can be derived inductively as follows. For all $s \in S$, $b_0(s) = s_{\text{init}}(s)$ and

$$b_t(s) = \frac{O(s, o_t) \sum_{s' \in S} b_{t-1}(s') T_\mathcal{P}(s', a_t, s)}{\sum_{s \in S} O(s, o_t) \sum_{s' \in S} b_{t-1}(s') T_\mathcal{P}(s', a_t, s)}. \quad (1)$$

The transition of belief states with respect to actions can be treated as a continuous state MDP called *belief MDP*.

Definition 2: The belief MDP $\mathcal{M}_\mathcal{B}$ of the POMDP $\mathcal{M}_\mathcal{P} = (S, A, T_\mathcal{P}, s_{\text{init}}, \Omega, O)$ is defined by $\mathcal{M}_\mathcal{B} = (B, A, T_\mathcal{B}, b_0)$ where i) $B := \Delta(S)$ is the continuous belief space, ii) A is the finite set of actions, iii) $b_0 = s_{\text{init}}$, iv) $T_\mathcal{B} : B \times A \times B \to$

[1] The optimal policy for an iLTL objective on a POMDP is not necessarily unique.

$[0, 1]$ is the transition probability function

$$T_{\mathcal{B}}(b, a, b') = \sum_{o \in O} \sum_{s' \in S} \sum_{s \in S} \eta(b, o, b') O(s', o) T_{\mathcal{P}}(s, a, s') b(s) \tag{2}$$

with

$$\eta(b, o, b') = \begin{cases} 1, & \text{if belief update (1) for } b, o \text{ returns } b' \\ 0, & \text{otherwise.} \end{cases}$$

A sequence of belief states $\sigma : \mathbb{N} \to \Delta(S)$ is a *belief path* of the belief MDP if for any $t \in \mathbb{N}$, there exists $a \in A$ such that $T_{\mathcal{B}}(\sigma_t, a, \sigma_{t+1})) > 0$.

Whereas the belief MDP has a continuous state space containing an infinite number of beliefs, we can refine the state space into a finite set of beliefs that are reachable from the initial belief within a finite number of steps.

Definition 3: The history MDP $\mathcal{M}_{\mathcal{H}}$ of the POMDP $\mathcal{M}_{\mathcal{P}} = (S, A, T_{\mathcal{P}}, s_{\text{init}}, \Omega, O)$ is defined by $\mathcal{M}_{\mathcal{H}} = (H, A, T_{\mathcal{H}}, o_0)$ where i) H is the discrete history state space containing all the possible history, ii) A is the finite set of actions, iii) $T_{\mathcal{H}} : H \times A \times H \to [0, 1]$ is the transition probability function $T_{\mathcal{H}}(h, a, hao) = \sum_{s' \in S} \sum_{s \in S} O(s', o) T_{\mathcal{P}}(s, a, s') \mathcal{B}(h)(s)$ iv) $\emptyset$ is the initial history state.

A deterministic memoryless[2] *policy* is $\pi(b_t) \in A$, which uses belief as input and outputs an action. The policy $\pi(\mathcal{B}(h_t))$ returns an action given a history state.

III. Problem Formulation and Main Result

Now, we introduce co-safe linear inequality temporal logic (sc-iLTL) specifying POMDP control objectives. First, we present an example to show that some objectives can be specified on the belief path σ but are hard to specify on the hidden state path $\sigma_{\mathcal{P}}$. Then, we formally define our control objective and show the main result of solving it.

A. Motivating example

The common way we specify the control objectives on POMDP is through hidden states, which may not be expressive enough for a perception-related task.

Example 1: Consider a drone probing task in a grid world (inspired by [20]). The drone needs to locate a ground target on the grid using an imperfect sensor. The **sensor** only provides the respective quadrant of the target within the field of view, that is, adjacent to or under the drone. The set of observations is $\Omega = \{\mathrm{SW}, \mathrm{NW}, \mathrm{NE}, \mathrm{SE}, \mathrm{None}\}$, where the first four observations stand for the respective quadrant and "None" means the target is not in the field of view. Suppose the target is at the adjacent grid north of the drone, then the sensor returns "NE" or "NW" with equal probability, and if the target is under the drone, then the sensor returns an observation in $\{\mathrm{SW}, \mathrm{NW}, \mathrm{NE}, \mathrm{SE}\}$ with equal probability.

When we model the environment using POMDP, the hidden state space contains all possible positions of the drone and ground target. Whereas the belief state is the optimal estimation of the distribution of ground target position given all previous actions and observations. We can specify such a task as increasing the maximum element of the belief state to a confidence level c.

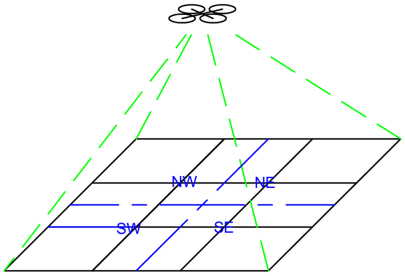

Fig. 1. Schematic diagram illustrating the limited field of view and imperfect sensing on a 3×3 grid. The sensor can only detect the target if it is inside the field of view, which is marked by green lines. Meanwhile, the sensor returns only relative quadrant observations $\{\mathrm{SW}, \mathrm{NW}, \mathrm{NE}, \mathrm{SE}\}$ when the target is within the field of view.

$$\|b\|_\infty \geq c$$

It is difficult to specify this task using objectives defined on hidden states; meanwhile, it can be specified using objectives defined on belief states. Setting $\|b\|_\infty \geq c$ implicity requires the drone to approach the target in a specific way that can increase $\|b\|_\infty$ by belief update (1). Traditional control objectives usually use reward $r(s, a) : S \times A \to \mathbb{R}$ with hidden states as input or LTL with atomic propositions defined on hidden states. However, even if the drone is right above the target, $\|b\|_\infty$ may still be below the confidence c. Traditional control objectives has to explicitly encode the belief update to define such a perception-related objective.

B. Sc-iLTL

We express the perception-related objective by a sc-iLTL formula constructed via atomic propositions defined on belief states. The definition and verification of the iLTL formula [1] are similar to the standard LTL formula [3] except for atomic propositions. An iLTL formula is derived recursively from the rules

$$\varphi ::= \text{true} \mid \text{ineq} \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \mathcal{U} \varphi_2, \ \text{ineq} \in \Lambda \tag{3}$$

Other propositional and temporal operators can be derived from previous operators, e.g., (or) $\varphi_1 \vee \varphi_2 := \neg(\neg\varphi_1 \wedge \neg\varphi_2)$, (eventually) $\Diamond\varphi := \text{true} \cup \varphi$.

In this work, we focus on the co-safe fragment of the iLTL formula, which can be verified via finite prefixes. We say that the iLTL formula φ is co-safe [21] if all the infinite sequences $\sigma \models \varphi$ satisfy: there exists a truncated finite sequence $\sigma_{\leq k} = b_0 b_1 \cdots b_k$, $k \in \mathbb{N}$ such that $\sigma_{\leq k} \cdot \sigma' \models \varphi$ for any infinite sequence σ'. Sc-iLTL requires $\neg$ only to apply directly to atomic propositions, and the formula only uses the temporal operators $\bigcirc$, $\mathcal{U}$, and $\Diamond$. Let $\sigma : \mathbb{N} \to \Delta(S)$ be a belief path of the POMDP. The satisfaction (denoted by $\models$) of an sc-iLTL formula follows the standard rule for co-safe LTL [21] except for

$$\sigma \models \text{ineq iff } p^T \sigma_0 > c \text{ with } p \in \mathbb{R}^{|S|}, c \in \mathbb{R}.$$

[2] Here, memoryless stands for only using current belief as input. We only consider deterministic policy in this work.

> **Problem Formulation:** For a POMDP $\mathcal{M}_\mathcal{P} = (S, A, T_\mathcal{P}, s_{\text{init}}, \Omega, O)$ and a sc-iLTL objective φ specifing the belief path σ. Find an optimal policy π that maximizes the satisfaction probability $Pr_\pi(\sigma \models \varphi)$.

Solving this control problem with an objective specifying the belief path is challenging due the memory-dependency and scalarability. Whereas an optimal policy satisfying the sc-iLTL typically is memory-dependent since the sc-iLTL involves reasoning of sequential ordering of events. We can transform the control problem into a maximum reachability problem on a continuous state belief MDP, which accepts a memoryless policy to be the optimal. Such a transformation is done by constructing the product of the belief MDP and the graph representation of a sc-iLTL objective. Meanwhile, control synthesis in belief space inherently faces scalability issues as the continuous state space contains an infinite number of belief states.

We use deterministic finite automata (DFAs) as a graph representation of a sc-iLTL objective thus simplifying the complex rule-based objective into reachability. Given a sc-iLTL objective φ, we can construct an DFA $\mathcal{A}_\varphi$ (with labels $\Sigma = 2^\Lambda$) such that a belief path $\sigma \models \varphi$ if and only if σ is accepted by the DFA $\mathcal{A}_\varphi$ which has a reachability objective. *A DFA is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ where i) Q is a finite set of automaton states, ii) Σ is a finite set of alphabets, iii) $\delta : Q \times \Sigma \to 2^Q$ is a (partial) transition function, iv) $q_{\text{init}} \in Q$ is the initial automaton state, v) $F \subseteq Q$ is a set of final automaton states.* The DFA verifies the belief path by a labeling function $L : B \to 2^\Lambda$, where $\text{ineq} \in L(b)$ if and only if ineq holds on b. The labeling on the belief path moves the automation states. A belief path σ is accepted by $\mathcal{A}_\varphi$ if and only if there exists $k \in \mathbb{N}$ such that the prefix $\sigma_{\leq k}$ moves the initial automaton state q_0 to a final state $q \in F$.

MCTS [8], [22] gives us the ability to solve the maximum reachability problem from the history MDP point of view. Since we specify the behavior of a finite prefix, we are dealing with a finite horizon. We only need to argue a finite number of belief states that are reachable from b_0. Thus, our problem can be treated as a reachability problem on a history MDP with finite state space. Meanwhile, MCTS can mitigate the scalability issue brought by the exponential growth of the number of history states via best-first search. Specifically, given a finite horizon POMDP, the value function estimated by MCTS converges in probability to the optimal value function. We can apply MCTS to our product belief MDP and find the maximal reachability probability with a convergence guarantee.

Theorem 1: Our algorithm 1 finds the optimal policy on the POMDP that maximizes the satisfaction probability of the given sc-iLTL objective.
We finish the proof of our main result at the end of Section IV-C.

IV. MCTS for Sc-iLTL

In this section, we introduce our algorithm to solve the control synthesis problem for the sc-iLTL obejective on POMDPs. First, in order to find a memory-dependent optimal policy, we transform the sc-iLTL control problem into a reachability problem where a memoryless policy can be optimal by constructing the product belief MDP and show that both problems share equivalent optimal policies. Then, We apply MCTS to find the optimal policy.

A. Product belief MDP

We translate the control problem for the POMDP into a reachability problem on the product belief MDP.

Definition 4: A product belief MDP $\mathcal{M}^\times_\mathcal{B} = (B^\times, A^\times, P^\times, b_0^\times, F^\times)$ of an belief MDP $\mathcal{M}_\mathcal{B} = (B, A, T_\mathcal{B}, b_0)$ and an DFA $\mathcal{A} = (Q, \Sigma, \delta, q_{\text{init}}, F)$ is defined by i) the set of states $B^\times = B \times Q \cup \text{sink}$, ii) the set of actions $A^\times = A$, iii) the the set of final states $F^\times = \{\langle b, q\rangle \in B^\times | q \in F\}$, iv) the initial product belief state $b_0^\times = \langle b_0, q_{\text{init}}\rangle$, v) the transition probability function

$$T^\times_\mathcal{B}(b^\times, a, b^{\times\prime}) = \begin{cases} T_\mathcal{B}(b, a, b') & b^\times = \langle b, q\rangle \notin F^\times, \\ & b^{\times\prime} = \langle b', q'\rangle, \\ & q' = \delta(q, L(b)) \\ 1 & b^\times \in F^\times,\ b^{\times\prime} = \text{sink} \\ 1 & b^\times = \text{sink},\ b^{\times\prime} = \text{sink} \\ 0 & \text{else.} \end{cases} \tag{4}$$

The transitions of the product MDP $\mathcal{M}^\times_\mathcal{B}$ are derived by combining the transitions of the belief MDP $\mathcal{M}_\mathcal{B}$ and the DFA $\mathcal{A}$. The path on the product belief MDP falls to the sink once $F^\times$ is met. Each belief path $\sigma \models \varphi$ will have a corresponding path $\sigma^\times$ on the product belief MDP $\mathcal{M}^\times$ visiting $F^\times$ in finite steps and vice versa. Here, we use $\sigma^\times \models \Diamond F^\times$ to describe the event of visiting $F^\times$ in finite steps. We introduce the sink since we only care about the finite prefix of $\sigma^\times$ before visiting $F^\times$. Adding the sink helps in simplifying all the transitions after $F^\times$ is visited.

B. Reachability problem on the product belief MDP

Now we formally formulate the reachability problem on the product belief MDP.

We introduce an undiscounted reward function

$$r(b^\times) = \begin{cases} 1 & b^\times \in F^\times \\ 0 & \text{else,} \end{cases} \tag{5}$$

such that the expected return of the cumulative reward along all paths starting at $b_0^\times$ also know as the value function $V_{\pi^\times}(b_0^\times) := \mathbb{E}_\pi \sum_{i=0}^\infty r(b_i^\times)$ equals the reachability probability.

Lemma 1: Given a policy $\pi^\times$, belief state b and DFA state q, it holds that

$$V_{\pi^\times}(\langle b, q\rangle) = Pr_{\pi^\times}(\sigma^\times \models \Diamond F^\times | \sigma_0^\times = \langle b, q\rangle). \tag{6}$$

Proof: Along a belief path visiting $F^\times$, reward 1 will be collected only once. Along a belief path not visiting $F^\times$, no positive reward will be collected. Thus $V_{\pi^\times}(b_0^\times) := \mathbb{E}_\pi \sum_{i=0}^\infty r(b_i^\times)$ equals the reachability probability to $F^\times$. ■

Given a policy, the value function satisfies the recursive formulation as the Bellman equation, when $b^{\times} \neq \text{sink}$,

$$V_{\pi^{\times}}(b^{\times}) = r(b^{\times}) + \sum_{b^{\times'}} T_{\mathcal{B}}^{\times}(b^{\times}, \pi(b^{\times}), b^{\times'}) V_{\pi^{\times}}(b^{\times'}), \quad (7)$$

when $b^{\times} = \text{sink}$,

$$V_{\pi^{\times}}(b^{\times}) = 1.$$

The optimal value function representing the maximum reachability probability satisfies the Bellman optimality equation, when $b^{\times} \neq \text{sink}$,

$$V^*(b^{\times}) = r(b^{\times}) + \max_{a \in A} \sum_{b^{\times'}} T_{\mathcal{B}}^{\times}(b^{\times}, a, b^{\times'}) V^*(b^{\times'}), \quad (8)$$

when $b^{\times} = \text{sink}$,

$$V^*(b^{\times}) = 1.$$

The policy $\pi^{\times}(\langle b, q \rangle)$ requires us to update the belief states and the DFA state at the same time. Given current belief and DFA state b_t, q_t, the action is $a_t = \pi^{\times}(\langle b_t, q_t \rangle)$.

Lemma 2: Given sc-iLTL objective φ and POMDP $\mathcal{M}_{\mathcal{P}}$, the optimal policy $\pi^{\times}(\langle b, q \rangle)$ on $\mathcal{M}_{\mathcal{B}}^{\times}$ maximizing reachability probability is the optimal policy $\pi^{\times}(\langle b, q \rangle)$ on $\mathcal{M}_{\mathcal{P}}$ maximizing satisfication probability of φ.

Proof: Given DFA $\mathcal{A}_{\varphi}$ with initial DFA state $q_0 = q_{\text{init}}$, following the same policy $\pi^{\times}$, the satisfaction probability on POMDP equals the reachability probability on product belief MDP, $Pr_{\pi^{\times}}(\sigma \models \varphi) = Pr_{\pi^{\times}}(\sigma^{\times} \models \Diamond F^{\times} | \sigma_0^{\times} = \langle \sigma_0, q_0 \rangle)$. Since each belief path σ has a unique corresponding path $\sigma^{\times}$ where $\sigma_0^{\times} = \langle \sigma_0, q_0 \rangle$, Lemma 1 and (8) finishes the proof. ■

C. Monte-Carlo tree search

Here, we use MCTS to find the optimal policy on the product belief MDP. We use a discrete history state to represent a continuous belief state similar to [8]. In this way, MCTS only searches a finite number of belief states that are reachable from b_0.

We use an equivalent product history state $h^{\times} = \langle h, q \rangle$ to represent each product belief state $b^{\times} = \langle \mathcal{B}(h), q \rangle$ by (1). The transition between the product history states is given as

$$T_{\mathcal{H}}^{\times}(h^{\times}, a, h^{\times'}) = \begin{cases} T_{\mathcal{B}}^{\times}(b^{\times}, a, b^{\times'}) & h^{\times} = \langle h, q \rangle,\ h^{\times'} = \langle hao, q' \rangle, \\ & q' = \delta(q, L(\mathcal{B}(h))), \\ & b^{\times} = \langle \mathcal{B}(h), q \rangle,\ b^{\times'} = \langle \mathcal{B}(hao), q' \rangle) \\ 1 & h^{\times} = \langle h, q \rangle, q \in F,\ h^{\times'} = \text{sink} \\ 1 & h^{\times} = \text{sink},\ h^{\times'} = \text{sink} \\ 0 & \text{else}, \end{cases} \quad (9)$$

MCTS extends the UCB1 method from the Bandit problem to a decision tree. The decision tree is constructed based on the Bellman optimality equation (8). Each state node $G(h^{\times}) = \langle N(h^{\times}), \hat{V}(h^{\times}), \mathcal{B}(h^{\times}) \rangle$ [3] contains $N(h^{\times})$ counts the number of times that $h^{\times}$ has been visited, $\hat{V}(h^{\times})$ is the value estimation of $\mathcal{B}(h^{\times})$ by the mean return of all simulations starting with $\mathcal{B}(h^{\times})$. New nodes are initialized to $\langle \hat{V}_{init}(h^{\times}), N_{init}(h^{\times}), \mathcal{B}(h^{\times}) \rangle$ if the knowledge is available, and to $\langle 0, 0, \mathcal{B}(h^{\times}) \rangle$ otherwise. The children to state node $G(h^{\times})$ are action nodes $G(h^{\times}a) = \langle N(h^{\times}, a), \hat{V}(h^{\times}a), \mathcal{B}(h^{\times}) \rangle$. The children to the action node are state node $G(h^{\times'})$ where $h^{\times'}$ is reachable from $h^{\times}$ via action a.

Each simulation on the tree starts in a hidden state $s \sim \mathcal{B}(h^{\times})$. In the simulation, if we have knowledge of all the child nodes, actions are selected by $\arg\max_a \left\{ \hat{V}(h^{\times}a) + c\sqrt{\frac{\log N(^{\times})}{N(h^{\times}, a)}} \right\}$. Otherwise, a rollout policy $\pi_{rollout}$ (uniform random action selection) is applied. Each simulation will add one new node to the tree. When the search is complete, the algorithm returns the optimal action for $h^{\times}$.

V. Guaranteed Convergence to the Optimal Policy

The key challenge to finding the optimal policy is the large number of historical states, which grows exponentially with respect to the length of the path. MCTS handles such a large state space as it uses best-first search; thus, its complexity is not related to the size of the state space.

Further, the finding of the optimal policy on POMDP for the iLTL objective is guaranteed as follows. First, MCTS finds the optimal policy for the reachability problem on the product belief MDP. Then, the optimal policy on the original POMDP is recovered.

Theorem 2: MCTS returns the optimal memoryless policy for the reachability problem on the product belief MDP $\mathcal{M}_{\mathcal{B}}^{\times}$. In the sense that the probability of MCTS failed to return the optimal action given the history state $h^{\times}$ converges to zero as

$$\lim_{N(h^{\times}) \to \infty} Pr(\pi(\mathcal{B}(h^{\times}) \neq a^*(h^{\times})) = 0, \quad (10)$$

meanwhile, the estimation error of the value function is bounded by

$$|\mathbb{E}[V^*(\mathcal{B}(h^{\times})) - \hat{V}(h^{\times})]| \leq O\left(\frac{\log(N(h^{\times})}{N(h^{\times})}\right) \quad (11)$$

Proof: This proof can be done by applying Theorem 1 of [8] to the product belief MDP. We provided a detailed version of the proof in the appendix. Applying the UCB1 method to a non-stationary bandit problem shows the guaranteed convergence to an optimal value function and optimal action. Suppose we fix $V(b^{\times'})$, then the bandit problem is stationary and can be solved by the UCB1 method. However, when we search the root, value estimations on the follow-up nodes also change, so we call this non-stationary. [22] transform an MDP problem into a non-stationary bandit problem. They utilize the convergence results on applying the UCB1 method to the non-stationary bandit problem in [23] to get the result on MDPs.

MCTS finds the optimal policy for the reachability problem on the product belief MDP. The reachability problem

[3] Here we use $\mathcal{B}(h^{\times})$ to represent $\mathcal{B}(h)$ where h is inside $h^{\times}$.

Algorithm 1 MCTS for the Product Belief MDP

```
Procedure Search($h^\times = \langle h, q \rangle$)
  repeat
    if $h = \emptyset$ then
      $s \sim s_{\text{init}}$
    else
      $s \sim \mathcal{B}(h)$
    end if
    Simulate($s, h^\times, 0$)
  until Timeout()
  return $\arg\max_a \hat{V}(h^\times a)$
end procedure

Procedure Rollout($s, h^\times, depth$)
  if $depth \geq d_{\text{max}}$ or $h^\times = \text{sink}$ then
    return 0
  end if
  $a = \pi_{rollout}(\mathcal{B}(h))$
  $s' \sim T_{\mathcal{P}}(s, a, \cdot)$
  $h^{\times'} \sim P^\times_{\mathcal{H}}(h^\times, a, \cdot)$
  return $r(\mathcal{B}(h)) + \text{Rollout}(s', h, \text{depth} + 1)$
end procedure

Procedure Simulate($s, h^\times, depth$)
  if $depth \geq d_{\text{max}}$ or $h^\times = \text{sink}$ then
    return 0
  end if
  if $h^\times \notin G$ then
    for all $a \in A$
      $G(h^\times, a) \leftarrow \langle \hat{V}_{init}(h^\times), N_{init}(h^\times), \mathcal{B}(h^\times) \rangle$
    end for
    return Rollout($s, h^\times, depth$)
  end if
  $a \leftarrow \arg\max_d \left( \hat{V}(h^\times, d) + c \cdot \sqrt{\frac{\log N(h^\times)}{N(h^\times, d)}} \right)$
  $s' \sim T_{\mathcal{P}}(s, a, \cdot)$
  $h^{\times'} \sim P^\times_{\mathcal{H}}(h^\times, a, \cdot)$
  $R \leftarrow r(\mathcal{B}(h)) + \text{Simulate}(s', h^{\times'}, depth + 1)$
  $N(h^\times) \leftarrow N(h) + 1$
  $N(h^\times, a) \leftarrow N(h, a) + 1$
  $\hat{V}(h^\times, a) \leftarrow \hat{V}(h, a) + \frac{R - \hat{V}(h^\times, a)}{N(h^\times, a)}$
  return $R$
end procedure
```

is defined by maximizing the expected return of an undiscounted bounded reward function. By apply Theorem 3,5 in [22], we show

$$\lim_{N(h^\times) \to \infty} Pr(\pi(\mathcal{B}(h^\times) \neq a^*(h^\times)) = 0$$

and

$$|\mathbb{E}[V^*(\mathcal{B}(h^\times)) - \hat{V}(h^\times)]| \leq O(\frac{\log(N(h^\times)}{N(h^\times)})$$

■

Now we we finish the proof for our main result Theorem 1. Given the result of MCTS on product belief MDP. We can recover the policy on the original POMDP which is optimal for the given sc-iLTL objective.

Proof: [Proof of Theorem 1] The finding of memory-dependent policy on the POMDP for the iLTL objective is guaranteed by i) Lemma 2 shows the optimal policy on product belief MDP and is the optimal policy on POMDP. ii) Theorem 2 guarantees that MCTS returns the optimal memoryless policy $\pi^\times(\langle b, q \rangle)$ on the product belief MDP. iii) the optimal memoryless policy on the product belief MDP is the optimal policy. The reason is sc-iLTL objective, verified via finite prefix, can be translated into a reachability problem considering finite horizon. Considering the exponential growth of history states, only a finite number of belief states will be visited for a path starting from the initial belief state given arbitrary policy. For an MDP with finite state space, the memoryless policy is sufficient to be the optimal policy. Thus, the memoryless policy is sufficient for the reachability problem on the product belief MDP.

Thus, MCTS finds the optimal policy for sc-iLTL objective φ on POMDP. ■

VI. Case Study

To show the expressiveness of the iLTL objective, we introduce the drone-probing problem. We apply our MCTS algorithm on the problem 100 times. We evaluated the performance of the MCTS algorithm by the percentage of success simulations.

A. Drone-probing problem and iLTL objective

The drone-probing problem is in a 4×4 gride world with 256 hidden states. The **drone** on the grid has an action set $\{\text{N}, \text{S}, \text{E}, \text{W}, \text{X}\}$ stands for moving to the four different directions or staying in the current grid. Given the action, the movement is deterministic, and the drone stays in the current grid if a movement hits the edge. The drone is initialized on the grid $(0, 0)$. The **ground target** will move randomly on the grid as We assign equal probability to all the possible following locations of the target. The target is initialized on the grid, excluding the corner with equal probability. The **sensor** has a limited field of view and only returns respective quadrant from $\Omega = \{\text{SW}, \text{NW}, \text{NE}, \text{SE}, \text{None}\}$ as described in section III-A.

The expressiveness of the sc-iLTL formula allows us to specify the following task. Suppose we want the drone to use its imperfect sensor to accurately locate the ground target and then move to the landing zone $(3, 3)$. Reaching the landing zone before getting an accurate measure is seen as a failure. Not reaching the landing zone until within the simulation horizon 100 is seen as a failure. We define the control objective with the following iLTL formula,

$$\begin{aligned}\varphi =& \Diamond \Big(\bigvee_{i=1}^{256} \text{ineq}_{\text{measure},i} \Big) \wedge \Diamond (\text{ineq}_{\text{goal}}) \\ & \wedge (\neg \text{ineq}_{\text{goal}} \mathcal{U} \bigvee_{i=1}^{256} \text{ineq}_{\text{measure},i}),\end{aligned} \tag{12}$$

where $\Lambda = \{\text{ineq}_{\text{measure},1}, \text{ineq}_{\text{measure},2}, ..., \text{ineq}_{\text{measure},256}, \text{ineq}_{\text{goal}}\}$ is the set of atomic propositions, each $\text{ineq}_{\text{measure},i} := p_i^T b > 0.9$ is a linear inequality over the belief space measures the confidence of the location of the ground target, p_i is a vector with all zero elements expect the i-th entry to be 1, b is the belief state, $\text{ineq}_{\text{goal}} := p_g^T b \geq 1$ stands for the event of reaching the landing zone.

It is hard to specify the task with standard LTL formula defined over hidden state space. Satisfying $\Diamond\big(\bigvee_{i=1}^{256} \text{ineq}_{\text{measure},i}\big)$ requires the drone to chase the target in a manner that can reason about the actual location of the target. Even if the drone is right above the target, $\|b\|_\infty$ may still be below the confidence 0.9.

B. Experiment

The experiments are conducted on a Windows 11 machine equipped with an Intel i9-14900K processor. The implementation is done using Python. We use the mona package [24] to translate the LTL objective into DFA. The DFA has 4 states, thus increasing the number of all product history states by 4 times.

We run the experiment for 100 times. In each experiment, the drone and ground target are spawned based on the initial belief. Then, we apply MCTS to get an action. After receiving an observation and moving to the hidden state, we apply MCTS again to get action. For each MCTS search, we apply 2000 simulations and set the depth of the tree d_max to 20.

We show the performance of our algorithm using a histogram and the average change of $\|b\|_\infty$ in fig 2. 87 out of 100 experiments are successful. The drone needs an average of 40.71 steps to finish the task. 4 out of 13 failed experiments are due to not returning to the landing zone before the horizon 100. Other failed experiments are due to the randomness of MCTS output.

VII. Conclusion

This work proposes an sc-iLTL objective for POMDPs which the optimal policy can be synthesised via MCTS. The iLTL objective defined in belief space is more expressive than the LTL objective defined using hidden states. Specifically, we utilize inequality in belief space to specify objectives on POMDPs. Sc-iLTL is suitable for objectives related to safety or surveillance. The transformation of the control for the sc-iLTL objective into a reachability problem on the product belief MDP enables us to leverage MCTS to find optimal policies effectively. Experiments in the drone-probing problem demonstrate the expressiveness of the sc-iLTL objective and the performance of our MCTS method.

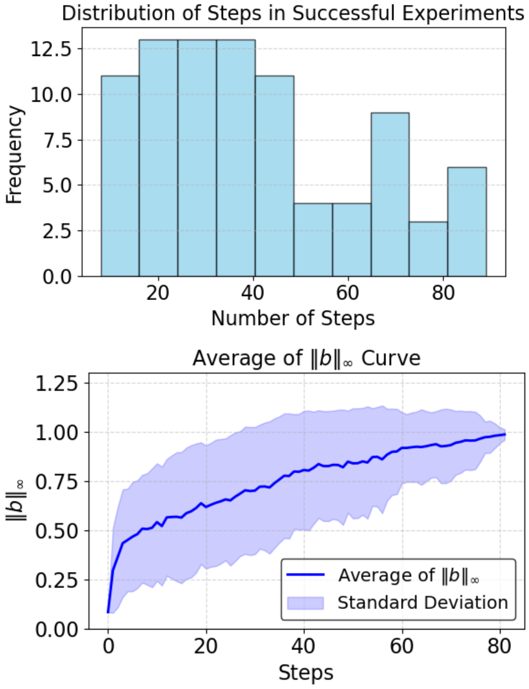

Fig. 2. Histograms of steps in success runs (left) and change of average $\|b\|_\infty$ during success runs (right). We fix $\|b\|_\infty$ once it goes beyond 0.9. Each action is returned by MCTS using 2000 simulations with depth $d_{max} = 20$. Our MCTS algorithm achieves 87% success rate for φ. We see the MCTS algorithm tries to increase $\|b\|_\infty$ up to the confidence predefined and finishes the task in an average of 40.71 steps.

References

[1] Y. Kwon and G. Agha, "Linear Inequality LTL (iLTL): A Model Checker for Discrete Time Markov Chains," in *Formal Methods and Software Engineering*, J. Davies, W. Schulte, and M. Barnett, Eds. Berlin, Heidelberg: Springer, 2004, pp. 194–208.

[2] O. Kupferman and M. Y. Vardi, "Model Checking of Safety Properties," *Formal Methods in System Design*, vol. 19, no. 3, pp. 291–314, Nov. 2001.

[3] C. Baier and J.-P. Katoen, *Principles of Model Checking*. The MIT Press, 2008.

[4] A. Lavaei, S. Soudjani, A. Abate, and M. Zamani, "Automated verification and synthesis of stochastic hybrid systems: A survey," *Automatica*, vol. 146, p. 110617, Dec. 2022.

[5] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs," in *International Joint Conference on Artificial Intelligence*, Aug. 2003.

[6] M. Bouton, J. Tumova, and M. J. Kochenderfer, "Point-Based Methods for Model Checking in Partially Observable Markov Decision Processes," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 06, pp. 10 061–10 068, Apr. 2020.

[7] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based POMDP solvers," *Autonomous Agents and Multi-Agent Systems*, vol. 27, no. 1, pp. 1–51, Jul. 2013.

[8] D. Silver and J. Veness, "Monte-Carlo Planning in Large POMDPs," in *Advances in Neural Information Processing Systems*, vol. 23. Curran Associates, Inc., 2010.

[9] H. Geffner and B. Bonet, "Solving Large POMDPs using Real Time Dynamic Programming," 1998.

[10] B. Bonet and H. Geffner, "Solving POMDPs: RTDP-Bel vs. Point-based Algorithms," in *International Joint Conference on Artificial Intelligence*, Jul. 2009.

[11] A. Jones, M. Schwager, and C. Belta, "Distribution temporal logic: Combining correctness with quality of estimation," in *52nd IEEE Conference on Decision and Control*, Dec. 2013, pp. 4719–4724.

[12] K. Chatterjee, M. Chmelík, R. Gupta, and A. Kanodia, "Qualitative analysis of POMDPs with temporal logic specifications for robotics applications," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 325–330.

[13] M. Ahmadi, R. Sharan, and J. W. Burdick, "Stochastic Finite State Control of POMDPs with LTL Specifications," no. arXiv:2001.07679. arXiv, Jan. 2020.

[14] R. Sharan and J. Burdick, "Finite state control of POMDPs with LTL specifications," in *American Control Conference*, Jun. 2014, pp. 501–508.

[15] R. Sharan, "Formal Methods for Control Synthesis in Partially Observed Environments: Application to Autonomous Robotic Manipulation," Ph.D. dissertation, California Institute of Technology, 2014.

[16] C.-I. Vasile, K. Leahy, E. Cristofalo, A. Jones, M. Schwager, and C. Belta, “Control in belief space with Temporal Logic specifications,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec. 2016, pp. 7419–7424.

[17] K. Leahy, E. Cristofalo, C.-I. Vasile, A. Jones, E. Montijano, M. Schwager, and C. Belta, “Control in belief space with temporal logic specifications using vision-based localization,” *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 702–722, May 2019.

[18] S. Haesaert, P. Nilsson, C. I. Vasile, R. Thakker, A. Agha-mohammadi, A. D. Ames, and R. M. Murray, “Temporal Logic Control of POMDPs via Label-based Stochastic Simulation Relations,” *IFAC-PapersOnLine*, vol. 51, no. 16, pp. 271–276, Jan. 2018.

[19] J. Li, “Reinforcement Learning-Based Motion Planning in Partially Observable Environments for Complex Tasks,” Ph.D. dissertation, United States – Iowa, 2024.

[20] M. Svoreňová, M. Chmelík, K. Leahy, H. F. Eniser, K. Chatterjee, I. Černá, and C. Belta, “Temporal logic motion planning using POMDPs with parity objectives: Case study paper,” in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC ’15. New York, NY, USA: Association for Computing Machinery, Apr. 2015, pp. 233–238.

[21] B. Lacerda, D. Parker, and N. Hawes, “Optimal and dynamic planning for Markov decision processes with co-safe LTL specifications,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2014, pp. 1511–1516.

[22] L. Kocsis and C. Szepesvári, “Bandit Based Monte-Carlo Planning,” in *Machine Learning: ECML 2006*, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Eds. Berlin, Heidelberg: Springer, 2006, pp. 282–293.

[23] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time Analysis of the Multiarmed Bandit Problem,” *Machine Learning*, vol. 47, no. 2, pp. 235–256, May 2002.

[24] N. Klarlund and A. Møller, “Mona version 1.4 user manual,” *B R I C S Notes Series*, no. NS-01-1, 2001.

Appendix

In the Appendix, we apply the proof in [22] to our product belief MDP.

MCTS treats the Bellman optimality equation

$$V^*(b^\times) = R(b^\times) + \max_{a\in A} \sum_{b^{\times\prime}} P^\times_{\mathcal{B}}(b^\times, a, b^{\times\prime}) V^*(b^{\times\prime}), \quad (13)$$

as a non-stationary bandit problem with $|A|$ arms. The bandit problem is defined by the sequence of random payoffs $X_{it}, i = 1, ..., K, t \geq 1$, where each i is the index of an action. Successive choose action i from the same state $b^\times$ yield the payoffs X_{i1}, X_{i2}, ... The payoff sequence is non-stationary since during the MCTS, the value estimation $\hat{V}(h^{\times\prime})$ for successive node changes when we estimate $\hat{V}(h^\times)$.

MCTS uses UCB1 to choose the action with the best upper confidence bound:

$$I_t = \arg_{i\in\{1,...,K\}]} \max \left\{ \bar{X}_{i,T_i(t-1)} + c_{t-1,T_i(t-1)} \right\} \quad (14)$$

where $T_i(n) = \sum_{s=1}^n \mathbb{I}(I_s = i)$ is the number of times action i was played up to time n, $I_t \in \{1, ..., K\}$ is the index of the action selected at time t, $c_{t,s} = 2C_p\sqrt{\ln t/s}$ is a bias sequence.

During the MCTS, the value estimation $\hat{V}(h^{\times\prime})$ for follow-up node changes when we estimate $\hat{V}(h^\times)$. Thus, the payoff sequence is non-stationary.

MCTS allows the payoff sequence X_{it} to be the non-stationary as long as it follows the drift condition. That is, the averages $\bar{X}_{in} := 1/n \sum_{t=1}^n X_{it}$ converges to the expectation μ_i

$$\begin{aligned} \mu_i &= \lim_{n\to\infty} \mu_{in} \\ \mu_{in} &= \mathbb{E}[\bar{X}_{in}] = \mu_i + \delta_{in}. \end{aligned} \quad (15)$$

For a finite horizon MDP with bounded reward function, or in our case, the product belief MDP with reward function being either 0 or 1, we can show that the drift condition always holds.

Lemma 3: Consider the product belief MDP of the belief MDP and a DFA, let k be the length of the truncated finite sequence $\sigma_{\leq k} = b_0 b_1 \cdots b_k$ used to verify the sc-iLTL objective. Then the bias of the esimated expected payoff $\bar{X}_n$ is $O((k|A|\log n + |A|^k)/n)$

If the drift condition holds, the expected estimation error is bounded as

Lemma 4: [22, Theorem 3] Let $\bar{X}_n = \sum_{i=1}^K \frac{T_i(n)}{n} \bar{X}_{i,T_i(n)}$, let μ^* be the expectation for μ_i where i is the optimal action and $\delta^*_n = \mu^*_n - \mu^*$,

$$|\mathbb{E}[\bar{X}_n] - \mu^*| \leq |\delta^*_n| + O(\frac{K(C_p^2 \ln(n) + N_0)}{n}). \quad (16)$$

The estimated optimal payoff concentrates on the mean by

Lemma 5: [22, Theorem 5] Fix $\delta > 0$ and let $\Delta_n = 9\sqrt{2n\ln(2/\delta)}$. The following bounds hold true provided that n is sufficiently large: $Pr(n\bar{X}_n \geq n\mathbb{E}[\bar{X}_n] + \Delta_n) \leq \delta$, $Pr(n\bar{X}_n \leq n\mathbb{E}[\bar{X}_n] - \Delta_n) \leq \delta$.

Proof of Lemma 3

[22, Theorem 7] The proof is done by applying Lemma 4, 5 inductively on the length of the truncated finite sequence k. Consider setting the case $k = 1$. Hoeffding's inequality holds the assumptions on the payoffs.

Assume the result holds for the tree up to depth $d - 1$ and consider a tree of depth d. Consider the root node. The result followed by Lemma 4, 5 still holds. ■

With the guarantee of drift condition, the estimated optimal payoff concentrates around its mean,

Lemma 6: [22, Theorem 6] we can get the optimal action with an accurate guarantee

$$\lim_{t \to \infty} Pr(I_t \neq i^*) = 0. \tag{17}$$

Proof of Theorem 2

Equation (17) holds for all nodes due to Lemma 3. Thus, MCTS returns the optimal action with probability 1 when n is sufficiently large. ■