2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022)
October 23-27, 2022, Kyoto, Japan

Planning with Intermittent State Observability: Knowing When to Act Blind

Connor Basich[1], John Peterson[1], and Shlomo Zilberstein[1]

Abstract— **Contemporary planning models and methods often rely on constant availability of free state information at each step of execution. However, autonomous systems are increasingly deployed in the open world where state information may be costly or simply unavailable in certain situations. Failing to account for sensor limitations may lead to costly behavior or even catastrophic failure. While the *partially observable Markov decision process* (POMDP) can be used to model this problem, solving POMDPs is often intractable. We introduce a planning model called a *semi-observable Markov decision process* (SOMDP) specifically designed for MDPs where state observability may be intermittent. We propose an approach for solving SOMDPs that uses *memory states* to proactively plan for the potential loss of sensor information while exploiting the unique structure of SOMDPs. Our theoretical analysis and empirical evaluation demonstrate the advantages of SOMDPs relative to existing planning models.**

I. Introduction

As AI and robotics have advanced in recent years, attention has shifted to the deployment of autonomous systems in the open world. The increased complexity and uncertainty exhibited in open-world domains challenges some of the typical assumptions made in planning, such as domain stationarity, unexpected scenarios, and sensor and perception reliability. *Markov decision processes* (MDPs) have been shown to be effective in a wide array of domains [1], but they rely on exact state information at each step of execution, often acquired from sensor data and processed by perception algorithms [2]. The acquisition of state information, whether full or partial, is often assumed to be free and consistently available. However, practical limitations may constrain the agent's ability to observe the state of the environment.

Constraints on the availability of state information could arise in a variety of ways. Sensing may have a non-negligible cost that makes it prohibitive. For example, an extraterrestrial science robot may have a finite and non-repletable battery supply that sensing actions consume [3]. Sensor information may simply be unavailable at various times during a system's deployment due to technical limitations or by design. For example, location obtained using a GPS may be unavailable when the system is underground. Finally, sensor information may be available, and even free, but it may be periodically completely unreliable due to interference, resulting in potentially misleading information. For example, an object in front of a robot may not be recognizable due to glare [4]. Failing to *proactively* account for such situations can lead to costly erratic behavior or critical failures [4]. Recent work in introspective perception examines ways to learn to detect various forms of sensor failures [4], [5]. Hence, it is important to develop complementary planning algorithms that can take into account the aforementioned limitations.

The *partially observable Markov decision process* (POMDP) [6] can be used to model this problem, including various forms of sensor noise or the entire loss of sensor information, which could be indicated by a special observation. However, POMDP solvers are notoriously complex [7] and are not designed for the unique properties of planning with intermittent state observability (i.e., the state is always either fully observable or unobservable). Hence, we propose a more cost-effective approach for this class of problems.

Specifically, we propose the *semi-observable Markov decision process* (SOMDP), which models domains where the system's current state is only intermittently observable. We demonstrate that there are distinct computational advantages for using our model in domains with intermittent observability, compared to POMDPs that are computationally harder to solve [8]. In particular, we show how to approximate the SOMDP with a depth-limited *memory-state MDP* (MSMDP) [9], which can be efficiently solved using standard heuristic search techniques developed for fully-observable settings. In the settings we study with intermittent state observability, our approach quickly produces high-quality policies on problems that are not solvable by exact POMDP algorithms with a generous time allocation. Furthermore, even well-known approximate POMDP solvers are not competitive with our approach as they often produce worse solutions and require significantly more runtime. Finally, we prove that increasing the depth limit can never decrease performance, and provide a simple test to determine if a given depth yields the optimal SOMDP policy.

Our paper is structured as follows: in Section III we introduce the SOMDP; in Section IV we introduce our methodology for solving SOMDPs based on memory states; in Section V we introduce our domain-independent heuristic for efficiently solving MSMDPs; and in Sections VI and VII we present and discuss our experimental results.

II. Preliminaries

A *Markov decision process* (MDP) is represented by the tuple $\langle S, A, T, R\rangle$ where S is a finite set of states, A is a finite set of actions, $T : S \times A \times S \rightarrow [0, 1]$ is a transition function representing the probability of arriving in state s' having taken the action a in state s, and $R : S \times A \rightarrow \mathbb{R}$ is a reward function representing the immediate expected reward

Supported by the National Science Foundation grant IIS-1954782 and the Alliance Innovation Lab Silicon Valley.

[1]University of Massachusetts Amherst, Amherst, Massachusetts, {cbasich, jrpeterson, shlomo}@cs.umass.edu

978-1-6654-7927-1/22/$31.00 ©2022 IEEE

of taking action a in state s. A solution to an MDP is a *policy*, denoted $\pi : S \rightarrow A$, which maps states to actions. A policy π induces the state-value function $V^\pi : S \rightarrow \mathbb{R}$, defined as $V^\pi(s) = R(s, \pi(s)) + \sum_{s' \in S} T(s, \pi(s), s')V^\pi(s')$, which represents the expected cumulative reward when starting in the state s and following the policy π. Similarly, a policy π induces the action-value function $q^\pi : S \times A \rightarrow \mathbb{R}$, defined as $q^\pi(s, a) = R(s, a) + \sum_{s' \in S} T(s, a, s')V^\pi(s')$, representing the expected cumulative reward when starting in state s, taking action a, and then following policy π.

A policy which maximizes these functions is called an *optimal policy*. Without loss of generality, we assume that there is a unique optimal policy, denoted π^*, unless stated otherwise. Given π^*, the optimal state-value function following π^* is defined as $V^*(s) = \max_{a \in A} q^*(s, a)$ where q^* is the action-value function under the policy π^*.

III. Semi-Observable Planning

In a traditional MDP, a system computes a policy π and acts according to the policy in the following way: it senses to observe its current state, queries its policy to determine the action to take in its current state, and then executes that action. However, an autonomous system may not be able to always observe its current state, for any of several reasons discussed above. In some cases, such as in a traditional partially observable Markov decision process or mixed-observability Markov decision process, the system may have *some* degree of state observability in the form of partial states, noisy state feature information, or both. However, in this paper we consider the special case in which the system has either *full observability* of its state, or it may *completely lose observability* of its state. In practice, various forms of sensor failures that produce uninterpretable information may effectively lead to the loss of observability [10].

Definition 1. *A* semi-observable Markov decision process *(SOMDP) is represented by the tuple* $\langle S, A, T, \eta, R\rangle$ *where*

- S *is a finite set of states,*
- A *is a finite set of actions,*
- $T : S \times A \times S \rightarrow [0, 1]$ *is a state transition function,*
- $\eta : A \times S \rightarrow [0, 1]$ *is an* observability profile *that represents the likelihood of state observability after performing action* a *and transitioning to state* s'*, and*
- $R : S \times A \rightarrow \mathbb{R}$ *is a reward function.*

As with an MDP, the solution of a SOMDP is a mapping $\pi : S \rightarrow A$ called a *policy*, which similarly induces both a state- and action-value function. The objective remains to find a policy maximizing these functions, called an *optimal policy*. However, unlike an MDP, in a SOMDP the system does not necessarily observe its state at each time step. We first observe that this problem could naively be modeled as a POMDP where the observation set is simply $S \cup \{\emptyset\}$, where $\emptyset$ is the null observation.

Proposition 1. *Every SOMDP is a POMDP.*

Unfortunately, it is known that solving POMDPs exactly, and often even approximately, is intractable [8]. This computational burden is driven by the need to maintain a belief, or distribution over state likelihood, conditioned on the last belief state, the previous action, and the most recent observation. On the other hand, MDPs can be solved efficiently, and often optimally, even for fairly large domains, which makes them an attractive model to use in sequential decision making with full observability.

IV. Solving SOMDPs

We propose an approach to solving SOMDPs that exploits their unique problem structure to remove the need to maintain a belief state at each step, allowing us to employ efficient fully-observable solution methods that return high-quality approximate results which converge to the true optimal policy. To do this, the question we must address is, how should the system behave in an unobserved state, given that we will not track observations to maintain a belief over the whole statespace? The naive solution would be to simply query an oracular supervisory sensor [11] or a human supervisor [12] immediately upon losing state observability to reveal the state to the system; we can indeed improve upon such an approach. To do so, we turn to memory states.

A. Memory States

Our definition of a *memory state* is based on Hansen *et al.*'s work on mixed open-loop/closed-loop control [9] where, at each step, the agent either performs a sensing action that reveals perfect state information or performs a control action which provides no state information. In this context, a memory state represents uncertain knowledge about the environment given a sequence of open-loop control actions by the agent. Specifically, memory states capture the last fully observed state that the agent was in and the control action(s) it took since then, which is all the relevant information needed to infer the state of the world in a Markovian environment. Translating a memory state to a *belief state* (i.e., distribution of possible world states) can be done using Bayesian updating as in POMDPs [13]. In fact, the translation can be viewed as a special case of belief computation in a POMDP starting in a collapsed belief state, with only a null observation after each action.

Definition 2. *Let* S *be a set of states,* A *be a set of actions, and let* $\mathcal{F}$ *be the forest created as follows: set each unique state* $s \in S$ *to be the root of a unique tree and let each branch corresponds to an action* $a \in A$*. A* ***memory state*** *is any positive-length connected path in* $\mathcal{F}$ *rooted at the root node of a tree, of the form* $sa_1...a_k$*.*

If we bound the maximal depth of each tree in $\mathcal{F}$ by some finite constant value $\delta \in \mathbb{Z}^+$, we will ensure that the number of memory states is finite. We denote by $\mathcal{F}_\delta(S, A)$ the set of memory states for the state and action sets S and A with finite depth δ, and $\mathcal{F}(S, A) = \mathcal{F}_\infty(S, A)$.

B. Memory-State MDP

In this section we discuss how we can use memory states to allow us to solve a SOMDP without the need to maintain a belief state and observation set. First, we provide intuition. Recall that when the agent performs action a in state s and transitions to state s', s' is either observed by the agent or it is fully unobserved. If it is fully observed, nothing needs to be done; if it is fully unobserved, we can simply transition the agent to the equivalent memory state, sa. Hence, similar to how a POMDP can be viewed as a belief-state MDP, a SOMDP can be represented by a memory-state MDP.

Definition 3. *Let $\mathcal{M} = \langle S, A, T, \eta, R\rangle$ be a SOMDP. We represent the corresponding* ***memory-state MDP*** *(MSMDP), $\overline{\mathcal{M}}$, by the tuple $\langle \overline{S}, A, \overline{T}, \overline{R}\rangle$ where:*

- *$\overline{S} = S \cup \mathcal{F}(S, A)$ is a set of states,*
- *$\overline{A} = A \cup \{\texttt{Reveal}\}$ is a set of actions,*
- *$\overline{T} : \overline{S} \times A \times \overline{S} \to [0, 1]$ is a state transition function,*
- *and $\overline{R} : \overline{S} \times A \to \mathbb{R}$ is a reward function.*

Similar to a belief MDP, we can compute the *belief state* exactly given a memory state, i.e. the agent's last observed state, $s \in S$, and its action history since then, $a_1...a_k$ where each $a_i \in A$. Let $\overline{s} = sa_1...a_k \in \overline{S}$ be a memory state with $k \geq 1$, then the belief of state $s' \in S$ given $\overline{s}$, denoted $b(s'|\overline{s})$, is defined as $b(s'|\overline{s}) = \sum_{s'' \in S} b(s''|sa_1...a_{k-1})T(s'', a_k, s')$. If $\overline{s} \in S$, the belief is either 1.0 or 0.0. Observe that the probability of entering a memory state is exactly determined by the observability profile, η in $\mathcal{M}$.

Additionally, similar to the action Sense in [9], we require the inclusion of an action Reveal in A that deterministically grants the agent observability of its current state (at possibly high cost). While it may not be the case that, in certain memory states, the value of information is greater than the expected value of acting in open-loop, this addition is also practically useful in that it allows us to limit the total number of memory states in our state space to some finite value. Specifically, given a MSMDP $\overline{\mathcal{M}}$ and a positive integer $\delta \in \mathbb{Z}^+$, we denote the *depth-limited* MSMDP by $\overline{\mathcal{M}}_\delta$, where we require that the agent performs action Reveal whenever it is in a depth-δ memory state, restricting the maximal length of any open-loop control sequence to at most δ steps.

In general, the *depth-limited* MSMDP will be an approximate model for the true SOMDP. However, we can show that given two MSMDPs for the same SOMDP but with different maximal tree depths δ and δ', an optimal policy for the MSMDP with greater depth will be at least as good as the other when evaluated as a (possibly suboptimal) policy for the SOMDP. In other words, expected performance monotonically increases with the maximum allowed length of open-loop control.

Proposition 2. *Let $V^*_\delta : S \cup \mathcal{F}_\delta(S, A) \to \mathbb{R}$ be the optimal value function for $\overline{\mathcal{M}}_\delta$. For any $\delta' > \delta$, and any $s \in S \cup (\mathcal{F}_\delta(S, A) \cap \mathcal{F}_{\delta'}(S, A))$, $V^*_{\delta'}(s) \geq V^*_\delta(s)$.*

Proof. First, let π^*_δ be an optimal policy for $\overline{\mathcal{M}}_\delta$. Observe that $\mathcal{F}_\delta(S, A) \subset \mathcal{F}_{\delta'}(S, A)$, and as such, it is clearly the case that we can construct a policy $\pi_{\delta'}$ for $\overline{\mathcal{M}}_{\delta'}$ where for every $s \in S \cup \mathcal{F}_\delta(S, A)$, $\pi_{\delta'}(s) = \pi^*_\delta(s)$. Furthermore, observe that under this policy, no memory state $s \in \mathcal{F}_{\delta'}(S, A) \backslash \mathcal{F}_\delta(S, A)$ is reachable, as by definition, for *any* memory state $s \in \mathcal{F}_\delta(S, A)$ of depth δ, $\pi^*_\delta(s) = \texttt{Reveal}$, and by construction $\pi_{\delta'}(s) = \texttt{Reveal}$. Hence, the only reachable states in $\overline{\mathcal{M}}_{\delta'}$ under $\pi_{\delta'}$ (memory or otherwise) are fully contained in the state space of $\overline{\mathcal{M}}_\delta$, so $V^\pi_{\delta'}(s) = V^*_\delta(s)$ for every state $s \in S$. Hence $V^*_{\delta'}(s) \geq V^\pi_{\delta'}(s) = V^*_\delta(s)$ for every state $s \in S \cup (\mathcal{F}_\delta(S, A) \cap \mathcal{F}_{\delta'}(S, A))$. □

C. Optimality of MSMDP Mapping

Proposition 2 suggests that increasing the maximal depth, δ, will cause the optimal value function of the depth-limited MSMDP to tend towards the optimal value of the infinite-depth MSMDP in the limit, which is equivalent to the optimal value of the unconstrained SOMDP itself. However, due to practical concerns, we are particularly interested in knowing when a SOMDP admits an optimal *finite* depth-limited MSMDP; i.e., when there exists a depth-limited MSMDP for some finite depth that returns that true optimal solution to the original SOMDP. In the case of a finite-horizon problem, the answer will always be yes.

Theorem 1. *Any finite-horizon SOMDP, for horizon $H \in \mathbb{N}$, admits an optimal finite depth-limited memory state MDP.*

Proof. This is trivial to observe by setting $\delta = H$. □

However, it is straightforward to observe that the above claim does not hold for an arbitrary infinite-horizon SOMDP. Consider the counterexample where there is a single state s and a single action a in addition to Reveal that deterministically self-loops and never has observability; if $R(s, a) > R(s, \texttt{Reveal})$, any finite depth limit will result in a suboptimal policy for the SOMDP. However, in practice it is likely the case that there *is* a finite depth, δ^*, for which the MSMDP does admit an optimal policy due to the risk and uncertainty associated with open-loop behavior in the real world. Therefore, when this is the case, we would like to know if we can identify if we have reached such a depth, as it is unknown *a priori*.

Definition 4. *Let $\mathcal{M}$ be a SOMDP that admits an optimal depth-limited MSMDP for (unknown) finite depth $\delta^* \in \mathbb{N}$. Assume that when optimal policies for each depth-limited MSMDP are not unique, selected policies always break action ties in favor of* Reveal, *break all other action ties in favor of the lower indexed action, and that the action* Reveal *has fixed reward $\rho \in \mathbb{R}$. The* Optimal-Depth Test *for depth δ is the following:*

1) Solve $\overline{\mathcal{M}}_\delta$ and $\overline{\mathcal{M}}_{\delta+1}$ for opt. policies π^*_δ and $\pi^*_{\delta+1}$,
2) if $\pi^*_\delta(\overline{s}) = \pi^*_{\delta+1}(\overline{s})$ for every $\overline{s} \in S_\delta$ return TRUE,
3) if $\pi^*_\delta(\overline{s}) \neq \pi^*_{\delta+1}(\overline{s})$ for any $\overline{s} \in S_\delta$ return FALSE.

Here, $\overline{S}_\delta$ denotes the statespace for $\overline{\mathcal{M}}_\delta$. To show correctness, we first need the following lemma, which intuitively states that if we reach a depth δ, where increasing δ by one does not change the optimal policy, then increasing the depth further will continue to not change the optimal policy. Note that in the following, optimal MSMDP policies are selected from the associated set of optimal policies via the tie-breaking strategy in Definition 4.

Lemma 1. *If $\pi^*_\delta(\overline{s}) = \pi^*_{\delta+1}(\overline{s})$ for all states $\overline{s} \in \overline{S}_\delta$, then for every $\delta' > \delta$, $\pi^*_\delta(\overline{s}) = \pi^*_{\delta'}(\overline{s})$ for all states $\overline{s} \in \overline{S}_\delta$.*

Proof. First, we denote by S_δ the set of all states in $\overline{\mathcal{M}}_\delta$. Now, suppose for contradiction that the claim does not hold, i.e. $\pi^*_\delta = \pi^*_{\delta+1} \neq \pi^*_{\delta+2}$ for all $\overline{s} \in S_\delta$ where π^* here denotes the optimal policy selected according to our tie-braking strategy. Then, there exists some state $\overline{s} \in S_\delta$ such that $\pi^*_\delta(\overline{s}) \neq \pi^*_{\delta+2}(\overline{s})$. First, observe that by construction of MSMDPs, no memory state of depth $\delta + 1$ is reachable in $\overline{\mathcal{M}}_{\delta+1}$ under $\pi^*_{\delta+1}$, since in π^*_δ all depth δ states must have the action `Reveal` assigned.

Hence there exists a depth δ memory state, $\overline{s}$, such that $\pi^*_{\delta+1}(\overline{s}) \neq \pi^*_{\delta+2}(\overline{s})$, since, if that were not the case, all reachable states would be identical under each policy, and hence it cannot be the case that the policy differs for another state in S_δ given the tie-braking of optimal policy strategy assumption. Let $a_1 = \pi^*_\delta(\overline{s}) = \pi^*_{\delta+1}(\overline{s})$ and $a_2 = \pi^*_{\delta+2}(\overline{s})$. For similar logic as above, it must be the case that $a_1 =$ `Reveal`, and hence that $a_2 \neq$ `Reveal`.

By assumption of contradiction, we know that

$$q^*_\delta(\overline{s}, \texttt{Reveal}) > q^*_\delta(\overline{s}, a_2) \text{ and}$$

$$q^*_{\delta+2}(\overline{s}, \texttt{Reveal}) < q^*_{\delta+2}(\overline{s}, a_2).$$

Observe that the lack of equality condition here is due to the tie-braking strategy assumption; i.e. if the values were in fact equal, the other action would have been taken. This means that $q^*_\delta(\overline{s}, \texttt{Reveal}) = \sum_{s \in S} b(s|\overline{s}) V^*_\delta(s)$ and although the math is omitted due to space, we get that

$$\sum_{s \in S} b(s|\overline{s}) \Big[V^*_\delta(s) - V^*_{\delta+2}(s) \Big]$$
$$> \sum_{s \in S} b(s|\overline{s}) \sum_{s' \in S} T(s, a_2, s') \Big[V^*_\delta(s') - V^*_{\delta+2}(s') \Big].$$

Additionally, by Proposition 2 we know:

$$(1) \sum_{s \in S} b(s|\overline{s}) \Big[V^*_\delta(s) - V^*_{\delta+2}(s) \Big]$$
$$\leq \sum_{s \in S} b(s|\overline{s}) \Big[V^*_{\delta+2}(s) - V^*_{\delta+2}(s) \Big] = 0$$

$$(2) \sum_{s \in S} b(s|\overline{s}) \sum_{s' \in S} T(s, a_2, s') \Big[V^*_\delta(s') - V^*_{\delta+2}(s') \Big]$$
$$\geq \sum_{s \in S} b(s|\overline{s}) \sum_{s' \in S} T(s, a_2, s') \Big[V^*_\delta(s') - V^*_\delta(s') \Big] = 0$$

Hence, $0 < 0$, which is a contradiction. □

Theorem 2. *The* Optimal-Depth Test *returns TRUE if and only if $\delta \geq \delta^*$.*

Proof. Let δ^* be the smallest optimal finite depth; it follows immediately from Proposition 2, the definition of optimality, and our optimal policy tie-breaking strategy, that for every $\delta > \delta^*$, $\pi^*_\delta = \pi^*_{\delta^*}$, and hence if $\delta > \delta^*$, the algorithm will return TRUE.

By Lemma 1, given δ, if $\pi^*_\delta = \pi^*_{\delta+1}$, then $\pi^*_\delta = \pi^*_{\delta+2}$. Hence, if our algorithm returns TRUE, it must be the case that for every $\delta' > \delta$, by simple induction, $\pi^*_\delta = \pi^*_{\delta'}$; hence δ must be at least as big as δ^* or we will have contradicted the assumption that δ^* is the smallest optimal finite depth. □

V. Efficient Planning

While memory states allow us to model problems with state unobservability without requiring complex belief-updates, the size of the forest that defines the set of memory states, $\mathcal{F}_\delta(S, A)$, is $|S||A|^\delta$, and hence the complexity of solving the model grows as we increase the depth limit. Hansen et al. [9] handle this in the context of Q-learning by pruning branches in $\mathcal{F}_\delta(S, A)$ during exploration where the value of information in that memory state is greater than or equal to its cost. In our case, we are performing optimal model based planning, and can apply standard algorithms for MDPs to depth-limited MSMDPs. For many problems, as with MDPs, heuristic search algorithms such as LAO* [14] can, with an admissible heuristic,efficiently compute optimal policies without evaluating the entire state space.

We introduce a domain-independent heuristic, h_{V^*}, which we prove is admissible for all MSMDPs with non-positive rewards, allowing us to efficiently solve a depth-limited MSMDP using LAO* for its optimal policy. The heuristic is based on the optimal value function of the "ideal" version of the SOMDP where the agent has observability everywhere with probability 1.0 (or, equivalently, where the action `Reveal` has no cost). As this is an always-optimistic version of the problem, solving it provides a (fairly tight) lower bound on the value of any state in the MSMDP, giving us our admissible heuristic (proved below); evidence of the empirical benefit is provided in Section VII. We formally define the heuristic below.

Definition 5. *Let $\mathcal{M} = \langle S, A, T, \eta, R \rangle$ be a SOMDP where all rewards are non-positive, $\mathcal{M}^*$ be $\mathcal{M}$ where $\eta[A \times S] = \{1.0\}$, and $\overline{\mathcal{M}} = \langle \overline{S}, \overline{A}, \overline{T}, \overline{R} \rangle$ be the MSMDP on $\mathcal{M}$. Given the optimal value function for $\mathcal{M}^*$, $V^* : S \to \mathbb{R}$, we define the heuristic function $h_{V^*} : \overline{S} \to \mathbb{R}$, as follows:*

$$h_{V^*}(\overline{s}) = \begin{cases} V^*(\overline{s}) & \textit{if } \overline{s} \in S \\ \sum_s b(s|\overline{s}) V^*(s) & \textit{otherwise} \end{cases}$$

Ultimately, We would like to prove that h_{V^*} is an admissible heuristic for the MSMDP with finite δ. Throughout the rest of the section, we will be assuming that all rewards are negative, and may interchangeably utilize the terminology "cost" in the context of cost-minimization, understanding that every non-positive reward maximizing MDP can be converted into an equivalent non-negative cost minimizing MDP. This would enable us to guarantee, for a certain class of heuristic search planning algorithms, that the algorithm will converge to the optimal solution when using h_{V^*} as its heuristic. To prove this, we need a few preliminary results.

First, we must show that when the action `Reveal` has zero cost in memory states (i.e. it is a "free" action) the q-value of `Reveal` will be at least as large as the q-value of any other available action. Intuitively, this means that Reveal can be assumed to be taken immediately upon entering a memory-state or, equivalently, immediately upon entering an unobservable state. Formally:

Lemma 2. *Let $\overline{\mathcal{M}}$ be a non-positive reward MSMDP where*

$$\overline{R}(\overline{s}, \texttt{Reveal}) = \begin{cases} 0 & \textit{if } \overline{s} \in \mathcal{F}_\delta(S, A) \\ -\infty & \textit{otherwise} \end{cases}$$

Then, given a policy π, $q^\pi(\overline{s}, a) \leq q^\pi(\overline{s}, \texttt{Reveal})$ for any $a \neq \texttt{Reveal}$ when $\overline{s} \in \mathcal{F}_\delta(S, A)$, where $q^\pi(s, a)$ denotes the q-value for taking action a in state s and following the policy π in all future states.

Proof. Let $\overline{s} \in \overline{S}$ be a memory state. Then $q^\pi(\overline{s}, a) - q^\pi(\overline{s}, \texttt{Reveal})$

$$= \overline{R}(\overline{s}, a) + \sum_{s \in S} b(s|\overline{s}) \sum_{\overline{s}' \in \overline{S}} \overline{T}(s, a, \overline{s}') V^\pi(\overline{s}') - \sum_{s \in S} b(s|\overline{S}) V^\pi(s)$$
$$= \sum_{s \in S} b(s|\overline{s}) \Big[R(s, a) + q^\pi(s, a) - R(s, a) - V^\pi(s)]$$
$$= \sum_{s \in S} b(s|\overline{s}) \Big[q^\pi(s, a) - \max_{a^* \in \overline{A}} q^\pi(s, a^*) \Big] \leq 0$$

□

Next, we observe that when there is no probability of state unobservability, given that the agent starts in a reliable state, the agent will achieve its best performance in expectation.

Proposition 3. *Let $\mathcal{M} = \langle S, A, T, \eta, R \rangle$ be a non-positive reward SOMDP with corresponding MSMDP $\overline{\mathcal{M}} = \langle \overline{S}, \overline{A}, \overline{T}, \overline{R} \rangle$. The optimal value function for $\overline{\mathcal{M}}$ defined as*

$$\overline{V}^*(\overline{s}) = \overline{R}(\overline{s}, \overline{\pi}^*(\overline{s})) + \sum_{\overline{s}' \in \overline{S}} \overline{T}(\overline{s}, \overline{\pi}^*(\overline{s}), \overline{s}') \overline{V}^*(\overline{s}')$$

is maximized when $\eta[A \times S] = \{1.0\}$.

Proof. It is straightforward to see that the behavior of the agent (e.g. its action trace) when $\eta[A \times S] = \{1.0\}$ will be the same as when $\eta[A \times S] \subset [0, 1]$ and

$$\overline{R}(\overline{s}, \texttt{Reveal}) = \begin{cases} 0 & \text{if } \overline{s} \in \mathcal{F}_\delta(S, A) \\ -\infty & \text{otherwise} \end{cases}$$

up to the execution of the action Reveal. Upon entering a memory state, $\overline{s}$, by Lemma 2, Reveal will have the highest q-value (up to ties which we may assume by construction are broken in favor of Reveal), and hence the agent will always immediately execute the action Reveal to observe its state before acting in any optimal policy, which can be viewed as an addition to the original action. This is the same as never *needing* to execute Reveal as the reward of the action is 0 in a memory state and there is no discounting. Hence, any negative adjustment to the reward of Reveal will decrease the expected cumulative reward of a memory state, leading to the same or lower value for every state in the domain under the optimal policy (as all rewards are non-positive). Hence, if $\eta[A \times S] \subset [0, 1]$ and $\overline{R}(\overline{s}, \texttt{Reveal}) < 0$, $\overline{V}^*(s)$ will be the same or less as when $\eta[A \times S] = \{1.0\}$. □

Theorem 3. *$h_{V^*} : \overline{S} \to \mathbb{R}$ is an admissible heuristic for $\overline{\mathcal{M}}_\delta$ where $\delta \geq 1$ and $\overline{\mathcal{M}}$ is a non-positive reward MSMDP.*

Proof. Let $\overline{s} \in \overline{S}$ and suppose $\overline{\pi}^*(s) = a$. First, assume $\overline{s} \in S$ (note that $a \neq \texttt{Reveal}$ in such a case). Then:

$$\begin{aligned} \overline{V}^*(\overline{s}) &= \overline{R}(\overline{s}, a) + \sum_{\overline{s}' \in \overline{S}} \overline{T}(\overline{s}, a, \overline{s}') \overline{V}^*(\overline{s}') \\ &\leq R(\overline{s}, a) + \sum_{s' \in S} T(\overline{s}, a, s') \overline{V}^*(s') \\ &\leq R(\overline{s}, a) + \sum_{s' \in S} T(\overline{s}, a, s') V^*(s') \text{ by Prop. 3} \\ &= V^*(\overline{s}) = h_{V^*}(\overline{s}). \end{aligned}$$

Second, assume $\overline{s} \notin S$, and $a = \texttt{Reveal}$. Then:

$$\begin{aligned} \overline{V}^*(\overline{s}) &= \overline{R}(\overline{s}, a) + \sum_{\overline{s}' \in \overline{S}} \overline{T}(\overline{s}, a, \overline{s}') \overline{V}^*(\overline{s}') \\ &\leq \sum_{\overline{s}' \in \overline{S}} \overline{T}(\overline{s}, a, \overline{s}') \overline{V}^*(\overline{s}') \\ &= \sum_{s \in S} b(s|\overline{s}) \overline{V}^*(s) \\ &\leq \sum_{s \in S} b(s|\overline{s}) V^*(s) \text{ by Prop. 3} \\ &= h_{V^*}(\overline{s}). \end{aligned}$$

Finally, assume $\overline{s} \notin S$ and that $a \neq \texttt{Reveal}$. Then:

$$\begin{aligned} \overline{V}^*(\overline{s}) &= \overline{R}(\overline{s}, a) + \sum_{\overline{s}' \in \overline{S}} \overline{T}(\overline{s}, a, \overline{s}') \overline{V}^*(\overline{s}') \\ &= \sum_{s \in S} b(s|\overline{s}) \Big[R(s, a) + \sum_{s' \in S} T(s, a, s') \overline{V}^*(s') \Big] \\ &\leq \sum_{s \in S} b(s|\overline{s}) \Big[R(s, a) + \sum_{s' \in S} T(s, a, s') V^*(s') \Big] \text{ by Prop. 3} \\ &= \sum_{s \in S} b(s|\overline{s}) V^*(s) = h_{V^*}(s). \end{aligned}$$

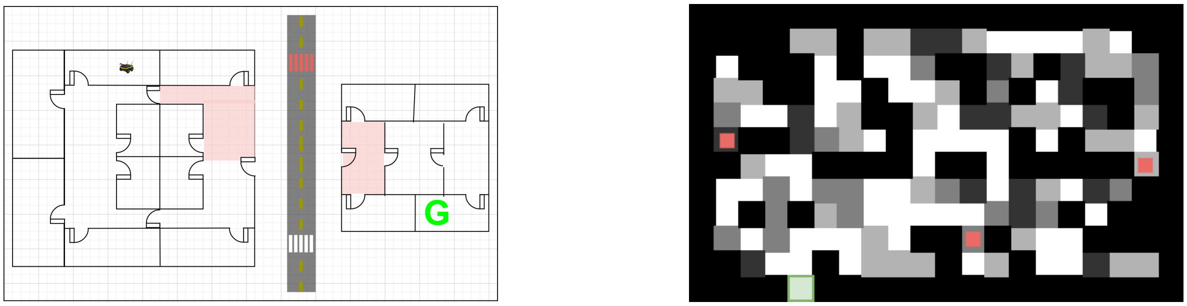
Hence h_{V^*} is admissible. □

VI. Experimental Domains

A. Campus Robot

In this domain, a robot equipped only with a camera operates in a known map and needs to deliver a package from one office to another in a campus environment. The robot must safely navigate the environment, which includes closed doors and crosswalks across a main road. States $s \in S$ are represented by the tuple $\langle x, y, \theta, o \rangle$ where: x, y, and θ is the robot's pose and o is the type of obstacle (or none) at the robot's current position. The robot can perform the following actions: Move (in direction), which has a 20% probability of failing (i.e. the robot stays in the same location, Open, which does nothing if the robot is not at a door, but otherwise deterministically opens it, Wait, which does nothing unless the robot is at a crosswalk, in which case there is a chance of the traffic condition changing, and Cross, which does nothing unless the robot is at a crosswalk, where the robot deterministically crosses if there is no traffic, crosses successfully with 50% probability with light traffic, and if there is heavy traffic crosses with 10% probability and crashes with 10% probability leading to a dead end (and otherwise does not move).

Unit negative reward is incurred on each time step when the robot is moving and does not have a collision. If the



(a) An illustration of the Campus Robot domain. The agent, represented by the Jackal image, must navigate from its location to the goal state, represented by the green ‘G’ while managing obstacles, perception failures, and its supervisory sensor. Red denotes areas with low likelihood of observability due to environmental factors.

(b) An illustration of the Disaster Relief domain. Black represents walls or debris; white represents no smoke; each gradation of grey represents a level of smoke increasing with the darkness of the gradation. Small red squares represent the location of people who are trapped; the green represents the entryway into the area.

Fig. 1: Empirical Domains

robot collides with a wall or a closed door, or attempts to cross the road outside of a crosswalk, a negative reward of -5 is incurred. The primary impact on the cumulative reward across a trajectory is the number of time steps taken for the robot to successfully deliver its package.

In the semi-observable setting, both for the SOMDP and relevant POMDP baseline, the probability of observability, η, is dependent on the location of the state the robot enters; red areas in the map (Fig. 1a) have likelihood of 0.1, and 0.9 elsewhere. For instance, in certain areas the likelihood of observability is very low due to glare or darkness. At any time, the robot can perform the action `Reveal` to request aid from a remote supervising human at a fixed negative reward (-3) to deterministically reveal its state.

B. Disaster Relief

In this domain, a robotic drone must deliver relief packages containing respirators to people in known locations in a burning building. The objective of the agent is to bring gas masks to each person who is trapped so they do not inhale deadly gas, smoke, and debris while waiting for rescue. States $s \in S$ are represented by the tuple $\langle x, y, \theta, l, \mathcal{P} \rangle$ where: x, y, and θ is the robot's pose, $l \in \mathbb{N}$ is the smoke level in the agent's current location, and $\mathcal{P}$ is a vector of integers, where each element is the smoke level at the location of one of the humans. The robot can perform the following actions: `Move` (in direction) which has a 20% probability of failing (i.e. the robot stays in the same location), `Aid` which deterministically gives a relief package to a human when the robot *observes* a human in its location, and is prohibited otherwise. When the person receives the package, the effective smoke level for that person is 0. Additionally, if the agent collides with a wall, with probability 0.2 it drops its aid packages and must return to the start location outside the building to get a new set of relief packages. Note that this is not a dead end, simply a high-cost negative outcome. At each time step, a negative reward is incurred equal to the sum of the smoke level across un-aided human locations divided by 10. If the agent drops its aid packages, they incur a large negative reward equal to 10 times a single step reward.

In the semi-observable setting, the probability of observability, η, is dependent solely on the smoke level of the entered state, and is equal to $1 - (0.3 \cdot l)$, where l is the smoke level. As an example, if the agent moves into a location with a smoke level of 3, the maximum, it observes its new state with probability 0.1 and fails to observe the state with probability 0.9. In the SOMDP, losing observability results in transitioning to the corresponding memory state. In the POMDP baseline, losing observability is equivalent to receiving the null observation on that time step. The robot can naturally regain observability simply by moving around into new locations. However, the robot also has the option to perform the action `Reveal` during which it stops moving and takes extra time to localize its position in the world. This action deterministically reveals the agent's state and incurs negative reward equivalent to two time steps of smoke exposure for the un-aided humans.

VII. Empirical Evaluations

In this section, we discuss evaluations of our two primary contributions: (1) the SOMDP model, and (2) the h_{V^*} heuristic. To evaluate the first contribution, we implemented our SOMDP model, with an approximate MSMDP model on both domains, with four different depth limits, $\delta = 1, 2, 3$ and 4. To solve the MSMDPs, we implemented the algorithm LAO* [14], which is a well known heuristic search algorithm which converges to an optimal solution under an admissible heuristic. The LAO* results reported in Table I are computed using our heuristic, h_{V^*}, and hence are optimal for the respective depths. We compared our approach against POMDP implementations of the same domains. For the POMDPs, we considered three offline POMDP solvers and two online POMDP solvers from the open-source JULIA library POMDP.jl [15]: (1) QMDP [6], an approximate solution technique which assumes that belief collapses after each step; (2) SARSOP [16], (3) Point Based Value Iteration (PBVI), (4) Partially Observable Monte-Carlo Planning (POMCP) [17], and (5) ARDESPOT [18].

Results comparing the performance between the approximate depth-limited MSMDP model for each depth and the POMDP methods are reported in Table I for both domains.

		Campus Robot		Disaster Relief	
Model	Algorithm	Time (s)	Reward	Time (s)	Reward
UB		—	-32.64	—	-18.50
SOMDP	LAO^*_1	0.45	-69.26 ± 9.23	0.37	-30.36 ± 2.67
	LAO^*_2	0.44	-50.40 ± 8.25	0.46	-26.42 ± 4.76
	LAO^*_3	2.44	-46.30 ± 7.36	1.77	-24.35 ± 3.66
	LAO^*_4	18.32	-43.30 ± 7.43	10.79	-23.77 ± 2.67
POMDP	QMDP[1]	3.41	-50.53 ± 15.05	22.63	-35.13 ± 14.92
	SARSOP[1]	—	—	264.28	-23.41 ± 4.50
	PBVI[1]	—	—	—	—
	POMCP[2]	—	—	—	—
	ARDESPOT[2]	—	—	—	—

TABLE I: Performance comparison of solution methods on both domains. Time depicts runtime in seconds. Reward depicts the mean and standard deviation over 100 trials. [1] denotes an offline algorithm and [2] denotes an online algorithm. Bars denote algorithms that did not finish within the time limit (3 hours) or for the online algorithms failed to find the goal within 1000 steps.

	Campus Robot			Disaster Relief		
δ	Likelihood	Residual	Impact	Likelihood	Residual	Impact
1	0.27	-36.09	52.11%	0.21	-16.33	53.79%
2	0.11	-12.40	24.6%	0.05	-2.84	10.75%
3	0.05	-5.12	11.1%	0.01	-0.77	3.16%
4	0.03	-3.09	7.13%	0.003	-0.15	0.64%

TABLE II: Effect of maximal depth limit on performance showing that by depth 4, less than 8% and 1% of (negative) reward is attributable to the finite depth limit in each domain respectively.

UB refers to the upper bound; this is computed by solving the problem for the best-case SOMDP where $\eta[A \times S] = \{1.0\}$; this is a loose upper bound that is generally unobtainable.

Policies computed under our approach outperformed those computed by QDMP for all values of δ on both domains, with the sole exception of $\delta = 1$ on the Campus Robot Domain, taking an order of magnitude less time for $\delta < 4$. Additionally, SARSOP required 1-2 orders of magnitude more time to compute a solution in Disaster Relief compared to our approach to achieve comparable performance to LAO^*_4 with twice the standard deviation. SARSOP failed to compute a solution in Campus Robot within 3 hours, and PBVI fails to converge within 3 hours in both domains. Finally, both online POMDP solvers, POMCP and ARDESPOT, performed significantly below all other methods considered, including ours with $\delta = 1$, failing to ever reach the goal. After considering sampled trajectories, we observed that the solvers prioritize myopic cost-minimizing behavior that fails to advance the agent towards its goal. For example, never leaving initial 0-level smoke states in Disaster Relief to avoid the possibility of crashing. We hypothesize that these solutions undervalue or ignore subtasks required for goal satisfaction (e.g. "aid" actions for people in Disaster Relief) because of the risk of additional incurred cost and total number of required steps before subtasks can be completed and incurred cost can be reduced. Overall, this shows that our approach quickly converges to high quality solutions in the intermittently-observable domain, while requiring significantly less time to do so than the comparable POMDP approaches.

		Campus Robot			Disaster Relief		
δ	Heuristic	$\lvert\bar{S}\rvert$	Time (s)	N.E.	$\lvert\bar{S}\rvert$	Time (s)	N.E.
1	h_0	8200	1.01	4676	24048	1.17	5676
	h_{V^*}		0.45	3791		0.37	3417
2	h_0	58425	0.96	24955	124248	1.37	23614
	h_{V^*}		0.44	11154		0.46	10849
3	h_0	410000	5.89	77641	625248	7.48	64459
	h_{V^*}		2.44	35539		1.77	22781
4	h_0	2871025	32.06	277658	3130248	13.9	147949
	h_{V^*}		18.32	162874		10.79	43653

TABLE III: Efficiency comparison of $\mathbf{LAO^*}$ with the null heuristic and the $\boldsymbol{h_{V^*}}$ heuristic for four maximum memory state depths on both experimental domains. N.E. denotes the number of nodes expanded by $\mathbf{LAO^*}$ to converge to the optimal policy.

Table II shows the convergent behavior of our model with respect to δ, and that we quickly approach near-optimality. We simulated each domain 1000 times for each depth and examined the sample likelihood of a state being max-depth. In both domains the likelihood diminishes by an order of magnitude or more from $\delta = 1$ to $\delta = 4$, approaching 0 in the Disaster Relief domain. *Residual* is the *maximum* possible residual reward lost by limiting open-loop control to δ steps, based on the average trace length over the 1000 trials. This also shrinks by an order of magnitude or more, and approaches zero in Disaster Relief, by $\delta = 4$. *Impact* is simply the ratio of Residual to the mean cost in Table I and shows the relative performance impact of the forced action `Reveal`. Overall this demonstrates that even though we do not pass the Optimal-Depth Test, guaranteeing a globally (with respect to δ) optimal solution, our approach results in excellent performance empirically without the optimal depth.

Table III compares the efficacy of the h_{V^*} heuristic (Definition 5) relative to the null heuristic (h_0), a standard domain-independent admissible heuristic baseline. To do so, we ran LAO^* with each admissible heuristic, guaranteeing an optimal solution in both cases, for the MSMDP with $\delta = 1, 2, 3$ and 4. We recorded both the runtime and the number of nodes expanded by LAO^*; LAO^* was about two to three times as efficient when using h_{V^*}, expanding roughly between half to one third of the nodes and taking about half or less the time to find the optimal solution as when it used h_0 in most cases. We emphasize that h_{V^*} is completely domain-independent, and had similarly effective performance in both domains.

VIII. Related Work

The notion of what we refer to as a *memory state* is related to the general problem of representing states in systems with imperfect state observability or where knowledge of the current state can be determined entirely from prior information. These representations are generally based on histories of some combination of actions, observations, and known states, depending on individual availability. Exact belief states in POMDPs are updated based on histories of sequences of actions and observations, and are sufficient statistics for planning in POMDPs [8]. Finite state controllers for POMDPs [19] compress belief states into a finite set of nodes in an automaton representing an approximate policy.

Active perception [20] is concerned with the problem of designing and managing perception systems that are themselves active dynamic systems that can be altered or can change their behavior online as a means of influencing the information received by the acting agent, and ultimately said agent's behavior [21]. In fact, it is readily observed that the question faced by [9] of "to sense or not to sense" is itself a form of active perception. Although this work is primarily focused on the question of handling failure cases of perception through decision making, rather than modulating perception itself, approaches in active perception are symbiotic with what we present here and present interesting directions for future research.

Finally, *introspective perception* is a recent, rich line of work that allows a robot to "know when it doesn't know" by modeling the uncertainty and quality of the *outputs* of its perception systems [4], [5]. Introspective perception therefore offers a means of learning the likelihood of losing state observability for situations of perception failure driven state unobservability. This work offers complementary planning capabilities that can work with introspective perception to improve the overall reliability of a robotic system deployed in the open world.

IX. Conclusion

In this paper, we propose a novel planning model for stochastic sequential decision making problems, called a *semi-observable Markov decision process* (SOMDP), to better handle problems where state observability is available only in an intermittent capacity. This phenomenon could be due to prohibitively expensive sensing, unavailable sensory information, or unreliable state updates from perception. We present a solution approach for SOMDPs based on the use of memory states, which enables us to use efficient solution methods for fully observable problems to solve for high-quality SOMDP policies. We provide several theoretical properties of the SOMDP and corresponding depth-limited MSMDPs, showing that expected performance increases monotonically with memory state depth and providing a test to determine if a given depth admits the true optimal policy.

To address the added model complexity due to the inclusion of memory states, we introduce the novel h_{V^*} heuristic based on a SOMDP in which the probability of observability is always 1. We prove admissibility of our heuristic, guaranteeing that optimal heuristic search algorithms such as LAO* converge to optimal policies under h_{V^*}. We further show empirically that heuristic estimates from h_{V^*} provide valuable information to improve search efficiency, with LAO* expanding roughly half the nodes with the h_{V^*} heuristic compared to with the null heuristic.

More importantly, we show that policies computed using our approach are at least competitive with, and often significantly outperform, the solutions to the corresponding POMDP formulation in one or both of runtime and solution quality. Our approach quickly converges to solutions which are comparable to the high-quality approximate solutions computed by SARSOP while requiring significantly less time to converge. Additionally, our approach with only a shallow depth of 2 outperformed approximate solutions from QMDP, a common fast approximate POMDP algorithm, and in all cases significantly outperformed two recent online POMDP solvers, POMCP and ARDESPOT.

Hence, our approach is able to effectively exploit the structure of the problems considered to efficiently compute high quality solutions where the exact POMDP solver failed to return a solution within a three-hour timeout window, and outperformed both approximate POMDP solvers in both runtime and solution quality.

References

[1] E. A. Feinberg and A. Shwartz, *Handbook of Markov decision processes: Methods and applications*. Springer, 2012, vol. 40.

[2] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2020.

[3] J. Dooley, "Mission concept for a Europa Lander," in *IEEE Aerospace Conference*, 2018.

[4] S. Rabiee and J. Biswas, "IVOA: Introspective vision for obstacle avoidance," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[5] S. Daftry, S. Zeng, J. A. Bagnell, and M. Hebert, "Introspective perception: Learning to predict failures in vision systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[6] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *International Conference on Machine Learning (ICML)*, 1995.

[7] H. Kurniawati, "Partially observable Markov decision processes (POMDPs) and robotics," *arXiv preprint arXiv:2107.07599*, 2021.

[8] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, 1998.

[9] E. A. Hansen, A. G. Barto, and S. Zilberstein, "Reinforcement learning for mixed open-loop and closed-loop control," in *Neural Information Processing Systems Conference (NIPS)*, 1996.

[10] A. Gaddam, T. Wilkin, M. Angelova, and J. Gaddam, "Detecting sensor faults, anomalies and outliers in the internet of things: A survey on the challenges and solutions," *Electronics*, 2020.

[11] S. Rabiee, C. Basich, K. Wray, S. Zilberstein, and J. Biswas, "Competence-aware path planning via introspective perception," *IEEE Robotics and Automation Letters*, 2022.

[12] C. Basich, J. Svegliato, K. H. Wray, S. Witwicki, J. Biswas, and S. Zilberstein, "Learning to optimize autonomy in competence-aware systems," in *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2020.

[13] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based POMDP solvers," *Autonomous Agents and Multi-Agent Systems*, vol. 27, 2013.

[14] E. A. Hansen and S. Zilberstein, "LAO*: A heuristic search algorithm that finds solutions with loops," *Artificial Intelligence*, vol. 129, 2001.

[15] M. Egorov, Z. N. Sunberg, E. Balaban, T. A. Wheeler, J. K. Gupta, and M. J. Kochenderfer, "POMDPs.jl: A framework for sequential decision making under uncertainty," *Journal of Machine Learning Research*, 2017.

[16] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Robotics: Science and systems (RSS)*, 2008.

[17] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," *Advances in neural information processing systems*, vol. 23, 2010.

[18] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "DESPOT: Online POMDP planning with regularization," *Advances in neural information processing systems*, vol. 26, 2013.

[19] N. Meuleau, L. Peshkin, K.-E. Kim, and L. P. Kaelbling, "Learning finite-state controllers for partially observable environments," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 1999.

[20] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, 2018.

[21] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, 1988.