Planning with Inconsistent Sensors: Knowing When to Act Blind

Connor Basich* **John R. Peterson** **Shlomo Zilberstein**
University of Massachusetts Amherst
{cbasich, jrpeterson, shlomo}@cs.umass.edu

Abstract

As mobile robots are deployed in increasingly complex domains in the open world, the level of detail demanded by the robot's decision-making model to ensure reliable operation increases. To support this, mobile robotic agents are fixed with a wider array of more informative sensor equipment and downstream perception systems that convert such sensors' information into usable representations by the agent's planning models. Automated decision making models often assume free and consistent access to such information. However, in the context of a mobile robot, this assumption may not hold, and failing to account for this in the planning model may lead to costly behavior or even failure. In this paper, we propose a mixed open-loop/closed-loop planning model based on memory states that integrates knowledge about limitations on sensory feedback in order to proactively plan around these limitations or exploit situations where costly sensing is unnecessary. We provide both theoretical properties as well as empirical evaluations on a simulated mobile robot domain.

1 Introduction

As AI and robotics have advanced in recent years, attention has shifted from the deployment of mobile robotic agents in well-understood stationary domains to deployment in the complex and dynamic "open world". The increased complexity and uncertainty that the open world exhibits often demands a more finely detailed model of the domain to ensure reliable performance by the system. This includes ensuring system safety [Svegliato *et al.*, 2019], adhering to ethical constraints [Svegliato *et al.*, 2021], understanding and managing system competence [Basich *et al.*, 2020], and mitigating negative side effects [Saisubramanian *et al.*, 2020].

However, common to these approaches, and most work in automated decision making, are the following (often implicit) assumptions on the sensory feedback which provides the state information necessary for planning and control: (1) the feedback is *free* or has negligible cost compared to control; (2) the feedback is *available* on-demand; and (3) the feedback is *reliable*, by which we mean that it delivers an accurate (possibly incomplete) representation of the agent's environment. Unfortunately, in the open world, this assumption can not only be invalid, but can lead to costly erratic behavior and critical failures when violated [Rabiee and Biswas, 2019].

In general there are several reasons why these assumptions may not hold. Sensory feedback may be available but have a non-negligible cost that makes using it cost-ineffective; for example, an extraterrestrial science robot may have a finite and non-repletable battery supply which sensing actions consume [Dooley, 2018]. Alternatively, sensory feedback may simply be unavailable at various times during a system's deployment due to technical limitations or by design; for example, GPS may be unavailable while underground. Finally, sensory feedback may be available, and even free, but may provide unreliable information that is no better than no information, or worse, that is actively misleading; for example, in the presence of glare, an object in front of the robot may fail to be detected. Recent work in introspective perception has looked into using a secondary system to learn and monitor this behavior [Daftry *et al.*, 2016].

Consequently, while there are many facets to the problem of *reliable autonomy in the open world*, for autonomous decision making systems that depend on sensory feedback to update their internal state in order to act appropriately, it is necessary that they operate effectively even in the face of one of the aforementioned constraints. Such an agent should be able to operate reliably when faced with *unanticipated* limitations on sensory feedback, *proactively* plan around known or expected situations of limited sensory feedback, and *exploit* cases where sensory feedback is unnecessary.

To this end, we propose a general way to integrate perception reliability information derived from an *introspective perception system* into an existing planning model. We map states where perception is deemed unreliable to *memory states* in a separate augmented planning model, which have well-defined transition and cost dynamics analogous to belief states in a POMDP [Shani *et al.*, 2013]. This model enables the agent to perform open-loop planning when reliable perception is unavailable or is deemed cost ineffective, avoiding the need to automatically query an expensive supervisory sensor. By utilizing a mix of closed-loop and open-loop planing, the agent can better optimize its performance in the face of limited sensory feedback. To be able to handle the combi-

*Contact Author

natorial increase in model size induced by the inclusion of memory states, we introduce an admissible heuristic that significantly outperforms its naive baseline counterpart. We further prove that allowing for longer open-loop sequences can never worsen expected performance. Finally we provide empirical evaluations of a simulated agent's behavior.

2 Related Work

Most closely related to our work is that of Hansen *et al.* [1996], which introduces the notion of a *memory state*, a representation of the knowledge needed by the agent to infer its current state in its domain, and mixed open-loop and closed-loop control in the context of reinforcement learning. Our work generalizes that earlier work by considering a variety of sensor failure modes, sensor unavailability, or prohibitively expensive sensor information. Earlier work was designed for situations where sensing is available at a non-zero cost at every timestep, but when the agent takes a control action, it deterministically stays in a memory state. In this paper, we extend the approach to a wide variety of sensor limitations. For example, the agent may have access to free imperfect perception, but may alternatively take a costly action to query a supervisory sensor to transition from a memory state to a reliable state. Additionally, we allow for the possibility of naturally transitioning from a memory state to a fully observed state in the modeled domain without querying the supervisory sensor in the case where reliable perception naturally becomes available again after a period of sensor failure or unavailability.

Early work on anytime sensing [Zilberstein, 1996] demonstrated the ability to adapt sensing effort to the needs of the planning and execution architecture, so that less precise—and less costly—sensing is performed whenever it is sufficient for effective operation (e.g., navigating through uncluttered space). Unlike our proposed approach, the sensor reliability in this line of work is directly controlled by the time allocation to the anytime sensing process.

Active perception [Bajcsy *et al.*, 2018] is another related area of research that has received significant attention over the last several decades. Active perception 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 [Bajcsy, 1988]. In fact, it is readily observed that the question faced by Hansen *et al.* [1996] 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, we believe that approaches in active perception are symbiotic with what we present here and presents interesting directions for future research.

Addressing perception uncertainty and failures that can arise as a consequence has also been studied in recent years. Kaipa *et al.* [2016] investigate how perception uncertainty in robotic bin-picking can induce various failure modes which can propagate error through each stage of task execution, and propose an approach to characterize the perception uncertainty driven failures to determine if the robot should query for human intervention or to invoke one of a number of specific planners used for fine-motion strategies that improve accuracy at the cost of completion time.

Hanheide *et al.* [2017] propose an approach for enabling a robotic agent acting in a domain with uncertain sensing, uncertain actions, and incomplete information about its environment to methodically gather the information required to accomplish its task. However, their work primarily addresses an *a priori* lack of information and domain knowledge, and assumes constant and free access to an array of sensory feedback for various perceptual systems. In this sense, their approach focuses on building a coherent model of the agent's domain while avoiding failures, and ultimately accomplishing its task as efficiently as possible, and hence has more similarities to model based reinforcement learning and partially observable Markov decision processes.

More recently, Lee *et al.* [2020] have investigated the problem of a robot that is given a model-based policy, but can detect when there is large uncertainty in its perception and actuation systems indicating that its given policy may be unreliable and should not be applied. To address this, the authors use a policy-optimization approach to learn a local policy on raw sensory inputs in areas of large uncertainty in place of the model-based policy. As their approach fundamentally relies on consistent sensory information, it cannot be directly applied to the specific problem studied in this paper.

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 [Daftry *et al.*, 2016; Rabiee and Biswas, 2019]. Hence, this work offers complementary planning capabilities that can work with introspective perception.

3 Background

The primary objective of this work is to capture various forms of incompleteness or inconsistency in sensory feedback, either intended or unintended, and integrate them into a primary planning model in order to proactively handle these phenomena. We use a fully observable planning model, called a *stochastic shortest path problem*, to represent the base domain for several reasons. First, we explicitly assume that all information received from perception that is deemed reliable by an introspective perception system is fully observable; that is, it completely reveals the state to the agent. Second, any information received that is deemed unreliable is equivalent to a null observation in that it provides no state information to the agent. Finally, partially observable models which can capture similar issues of perception uncertainty, and are strict generalizations of the model we use, are significantly less tractable to solve than their fully observable counterparts. And, as they do not exploit problem specific information and assumptions, their use does not provide additional utility in the problem considered in this work.

A *stochastic shortest path problem* (SSP) is represented by the tuple $\langle S, A, T, C, s_0, s_g \rangle$ where S is a finite set of states, A is a finite set of actions, $T : S \times A \times S \to [0, 1]$ represents the probability of reaching state $s' \in S$ after performing action

$a \in A$ in state $s \in S$, $C : S \times A \to \mathbb{R}$ represents the immediate cost of performing action $a \in A$ in state $s \in S$, $s_0 \in S$ is an initial state, and s_g is the goal state such that $T(s_g, a, s_g) = 1 \wedge C(s_g, a) = 0 \; \forall a \in A$.

A solution to an SSP is a policy $\pi : S \to A$ that indicates that action $\pi(s) \in A$ should be taken in state $s \in S$. A policy π induces the state-value function $V^\pi : S \to \mathbb{R}$

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

that represents the expected cumulative cost $V^\pi(s)$ of reaching the goal state s_g from state s following the policy π, and the action-value function $q^\pi : S \times A \to \mathbb{R}$

$$q^\pi(s, a) = C(s, a) + \sum_{s' \in S} T(s, a, s') V^\pi(s')$$

that represents the expected cumulative cost of reaching the goal state s_g from state s given that the agent takes the action a in state s and follows the policy π thenceforth.

Any policy that minimizes these functions is referred to as an optimal policy. Without loss of generality we may assume that the optimal policy, denoted π^*, is unique unless explicitly stated otherwise. Given π^*, we can define the optimal state-value function following policy π^* using the *Bellman optimality equation* as follows

$$\begin{aligned} V^*(s) &= \min_{a \in A} \Big[C(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V^*(s') \Big] \\ &= \min_{a \in A} q^*(s, a) \end{aligned}$$

where q^* is the action-value function under the policy π^*.

Central to our approach is the concept of a *memory state* [Hansen *et al.*, 1996], which 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 in the base SSP that the agent was in and the control actions 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 [Shani *et al.*, 2013]. In fact, the translation can be viewed as a special case of a POMDP where each observation is a null observation.

Definition 1. *Let $\mathcal{M} = \langle S, A, T, C, s_0, s_g \rangle$ be an SSP. 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 connected path in $\mathcal{F}$.*

Note that if we bound the maximal depth of a tree in $\mathcal{F}$ by some finite constant value $\delta \in \mathbb{N}$, we will ensure that the number of memory states is finite. In order to enable this bound, we will assume that the agent has access to a sensing action called `Query` that fully reveals the agent's current state, and require that the only action allowed in a leaf node in $\mathcal{F}$ is `Query`. For notational convenience, we will henceforth use the notation $\mathcal{F}_\delta(S, A)$ to refer to the set of memory states for the state and action sets S and A with finite depth δ.

Given the definition of a memory state, and the action `Query`, we can define an augmented SSP that enables mixed open-loop/closed-loop planning.

Definition 2. *Given the SSP $\mathcal{M} = \langle S, A, T, C, s_0, s_g \rangle$, we define the augmented **memory SSP**, $\overline{\mathcal{M}}_\delta = \langle \overline{S}, \overline{A}, \overline{T}, \overline{C}, s_0, s_g \rangle$ as follows:*

- $\overline{S} = S \cup \mathcal{F}_\delta(S, A)$,
- $\overline{A} = A \cup \{\texttt{Query}\}$,
- $\overline{T} : \overline{S} \times \overline{A} \times S \to [0, 1]$,
- $\overline{C} : \overline{S} \times \overline{A} \to \mathbb{R}$,
- *s_0 and s_g are unchanged.*

Just as in a POMDP we can compute a belief state based on the agent's action and observation history. We can define the transition function in a recursive fashion to compute the transition probability of a successor state given a memory state and action:

$$\overline{T}(sa_1..a_m, a, s') = \sum_{s''} \overline{T}(sa_1..a_{m-1}, a_m, s'') T(s'', a, s')$$

except when $m = \delta$ in which case

$$\overline{T}(sa_1..a_m, a, s') = \begin{cases} 0 & \text{if } a \neq \texttt{Query} \\ \overline{T}(sa_1..a_{m-1}, a_m, s') & \text{if } a = \texttt{Query} \end{cases}$$

If $s \in S$ is not a memory state, then we simply define:

$$\overline{T}(s, a, s') = T(s, a, s')$$

Similarly, the cost function for a memory state is defined recursively as follows:

$$\overline{C}(sa_1..a_m, a) = \sum_{s''} \overline{T}(sa_1..a_{m-1}, a_m, s'') C(s'', a)$$

and in the case where $s \in S$ is not a memory state:

$$\overline{C}(s, a) = C(s, a)$$

4 Modeling Inconsistent Sensory Feedback

As stated earlier, the objective of this paper is to introduce a planning model that better addresses situations when sensory feedback is either *costly*, *unavailable*, or *unreliable*, as discussed in Section 1. In this section we focus on the problem of *unreliability*, but we note that our approach easily captures each case; we discuss how later in the Discussion.

Before formalizing the model, we need to consider three important factors: (1) how we *define* reliability, (2) how we *determine* reliability, and (3) how we *handle* a lack of reliability.

Intuitively, *reliability* in the context of perception should be a measure of how well the state produced by perception represents the agent's true state. Of course, such knowledge implies that the true state is known which obviates the very problem we are looking to address. Instead, we argue that *reliability* should be a measure of how much the state produced by perception *should be trusted*. It is worth clarifying

that this is a distinct notion from standard machine learning evaluations such as accuracy, precision, or recall which have well-defined semantics; however we remark that reliability itself may be conditioned (either explicitly or implicitly) on such information, if known, about the perception system.

In the simplest case *reliability* may be a binary variable that simply indicates if the current state information can be trusted or not; indeed, this is the case that is assumed for the remainder of this paper. However, in general, *reliability* may be more complex, for instance $[0,1]^{|F|}$ where F is the set of state features, representing a continuous reliability score on each state feature. We will refer to this space generally as $\mathcal{R}$.

Second, to determine reliability, we rely on methods from *introspective perception* [Rabiee and Biswas, 2019]. It is sufficient for our purposes to assume the existence of a (trained) introspective perception function $\mathcal{I}$ that will produce a reliability value $r \in \mathcal{R}$ at each timestep based on sensor feedback. We emphasize that reliability information is provided by $\mathcal{I}$ during online operation, and that interaction with $\mathcal{I}$ is not directly included in the offline planning model. However, as discussed below, a model of expected reliability can be incorporated into the planning model based on either expert knowledge or as a learned estimator of feedback from $\mathcal{I}$.

Finally, we assume the existence of a *supervisory sensor* that can be queried at high cost by the system to fully observe the agent's true state and is known to be perfectly reliable.

Given these three components, we can define a new SSP that is augmented with known information of perception unreliability.

Definition 3. *Formally, let $\mathcal{M} = \langle S, A, T, C, s_0, s_g \rangle$ be a ground SSP. We represent a **perception sensitive SSP**, $\tilde{\mathcal{M}}$, by the tuple $\langle \tilde{S}, \tilde{A}, \tilde{T}, \tilde{C}, s_0, s_g \rangle$ where:*

- $\tilde{S} = S \times \mathcal{R}$ *is the set of states where S is the set of ground states and $\mathcal{R}$ is the set/space of reliability values.*
- $\tilde{A} = A \cup \{\texttt{Query}\}$ *is the set of actions where A is the set of ground actions and* `Query` *is an action that queries a supervisory sensor,*
- $\tilde{T} : \tilde{S} \times \tilde{A} \times \tilde{S} \to [0,1]$ *is the transition function, and*
- $\tilde{C} : \tilde{S} \times \tilde{A} \to \mathbb{R}$ *is the cost function.*

We introduce the function $\eta : S \times A \to \Delta^{|\mathcal{R}|}$, called the *reliability profile*, which returns the probabilities over reliability values in the successor state given that the agent took action $a \in A$ in state $s \in S$. For instance, when reliability is a binary value, η returns the 1-step likelihood of perception failing (i.e. entering an unreliable state in the next timestep). In practice, η may be a learned estimator based on feedback from the introspective perception system $\mathcal{I}$. In this work, we assume that η is known a priori. In either case, η is defined over planning-level states, abstracting knowledge about sensor-level reliability to the state space of the SSP.

When $\mathcal{R} = \{r, \neg r\}$ denotes a binary valuation of reliability, the transition function would be the following:

$$\tilde{T}((s,r),a,(s',r)) = T(s,a,s')(1-\eta(s,a))$$
$$\tilde{T}((s,r),a,(s',\neg r)) = T(s,a,s')\eta(s,a)$$
$$\tilde{T}((s,\neg r),a,(s',\cdot)) = \tilde{T}_{\neg r}((s,\neg r),a,(s',\cdot))$$

Here, and throughout the rest of the paper when $\mathcal{R}$ is a binary set of values, for convenience of notation, we use $\eta(s,a)$ to represent the probability of entering an unreliable state in the next timestep given that the agent took action $a \in A$ in state $s \in S$. In other words, when $\mathcal{R}$ is binary we can express η as a function from $S \times A$ to $[0,1]$, indicating the probability of unreliability in the next timestep.

We note that we have explicitly not defined the function $\tilde{T}_{\neg r}$ that represents the transition dynamics in an unreliable state, in the above transition function. This is because by virtue of the agent being in an unreliable state, the agent does not know its true state and hence there is not an obvious well-defined model of the transition function for these states, short of learning one as is done by Lee *et al.* [2020]. The naive approach to handling this problem is to require that the agent query the supervisory sensor as soon as it enters a state of unreliable perception. We take a different approach that leverages memory states and mixed open-loop / closed-loop planning, as discussed in the next section.

5 Knowing When to Act Blind

In this section, we introduce an approach which improves on the naive solution discussed above. The approach uses memory states and mixed open-loop/closed-loop planning to avoid the need to query the supervisory sensor each time an unreliable state is encountered. Our approach relies on mapping a *perception sensitive SSP* which augments the base domain with perception reliability information, to a separate *memory SSP*. This mixed control provides the agent a recourse for acting that is more cost effective than immediately querying the expensive supervisory sensor, while still leveraging this ability in cost-critical situations.

The following observation is the crux of our approach: there is an injective mapping from unreliable states in $S \times \mathcal{R}$ to memory states in $\mathcal{F}(S,A)$ when $\mathcal{R}$ is binary. The mapping is as follows: upon arriving in an unreliable state, the agent deterministically transitions to a memory state sa where s is the last reliable state the agent was in and a is the action the agent just performed. The agent can then continue to operate in open-loop control without querying the supervisory sensor as this mapping gives us a well defined transition function for the state that the agent is in. We can bound the depth of each memory state tree by bounding the number of actions that the agent can take in open-loop control before querying the supervisory sensor to ensure a finite state space.

However, this model shift introduces a particular complexity. Namely, when operating in open-loop control, it is entirely possible that perception unreliability may resolve itself (or be resolved through the actions that were performed in open-loop). As an example, the agent may enter a part of a hallway with glare from the window causing $\mathcal{I}$ to judge perception as unreliable. The agent may elect to move forward several meters in open-loop control instead of querying its supervisory sensor as there is low likelihood of a problem occurring. After doing so, it may pass the problematic area and perception may once again be judged reliable, providing full state observability to the agent.

To account for this, we need to update our reliability pro-

file, η, to a new function $\overline{\eta} : \overline{S} \times A \to \Delta^{|\mathcal{R}|}$. We note that, for any $\overline{s} \in S$, and any $a \in A$, $\overline{\eta}(\overline{s}, a) = \eta(\overline{s}, a)$, $\overline{\eta}$ simply models the fact that the system can not only fall *out of* reliable perception, but can actually fall back *into* reliable perception without querying the supervisory sensor. However, for any memory state, the function $\overline{\eta}$ will have to be learned or approximated as the agent does not necessarily have any way of knowing what state it was in when its reliability status changed (several possible states that it could have been in may, under the same action, lead with nonzero probability the state that it arrives in). Note that we continue the notational convention introduced in Section 4 regarding the representation of η when $\mathcal{R}$ is binary for $\overline{\eta}$ as well.

An important observation to make is that when there is no probability of perception being unreliable, given that the agent starts in a reliable state, the agent will achieve its best performance in expectation. Or, in other words, perception unreliability can only diminish the agent's expected performance. We formalize this below.

Proposition 1. *Let $\mathcal{M} = \langle S, A, T, C, s_0, s_g \rangle$ be an SSP, and let $\overline{\mathcal{M}}_\delta$ be the corresponding memory SSP with depth δ. The optimal value function for $\overline{\mathcal{M}}_\delta$ defined as*

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

is minimized when $\overline{\eta}[S \times A] = \{0\}$, given that $s_0, s_g \in S$.

To prove this, we first need to show that when the action `Query` has zero cost in memory states (i.e. it is a "free" action) the q-value of `Query` will be no higher than the q-value of any other available action. Intuitively, this means that `Query` can be assumed to be taken immediately upon entering a memory-state or, equivalently, immediately upon entering an unreliable state. Formally:

Lemma 1. *Let $\overline{\mathcal{M}}$ be a memory SSP where*

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

Then, given a policy π, $q^\pi(\overline{s}, a) \geq q^\pi(\overline{s}, \texttt{Query})$ for any $a \neq$ `Query` 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 $\overline{s} = sa_1 \cdots a_k \in \overline{S}$ be a memory state. Then $q(\overline{s}, a) - q(\overline{s}, \texttt{Query})$

$$= \overline{C}(\overline{s}, a) + \sum_{\overline{s}' \in \overline{S}} \overline{T}(\overline{s}, a, \overline{s}')V^\pi(\overline{s}') - \Big[\overline{C}(\overline{s}, \texttt{Query}) + \sum_{\overline{s}' \in \overline{S}} \overline{T}(\overline{s}, \texttt{Query}, \overline{s}')V^\pi(\overline{s}')\Big]$$

$$= \overline{C}(\overline{s}, a) + \sum_{\overline{s}' \in \overline{S}} \sum_{s' \in S} \overline{T}(sa_1 \cdots a_{k-1}, a_k, s')\overline{T}(s', a, \overline{s}')V^\pi(\overline{s}') - \sum_{s' \in S} \overline{T}(sa_1 \cdots a_{k-1}, a_k, s')V^\pi(s')$$

$$= \sum_{s' \in S} \overline{T}(sa_1 \cdots a_{k-1}, a_k, s')C(s', a) + \sum_{s' \in S} \overline{T}(sa_1 \cdots a_{k-1}, a_k, s')\Big[\sum_{\overline{s}' \in \overline{S}} \overline{T}(s', a, \overline{s}')V^\pi(\overline{s}') - V^\pi(s')\Big]$$

$$= \sum_{s' \in S} \overline{T}(sa_1 \cdots a_{k-1}, a_k, s')\Big[C(s', a) + q^\pi(s', a) - C(s', a) - V^\pi(s')\Big]$$

$$= \sum_{s' \in S} \overline{T}(sa_1 \cdots a_{k-1}, a_k, s')\Big[q^\pi(s', a) - \min_{a' \in A} q^\pi(s', a')\Big] \geq 0$$

□

Given Lemma 1 we now have what we need to prove Proposition 1.

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

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

up to the execution of the action `Query`. Upon entering a memory state, $\overline{s}$, by Lemma 1, `Query` will have the lowest q-value, and hence the agent will always immediately query the supervisory sensor 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 not ever needing to query as the cost of the action is 0 in a memory state and there is no discounting. Hence, any positive adjustment to the cost of `Query` will increase the expected cumulative cost of a memory state, leading to the same or greater value for every state in the domain under the optimal policy. Hence, if $\overline{\eta}[S \times A] \subset [0, 1]$ and $\overline{C}(\overline{s}, \texttt{Query}) > 0$, $\overline{V}^*(s)$ will be at least as large as when $\overline{\eta}[S \times A] = \{0\}$. □

6 Efficient Planning

The main difficulty faced when planning on memory states is the explosion in the size of the state space which is combinatorial in the number of actions and the maximal depth δ. Hansen *et al.* [1996] handle this issue in the context of Q-learning by pruning branches in $\mathcal{F}(s, a)$ during exploration where the value of information in that memory state is greater than or equal to its cost.

In our case, as we are performing optimal model based planning, we employ an optimal search-based planning algorithm, LAO* [Hansen and Zilberstein, 2001], using a heuristic in place of the above pruning rule to guide search. LAO* is known to be converge to an optimal policy given an admissible heuristic. Our heuristic is based on the optimal value function of the base SSP; in particular, we observe that planning in the base domain will exhibit the same behavior as

planning in the memory domain where sensing never fails and has no cost. In other words, it is an 'ideal' instance of the problem, and therefore solving it provides a (fairly tight) lower bound on the value of any state in the memory state version of the problem, giving us our heuristic. We formally define it below.

Definition 4. *Let $\mathcal{M} = \langle S, A, T, C, s_0, s_g \rangle$ be an SSP, and let $\overline{\mathcal{M}}_\delta$ be the corresponding memory SSP for the perception sensitive extension of $\mathcal{M}$, $\tilde{\mathcal{M}}$, given some $\delta \in \mathbb{N}$. 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:*

$$h_{V^*}(\overline{s}) = \begin{cases} V^*(\overline{s}) & \text{if } \overline{s} \in S \\ \sum_{s'} \overline{T}(sa_1..a_{m-1}, a_m, s')V^*(s') & \text{otherwise} \end{cases}$$

Theorem 1. *$h_{V^*} : \overline{S} \to \mathbb{R}$ is an admissible heuristic for $\overline{\mathcal{M}}_\delta$ where $\delta \geq 1$.*

Proof. Consider $\overline{\mathcal{M}}_\delta$ for any positive integer δ. First, observe that if $\overline{\eta}[S \times A] = \{0\}$, under any well-defined policy for $\overline{\mathcal{M}}_\delta$ the set of reachable states is exactly S. Hence, $\overline{\mathcal{M}}_\delta$ with such an $\overline{\eta}$ is equivalent in behavior to that of $\mathcal{M}$ since the action `Query` will never be taken by an optimal policy, and hence will emit the same behavior meaning that $V^*(s) = \overline{V}^*(s)$ for all $s \in S$. Furthermore, by Proposition 1, we know that $\overline{V}^*(s)$ is minimized when $\overline{\eta}[S \times A] = \{0\}$, which implies that $V^*(s)$ is a minimal bound on the value function for every state $s \in S$ in $\overline{\mathcal{M}}$.

For any $\overline{s} \in \mathcal{F}_\delta(S, A)$,

$$\begin{aligned} \overline{V}^*(\overline{s}) &= \overline{C}(\overline{s}, \overline{\pi}^*(\overline{s})) + \sum_{\overline{s}' \in \overline{S}} \overline{T}(\overline{s}, \overline{\pi}^*(\overline{s}), \overline{s}')\overline{V}^*(\overline{s}') \\ &\geq \min_{\overline{s}' \in \overline{S} | \overline{T}(\overline{s}, \overline{\pi}^*(\overline{s}), \overline{s}') > 0} \overline{V}^*(\overline{s}') \\ &\geq \min_{s' \in S | \overline{T}(\overline{s}, \overline{\pi}^*(\overline{s}), s') > 0} \overline{V}^*(s') \\ &\geq \min_{s' \in S | \overline{T}(\overline{s}, \overline{\pi}^*(\overline{s}), s') > 0} V^*(s') \end{aligned}$$

by Lemma 1 and the above logic. □

We can also show that given two memory SSPs of the same domain but with different maximal tree depth δ, the one with the greater depth will emit an optimal policy, and hence optimal value function, that is at least as good as that of the other. In other words, increasing the maximum allowed duration of open-loop control can never increase expected cost.

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 $s \in S \cup (\mathcal{F}_\delta(S, A) \cap \mathcal{F}_{\delta'}(S, A))$, $V^*_{\delta'}(s) \leq V^*_\delta(s)$.*

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, $\pi^*_\delta(s) =$ `Query`, and by construction $\pi_{\delta'}(s) =$ `Query`. 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 $V^*_{\delta'}(s) \leq V^\pi_{\delta'}(s) = V^*_\delta(s)$ for every state $s \in S \cup (\mathcal{F}_\delta(S, A) \cap \mathcal{F}_{\delta'}(S, A))$. □

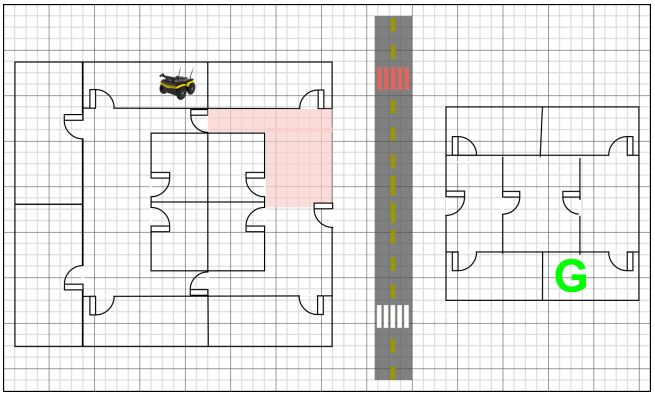

Figure 1: An illustration of the simulated domain used in our experiments. The agent, represented by the Jackal image, must navigate from where it is to the goal state, represented by the green 'G' while managing obstacles, perception failures, and its supervisory sensor. The Jackal, doors, and crosswalks are not drawn to scale.

7 Empirical Evaluations

We test our approach in a simulated robotic domain where the objective is for a robot to deliver a package from one office in a campus environment to another office in a different building on the same campus. In order to reach its destination, the robotic agent needs to navigate through the world while also successfully interacting with both building doors and traffic-ridden crosswalks.

An illustrated example of the domain can be seen in Figure 1. Each grid cell represents a unique location in the world which corresponds to some state or states in the domain model. There are two sources of stochasticity in the domain. When moving, there is a probability that the robot gets stuck and does not move. When waiting at a crosswalk, the traffic conditions stochastically change at each timestep.

In addition to navigating through the domain as described, the robot is faced with limitations on its regular sensory feedback. In particular, at all times there is a small (0.1) likelihood of perception failing, and in certain parts of the domain (denoted in shaded red) there is a higher (up to 0.9) likelihood of perception failing due to environmental factors.

If the agent tries to move into a wall or door, it suffers high cost, and if it attempts to cross the road when there is oncoming traffic, there is a chance of getting hit resulting in extremely high cost. Hence, the agent must proactively plan to operate in open-loop control when there is low likelihood of unreliability, while either avoiding areas of high likelihood of perception unreliability or querying its supervisory sensor at the appropriate time to avoid these failure cases.

To solve for the agent's policy, we employ the algorithm LAO*. We compare the efficiency of LAO* when using two different admissible heuristics: the null heuristic (which is always admissible), and the heuristic defined in Definition 4, h_{V^*}. As both heuristics lead to optimal policies, we use the

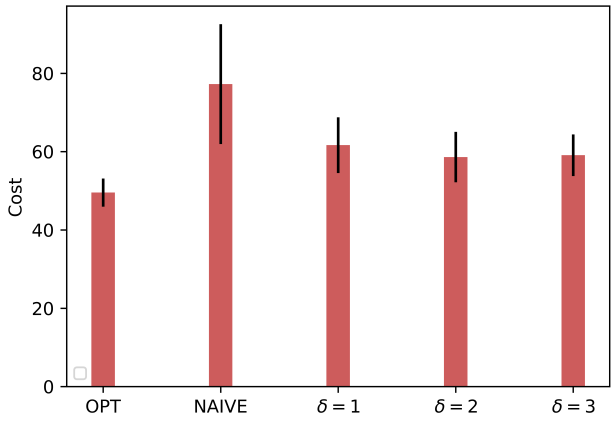

Figure 2: The true incurred cost (mean and std) for base domain (OPT), base domain with required and unexpected sensing (NAIVE), and the optimal policy for the memory SSP with delta = 1, 2, 3

policy derived from the the h_{V^*} heuristic without loss of generality in our plots.

To evaluate our approach, we compare the optimal performance in the following scenarios. First, as a baseline, the base model $\mathcal{M}$ where there is no probability of unreliable perception (OPT). Second, we introduce unreliable perception, but the agent always follows the optimal policy on $\mathcal{M}$ and immediately takes the action `Query` in a memory state; this simulates the situation where the agent does not proactively account for perception unreliability in its model and its only recourse is to query its supervisory sensor (NAIVE). The remaining three scenarios are when the agent uses the optimal policy for the memory SSP with $\delta = 1, 2$, and 3.

Figure 2 illustrates that our approach can lead to a decrease in average cost incurred by the agent in performing its task, and in particular is closest to the cost when perception is never unreliable (OPT). Notably, the biggest gain comes from the basic inclusion of perception reliability knowledge into the planning model, which enables the agent to proactively plan to avoid parts of the domain where $\overline{\eta}$ is high. Simply adding memory states of depth 1, which increases the state space by only a factor of $|A| \ll |S|$, leads to a 20.17% decrease in the average incurred cost for the same task, and a 50% reduction in standard deviation, indicating that performance is significantly more reliable. However, the graph also illustrates empirically what was proved in Proposition 2, namely that increasing δ can improve performance by up to as much as 4.93% in the domain tested. While the benefit is not as significant as the 20% gained above, we expect that the benefit may be even greater in domains where high cost outcomes are less likely (e.g. driving in a large open area) than the domain considered here where the agent operates in a fairly confined space and runs the risk of crashing if it operates in open-loop control for too long.

Table 1 shows that LAO* with our heuristic converges up to 5 times faster than with the null heuristic, and expands roughly half the number of nodes. In the case of $\delta = 3$, LAO* fails to converge within 12 hours at which point it was terminated.

δ	$\lvert\overline{S}\rvert$	Heuristic	Time (s)	Nodes Expanded
1	3048	h_0	268.32	1025
		h_{V^*}	49.97	561
2	21717	h_0	2277.98	5223
		h_{V^*}	675.319	2573
3	152400	h_0	—	—
		h_{V^*}	22486.81	13998

Table 1: Efficiency Comparison of LAO* with the null heuristic and the h_{V^*} heuristic introduced in this paper for three different maximum depths of memory states.

8 Discussion

Robust and reliable planning in the open world is an important area of work in AI and robotics, which involves many challenges. In this paper, we focused on the problem of inconsistent sensory feedback that can take the form of costly sensing, unavailable sensing, and unreliable perception, with an emphasis on the third problem which we consider the most difficult to plan around.

To address this problem, we proposed an augmentation of a standard stochastic shortest path problem that is sensitive to unreliable perception by including a measure of reliability in the state representation. To enable this, we assumed that the agent has access to both an introspective perception system which notifies the agent when perception is unreliable, and a costly supervisory sensor that can fully reveal the state to the agent whenever it is queried. We showed that we can in fact do better than naively querying the supervisory sensor immediately upon reaching an unreliable state by mapping the planning model augmented with perception reliability information, called a *perception sensitive SSP* to another extended stochastic shortest path model called a *memory SSP* that allows the agent to plan for and perform mixed open-loop/closed-loop control. This mixed control enables the agent a recourse for acting that is more cost effective than immediately querying the expensive supervisory sensor, while still leveraging this ability in cost-critical situations.

It is straightforward to see that our approach also handles problems where standard sensing is always available and reliable, but has non-negligible cost, as considered by Hansen *et al.* [1996], or when base sensing is always free and reliable but not always available. To model the former, we can set the reliability profile η so that the model deterministically transitions to an unreliable state when the action `Query` is not taken; in this case, querying is equivalent to the standard sensing action considered by Hansen *et al.* [1996]. To address the latter, we can simply re-frame the boolean reliability indicator as *availability* and use the model as described, where η is simply a predictor of unavailable sensing instead.

We provided several theoretical results about our approach including an admissible heuristic which, we showed, speeds up the runtime of the planning algorithm by as much as 5 times. Additionally, we empirically demonstrated the value gained by including memory states and mixed open-loop/closed-loop control, showing that it decreased average incurred cost by up to 20%, and that increasing the length of

open-loop control action sequences leads to improved performance (up to 5%).

There are several important avenues for future work. Most notably, it would be beneficial to develop faster solution techniques, as the main bottleneck of this approach is a combinatorial increase in the size of the state space. Methods for reducing the number of states considered via pruning mechanisms, planning on macro actions, and alternative planning algorithms such as short-sighted approaches or Monte Carlo tree search may be the key to improving the efficiency of solving these problems. Additionally, we would like to further investigate the notion of unreliability in perception; while this paper focuses solely on the case where reliability is an all-or-nothing concept, in practice it may instead be binary for each state feature, or, in the most general case, some real number for each state feature that indicates *how* reliable that feature is. Finally, we would like to deploy this on a real mobile robot to demonstrate that this approach is feasible in a real open-world setting.

Acknowledgments

This work was supported in part by the National Science Foundation grants IIS-1724101 and IIS-1954782.

References

Ruzena Bajcsy, Yiannis Aloimonos, and John K. Tsotsos. Revisiting active perception. *Autonomous Robots*, 2018.

Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 1988.

Connor Basich, Justin Svegliato, Kyle Hollins Wray, Stefan Witwicki, Joydeep Biswas, and Shlomo Zilberstein. Learning to optimize autonomy in competence-aware systems. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2020.

Shreyansh Daftry, Sam Zeng, J. Andrew Bagnell, and Martial Hebert. Introspective perception: Learning to predict failures in vision systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

Jennifer Dooley. Mission concept for a Europa Lander. In *IEEE Aerospace Conference*, 2018.

Marc Hanheide, Moritz Göbelbecker, Graham S. Horn, Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Patric Jensfelt, Charles Gretton, Richard Dearden, Miroslav Janicek, et al. Robot task planning and explanation in open and uncertain worlds. *Artificial Intelligence*, 2017.

Eric A. Hansen and Shlomo Zilberstein. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 2001.

Eric A. Hansen, Andrew G. Barto, and Shlomo Zilberstein. Reinforcement learning for mixed open-loop and closed-loop control. In *Neural Information Processing Systems Conference (NIPS)*, 1996.

Krishnanand N. Kaipa, Akshaya S. Kankanhalli-Nagendra, Nithyananda B. Kumbla, Shaurya Shriyam, Srudeep Somnaath Thevendria-Karthic, Jeremy A. Marvel, and Satyandra K. Gupta. Addressing perception uncertainty induced failure modes in robotic bin-picking. *Robotics and Computer-Integrated Manufacturing*, 2016.

Michelle A. Lee, Carlos Florensa, Jonathan Tremblay, Nathan Ratliff, Animesh Garg, Fabio Ramos, and Dieter Fox. Guided uncertainty-aware policy optimization: Combining learning and model-based strategies for sample-efficient policy learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

Sadegh Rabiee and Joydeep Biswas. IVOA: Introspective vision for obstacle avoidance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

Sandhya Saisubramanian, Ece Kamar, and Shlomo Zilberstein. A multi-objective approach to mitigate negative side effects. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

Guy Shani, Joelle Pineau, and Robert Kaplow. A survey of point-based POMDP solvers. *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2013.

Justin Svegliato, Kyle Hollins Wray, Stefan J. Witwicki, Joydeep Biswas, and Shlomo Zilberstein. Belief space metareasoning for exception recovery. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

Justin Svegliato, Samer B. Nashed, and Shlomo Zilberstein. Ethically compliant sequential decision making. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Shlomo Zilberstein. Resource-bounded sensing and planning in autonomous systems. *Autonomous Robots*, 3(1):31–48, 1996.