Minimally sufficient structures for information-feedback policies

Basak Sakcak, Vadim K. Weinstein, Kalle G. Timperi, and Steven M. LaValle *

Center for Ubiquitous Computing
Faculty of Information Technology and Electrical Engineering
University of Oulu, Finland
{firstname.lastname}@oulu.fi

Abstract. In this paper, we consider active tasks which require planning and control to be achieved. Such tasks require a desirable outcome in the physical world resulting from the interactions between an internal system, corresponding to a filter and a policy, and an external system, representing the physical world. A filter is seen as the robot's perspective of the external world based on limited sensing, memory, and computation and it is represented as a transition system over a space of information states. The interactions result from the coupling of these two systems through a sensor-mapping and an information-feedback policy. Within this setup, we look for sufficient structures, that is, sufficient internal systems and sensors, for a given task. We establish necessary and sufficient conditions for these structures to satisfy so that feasible information-feedback policies exist that can be defined over the states of an internal system. We also show that under mild assumptions, minimal internal systems that can represent a particular plan/policy described over the action-observation histories exist and are unique. Finally, the results are applied to determine sufficient structures for distance-optimal navigation in a polygonal environment.

Keywords: Planning, Transition Systems, Information Spaces, Sensing Uncertainty, Theoretical Foundations.

1 Introduction

Planning and control is a fundamental problem in robotics and requires determining a sequence of actions which would result in accomplishing a particular task or a set of tasks in a space of environments. Given a well-defined task structure, and particular robot hardware, solving this problem requires designing a filter and a policy over the filter states. Therefore, whether filters and policies are computed, or learned, it is clear that these two structures should be analyzed and designed together.

* This work was supported by a European Research Council Advanced Grant (ERC AdG, ILLUSIVE: Foundations of Perception Engineering, 101020977), Academy of Finland (projects PERCEPT 322637, CHiMP 342556).

A significant difference between a pure inference task, which corresponds to keeping track of the physical world state, and a planning or control task is that in the latter case we are only interested in distinguishing the action to take at a particular state and not the state of the physical world itself. Therefore, a filter that correctly predicts the outcomes of actions in terms of observations may not be meaningful if all we need is a way to distinguish which action to take.

Most work in the literature separates designing filters (or observers) from respective policies. Typically, filters are designed to estimate the state and policies are determined over the estimated state of the physical world (see [13],[26]). For problems in which the state is not fully observable, *partially observable Markov decision processes* (POMDPs) [10],[15] and *belief spaces* [23],[1] have been considered for planning. Considering mapping observations to actions, early work characterized action-based sensors which provide only the information that is necessary and sufficient, exactly what is needed for an action to be determined [7]. In this case, states can be grouped through equivalence relations induced by an action-based sensor such that the same action is applied for any state within an equivalence class. This assumes that at each instant of determining an action, relevant information can be extracted from the environment. Hence, yielding *memoryless* or *reactive* policies. However, if this is not the case or if the robot sensors are fixed, then, the decisions need to be based on the history of previous actions and observations. At one end of the spectrum, assuming unlimited memory, these decisions can be made based on full histories. However, in general, this is computationally unfeasible. Therefore, relevant information needs to be extracted from the histories by a filter, allowing a feasible policy to be defined over its states.

In this work, we analyze the relationship between active tasks and a particular filter together with a policy defined over its states. Many of the concepts will build upon our previous work [24,18,17], in which we introduced a general framework built from input-output relationships between two or more coupled dynamical systems. In the basic setting of a robot embedded in an environment, these two dynamical systems correspond to an *internal system* (a centralized computational component) and an *external system* (robot body and the environment). Given particular robot hardware, that is, fixing the robot sensors and actuators, input-output relations correspond to actions and sensor observations.

The internal system is formally described as a transition system, named an *information transition system* (ITS), with a state space that is an *information space* (I-space). I-spaces are introduced in [11, Chapter 11] as a way of analyzing the information requirements of robotic tasks. These were inspired by games with hidden information [2]. The term *information* is related to the von Neumann-Morgenstern notion of information and not to the later notion introduced by Shannon. We see an ITS as a filter and a policy is defined over its states. Derived I-spaces and quotient ITSs are obtained from action-observation histories using *information mappings* (I-maps) that are many-to-one. A derived I-space constitutes the state space of a quotient (derived) ITS. Within this framework, we analyze conditions that these derived ITSs should satisfy so that feasible, that is, task-accomplishing, policies can be described over their states. Planning and control tasks, termed *active tasks*, were already considered in [18,17]. The results there provided a scaffolding but lacked in establishing the necessary and sufficient conditions and a characterization of sufficient filter-policy pairs, which we do in this paper. In particular, we will consider two cases; fixing the sensor-mapping and analyzing a sufficient ITS, and fixing a particular class of policy and analyzing sufficient sensors for that class.

There is a limited literature that studied the information requirements for active tasks. This corresponds to determining the weakest notion of sensing or filtering that is sufficient to accomplish a task. A notable early work showed, especially for manipulation, that one can achieve certain tasks even in the absence of sensory observations [6]. Considering specific problems in mobile robot navigation [4],[21] addressed minimal sensors and filters that are sufficient for navigation. In [25], the authors characterize all possible sensor abstractions that are sufficient to solve a planning problem. Closely related to our work, a language-theoretic formulation appears in [16], in which, Procrustean-graphs (p-graphs) were proposed as an abstraction to reason about interactions between a robot and its environment. Following up from [7], [14] analyzes conditions for the existence of action-based sensors encoding a particular plan. The authors also propose an algorithm that decomposes those plans for which no action-based sensor exists into subpieces for which one does exist.

We focus on sufficient internal systems that can result in task accomplishment once coupled to the external system. Therefore, our treatment can be seen as characterizing a sufficient (or minimal) plan, that is, establishing conditions that an ITS should satisfy so that a feasible policy can be defined over its states. This is different than characterizing a sufficient ITS for planning, in which case an ITS should allow a policy to be computed. A recent work [19] addresses the latter case by establishing sufficient conditions that an *information mapping* (I-map) should satisfy for it to allow a dynamic programming formulation.

The rest of the paper is organized as follows. Section 2 briefly reviews mathematical underpinnings and develops notions related to the coupling of internal and external systems and to task accomplishment. Section 3 applies the general concepts to develop the notion of a feasible policy and what it means for an ITS to support a feasible policy. In this section we establish necessary and sufficient conditions for ITSs considering fixed sensors, and for sensors considering reactive policies. Section 4 applies the developed ideas to the problem of distance optimal navigation in plane. Finally, Section 5 concludes the paper by summarizing the results and indicating exciting avenues for future research that could arise from this paper.

2 Information Transition Systems

We consider a robot embedded in an environment and model the interaction between a decision making entity and the physical world as coupled *internal* and *external* systems (see Figure 1).

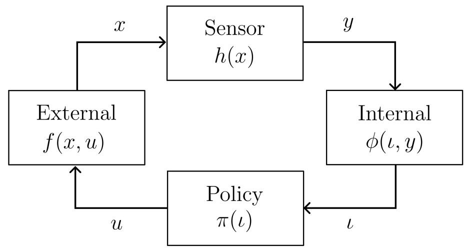

Fig. 1. Internal $(\mathcal{I}, Y, \phi)$ and external (X, U, f) systems coupled through coupling functions h and π, the sensor mapping and the information-feedback policy, respectively.

The external system corresponds to the totality of the physical environment, including the robot body. Let X denote the set of external states and let U be the set of actions. When applied at a state $x \in X$, a control $u \in U$ causes x to change according to a state transition function $f\colon X \times U \to X$.

The internal system represents the perspective of a decision maker. The states of this system correspond to the retained information gathered through the outcomes of actions in terms of sensor observations. To this end, the basis of our mathematical formulation of the internal system is the notion of an *I-space* presented in [11, Chapter 11]. Let $\mathcal{I}$ be an information space. We will use the term information state *(I-state)* to refer to the elements of $\mathcal{I}$ and denote them by ι.

We model both the internal and the external systems as transition systems. Following the terminology introduced in [18], an internal system will be referred to, more generally, as an ITS.

Definition 1 (Information Transition System (ITS)). An ITS is the quadruple $S = (\mathcal{I}, \Lambda, \phi, \iota_0)$, in which $\mathcal{I}$ is an information space corresponding to the states of the transition system, Λ is the set of edge labels, $\phi\colon \mathcal{I} \times \Lambda \to \mathcal{I}$ is the information transition function, and ι_0 is the initial state.

In particular, we will consider two sets of edge labels, namely $\Lambda = U \times Y$ and $\Lambda = Y$, in which U and Y are the sets of actions and observations, respectively.

Definition 2 (State-relabeled ITS). Given an ITS $S = (\mathcal{I}, \Lambda, \phi, \iota_0)$ a state-relabeled ITS is the 6-tuple $S_\ell = (\mathcal{I}, \Lambda, \phi, \iota_0, \ell, L)$, in which $\ell : \mathcal{I} \to L$ is a labeling function that attributes to each state $\iota \in \mathcal{I}$ a unique label from set L.

The external system is modeled as a state-relabeled transition system as well, for this case labels correspond to the observations.

Definition 3 (External System). An external system is a state-relabeled transition system $\mathcal{X}_h = (X, U, f, h, Y)$, in which X is the state space, U is the set of edge labels corresponding to the set of actions, $f : X \times U \to X$ is the (external) state transition function, and $h : X \to Y$ is a labeling function corresponding to the sensor-mapping.

In our framework, a labeling function defined over the states of an ITS has two purposes: (i) deriving quotient ITSs and (ii) acting as a coupling function that maps the output of an ITS to the input of an external system (see Figure 1). The latter case corresponds to the policy $\pi : \mathcal{I} \to U$.

We now focus on the former case of quotient systems derived from a state-relabeled ITS. Let $\mathcal{I}_{der}$ be a derived I-space and let $\kappa : \mathcal{I} \to \mathcal{I}_{der}$ be an I-map that is a labeling function defined over the states of an ITS $S = (\mathcal{I}, \Lambda, \phi)$. Preimages of κ introduce a partitioning of $\mathcal{I}$ creating equivalence classes. Let $\mathcal{I}/\kappa$ be the equivalence classes $[\iota]_\kappa$ induced by κ such that $\mathcal{I}/\kappa = \{[\iota]_\kappa \mid \iota \in \mathcal{I}\}$ and $[\iota]_\kappa = \{\iota' \in \mathcal{I} \mid \kappa(\iota') = \kappa(\iota)\}$. Then, through these equivalence classes, we can define a new ITS, called the *quotient of S by κ*, denoted by S/κ. It is defined as $S/\kappa = (\mathcal{I}/\kappa, \Lambda, \phi/\kappa)$, in which

$$\phi/\kappa := \big\{ \big([\iota]_\kappa, \lambda, [\iota']_\kappa\big) \mid (\iota, \lambda, \iota') \in \phi \big\}.$$

Above, the map $\phi : \mathcal{I} \times \Lambda \to \mathcal{I}$ is treated as a subset of $\mathcal{I} \times \Lambda \times \mathcal{I}$. An important notion when obtaining quotient ITSs through labeling functions is *sufficiency*.

Definition 4 (Sufficiency). A labeling function $\kappa \colon \mathcal{I} \to \mathcal{I}_{der}$ defined over the states of a transition system $(\mathcal{I}, \Lambda, \phi, \iota_0)$ is called *sufficient*, if for all $s, t, s', t' \in \mathcal{I}$ and all $\lambda \in \Lambda$, the following implication holds:

$$\kappa(s) = \kappa(t) \wedge s' = \phi(s, \lambda) \wedge t' = \phi(t, \lambda) \implies \kappa(s') = \kappa(t').$$

In [24,17], it was shown that the quotient of an ITS is also an ITS as in Definition 1 if and only if κ is sufficient. This is because given the label of the current state the edge label the label of the next state can be uniquely determined if the labeling function is sufficient, ensuring that the state transitions of the quotient system are deterministic. Hence, ϕ/κ represents a function.

Given a labeling function κ, we might be interested in a finer labeling function which distinguishes the states distinguished by κ but at a higher resolution. This is achieved by the notion of a refinement of κ that is defined in the following.

Definition 5 (Refinement of an I-map). An I-map κ' is a *refinement* of κ, denoted by $\kappa' \succeq \kappa$, if for all $A \in \mathcal{I}/\kappa'$ there exists $B \in \mathcal{I}/\kappa$ such that $A \subseteq B$.

The history ITS is a special type of ITS from which others will be derived through sufficient I-maps. Let $(A)^{<\mathbb{N}}$ denote the set of all finite-length sequences of elements of A. The elements of the history information space, denoted by $\mathcal{I}_{hist}$, are finite sequences of alternating actions and observations which build upon the initial state $\eta_0 = () \in \mathcal{I}_{hist}$, therefore, $\mathcal{I}_{hist} = (U \times Y)^{<\mathbb{N}}$.

Definition 6 (History ITS). The history ITS $S_{hist} = (\mathcal{I}_{hist}, U \times Y, \phi_{hist}, \eta_0)$ is an ITS with state space $\mathcal{I}_{hist}$ and the information transition function ϕ_{hist} is defined starting from $\eta_0 = ()$ through the concatenation operation, that is,

$$\eta_k = \eta_{k-1}{}^\frown(u_{k-1}, y_k).$$

External and internal systems can be coupled through the coupling functions that map the input of one to the output of the other and vice versa. For us, the sensor mapping $h : X \to Y$ and the policy $\pi : \mathcal{I} \to U$ are two coupling functions. The coupled system of internal and external described this way is an autonomous system (a closed system), meaning that given an initial state $(\iota_0, x_1) \in \mathcal{I} \times X$ there exists a unique trajectory. We denote the function $(\iota, x) \mapsto (\iota', x')$ by $\phi *_{\pi,h} f$. Then, the coupled system, denoted by $S_\pi \star \mathcal{X}_h$, is the pair $S_\pi \star \mathcal{X}_h = (\mathcal{I} \times X,\, \phi *_{\pi,h} f)$. It is also possible to have a single coupling function, in which case the coupled internal-external system admits an input. If no policy is determined over the internal system, the coupled system is determined by $(\mathcal{I} \times X, U, \phi *_h f)$, in which $\phi *_h f : \mathcal{I} \times X \times U \to \mathcal{I} \times X$ is the state transition function. This can be seen as the perspective that one has for planning purposes which allows to evaluate any possible action at a given I-state.

Consider the coupling of a history ITS $(\mathcal{I}_{hist}, U \times Y, \phi_{hist})$ with an external system (X, U, f) through the coupling function h, that is, $(\mathcal{I}_{hist} \times X, U, \phi_{hist} *_h f)$. Let $U^{<\mathbb{N}}$ be the set of all finite-length action sequences that can be fed as an input to this coupled system. A coupled state $(\eta, x) \in \mathcal{I}_{hist} \times X$ is *reachable* from $((), x_1)$ if there exists some $\tilde{u} \in U^{<\mathbb{N}}$ such that the state of the coupled system becomes (η, x) when $\tilde{u}$ is applied starting from an initial state $((), x_1) \in \mathcal{I}_{hist} \times X$.

Definition 7 (Set of attainable histories). Given a coupling of a history ITS with an external system $\mathcal{X}_h = (X, f, U, h, Y)$, a history $\eta \in \mathcal{I}_{hist}$ is called *attainable* if there exist $x, x_1 \in X$ such that the coupled state (η, x) is reachable from the initial state $((), x_1)$. We denote by $\mathcal{I}_{hist}^{\mathcal{X}_h}$ the set of attainable histories.

The coupling with an external system $\mathcal{X}_h = (X, f, U, h, Y)$ induces a labeling function $\kappa_{att} : \mathcal{I}_{hist} \to \{0, 1\}$ over the histories through ${\kappa_{att}}^{-1}(1) = \mathcal{I}_{hist}^{\mathcal{X}_h}$. If a history I-state η_K up to some stage K is unattainable, then any history I-state η_N up to some stage $N > K$ that builds upon η_K will also be unattainable. This is stated in the following lemma, which follows directly from Definition 7.

Lemma 1. *For any $\eta \in {\kappa_{att}}^{-1}(0)$, and any $(u, y) \in U \times Y$, the next history I-state satisfies $\phi_{hist}(\eta, (u, y)) \in {\kappa_{att}}^{-1}(0)$.*

3 Sufficient structures for solving active tasks

We focus on solving *active tasks* which entail executing an information-feedback policy that forces a desirable outcome in the external system. In [17], we showed that given a feasible policy defined over the history I-space, minimal ITSs exist that can support that policy. In this section, we will expand on this idea and formally define what supporting a policy means. Since our formulation treats ITSs in conjunction with respective policies, we will consider two cases:

– Fixing the sensor-mapping h, which corresponds to fixing S_{hist}, and characterizing the sufficient ITSs that support a particular feasible or optimal policy defined over $\mathcal{I}_{hist}$.

– Fixing the particular ITS, $S = (\mathcal{I}, Y, \phi)$ and characterizing the sensors $h : X \to Y$ that are sufficient for this ITS to support a particular class of feasible policies, that is, mappings from $\mathcal{I}$ to U. In particular, we will consider *reactive policies* such that the ITS is simply $S = (Y, Y, \mathrm{id}_Y)$ in which id_Y is the identity function.

3.1 Task Description

A task is encoded through a *task-induced labeling* function $\kappa_{task} : \mathcal{I}_{hist} \to \{0, 1\}$, meaning that ${\kappa_{task}}^{-1}(1)$ is the set of histories that are task accomplishing. A task-induced labeling can be given, learned, or specified through a logical language over $\mathcal{I}_{hist}$ or X (see [17] for a discussion on the ways of determining κ_{task}).

When tasks are specified using a logical language over $\mathcal{I}_{hist}$, the resulting sentences of the language involve combinations of predicates that assign truth values to subsets of $\mathcal{I}_{hist}$. This implicitly defines $\kappa_{task} : \mathcal{I}_{hist} \to \{0, 1\}$, in which 0 stands for `false` and 1 stands for `true`. When a task description is determined as a logical language over X, the resulting sentences of the language involve combinations of predicates that assign truth values to subsets of X (see [8,3] for examples using linear temporal logic). Given a sequence of actions $\tilde{u} = (u_1, u_2, \ldots, u_N)$ and an initial state x_1, an external state trajectory $x_1 \diamond \tilde{u}$ is a sequence of external system states defined as

$$x_1 \diamond \tilde{u} = (x_1, x_2 = f(x_1, u_1), \ldots, x_{N+1} = f(x_N, u_N)).$$

Under a sensor-mapping h, an external system trajectory $x_1 \diamond \tilde{u}$ corresponds to a unique action-observation history, that is $(h(x_1), u_1, h(x_2), \ldots, u_N, h(x_{N+1}))$. Note that the inverse is not necessarily true, since the sensor mapping $h : X \to Y$ is not necessarily invertible (it is not one-to-one). Thus, the same action-observation history can lead to different external system trajectories depending on the particular initial state $x_1 \in X$. In this case it may not be possible to determine whether a given history satisfies the task description, as this depends on the particular sensing and actuation setting in relation to the task description given over X. Let $g : \mathcal{I}_{hist} \to \mathrm{pow}(\tilde{X})$ be a function that maps a history to the corresponding set of possible external system trajectories. Then, whether a history η satisfies a task description given over X can be determined based on whether all $\tilde{x} \in g(\eta)$ satisfy the task description.

Accomplishing an active task requires that a sentence of interest becomes true as a result of the execution of a policy defined over the internal system states, that is, the resulting history η belongs to ${\kappa_{task}}^{-1}(1)$. Let $S_\pi = (\mathcal{I}, Y, \phi, \pi, U)$ and $\mathcal{X}_h = (X, U, f, h, Y)$ be a policy-labeled ITS and an external system, respectively. The corresponding coupled system is $S_\pi \star \mathcal{X}_h = (\mathcal{I} \times X, \phi \star_{\pi,h} f)$. We consider tasks that are defined over finite-length histories for which the satisfaction of sentences can be determined in finite time [1]. We will consider tasks that have

[1] For a discussion on infinitary tasks and how they can be transcribed as tasks over finite-length histories see Section 4.1 in [17].

a termination condition so that once a sentence becomes true, the interaction $S_\pi \star \mathcal{X}_h$ stops resulting in the history $\eta_N \in {\kappa_{task}}^{-1}(1)$ for some N.

Definition 8 (Feasible Policy). A policy $\pi : \mathcal{I} \to U$ defined over the states of $S = (\mathcal{I}, Y, \phi)$ is *feasible* if for all $x_1 \in X$ the coupled system $S_\pi \star \mathcal{X}_h$ initialized at (ι_0, x_1) results in $\eta_N \in {\kappa_{task}}^{-1}(1)$, in which N may depend on (ι_0, x_1).

In the above definition, we have assumed for simplicity that task accomplishment can be achieved for any $x_1 \in X$. This may not be the case in general. In case it is not, a feasible policy should be defined for all $x_1 \in X' \subseteq X$, in which X' is the set of states for which there exists a task-accomplishing history.

3.2 ITSs Sufficient for Feasible Policies

In this section we derive conditions that a state-relabeled ITS $(\mathcal{I}, Y, \phi, \iota_0)$ needs to satisfy in order to support a particular feasible policy π_{hist} described over histories. In achieving this objective, we first define the restriction of a history ITS by a π_{hist} which drops the dependency on actions in transitions. Consequently, the quotient system is required to distinguish the actions distinguished by the policy.

Consider the history ITS $(\mathcal{I}_{hist}, U \times Y, \phi_{hist}, ())$ coupled to the external system $\mathcal{X}_h = (X, f, U, h, Y)$. This leads to the set of attainable histories $\mathcal{I}_{hist}^{\mathcal{X}_h}$ and the corresponding labeling function κ_{att}, see Definition 7. Then, any policy $\pi_{hist} : \mathcal{I}_{hist}^{\mathcal{X}_h} \to U$ defined on $\mathcal{I}_{hist}^{\mathcal{X}_h}$ further restricts the set of all histories to the subset containing those that are achieved by applying it. Due to π_{hist}, at each history I-state η, only a single action is possible. We will call this the *restriction of* $\mathcal{I}_{hist}^{\mathcal{X}_h}$ *by* π_{hist}, denoted by $\mathcal{I}_{hist}^{\mathcal{X}_h}\restriction\pi_{hist}$, that is,

$$\mathcal{I}_{hist}^{\mathcal{X}_h}\restriction\pi_{hist} := \left\{\eta^\frown(u, y) \in \mathcal{I}_{hist}^{\mathcal{X}_h} \mid (\eta, y) \in \mathcal{I}_{hist}^{\mathcal{X}_h} \times Y \wedge u = \pi_{hist}(\eta)\right\}. \quad (1)$$

Notice that π_{hist} also restricts the transitions to only those that can be realized following this policy. Let

$${\phi_{hist}}' := \left\{(\eta, (u, y), \eta') \in \phi_{hist} \mid \eta' \notin \mathcal{I}_{hist}^{\mathcal{X}_h}\restriction\pi_{hist}\right\} \quad (2)$$

be the set of transitions that cannot be realized following π_{hist}.[2] Then, the set of transitions achievable under π_{hist} is simply the set difference

$$\phi_{hist}\restriction_{\pi_{hist}} := \phi_{hist} \setminus {\phi_{hist}}'. \quad (3)$$

Notice that $\phi_{hist}\restriction_{\pi_{hist}}$ describes a function with domain $\mathcal{I}_{hist}^{\mathcal{X}_h}\restriction\pi_{hist} \cup {\kappa_{att}}^{-1}(0)$.

Let $U^\xi := U \cup \{\xi\}$, in which ξ serves as a dummy label indicating that an I-state is not attainable. We encode this by defining a labeling function $\kappa_\pi : \mathcal{I}_{hist}^{\mathcal{X}_h}\restriction\pi_{hist} \cup {\kappa_{att}}^{-1}(0) \to U^\xi$ via

$$\kappa_\pi(\eta) := \begin{cases} \pi_{hist}(\eta) & \text{if } \kappa_{att}(\eta) = 1 \\ \xi & \text{otherwise.} \end{cases} \quad (4)$$

[2] For notational convenience, we treat ϕ_{hist} in (2) as a subset of $\mathcal{I}_{hist} \times (U \times Y) \times \mathcal{I}_{hist}$.

Note that κ_π encodes both the labeling κ_{att}, which distinguishes unattainable histories, and the policy π_{hist}, which distinguishes histories in terms of actions to take.

Lemma 2. *Let $(\eta,(u,y),\eta'),(\eta,(u',y),\eta'') \in \phi_{hist}\restriction_{\pi_{hist}}$. Then $\kappa_\pi(\eta') = \kappa_\pi(\eta'')$.*

Proof. Suppose $\eta \in {\kappa_{att}}^{-1}(1)$. Then $u = u' = \pi_{hist}(\eta)$ and since for all y there is a unique η' by the construction of $\phi_{hist}\restriction_{\pi_{hist}}$ (see Eq.(3)) it follows that $\eta' = \eta''$. Suppose $\eta \in {\kappa_{att}}^{-1}(0)$. Then, for any $(u,y) \in U \times Y$, the resulting η', η'' satisfy $\eta', \eta'' \in {\kappa_{att}}^{-1}(0)$ due to Lemma 1. Therefore, $\kappa_\pi(\eta') = \kappa_\pi(\eta'') = \xi$. ☐

Let $\kappa_{\tilde{Y}} : \mathcal{I}_{hist} \to Y^{<\mathbb{N}}$ be an I-map that maps each action-observation history $\eta = (y_1, u_1, \ldots, u_{N-1}, y_N)$ to the corresponding observation history $\tilde{y} = (y_1, \ldots, y_N)$.

Lemma 3. *The I-map $\kappa_{\tilde{Y}}$ with its domain restricted to $\mathcal{I}^{\mathcal{X}_h}_{hist}\restriction\pi_{hist} \cup {\kappa_{att}}^{-1}(0)$ is a refinement of κ_π, that is, $\kappa_{\tilde{Y}} \succeq \kappa_\pi$.*

Proof. Let $[\tilde{y}]_{\kappa_{\tilde{Y}}}$ be an equivalence class induced by $\kappa_{\tilde{Y}}$. By Lemma 2, the preimage of ${\kappa_{\tilde{Y}}}^{-1}(\tilde{y})$ is either a singleton or it satisfies $\kappa_\pi(\eta) = \xi$ for all $\eta \in {\kappa_{\tilde{Y}}}^{-1}(\tilde{y})$. This proves that for all $A \in \mathcal{I}^{\mathcal{X}_h}_{hist}\restriction\pi_{hist} \cup {\kappa_{att}}^{-1}(0)/\kappa_{\tilde{Y}}$ there exists a $B \in \mathcal{I}^{\mathcal{X}_h}_{hist}\restriction\pi_{hist} \cup {\kappa_{att}}^{-1}(0)/\kappa_\pi$ such that $A \subseteq B$. ☐

Thanks to Lemma 3, we can define the restriction of the history ITS by π_{hist} as a quotient of history ITS by $\kappa_{\tilde{Y}}$, that is, $S_{hist}/\kappa_{\tilde{Y}}$, together with a labeling function $\pi : Y^{<\mathbb{N}} \to U^\xi$ which is defined through κ_π. Furthermore, due to Lemmas 2 and 3, the I-state transitions need to depend only on elements of Y. Therefore, we will define the set of transitions $\phi_{\tilde{Y}}$ by taking the projection of ${\phi_{hist}}'$ onto $Y^{<\mathbb{N}} \times Y \times Y^{<\mathbb{N}}$.

Definition 9 (Restriction of history ITS by π_{hist}). The restriction of a history ITS by π_{hist} is the state-relabeled ITS, $\mathcal{S}_{hist}\restriction_{\pi_{hist}} = (Y^{<\mathbb{N}}, Y, \phi_{\tilde{Y}}, (), \pi, U^\xi)$ such that under $\pi : Y^{<\mathbb{N}} \to U^\xi$, $\tilde{y} \mapsto \kappa_\pi(\eta)$, in which $\eta \in {\kappa_Y}^{-1}(\tilde{y})$. Note that for all $\eta, \eta' \in {\kappa_{\tilde{Y}}}^{-1}(\tilde{y})$, $\kappa_\pi(\eta) = \kappa_\pi(\eta')$ (Lemma 2).

Lemma 4. *The ITS corresponding to the restriction of the history ITS by π_{hist}, that is, $(Y^{<\mathbb{N}}, Y, \phi_{\tilde{Y}})$ is full*[3]*.*

Let $\Pi((\mathcal{I}, Y, \phi), \mu) : Y^{<\mathbb{N}} \to U^\xi$ be a function in which $(\mathcal{I}, Y, \phi)$ is an ITS and $\mu : \mathcal{I} \to U^\xi$ is a labeling function. Given an input sequence of some length N, that is, $\tilde{y} = (y_1, y_2, \ldots, y_N)$, $\tilde{y} \mapsto u_N$ under $\Pi((\mathcal{I}, Y, \phi), \mu)$, in which, u_N is the last element of the respective output sequence $\tilde{u} = (\mu(\iota_1), \mu(\iota_2), \ldots, \mu(\iota_N))$ with $\iota_i = \phi(\iota_{i-1}, y_i)$ for $i = 1, \ldots, N$.

Definition 10 (Supports π_{hist}). Let $\left(Y^{<\mathbb{N}}, Y, \phi_{\tilde{Y}}, (), \pi, U^\xi\right)$ be the restriction of the history ITS $(\mathcal{I}_{hist}, U \times Y, \phi_{hist}, ())$ by a policy π_{hist}. Let $S = (\mathcal{I}, Y, \phi)$ be an ITS. S supports π_{hist} if there exists $\mu : \mathcal{I} \to U$ such that $\Pi(\mathcal{I}, Y, \phi, \mu) = \pi$.

[3] A transition system (S, Λ, T) is called *full*, if $\forall s \in S, \lambda \in \Lambda$ there exists at least one $s' \in S$ with $(s, \lambda, s') \in T$.

The following theorem establishes a necessary and sufficient condition that an ITS $(\mathcal{I}, Y, \phi, \iota_0)$ needs to satisfy in order to support the feasible policy π_{hist}.

Theorem 1. *Let $\left(Y^{<\mathbb{N}}, Y, \phi_{\tilde{Y}}, (), \pi, U^{\xi}\right)$ be the restriction of the history ITS by a feasible policy π_{hist}. An ITS $S = (\mathcal{I}, Y, \phi, \iota_0)$ supports π_{hist} if and only if S is the quotient of $\left(Y^{<\mathbb{N}}, Y, \phi_{\tilde{Y}}, ()\right)$ by some sufficient κ satisfying $\kappa \succeq \pi$.*

Proof. $\Longrightarrow$ **direction (If $\kappa \not\succeq \pi$, then there does not exist a μ):** Suppose κ is not a refinement of π. Then, there exist $\iota \in \mathcal{I}$ and $u \in U^{\xi}$ such that $\kappa^{-1}(\iota) \setminus \pi^{-1}(u) \neq \emptyset$. This implies that there exist $\tilde{y}$ and $\tilde{y}'$ such that $\kappa(\tilde{y}) = \kappa(\tilde{y}')$ and $\pi(\tilde{y}) \neq \pi(\tilde{y}')$. Then there does not exists a μ that satisfies $\Pi((\mathcal{I}, Y, \phi), \mu) = \pi$ since input sequences $\tilde{y}$ and $\tilde{y}'$ cannot be distinguished by κ.

$\Longleftarrow$ **direction (If $\kappa \succeq \pi$, then there exists a μ):** We will prove this by construction. Since κ is a refinement of π, every set in $Y^{<\mathbb{N}}/\kappa$ is a subset of $Y^{<\mathbb{N}}/\pi$ which implies that for all $\iota \in \mathcal{I}$ it is true that for each $\tilde{y}, \tilde{y}' \in \kappa^{-1}(\iota)$, $\pi(\tilde{y}) = \pi(\tilde{y}')$. Then, there exists a function $\mu : \mathcal{I} \rightarrow U^{\xi}$ such that $\mu(\kappa(\tilde{y})) = \mu(\kappa(\tilde{y}'))$ since $\pi(\tilde{y}) = \pi(\tilde{y}')$ for all $\tilde{y}, \tilde{y}' \in \kappa^{-1}(\iota)$. □

Clearly, $\kappa = \pi_{hist}$ satisfies this condition. However, in most cases it is not computationally feasible to define policies over entire histories. Therefore, we look for a minimal κ that satisfies this condition. This corresponds to finding a minimal sufficient refinement of π that gives out the minimal ITS that can support π_{hist} (see also Theorem 3 in [17]). This is stated in the following result.

Corollary 1. *Let $\left(Y^{<\mathbb{N}}, Y, \phi_Y, (), \pi, U^{\xi}\right)$ be the restriction $S_{hist}\restriction_{\pi_{hist}}$ and let $\bar{\pi}$ be a minimal sufficient refinement of π. A minimal ITS that supports π_{hist} is the quotient of $(Y^{<\mathbb{N}}, Y, \phi_{\tilde{Y}}, ())$ by $\bar{\pi}$. Furthermore, this minimal ITS is unique.*

Proof. Uniqueness follows from Theorem 4.19 in [24] which states that the minimal sufficient refinement of a labeling function defined over the states of a transition system is unique if the transition system is full. By Lemma 4, $S_{hist}\restriction_{\pi_{hist}}$ is full. □

Corollary 2. *Suppose that π_{hist} is a feasible policy and $(\mathcal{I}, Y, \phi, \iota_0, \mu, U^{\xi})$ is an ITS that supports π_{hist}. Then, the labeling function $\mu : \mathcal{I} \rightarrow U^{\xi}$ is a feasible policy as defined in Definition 8.*

The following example illustrates the introduced concepts.

Example 1 (Skewed Tetromino Environment). Consider the Skewed Tetromino environment given in Figure 2(a). The state space is $X = \{x_1, x_2, x_3, x_4\}$ with a sensor mapping $h(x) = 1$ for $x = x_4$ and $h(x) = 0$, otherwise. The action space $U = \{u_1, u_2, u_3\}$ such that u_1, u_2, u_3 correspond to stopping, moving one cell up and moving one cell to the right, respectively. The task is defined as reaching the state x_4 which corresponds to obtaining $y_k = 1$ for some k. Figure 2(b) shows the labeling determined by κ_π that distinguishes unattainable histories and histories labeled by π_{hist}. From κ_π we obtain the restriction of history ITS, that is, $\mathcal{S}_{hist}\restriction_{\pi_{hist}} = (Y^{<\mathbb{N}}, Y, \phi_Y, (), \pi, U^{\xi})$. Notice that π is not sufficient.

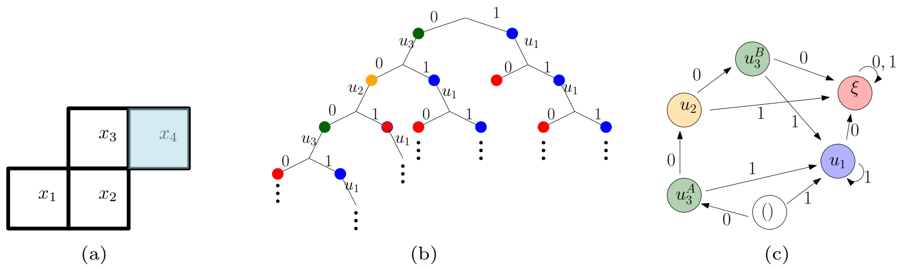

Fig. 2. (a) Skewed tetromino environment. (b)Labels attributed by κ_π with labels u_1 (blue), u_2 (orange), u_3 (green), and ξ (red). (c) S is the quotient of $S_{hist}\restriction_{\pi_{hist}}$ by κ.

Consider observation histories $\tilde{y} = (0)$ and $\tilde{y}' = (0, 0, 0)$ which satisfy $\pi(\tilde{y}) = \pi(\tilde{y}') = u_3$, however, $\pi(\tilde{y}\frown 0) \neq \pi(\tilde{y}'\frown 0)$. Figure 2(c) shows the quotient $S = (\mathcal{I}, Y, \phi, \iota_0)$ of $\mathcal{S}_{hist}\restriction_{\pi_{hist}}$ by κ which is the minimal sufficient refinement of π. This is the minimal ITS that can support π_{hist}. There exists a policy μ with

$$\mu(\iota) = \begin{cases} u_3 & \text{if } \iota \in \{u_3^A, u_3^B\} \\ \iota & \text{otherwise,} \end{cases} \tag{5}$$

such that $S_\mu \star \mathcal{X}_h$ accomplishes the task.

Let $\mathcal{I}_{ndet} \subseteq \text{pow}(X)$ be a nondeterministic I-space and consider the ITS $S^{ndet} = (\mathcal{I}_{ndet}, Y, \phi_{ndet}, X)$. Let $\hat{X}(A, y) = \{f(x, \mu'(A)) \mid x \in A\}$ and define $\phi_{ndet}(A, y) := \hat{X}(A, y) \cap h^{-1}(y)$. Furthermore, define

$$\mu'(\iota) = \begin{cases} () & \text{if } \iota = X \\ u_1 & \text{if } \iota = \{x_4\} \\ u_2 & \text{if } \iota = \{x_2\} \\ u_3 & \text{otherwise.} \end{cases}$$

Proposition 1. *S^{ndet} is isomorphic to S defined in Example 1.*

Proof. There exists a mapping ψ that maps the states of S^{ndet} to those of S such that $X \mapsto ()$, $\{x_1, x_2, x_3\} \mapsto u_3^A$, $\{x_2\} \mapsto u_2$, $\{x_3\} \mapsto u_3^B$, and $\emptyset \mapsto \xi$. Then, by checking the definitions, for any $A, B \in \mathcal{I}_{ndet}$ and any $\iota, \iota' \in \mathcal{I}$ if $\psi(A) = \iota$ and $\psi(B) = \iota'$ it is also true that $\phi(\iota) = \iota'$ implies $\phi_{ndet}(A) = B$.

From Proposition 1, it can be deduced that S^{ndet} is a minimally sufficient ITS that can support π_{hist} which takes the external system state to x_4. This is related to the *Good Regulator Theorem* [5] which roughly states that for any policy to regulate the system state to a target, the internal system should encode a model of the external system.

3.3 Multiple Policies

So far we have considered a single task and looked for an ITS that can support a feasible policy for that task. However, typically, robots are expected to achieve multiple tasks. Therefore, in this section we will focus on ITSs that can support a set of feasible policies satisfying a set of tasks.

Let $\mathcal{T} = \{\kappa_{task}^{T_i}\}_{i=1,\ldots,N}$ be a set of N number of tasks such that each task T_i induces a labeling function $\kappa_{task}^{T_i}$. Then, for each task there exists a feasible policy $\pi_{hist}^{T_i} : \mathcal{I}_{hist} \to U^{\xi}$ defined over the history I-space.

Theorem 2. *Let $S_{hist}\restriction_{\pi_{hist}^{T_i}} = (Y^{<\mathbb{N}}, Y, \phi_{\tilde{Y}}, (), \pi_i, U^{\xi})$ be the restriction of history ITS by a feasible policy $\pi_{hist}^{T_i}$ for $i = 1, \ldots, N$. Let $\pi^{\mathcal{T}}$ be the join (least upper bound) of $\{\pi_i\}_{i=1,\ldots,N}$. An ITS $S = (\mathcal{I}, Y, \phi, \iota_0)$ supports $\pi_{hist}^{T_i}$ for all $i = 1, \ldots, N$, if and only if S is the quotient of $(Y^{<\mathbb{N}}, Y, \phi_{\tilde{Y}}, ())$ by some sufficient κ satisfying $\kappa \succeq \pi^{\mathcal{T}}$.*

Proof. The set of policies defined over $Y^{<\mathbb{N}}$ forms a lattice. Hence $\pi^{\mathcal{T}}$ exists and is unique. Because $\pi^{\mathcal{T}}$ is the join of $\{\pi_i\}_{i=1,\ldots,N}$, $\kappa \succeq \pi^{\mathcal{T}}$ implies $\kappa \succeq \pi_i$ for any $i = 1, \ldots, N$. The rest of the proof follows from Theorem 1. □

3.4 Sufficient Sensors for Reactive Policies

In the previous section, we fixed the sensor mapping $h : X \to Y$ and looked for an ITS that can support a particular feasible policy defined over the action-observation histories. In this section, we fix the form of the ITS and its dependence on the sensor mapping h, and leave h free. This implies that the task-induced labeling κ_{task} also depends on the selected sensor mapping.

Let $\mathcal{H}$ be the set of all sensor mappings, or equivalently, the set of all partitions of X. In the following, Y is seen as the set of labels attributed to the subsets forming the partition (see [12]). Therefore, the particular range of h does not carry any importance as long as it induces the same partitioning. We fix the ITS to be of the form $S = (Y, Y, \mathrm{id}_Y, \iota_0)$, in which id_Y is the identity function, and fix the policy $\pi_Y : Y \to U$ leaving Y free, that is, leaving h free. This results in a set of ITSs and respective reactive policies characterized by the sensor mapping h. With reactive policy we mean a policy that maps the observation received to the action. Let $S_{\pi_Y} = (Y, Y, \mathrm{id}_Y, \iota_0, \pi_Y, U)$ and $\mathcal{X}_h = (X, U, f, h, Y)$ be an ITS labeled with the policy π_Y and an external system with a sensor mapping h, respectively. Their coupling $S_{\pi_Y} \star \mathcal{X}_h$ initialized at (ι_0, x_1) results in the following action-observation history

$$\eta_N = (h(x_1), (u_1, h(x_2)), \ldots (u_{N-1}, h(x_N))),$$

in which each $x_{i+1} = f(x_i, u_i)$ and $u_i = \pi_Y \circ \mathrm{id}_Y \circ h(x_i)$.

Definition 11 (Sufficient sensor for a reactive policy). A sensor mapping h is called *sufficient for a reactive policy* if any initialization (ι_0, x_1) of $S_{\pi_Y} \star \mathcal{X}_h$ results in a history $\eta_N \in {\kappa_{task}}^{-1}(1)$.

Consider the strongest possible sensor $h_{bij} : X \to Y$, i.e. a bijection. For notational simplicity, without loss of generality, we will assume $Y = X$ so that h_{bij} is the identity function. In this case, the history I-space corresponds to $\mathcal{I}_{hist} = (X \times U)^{<\mathbb{N}}$. Let π_{hist} be a feasible policy and let $\mathcal{I}_{hist}^{\mathcal{X}_h} \restriction \pi_{hist}$ be the restriction of $\mathcal{I}_{hist}^{\mathcal{X}_h}$ by π_{hist} as defined in Eqn.1. From π_{hist}, we define a new policy $\pi_{\tilde{X}} : \tilde{X} \to U$ by setting

$$\pi_{\tilde{X}}(\tilde{x}) := \pi_{hist}({\kappa_{\tilde{Y}}}^{-1}(\tilde{x})). \tag{6}$$

Due to Lemmas 2 and 3, $\kappa_{\tilde{Y}}$ is invertible for the domain $\mathcal{I}_{hist}^{\mathcal{X}_h} \restriction \pi_{hist}$ so that ${\kappa_{\tilde{Y}}}^{-1}(\tilde{x})$ is unique. In the case of a bijective sensor mapping, $\pi_{\tilde{X}}$ is a policy that maps $(x_1, \ldots, x_N)$ to some u_N.

Let $\kappa_Y : \tilde{Y} \to Y$ be an I-map such that $\kappa_Y(\tilde{y}) = y_N$, in which $\tilde{y} = (y_1, \ldots, y_N)$ is an observation sequence of some length N. Considering a bijective sensor, under κ_Y, $\tilde{x} \mapsto x_N$, in which $\tilde{x} = (x_1, \ldots, x_N)$ for some N.

Lemma 5. *There exists a feasible policy $\pi_X : X \to U$ if and only if there exists a feasible π_{hist} satisfying the following condition C1:*

$$\text{(C1)} \quad \pi_{\tilde{X}}(\tilde{x}) = \pi_{\tilde{X}}(\tilde{x}') \text{ for all } \tilde{x}, \tilde{x}' \in {\kappa_Y}^{-1}(x).$$

Proof. $\Longleftarrow$ **direction (Existence of π_{hist} satisfying C1 implies existence of π_X):** Suppose there exists a feasible π_{hist} which satisfies C1. Then, we can define a $\pi_X : X \to U$ as the policy $\pi_X = \pi_{\tilde{X}}(\tilde{x})$, in which $\tilde{x} \in {\kappa_Y}^{-1}(x)$, since by C1, $\pi_{\tilde{X}}(\tilde{x}) = \pi_{\tilde{X}}(\tilde{x}')$ for all $\tilde{x}, \tilde{x}' \in {\kappa_Y}^{-1}(x)$. Furthermore, since $\pi_{\tilde{X}}$ is feasible π_X is also feasible. $\Longrightarrow$ **direction (Existence of π_X implies existence of π_{hist} satisfying C1)** Suppose π_X exists. This means that for any $x \in X$ the coupled system $(X \times X, \mathrm{id}_X *_{\pi_X,h} f)$ initialized at (ι_0, x) will result in a task accomplishing history such that $\eta_N = (x_1, \pi_X(x_1), f(x_1, \pi_X(x1)), \ldots, x_N) \in$ for some N. Then there exists a feasible π_{hist} such that $\pi_{hist}(\eta_{k-1}) = u_k$ with $u_k = \pi_X(\kappa_{\tilde{Y}}(\kappa_Y(\eta_{k-1})))$. This proves that π_{hist} satisfies $\pi_{\tilde{X}}(\tilde{x}) = \pi_{\tilde{X}}(\tilde{x}')$. $\square$

We now derive a necessary and sufficient condition that a sensor mapping h needs to satisfy so that there exists a reactive policy π_Y satisfying $\pi_Y \circ h = \pi_X$ for some feasible π_X. We omit the proof since it is similar to the proof of Theorem 1.

Theorem 3. *Suppose there exists a π_X. A sensor h is sufficient for a reactive policy if and only if $h \succeq \pi_X$ for some π_X.*

Corollary 3. *A minimal sensor h for a reactive policy π_X satisfies $h = \pi_X$.*

The following is a direct consequence of Theorem 3 and relates to the existence result of action-based sensors when there are no crossovers (multiple actions applied at the same state) in the plan [14].

Corollary 4. *If there is no π_X then there is no reactive policy.*

4 Minimal structures for distance-optimal navigation

In this section, we apply the results from the previous sections considering the distance optimal navigation in a connected polygon to characterize sufficient structures for optimal policies.

Let $X \subseteq \mathbb{R}^2$ be a connected polygon. The external system state at time t is the point $x(t) = (q_x(t), q_y(t)) \in X$. We assume a point robot with system dynamics

$$\begin{aligned} \dot{q}_x(t) &= \sin(\theta(t)) \\ \dot{q}_y(t) &= \cos(\theta(t)), \end{aligned} \tag{7}$$

in which $\theta(t) \in S^1$ is the control input that indicates the direction under constant unit speed. Given a pair of initial and final points $x_I, x_G \in X$, an optimal action trajectory $\theta^* : [0,T] \to S^1$ is the one that minimizes the cost function $J = \int_0^T dt$ and satisfies the differential equation (7) with initial condition $x(0) = x_I$ and final time condition $x(T) = x_G$ with T being free. Note that under unit speed, T matches the path length. An optimal action trajectory θ^* is one that satisfies

$$\frac{G^*}{\partial x_1} f_1(x, \theta^*) + \frac{G^*}{\partial x_2} f_2(x, \theta^*) = -1, \tag{8}$$

in which $G^* : X \to \mathbb{R}_+$ is the optimal cost-to-go function, i.e. the length of the shortest path to x_G. The left side of Eqn. 8 indicates the derivative of the optimal cost-to-go function along the direction obtained when an optimal action is applied at x.

Let $\mathcal{V}$ be the set of polygon vertices and let U be a set of actions (similar to motion primitives) such that each $u \in U$ corresponds to applying constant control $\bar{\theta} \in S^1$ for a finite length of time until a point $x \in \mathcal{V}$ is reached. We first show that U is sufficient to represent the shortest path between any $(x_I, x_G) \in X \times \mathcal{V}$. Therefore, it allows an event-based discretization such that an optimal plan (policy) can be expressed in a stage-evolving fashion.

Lemma 6. *The optimal control trajectory θ^* resulting in the shortest path to x_G from an initial state x_I is obtained by applying a finite sequence of actions $(u_1, u_2, \ldots, u_N) \in U^N$ for some N.*

Proof. The cost-to-go function G^* is piecewise quadratic with level curves that are circular arcs [9]. Discontinuities in the gradient direction manifest only at the vertices of the polygon, of which there are finitely many. Then, for any x, there exist $s \in \mathcal{V}$ and $\bar{\theta} \in S^1$ which is the direction of the gradient $\nabla_x G^*$ evaluated at x such that forward integrating Eqn.(7) with u which is determined by s and $\bar{\theta}$, the direction of $\nabla_x G^*$ stays constant along the respective state trajectory. Then, any optimal path can be obtained by applying a sequence of elements of U. □

Note that this result also follows from the fact that the shortest path in a polygonal environment is a sequence of bitangent edges.

Lemma 7. *Suppose P is simply connected. Then, for every pair of $x_I, x_G \in P$, there exists a unique sequence $\tilde{u} = (u_1, \dots, u_N)$ for some N resulting in the shortest path connecting x_I and x_G.*

We will consider the gap sensor introduced in [20] and show that the gap navigation tree as an ITS introduced therein is sufficient for supporting a set of optimal policies and it is also minimal.

Definition 12 (Gap sensor [20]). Let Y be the set of all cyclic sequences. A gap is defined as a discontinuity in the distance to the boundary of X measured at x. A *gap sensor* $h : X \to Y$ reports gaps as the cyclic order $h(x) = [g_1, g_2, \dots, g_N]$, in which each g_i, $i = 1, \dots, N$, is a gap.

Lemma 8. *For a path $\sigma : [0,1] \to X$ there exists a finite number of intervals $\{[t_i, t_{i+1}]\}_{i=1,\dots,N-1}$ with $t_0 = 0$ and $t_N = 1$ such that $h(x) = h(x')$ for all $x, x' \in [t_i, t_{i+1}]$ and all $i \in \{1, \dots, N-1\}$ [20].*

Proposition 2. *The gap sensor is not sufficient for a reactive policy.*

Proof. A sensor h is called sufficient for a reactive policy if it satisfies $h \succeq \pi_X$ for some task accomplishing $\pi_X : X \to U$. Considering a simply connected polygon, there exists a unique policy π_X which results in distance optimal navigation. Preimages of π_X partition X into an uncountable set of line segments emanating from a subset of polygon vertices. Then there exist x, x' with $h(x) = h(x')$ such that $\pi_X(x) \neq \pi_X(x')$. This implies $h \not\succeq \pi_X$. Therefore h is not sufficient for a reactive policy. □

Thanks to Lemmas 6 and 8, a policy π^{υ}_{hist} resulting in the shortest path to $x_G \in \mathcal{V}$ can be described over $\mathcal{I}_{hist} = (U \times Y)^{<\mathbb{N}}$ considering the history ITS. Let $\mathcal{X}_h = (X, U, f, h, Y)$ be the external system such that $(x, u) \mapsto \upsilon \in \mathcal{V}$ under f. Let S_{hist} be the history ITS. We will consider multiple policies that correspond to distance optimal navigation to any point $\upsilon \in \mathcal{V}$. The set of such policies is $\{\pi^{\upsilon}_{hist}\}_{\upsilon \in \mathcal{V}}$. Each π^{υ}_{hist}, once executed, corresponds to the shortest path from an initial state $x_I \in X$ to $\upsilon \in \mathcal{V}$.

Gap Navigation Trees (GNT) are proposed in [22], [20] as minimal structures for visibility related tasks. A GNT is constructed by exploring a connected planar environment, through split and merge operations. Once constructed, it encodes a portion of the shortest-map graph rooted at the current position. In this section, we will consider its use in distance optimal navigation in a simply connected polygon. The following describes a GNT constructed for an environment as an ITS used for distance-optimal navigation.

Lemma 9 (GNT as an ITS). *A Gap Navigation Tree (GNT) is an ITS with $S_{GNT} = (\mathcal{I}_{tree}, Y, \phi_{GNT}, \iota_0)$ in which $\mathcal{I}_{tree}$ is a tree, Y is the set of observations of a gap sensor, and ϕ_{GNT} is defined through the appearance or disappearance of gaps (critical events) which correspond to observations, each of which results in a new tree ι_{i+1} that is obtained from ι_i, by changing the root.*

Proposition 3. *$S_{GNT} = (\mathcal{I}_{tree}, Y, \phi_{GNT}, \iota_0)$ supports π^{υ}_{hist} for all $\upsilon \in \mathcal{V}$.*

Sketch Proof. Let $S_{hist}\restriction_{\pi^{\upsilon}_{hist}} = (Y^{<\mathbb{N}}, Y, \phi_{\tilde{Y}}, (), \pi_{\upsilon}, U^{\xi})$ be the restriction of the history ITS by π^{υ}_{hist}. Let $\pi^{\mathcal{V}}$ be the least upper bound of $\{\pi_{\upsilon}\}_{\upsilon \in \mathcal{V}}$. We need to show that there exists $\kappa : Y^{<\mathbb{N}} \to \mathcal{I}_{tree}$ and satisfies $\kappa \succeq \pi^{\mathcal{V}}$. Existence of a κ follows from Lemma 9, such that each $Y^{<\mathbb{N}}$ is mapped to $\iota \in \mathcal{I}_{tree}$ by recursively applying ϕ_{GNT} starting from ι_0. Note that domain of κ for deriving S_{GNT} is restricted to attainable histories. However, since all the unattainable ones would be labeled with ξ we will ignore this aspect. Each π_{υ} distinguishes observation sequences that would result from moving through an optimal sequence of vertices. It was shown in [20] that following a gap results in the disappearance of the gap which happens when a vertex is reached. Therefore, κ distinguishes the same observation histories as $\pi^{\mathcal{V}}$.

Corollary 5. *$S_{GNT} = (\mathcal{I}_{tree}, Y, \phi_{GNT}, \iota_0)$ is the minimal ITS for distance optimal navigation using a gap sensor.*

5 Discussion

We considered solving active tasks, which requires determining an ITS and a respective policy. To this end, we have analyzed ITS and policy pairs, fixing either the particular sensor or the class of ITS characterized by a sensor mapping. For both cases, we have established the necessary and sufficient conditions for such structures to satisfy so that an ITS can support a feasible policy. These results were then applied to analyzing minimally sufficient structures for distance optimal navigation in the plane.

The results open up new avenues for research. In particular, the conditions for an ITS to satisfy is established in conjunction with a particular policy determined over histories. It is an interesting direction to characterize all such policies which in turn, would result in a characterization of all pairs of ITSs and respective policies. It would be interesting to define an ordering (total or partial) over these pairs so that depending on different design objectives, a preference can be made.

In Section 4, the selected actions corresponded to motion primitives. Considering the particular motion primitives, which were defined as a function of polygon vertices, a reactive sensor can be defined which establishes equivalence classes such that states at which the same direction is applied towards the same vertex belong to the same equivalence class. However, if we had defined motion primitives as time (distance) parameterized functions such that each $u_i : [0, \tau_i] \to \bar{\theta}_i \in U$, then the resulting reactive sensor would need to distinguish also the points according to their distance to a particular vertex. This shows that there is an interesting trade-off when it comes to determining motion primitives. We leave this interplay as an interesting future work.

Selection of sensors and motion primitives is also related to the definability and complexity of the task over histories. Defining a measure for task complexity is also an interesting future direction.

References

1. Agha-Mohammadi, A.A., Chakravorty, S., Amato, N.M.: FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements. The International Journal of Robotics Research **33**(2), 268–304 (2014)
2. Başar, T., Olsder, G.J.: Dynamic Noncooperative Game Theory, 2nd Edition. Society for Industrial and Applied Mathematics (1998). https://doi.org/10.1137/1.9781611971132, https://epubs.siam.org/doi/abs/10.1137/1.9781611971132
3. Belta, C., Bicchi, A., Egerstedt, M., Frazzoli, E., Klavins, E., Pappas, G.J.: Symbolic planning and control of robot motion [grand challenges of robotics]. IEEE Robotics & Automation Magazine **14**(1), 61–70 (2007)
4. Blum, M., Kozen, D.: On the power of the compass (or, why mazes are easier to search than graphs). In: Proceedings Annual Symposium on Foundations of Computer Science. pp. 132–142 (1978)
5. Conant, R.C., Ashby, W.R.: Every good regulator of a system must be a model of that system. International Journal of Systems Science **1**(2), 89–97 (1970)
6. Erdmann, M.A., Mason, M.T.: An exploration of sensorless manipulation **4**(4), 369–379 (Aug 1988)
7. Erdmann, M.: Understanding action and sensing by designing action-based sensors. The International journal of robotics research **14**(5), 483–509 (1995)
8. Fainekos, G.E., Girard, A., Kress-Gazit, H., Pappas, G.J.: Temporal logic motion planning for dynamic mobile robots. Automatica **45**(2), 343–352 (2009)
9. Hershberger, J., Suri, S.: Efficient computation of euclidean shortest paths in the plane. In: Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science. pp. 508–517. IEEE (1993)
10. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains **101** (1998)
11. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge, U.K. (2006), also available at http://lavalle.pl/planning/
12. LaValle, S.M.: Sensing and Filtering: A Fresh Perspective Based on Preimages and Information Spaces, Foundations and Trends in Robotics Series, vol. 1:4. Now Publishers, Delft, The Netherlands (2012)
13. Majumdar, A., Tedrake, R.: Funnel libraries for real-time robust feedback motion planning. The International Journal of Robotics Research **36**(8), 947–982 (2017)
14. McFassel, G., Shell, D.A.: Reactivity and statefulness: Action-based sensors, plans, and necessary state. The International Journal of Robotics Research **42**(6), 385–411 (2023)
15. Ross, S., Pineau, J., Paquet, S., Chaib-Draa, B.: Online planning algorithms for POMDPs. Journal of Artificial Intelligence Research **32**, 663–704 (2008)
16. Saberifar, F.Z., Ghasemlou, S., Shell, D.A., O'Kane, J.M.: Toward a language-theoretic foundation for planning and filtering. The International Journal of Robotics Research **38**(2-3), 236–259 (2019). https://doi.org/10.1177/0278364918801503, https://doi.org/10.1177/0278364918801503
17. Sakcak, B., Timperi, K.G., Weinstein, V., LaValle, S.M.: A mathematical characterization of minimally sufficient robot brains. The International Journal of Robotics Research p. 02783649231198898 (2023)
18. Sakcak, B., Weinstein, V., LaValle, S.M.: The limits of learning and planning: Minimal sufficient information transition systems. In: 2022 International Workshop on the Algorithmic Foundations of Robotics (WAFR) (2022)

19. Subramanian, J., Sinha, A., Seraj, R., Mahajan, A.: Approximate information state for approximate planning and reinforcement learning in partially observed systems. Journal of Machine Learning Research **23**(12), 1–83 (2022)
20. Tovar, B., Cohen, F., LaValle, S.M.: Sensor beams, obstacles, and possible paths. In: Proceedings Workshop on Algorithmic Foundations of Robotics (WAFR) (2008)
21. Tovar, B., Murrieta-Cid, R., LaValle, S.M.: Distance-optimal navigation in an unknown environment without sensing distances. IEEE Transactions on Robotics **23**(3), 506–518 (Jun 2007)
22. Tovar, B., Guilamo, L., LaValle, S.M.: Gap navigation trees: Minimal representation for visibility-based tasks. In: Algorithmic Foundations of Robotics VI, pp. 425–440. Springer (2005)
23. Vitus, M.P., Tomlin, C.J.: Closed-loop belief space planning for linear, gaussian systems. In: 2011 IEEE International Conference on Robotics and Automation. pp. 2152–2159. IEEE (2011)
24. Weinstein, V., Sakcak, B., LaValle, S.M.: An enactivist-inspired mathematical model of cognition. Frontiers in Neurorobotics **16** (2022). https://doi.org/10.3389/fnbot.2022.846982, https://www.frontiersin.org/articles/10.3389/fnbot.2022.846982
25. Zhang, Y., Shell, D.A.: Abstractions for computing all robotic sensors that suffice to solve a planning problem. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 8469–8475 (2020). https://doi.org/10.1109/ICRA40945.2020.9196812
26. Zhu, H., Alonso-Mora, J.: Chance-constrained collision avoidance for MAVs in dynamic environments. IEEE Robotics and Automation Letters **4**(2), 776–783 (2019). https://doi.org/10.1109/LRA.2019.2893494