 doi:10.20944/preprints202211.0511.v1

Article

Escaping Local Minima Via Appraisal Driven Responses

Malte Damgaard [1,*], Rasmus Pedersen [1] and Thomas Bak[1]

[1] Department of Electronic Systems, Automation and Control, Aalborg University, Denmark
* Correspondence: mrd@es.aau.dk

Abstract: Inspired by the reflective and deliberative control mechanisms used in cognitive architec-tures such as SOAR and Sigma, we propose an alternative decision mechanism driven by architectural appraisals allowing robots to overcome impasses. The presented work builds on and improves on our previous work on a generally applicable decision mechanism with roots in the Standard Model of the Mind and the Generalized Cognitive Hour-glass Model. The proposed decision mechanism provides automatic context-dependent switching between exploration-oriented, goal-oriented, and backtracking behavior, allowing a robot to overcome impasses. A simulation study of two applica-tions utilizing the proposed decision mechanism is presented demonstrating the applicability of the proposed decision mechanism.

Keywords: Cognitive Robotics; Cognitive Architecture; Appraisals; Reflective Control; Deliberate Control; Reactive Control; Variational Inference; Deadlocks; Probabilistic Programming Idiom; The Standard Model of the Mind

1. Introduction

Robotic technology has immense potential to change our daily life. In the industry, human-robot co-working is envisioned to play a key role in the next industrial revolution known as Industry 5.0 [1]. In healthcare robots also see increasing usage e.g. in personalized healthcare for providing assistance to patients, and the elderly [2,3], and during the COVID-19 pandemic, robots were deployed to disinfect common spaces, such as supermarkets and hospitals [4]. Common to the above is the increased need for autonomous robots that can safely and naturally interact with humans while solving different abstractly and/or vaguely defined tasks. Due to the uncertainties in such problems, pure goal-driven problem-solving architectures will often end up in local minima in the problem formulation also known as impasses. I.e., situations where the information or action selection strategy currently available to the robot is insufficient to solve the task. Thus, one core faculty of such robotic systems should be the ability to reflect on the current situation to timely deviate from one action selection strategy to try out other strategies or to retrieve new information about the task.

The next generation of cognitive architectures, based on modern machine learning techniques, has the potential to revolutionize robotics by allowing roboticists to develop such autonomous systems easily. In previous work, we proposed the Generalized Cognitive Hour-glass Model constituting a framework for developing cognitive architectures by composing them from generally applicable probabilistic programming idioms over which powerful general algorithms can perform inference [5]. The idiomatic approach to composing cognitive architectures, encouraged by this framework, allows researchers and practitioners to more easily cooperate by mixing and matching probabilistic programming idioms developed by others while being able to handcraft parts of a system for which current solutions do not suffice.

In another work, we proposed one such probabilistic programming idiom based on the "standard model of the mind" [6] for the task of Active Knowledge Search (AKS) in

 doi:10.20944/preprints202211.0511.v1

unknown environments[7]. This idiom defines a probabilistic decision process that encourages a robot to take actions to discover, i.e. obtain information about, its environment based purely on notions of progress and information gain while avoiding constraint violations. Simulations applying this idiom to the specific problem of active mapping and robot exploration showed promising results. However, limitations were also identified. The main limitation was that in specific situations the simulated robot would get "stuck" taking repetitive actions yielding no new information about the environment, thus hindering full exploration of the environment. As we will discuss in more detail in Section 4, this is essentially caused by the fixed strategy for action selection employed by the previous solution.

In the literature related to robot navigation, similar phenomena are commonly known as "the local minima issue" [8], "deadlocks" [9], "limit cycles" [10], "infinite loops"[11], "dead ends", "cyclic dead ends", or "trap-situations" [12]. Like the problem mentioned above, all of these terms refer to situations in which a fixed strategy for action selection results in no meaningful progress towards a goal state or compared to a measure of optimality. To resolve these situations solutions proposed by researchers within robotics usually rely on problem-specific information, e.g. geometric properties, to detect and/or resolve the impasse. As an example consider the approach used in [13] where a grid map is defined over the workspace with a counter attached to each of the cells keeping track of the number of times a given cell has been visited. Whenever this counter reaches a predefined threshold, it is registered as a limit cycle. When a limit cycle is detected a temporary waypoint is generated, guiding the robot out of the enclosure causing the limit cycle. Finally, when the robot gets outside the enclosure, a virtual wall is generated, ensuring that the robot does not enter the problematic enclosure again. As another example consider the approach used in [14] where deadlock loops are detected based on the periodicity of the distance to the goal. Whenever a deadlock loop is detected, the distance to the goal is stored, and a wall-following behavior is initiated until an escape condition has been met. Similarly, in [15], deadlocks are detected based on a preferred velocity magnitude, the actual velocity magnitude, and the unsigned distance between robots. Whenever a deadlock is detected a deadlock resolution strategy is initiated. While the solutions suggested above might work for specific problems, they do not easily generalize to other problems.

In the literature related to cognitive architectures, similar phenomena in which an agent is unable to make progress with the information that is currently available are often referred to as impasses [16,17]. As we will elaborate upon in Section 2 research in cognitive architectures is focused on generally applicable solutions opposite to the problem-specific solutions commonly proposed in the robotics literature. Nevertheless, solutions seem to follow the same pattern as those proposed by researchers in robotics. First, systems are made able to detect impasses. Secondly, systems are induced with some sort of reflective mechanism that based on the detected impasse can choose appropriate temporary decision strategies until the impasse has been resolved. However, as we will also elaborate upon in Section 2 the approaches taken by some of the most prominent cognitive architectures, SOAR[16] and Sigma[17], usually requires that controls are abstracted to symbolic representations that have to be decoded by an extra module external to the decision process. Thus, the fine level of control needed within robotics cannot be accounted for as an intrinsic part of the decision process.

For these reasons, our intention with this paper is to present our recent efforts toward implementing general reflective mechanisms similar to the ones found in cognitive architectures within the scope of the framework proposed in [5] in a way that is suitable for robotic applications. The main contributions of this paper are:

- A description and implementation of a control structure grounded in stochastic variational inference that is capable of deliberate and reflective control based on architectural appraisals.
- A demonstration of how such a control structure overcomes the limitations of the probabilistic programming idiom previously proposed in [7].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2022 doi:10.20944/preprints202211.0511.v1

- A demonstration of how such general control structure compares to problem-specific approaches commonly used in robotics.
- A discussion of the time complexity of the proposed control structure.

This paper is organized as follows: Section 2 reviews how some of the most prominent cognitive architectures have tackled impasse phenomena. Section 3 introduces the notation used within this paper. Section 4 shortly describes the previously proposed probabilistic programming idiom in more detail together with the impasse phenomenon observed. Modifications and extensions to the previously proposed idiom are presented in Section 5 and Section 6. Simulation results utilizing the modifications are provided in Section Section 7. Section 8 concludes the paper, and potential future directions are given in this section.

2. Impasses in SOAR and Sigma

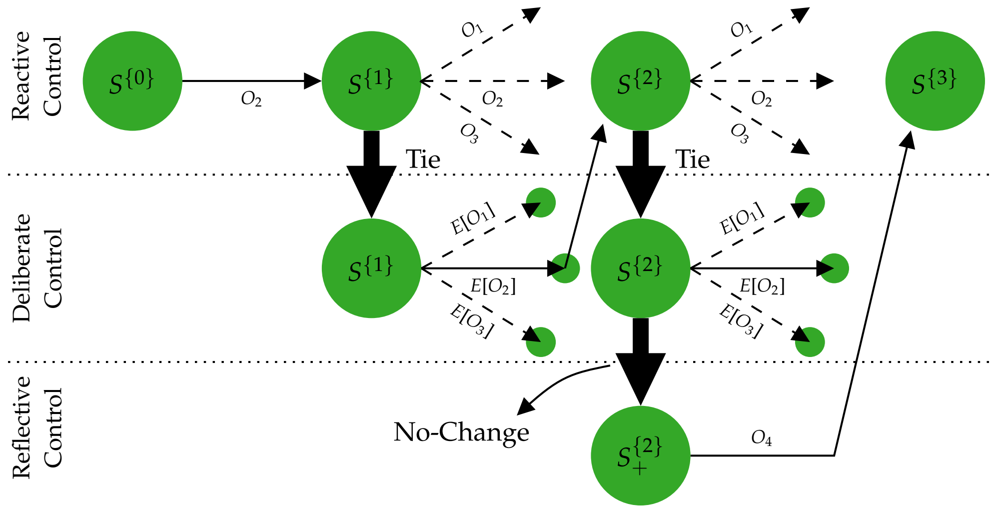


Figure 1. Illustration of the tri-level control structure employed by the two cognitive architectures SOAR[16] and Sigma[17]. For the state at time $t = 0$, S^0, only a single operator is relevant and a *reactive control* mechanism thus chooses to effectuate this operator which leads to a new state, S^1. In this new state, S^1, multiple operators, O_1, O_2, O_3, are proposed by the *reactive control* mechanism. Since no preference exists for these operators a tie impasse is detected, and the *deliberate control* mechanism simulates the expected outcome, $E(O_i)$, from effectuating each of the operators. In this state, S^1, the *deliberate control* mechanism is able to select an operator, O_2, based on preferences for the expected outcome, leading to a new state S^2. In S^2 a tie impasse also occurs, however, this time the expected result of applying any of the proposed operators, O_1, O_2, O_3, results in the next state, S^3, being similar to the current state, i.e. $S^3 = S^2$. Therefore, a no-change impasse is detected, and a *reflective control* mechanism is activated which changes the current state in a way such that a new operator, O_4, can be proposed.

Two of the most prominent cognitive architectures, SOAR[16] and Sigma[17], are both based on the problem space computational model[18], suggesting that problem spaces can be specified in terms of sets of states, S, and operators, O, and that goals can be reached by knowledge search in such problem spaces. However, both architectures acknowledge that direct knowledge search might not always be possible e.g., due to insufficient or ambiguous knowledge. Therefore, these architectures implement a nested tri-level control structure with higher levels activated by the detection of impasses as illustrated in Figure 1 [17]. At the base level, a *reactive control* mechanism proposes operators relevant to the current state based on available knowledge. If only one relevant operator is proposed this operator is effectuated. If multiple relevant operators, $O_1, ..., O_i$ are proposed it is detected as a *tie impasse* and the *deliberative control* mechanism can evaluate the expected outcome resulting from applying each of the proposed operators to the current state. Based on these outcomes the *deliberative control* mechanism might be able to settle on one of the relevant operators to effectuate. By controlling how the expected outcome influences the decision making the *deliberative control* mechanism can form sequential, knowledge-driven, or algorithmic behavior. Finally, whenever the *deliberative control* mechanism cannot pick a unique operator other types of impasses are detected. The impasses detected depend on the reason why no operator could be picked. Based on these impasses a *reflective control* mechanism can take additional actions in order to solve the impasse. The actions taken by the *reflective control* mechanism depend on the specific impasse detected and can include the generation of alternative sub-goals, the inclusion of additional information into the problem space, or even the generation of additional and entirely different problem spaces e.g., to perform meta-reasoning. Table 1 summarizes the different types of impasses detected by the two cognitive architectures SOAR and Sigma.

Table 1. Overview of the different types of impasses detected by the two cognitive architecture SOAR and Sigma.

Impasse	SOAR [16]	Impasse	Sigma [19]
Operator Tie	Occurs when multiple operators are proposed without any preference being able to select between them	**Tie**	Occurs when there are multiple candidate operators, but available knowledge is insufficient to choose among them.
Operator Conflict	Occurs when preferences for two proposed operators, O_1 and O_2, indicates that $O_1 \succ O_2$ and $O_2 \succ O_1$	**No-Change**[1]	Occurs when an operator is selected but no state change results.
Operator No-Change	An operator remains selected for consecutive decision cycles	**None**	Occurs when there are no candidate operators for selection.
State No-Change	No acceptable preferences or every candidate operator also has a reject preference		

[1] According to the description given in [19], however, in [17] it is stated that a no-change impasse occurs "when an operator remains selected for more than one cycle".

To summarize, in these tri-level control structures the *reflective control* mechanism uses the *deliberative control* mechanism as its inner loop which in turn uses the *reactive control* mechanism as its own inner loop. This approach assumes that there a priori exists a discrete/symbolic set of operators that re-actively can be either sorted out or proposed for further evaluation, thereby making detection of the impasses straightforward without any problem-specific knowledge. This approach has some clear benefits with respect to attention, i.e., the effective allocation of limited (computational) resources. Since each of the layers focuses computations on the information actually needed to solve a given problem in a specific context/state a lot of computations are saved. This is especially true when the approach is coupled with Appraisal Theory [20,21], such that the evaluations of operators initiated by the *deliberative control* mechanism are grounded in a limited set of appraisals variables. However, the approach also raises some difficulties in robotics, where a lot of the low-level control is more naturally described by means of continuous variables. As an example, consider the position control of a robot. In SOAR and Sigma, such controls are usually abstracted to symbolic representations such as "walk towards target object", "run towards target object", "pick up target object" or "walk towards random object"[17]. These symbolic representations then have to be decoded by an extra module external to the decision process, called the *motor buffer*, before they can be manifested in the environment. This layer of abstraction makes it hard to incorporate uncertainties resulting from low-level control into the decision process, which in the end will result in less optimal responses being picked.

3. Preliminaries

As in paper [7] we use the following notation. X is used to denote observed variables, Z is used to denote latent variables, and C is used to denote a collection of both types of variables. A superscript in curly brackets is used to indicate the index of a variable. For time indexes, the set of indexes of future variables is indicated as $\{t\}^+ = \{t+1, ..., t+\overline{T}\}$. Similarly, the set of indexes of past variables is indicated as $\{t\}^- = \{t-\underline{T}, ..., t\}$. Furthermore, within this paper the following approximate probabilistic logic

$$
\begin{aligned}
p(z \in \bar{z} \vee y \in \bar{y}) &\stackrel{\text{def}}{=} p(z \in \bar{z}) + p(y \in \bar{y}) - p(z \in \bar{z})p(y \in \bar{y}) \\
p(z \in \bar{z} \wedge y \in \bar{y}) &\stackrel{\text{def}}{=} p(z \in \bar{z}) \cdot p(y \in \bar{y}) \\
p\left(\bigwedge_{i=1}^{I} z^{\{i\}} \in \bar{z}^{\{i\}}\right) &\stackrel{\text{def}}{=} \prod_{i=1}^{I} p\left(z^{\{i\}} \in \bar{z}^{\{i\}}\right)
\end{aligned}
$$

is used, where $\wedge$ and $\vee$ denotes an approximate *and* and *or* operation, respectively. These approximate probabilistic logic rules constitute a probabilistic intersection and union with an independence assumption implied, respectively.

4. Problem elaboration

As stated in Section 1, the probabilistic programming idiom proposed in [7] defines a probabilistic decision process for Active Knowledge Search in unknown environments, based on the "standard model of the mind" [6]. This was done by first defining a probabilistic model relating the previous content of working memory, $Z_{\mathrm{WM}}^{\{t\}^-}$, with the future content, $C_{\mathrm{WM}}^{\{t\}^+}$, while taking variables stored in the long-term memory, Z_{LTM}, into account. In [7], the working memory was further sub-divided into variables relating to motoric actions i.e., the *motor buffer*, Z_{Mb}, variables related to the *perceptual buffer*, Z_{Pb}, *State variables*, Z_{s}, representing the state of the agent itself, the environment, and *decision variables* $C_{\mathrm{D}}^{\{t\}^+}$. From this a probabilistic decision model with the factorization

$$
\begin{aligned}
&p\left(C_{\mathrm{WM}\backslash\mathrm{b}}^{\{t\}^+}, Z_{\mathrm{Mb}}^{\{t-1\}^+}, Z_{\mathrm{Pb}}^{\{t\}^+} | Z_{\mathrm{WM}\backslash\mathrm{b}}^{\{t\}^-}, Z_{\mathrm{LTM}}\right) \\
&\stackrel{def}{=} \left[\prod_{\tau=t+2}^{t+\overline{T}} p\left(C_{\mathrm{D}}^{\{\tau\}} | Z_{\mathrm{s}}^{\{\tau\}}, Z_{\mathrm{WM}\backslash\mathrm{b}}^{\{t\}^-}, Z_{\mathrm{LTM}}\right) p\left(Z_{\mathrm{s}}^{\{\tau\}} | Z_{\mathrm{s}}^{\{\tau-1\}}, Z_{\mathrm{Mb}}^{\{\tau-1\}}\right) p\left(Z_{\mathrm{Mb}}^{\{\tau-1\}}\right)\right] \\
&\cdot p\left(C_{\mathrm{D}}^{\{t+1\}} | Z_{\mathrm{s}}^{\{t+1\}}, Z_{\mathrm{WM}\backslash\mathrm{b}}^{\{t\}^-}, Z_{\mathrm{LTM}}\right) p\left(Z_{\mathrm{s}}^{\{t+1\}} | Z_{\mathrm{s}}^{\{t\}}, Z_{\mathrm{Mb}}^{\{t\}}\right) p\left(Z_{\mathrm{Mb}}^{\{t\}}\right)
\end{aligned}
\tag{1}
$$

were derived, where $Z_{\mathrm{WM}\backslash\mathrm{b}} = Z_{\mathrm{WM}} \backslash \{Z_{\mathrm{Mb}}, Z_{\mathrm{Pb}}\}$. Inspired by the work on emotions in [22], a subset of the decision variables, x_{A}, was denoted *attention variables*. The purpose of these *attention variables* is to control how the decision process is influenced by the other decision variables: *progress*, z_p, *information gain*, z_i, and *constraints*, z_c, hereafter referred to as *appraisal variables*. In [7] this was done via the fixed relation

$$
\begin{aligned}
&p\left(x_A^{\{\tau\}} | Z_{\mathrm{s}}^{\{\tau\}}, Z_{\mathrm{WM}\backslash\mathrm{b}}^{\{t\}^-}, Z_{\mathrm{LTM}}\right) \\
&= Bernoulli\left(p\left(\left[z_{\mathrm{p}}^{\{\tau\}} = 1 \vee z_{\mathrm{i}}^{\{\tau\}} = 1\right] \wedge z_{\mathrm{c}}^{\{\tau\}} = 1 \middle| Z_{\mathrm{s}}^{\{\tau\}}, Z_{\mathrm{WM}\backslash\mathrm{b}}^{\{t\}^-}, Z_{\mathrm{LTM}}\right)\right)
\end{aligned}
\tag{2}
$$

which basically states that during the decision process attention should be given to future states that yield progress *or* new information *and* does not violate constraints. Having defined the model in Equation (1) and the relation in Equation (2), Stochastic Variational Inference was used to approximate the posterior over optimal future motoric actions given the attention variables, i.e.,

$$
q_{\phi_{\mathrm{Mb}}^{\{t-1\}^+,*}}\left(Z_{\mathrm{Mb}}^{\{t-1\}^+}\right) \approx p\left(Z_{\mathrm{Mb}}^{\{t-1\}^+} | x_{\mathrm{A}}^{\{t\}^+} = 1\right) \tag{3}
$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2022 doi:10.20944/preprints202211.0511.v1

The above was implemented as an abstract class utilizing the probabilistic programming language Pyro [23], thereby ensuring that the probabilistic programming idiom can be reused in multiple applications by implementing a few abstract methods defined by the abstract class.

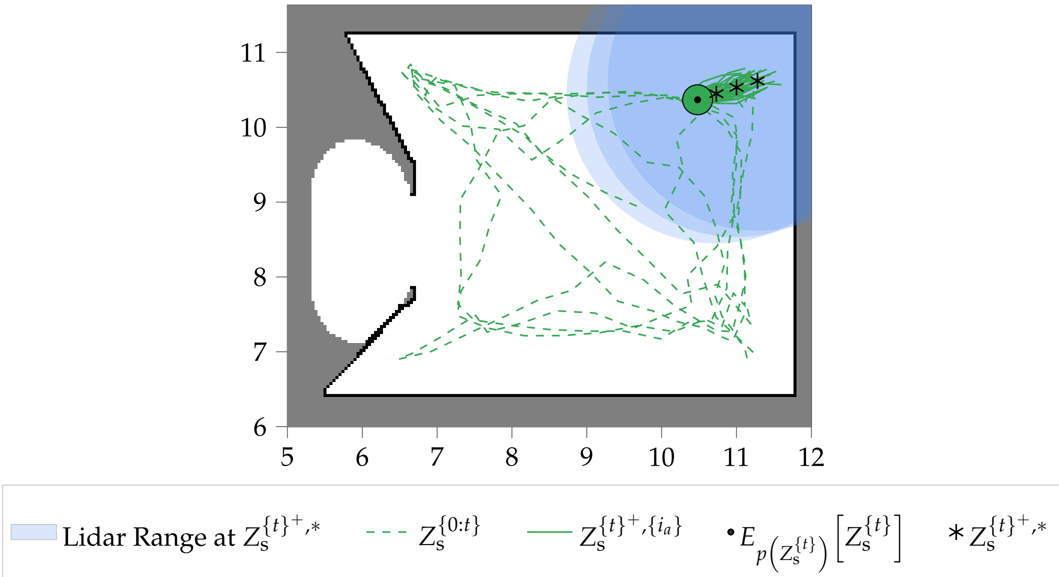


Figure 2. A simulated trajectory of using the method presented in [7] for the floor plan with ID "0a1b29dba355df2ab02630133187bfab" from the HouseExpo dataset [24]. The robot keeps driving around in the same room, without exploring the rest of its environment.

To investigate the performance of the idiom, it was used to implement an algorithm for autonomous robot exploration which was simulated on the full HouseExpo dataset [24], containing 35126 different floor plans. From these simulations, one of the observations was that the robot sometimes would end up taking repetitive actions purely driven by the *progress* appraisal variable. Whereby, the robot would not fully explore its environment as illustrated in Figure 2. In other words, the robot ended up at an impasse. It was further concluded that an alternative to the fixed decision strategy given by Equation (2) would be needed to overcome this problem.

5. Overall Idea

Even though both SOAR and Sigma are said to implement a tri-level control structure, the distinction between the *deliberative control* and the *reflective control* mechanisms seems architectural rather than conceptual. By that, we mean that they both simply comprise a specific architectural response to similar architectural stimuli, i.e., the detection of impasses. When we further consider the statement:

> *"Work in Sigma on appraisal, and its relationship to attention, has led to the conclusion that the detection of impasses should itself be considered as a form of appraisal"[17].*

It hints toward the possibility that similar functionality might be obtained from an architecturally simpler control structure. Based on this, and to overcome the limitations of the approach described in Section 2, we propose an alternative approach centered around appraisals. As illustrated in Figure 3, our proposal is to have a control structure consisting of a single architectural layer with decisions being the result of three main steps:

- Deliberate Attention Proposal
- Deliberate Attention Evaluations
- Affective Responses

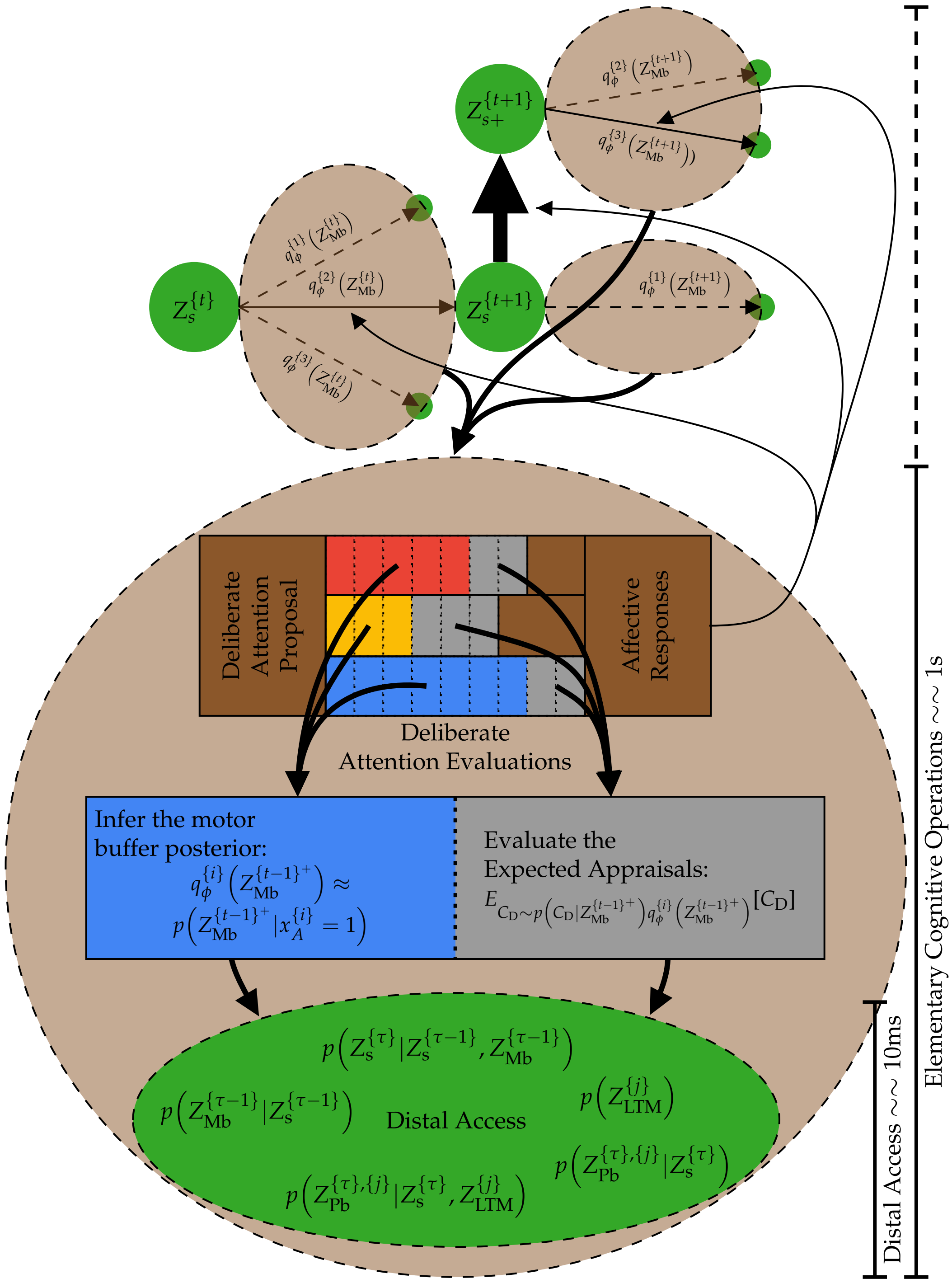


Figure 3. Illustration of the proposed approach with an indication of where we consider each element of the approach to fit into the approximate timescales at which humans make decisions set forward by Allen Newell in [25].

Considering the tri-level control structure described in Section 2 this resembles a combination of the deliberate and reflective mechanism. The main difference is that here attention mechanisms for choosing motoric actions similar to Equation (2) are proposed for evaluation rather than operators for which the outcome is known a priori. Each of the proposed deliberate attention mechanisms might consider a subset of and/or special combinations and weightings of the appraisal variables available to the robot. Thereby, promoting different behaviors. The posterior distributions describing the specific motoric actions corresponding to these proposed deliberate attention mechanisms first become available to the decision process as part of the *Deliberate Attention Evaluations* step. To obtain the posterior over motoric actions the *Deliberate Attention Evaluations* step follows the same steps described in Section 4 for each of the attention mechanisms proposed by the *Deliberate Attention Proposal* step. Besides inferring the motoric action posterior the *Deliberate Attention Evaluations* step also evaluates what the expected appraisals would be from effectuating each of them. Based on the expected appraisals of each of the motoric action posteriors the last step in the decision process can initiate different *affective responses*, such as effectuating one of the action posteriors or proposing additional attention mechanisms to evaluate. Thus, instead of treating the detection of and responses to impasses as distinctive architectural mechanisms, we propose that this is treated as *affective responses* to appraisals evaluated during the *Deliberate Attention Evaluations* step. While the proposed approach conceptually does support deliberate and reflective responses via the *affective responses*, it does not currently have support for reactive responses, since all motoric actions have to be inferred from the deliberate attention mechanisms. However, in Section 8 we will discuss how we imagine that reactive responses could be incorporated into the control structure. Furthermore, modern probabilistic programs such as Pyro [23] can combine stochastic variational inference with enumeration to infer the motoric action posterior. Thereby, making it possible to combine operators represented by both discrete/symbolic and continuous variables in the proposed control structure.

6. Idiom Modifications and Extensions

To test the approach proposed in Section 5 several modifications and extensions were made to the probabilistic programming idiom proposed in [7]. This includes additional appraisal variables, the possibility of adding and using additional deliberate attention mechanisms, together with an implementation of simple mechanisms for *deliberate attention proposal* and *affective responses*. In order to make the implementation reusable in the spirit of the framework presented in [5], all of this is implemented as a series of abstract python classes each constituting a probabilistic programming idiom available at [26].

6.1. Additional Appraisal Variables

Besides the *progress*, z_p, *information gain*, z_i, and *constraints*, z_c, appraisal variables defined in [7], a couple of new appraisal variables has been implemented. The first was the *accummulated constraints* appraisal,

$$\begin{aligned} p\Big(z_{\mathrm{Ac}} = 1 \Big| Z_{\mathrm{s}}^{\{t+1:\overline{T}\}}, Z_{\mathrm{WM}\backslash\mathrm{b}}^{\{t\}^-}, Z_{\mathrm{LTM}}\Big) \\ = Bernoulli\left(p\left(\bigwedge_{\tau=t+1}^{\overline{T}} z_{\mathrm{c}}^{\{\tau\}} = 1 \middle| Z_{\mathrm{s}}^{\{t+1:\overline{T}\}}, Z_{\mathrm{WM}\backslash\mathrm{b}}^{\{t\}^-}, Z_{\mathrm{LTM}}\right)\right), \end{aligned} \tag{4}$$

which was implemented due to a need to check constraint violations of a full state trajectory rather than at a single state. The second originated from a need to be able to define desirable/goal states that a robot should seek to attain. First, we approximate the KL-divergence between a desirable state, Z_{s}^*, and the state, $Z_{\mathrm{s}}^{\{\tau\}}$, after effectuating the motoric action, $Z_{\mathrm{Mb}}^{\{\tau-1\}}$, as

$$\begin{aligned} &D_{\mathrm{KL}}\Big[p(Z_{\mathrm{s}}^*)||p\Big(Z_{\mathrm{s}}^{\{\tau\}}|Z_{\mathrm{s}}^{\{\tau-1\}}, Z_{\mathrm{Mb}}^{\{\tau-1\}}\Big)\Big] \\ &= E_{\hat{Z}_{\mathrm{s}}^{\{\tau\}}}\left[log\left(\frac{p\Big(Z_{\mathrm{s}}^{\{\tau\}} = \hat{Z}_{\mathrm{s}}^{\{\tau\}}|Z_{\mathrm{s}}^{\{\tau-1\}}, Z_{\mathrm{Mb}}^{\{\tau-1\}}\Big)}{p\Big(Z_{\mathrm{s}}^* = \hat{Z}_{\mathrm{s}}^{\{\tau\}}\Big)}\right)\right] \end{aligned}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2022 doi:10.20944/preprints202211.0511.v1

$$
\begin{aligned}
&\approx \frac{1}{I}\sum_{i=1}^{I}\left(log\left(p\left(Z_{\mathrm{s}}^{*} = \hat{Z}_{\mathrm{s}}^{\{\tau\},\{i\}}\right)\right) - log\left(p\left(Z_{\mathrm{s}}^{\{\tau\}} = \hat{Z}_{\mathrm{s}}^{\{\tau\},\{i\}} | Z_{\mathrm{s}}^{\{\tau-1\}}, Z_{\mathrm{Mb}}^{\{\tau-1\}}\right)\right)\right) \\
&\approx ReLu\left(log\left(p\left(Z_{\mathrm{s}}^{*} = \hat{Z}_{\mathrm{s}}^{\{\tau\}}\right)\right) - log\left(p\left(Z_{\mathrm{s}}^{\{\tau\}} = \hat{Z}_{\mathrm{s}}^{\{\tau\}} | Z_{\mathrm{s}}^{\{\tau-1\}}, Z_{\mathrm{Mb}}^{\{\tau-1\}}\right)\right)\right) \\
&\stackrel{\mathrm{def}}{=} D_{Z_{\mathrm{s}}^{*}}\left(\hat{Z}_{\mathrm{s}}^{\{\tau\}}\right).
\end{aligned}
$$

Inspired by the optimality variable defined in [27] we then define the *desirability* appraisal, $z_{\mathrm{d},Z_{\mathrm{s}}^{*}}^{\{\tau\}}$, as

$$
\begin{aligned}
&p\left(z_{\mathrm{d},Z_{\mathrm{s}}^{*}}^{\{\tau\}} = 1 \middle| Z_{\mathrm{s}}^{\{t+1:\bar{T}\}} = \hat{Z}_{\mathrm{s}}^{\{t+1:\bar{T}\}}, Z_{\mathrm{WM\backslash b}}^{\{t\}^{-}}, Z_{\mathrm{LTM}}\right) \\
&\quad = \begin{cases} 0,0 & ;\text{if } p\left(z_{\mathrm{Ac}} = 1 \middle| Z_{\mathrm{s}}^{\{t+1:\bar{T}\}}, Z_{\mathrm{WM\backslash b}}^{\{t\}^{-}}, Z_{\mathrm{LTM}}\right) < 1 \\ Bernoulli\left(-e^{-\sigma_d \cdot D_{Z_{\mathrm{s}}^{*}}\left(\hat{Z}_{\mathrm{s}}^{\{\tau\}}\right)}\right) & ;\ else \end{cases}
\end{aligned}
\tag{5}
$$

where the subscript Z_{s}^{*} in $z_{\mathrm{d},Z_{\mathrm{s}}^{*}}^{\{\tau\}}$ is used to denote the dependency on $p(Z_{\mathrm{s}}^{*})$, and σ_d is a scaling factor. Equation (5) defines a pseudo probability for which states most similar to the desirable state, Z_{s}^{*}, has the highest probability, and states that are less similar have an exponentially lower probability, while states resulting from trajectories that violate constraints have zero probability. The dependency on the *accummulated constraint* appraisal was introduced to aid in overcoming a small probability of constraint violation observed in [7]. For the same reason the *progress*, z_p, and *information gain*, z_i, appraisals has also been modified as follows

$$
\begin{aligned}
&p\left(z_{i}^{\{\tau\}} = 1 \middle| Z_{\mathrm{s}}^{\{t+1:\bar{T}\}}, Z_{\mathrm{WM\backslash b}}^{\{t\}^{-}}, Z_{\mathrm{LTM}}\right) \\
&\quad = \begin{cases} 0,0 & ;\text{if } p\left(z_{\mathrm{Ac}} = 1 \middle| Z_{\mathrm{s}}^{\{t+1:\bar{T}\}}, Z_{\mathrm{WM\backslash b}}^{\{t\}^{-}}, Z_{\mathrm{LTM}}\right) < 1 \\ p\left(z_{\tilde{i}}^{\{\tau\}} = 1 \middle| Z_{\mathrm{s}}^{\{t+1:\bar{T}\}}, Z_{\mathrm{WM\backslash b}}^{\{t\}^{-}}, Z_{\mathrm{LTM}}\right) & ;\ else \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
&p\left(z_{p}^{\{\tau\}} = 1 \middle| Z_{\mathrm{s}}^{\{t+1:\bar{T}\}} = \hat{Z}_{\mathrm{s}}^{\{t+1:\bar{T}\}}, Z_{\mathrm{WM\backslash b}}^{\{t\}^{-}}, Z_{\mathrm{LTM}}\right) \\
&\quad = \begin{cases} 0,0 & ;\text{if } p\left(z_{\mathrm{Ac}} = 1 \middle| Z_{\mathrm{s}}^{\{t+1:\bar{T}\}}, Z_{\mathrm{WM\backslash b}}^{\{t\}^{-}}, Z_{\mathrm{LTM}}\right) < 1 \\ p\left(z_{\tilde{p}}^{\{\tau\}} = 1 \middle| Z_{\mathrm{s}}^{\{t+1:\bar{T}\}}, Z_{\mathrm{WM\backslash b}}^{\{t\}^{-}}, Z_{\mathrm{LTM}}\right) & ;\ else \end{cases}
\end{aligned}
$$

where $z_{\tilde{p}}$ and $z_{\tilde{i}}$ are the *progress*, z_p, and *information gain*, z_i, appraisals as defined in [7].

6.2. Deliberate Attention mechanisms

Based on the appraisals defined in Section 6.1 five different deliberate attention mechanisms have been implemented. All these can be defined as

$$
p\left(x_{A}^{\{\tau\}} | Z_{\mathrm{s}}^{\{t+1:\bar{T}\}}, Z_{\mathrm{WM\backslash b}}^{\{t\}^{-}}, Z_{\mathrm{LTM}}\right) = Bernoulli\left(p\left(\Phi(\tau) \middle| Z_{\mathrm{s}}^{\{t+1:\bar{T}\}}, Z_{\mathrm{WM\backslash b}}^{\{t\}^{-}}, Z_{\mathrm{LTM}}\right)\right)
$$

where $\Phi(\tau)$ defines the logic for combining appraisals as in Table 2. Each of these deliberate attention mechanisms promotes different behaviors.

6.3. Deliberate Attention Proposal And Affective Responses

Based on the *deliberate attention mechanisms* defined in Section 6.2, an *affective response* mechanism has been implemented with the purpose of making a robot effectively explore its environment and possibly navigate towards a goal state, Z_s^{*}, if defined. This *affective response* mechanism can be sub-divided into three parts responsible for different types of behaviors with pseudo-code given in Algorithm A1, Algorithm A2, and Algorithm A3.

Table 2. Definitions of $\Phi(\tau)$ used for each of the implemented deliberate attention mechanisms.

Deliberate Attention Mechanism	$\Phi(\tau)$ for $\tau \in [t+1; \overline{T}-1]$	$\Phi(\overline{T})$ for $\tau = \overline{T}$
ConstraintAvoidance - CA	$z_{\text{Ac}} = 1$	$z_{\text{Ac}} = 1$
StateReach - SR(Z_s^*)	$z_{\text{Ac}} = 1$	$\begin{cases} z_{\text{Ac}} = 1 & ; P(z_{\text{Ac}} = 1) < 1 \\ z_{\text{d},Z_s^*}^{\{\overline{T}\}} = 1 & ; \textit{else} \end{cases}$
StateReachWithProgress - SRP(Z_s^*)	$z_{\text{Ac}} = 1$	$\begin{cases} z_{\text{Ac}} = 1 & ; P(z_{\text{Ac}} = 1) < 1 \\ z_{\text{d},Z_s^*}^{\{\overline{T}\}} = 1 \wedge z_p^{\{\overline{T}\}} = 1 & ; \textit{else} \end{cases}$
StateReachWithExplore - SRE(Z_s^*)	$z_{\text{Ac}} = 1$	$\begin{cases} z_{\text{Ac}} = 1 & ; P(z_{\text{Ac}} = 1) < 1 \\ z_{\text{d},Z_s^*}^{\{\overline{T}\}} = 1 \wedge z_i^{\{\overline{T}\}} = 1 & ; \textit{else} \end{cases}$
Explore - E	$z_{\text{Ac}} = 1$	$\begin{cases} z_{\text{Ac}} = 1 & ; P(z_{\text{Ac}} = 1) < 1 \\ z_i^{\{\overline{T}\}} = 1 & ; \textit{else} \end{cases}$
ExploreWithProgress - EP	$z_{\text{Ac}} = 1$	$\begin{cases} z_{\text{Ac}} = 1 & ; P(z_{\text{Ac}} = 1) < 1 \\ z_i^{\{\overline{T}\}} = 1 \wedge z_p^{\{\overline{T}\}} = 1 & ; \textit{else} \end{cases}$

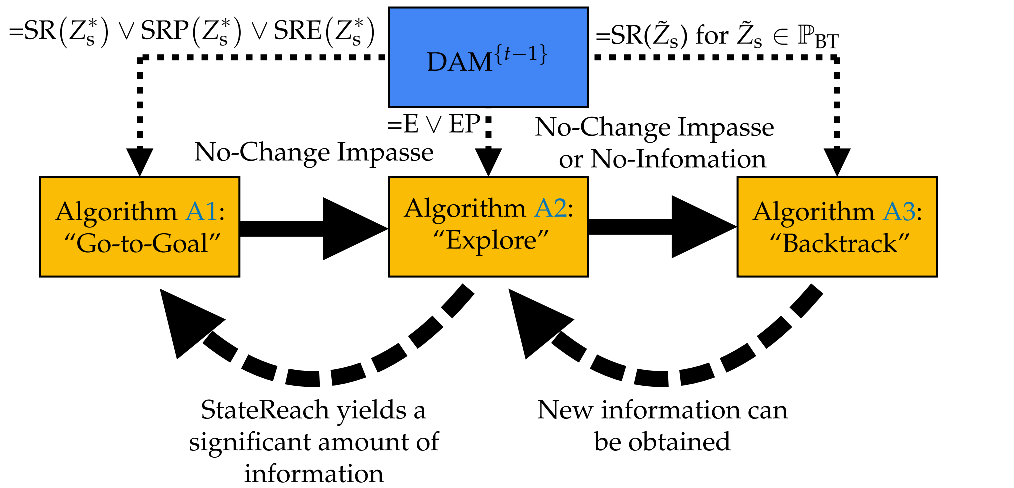


Figure 4. Overview of the implemented affective response mechanism sub-divided into 3 algorithms. Dotted lines indicate what algorithm is activated based on the which *deliberate attention mechanism*, DAM$^{\{t-1\}}$, resulted in the effectuation of a motoric action in the last time step. Solid lines indicate a reflective response resulting in a direct transition between the algorithms. Each direct transition requires a new *deliberate attention proposal* and *deliberate attention evaluation*. A dashed line indicates an indirect transition between the algorithms which takes effect in the next decision cycle. Z_s^* denotes a goal state, and $\mathbb{P}_{\text{BT}}$ denotes a path of previous states to backtrack.

Algorithm A1 yields behavior that strives for the goal state. Algorithm A2 yields behavior that strives to obtain new information about the environment. Finally, Algorithm A3 yields behavior that strives to backtrack. The combined affective response only depends on the appraisals defined in Section 6.1 and [7] which requires no problem-specific information, thereby, making this affective response mechanism general and reusable. As illustrated in Figure 4, each part of the *affective response* mechanism is activated either as a reflective response to another part of the *affective response* mechanism or based on the *deliberate attention mechanism* that caused a motoric action to be effectuated in the last decision cycle. E.g. if the *deliberate attention mechanism* "Explore (E)" caused the effectuation of $q_\phi^{\{E\}}\left(Z_{\text{Mb}}^{\{t-1\}}\right)$ at time $t-1$, then Algorithm A2 will be activated first at time t. In cases where the motoric action was caused by the *deliberate attention mechanism* "ConstraintAvoidance (CA)", the same algorithm is simply activated again. The intuition behind this affective response mechanism is as follows. If a goal state is known and if it is possible to attain it with the current knowledge directly, this should have first priority. If this is not possible new information should be sought after until the goal state can be attained. Finally, if in a state where new information cannot be obtained, the system should be able to bring itself back to a previous state in which new information can be obtained via backtracking. To support this *affective response* mechanism a *deliberate attention proposal* mechanism has been implemented that simply proposes the *deliberate attention mechanisms* required for each part of the *affective response* mechanism. When combined this exemplifies how both deliberate and reflective responses can be implemented grounded in the appraisals defined in Section 6.2. In particular, notice that something similar to the "no-change" impasse in SOAR and Sigma is obtained on the basis of the "Progress" appraisal in both Algorithm A1 and Algorithm A2.

7. Results

To test the effectiveness of the proposed approach two different simulation studies were performed. Both of these were done utilizing the Pseudo-SLAM simulator [24], and using the same implementation of abstract methods for the probabilistic programming idiom that was used in [7]. The exact parameters used for each of these simulations can be found at [26], which also contains scripts to replicate each of the experiments.

7.1. Pure Exploration

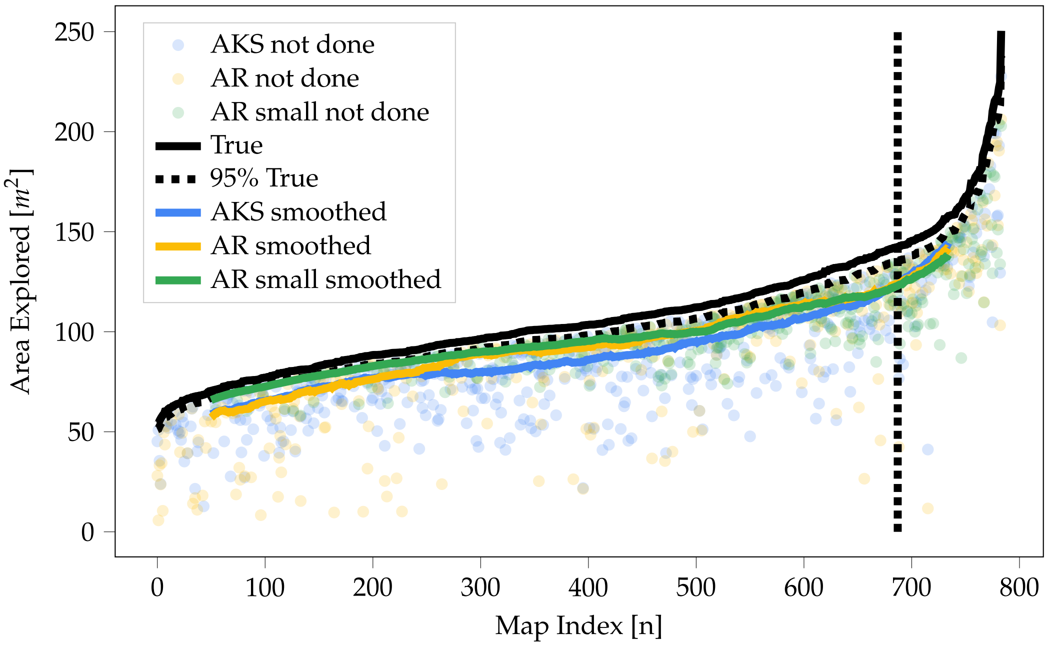


Figure 5. The area that the robot explored in each of the simulations. The indices of the 784 floorplans have been sorted by the true area of the map in ascending order. "AKS" shows results from using the method presented in [7]. "AR" shows results from using the *affective response* mechanism from Section 6.3. The "smoothed" curves show a moving average with a window size of 100 and shifted 50 indexes. The "not done" scatter shows the exact area explored by the simulations in which the robot did not manage to explore 95% of the map or more.

The first simulation study was done in order to compare with results from [7]. In [7] we tested the exploration capability of the proposed algorithm by simulating it in 35126 floor plans from the HouseExpo dataset[24]. However, (1) many of these were so small that

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2022 doi:10.20944/preprints202211.0511.v1

they were fully discovered in a few iterations, and (2) on the other end of the spectrum some of the floor plans were simply too big to be fully discovered within the maximum of 200 time-steps that was allowed in each simulation. Furthermore, (3) the problems of the previous solution discussed in Section 4 are only noticeable in floor plans with more than one room. Additionally, (4) it was found that for some of the floor plans openings between the rooms were physically too small for the robot to squeeze through. Thus, for the purpose of efficiently testing the approach proposed in this paper we selected a smaller subset of the HouseExpo dataset satisfying the following criteria.

1. The floor plans should have a bounding box larger than 100 m^2 to avoid spending time on simulations redundant due to (1).
2. The floor plans should be fully discovered in the experiment from [7], in order to minimize the influence from (2) and (4).
3. The floor plans should contain more than 3 rooms in order to provoke (3).

Based on this, a subset of the HouseExpo dataset consisting of 784 floor plans where selected. Figure 5 and Table 3 show the results of simulating our old approach, "AKS", again as well as the approach proposed within this paper, "AR". The simulations were performed with the same environmental and robot settings used in [7]. As no goal state was specified, only Algorithm A2 and Algorithm A3 were effectively used to drive the behavior of the "AR" method in these simulations. For each floor plan a random initial position where selected, and this initial position were utilized for both simulations. Notice, that even though one of the criteria for the selection of the subset of floor plans was that it should be fully explored by "AKS" in the experiment in [7], Figure 5 indicates that not all floor plans where fully explored by "AKS" in this new round of simulations. This is simply due to a difference in initial positions between simulations and illustrates a lack of robustness of "AKS". From Figure 5 it might seem that "AR" does not perform better than "AKS" for small floor plans. By visual inspection of the simulation trajectories, it was found that the reason for "AR" not being able to explore some floor plans fully was due to a lesser willingness to violate constraints compared to "AKS". This can be verified from the row "Collision pr. Timesteps" in Table 3. This is especially pronounced in small floor plans, where the openings between rooms tend to be smaller. To further verify that the lack of exploration by "AR" is indeed due to its unwillingness to violate constraints, the third series of simulations denoted "AR small" was performed. In these simulations, the size of the robot, the uncertainty in its initial position, and the assumed motion uncertainty was decreased. By changing these parameters, it becomes easier for the robot to take actions through narrow openings without constraint violations. From Figure 5 it is seen that "AR small" fully explores nearly all of the small floor plans, and performs similarly to "AR" for all other maps as expected. As the floor plans get bigger, the ability of all three methods to fully explore the floor plans to a greater extent depends on initial conditions, rather than the ability of the methods to escape local minima. As a result, from Figure 5 it is observed that as the floor plans get larger all methods perform very similarly. Nevertheless, from Table 3 it is evident that "AR" is indeed better for overcoming local minima and making the robot efficiently explore its environment.

7.2. Goal Seeking

The second series of simulations were performed in order to compare the proposed approach with more problem-specific approaches from [8]. To do so three of the test environments from [8] were recreated in the Pseudo-SLAM simulator as illustrated in Figure 6. These three environments are designed specifically with the purpose of causing local minima, and as such are perfect for testing the proposed approach. Since the approach proposed in this paper is based on probabilistic methods, some degree of variations in results should be expected. Therefore, 100 simulations were performed for each of these environments with the same initial conditions and goal state as in [8]. Since a goal state where specified for these simulations, the full capabilities of the *affective response* mechanism described in Section 6.3 were effectively in use. Table 4 summarizes the results from these

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2022 doi:10.20944/preprints202211.0511.v1

Table 3. Comparison of our approach with results for 6 different methods presented in [8].

Metric	AKS	RGS	RGS small
Maps Not Fully Explored	484	331	314
Mean Exploration Percentage	84,9%	87,3%	90,7%
Mean percentage explored for unfinished maps	78,6%	76,7%	84,4%
Maps with Collisions	18	0	0
Collisions	19	0	0
Collision pr. Timesteps	0,14‰	0,00‰	0,00‰

Table 4. The traveled distance in 3 different environments utilizing our approach, AR, compared with results for 6 other methods presented in [8]. As AR is based on probabilistic methods we ran 100 simulations and present the mean of the results together with the minimum and maximum values for each of the environments. The best result for each environment is highlighted with bold text.

Environment	Random[1]	Reflected Virtual Target[1]	Global Path Backtracking[1]	Half Path Backtracking[1]	Local Path Backtracking[1]	Wall-Following[1]	AR
C-shaped	55	**45**	59	101	59	1915	45,34 [**39,38**-73,97]
Double U-shaped	97	88	100	110	96	466	**47,51** [**28,07**-72,56]
V-shaped	38	**27**	29	31	28	111	52,91 [**22,91**-114,98]
Average	63,33	53,33	62,67	80,67	61	830,67	**48,59**

[1] Results from Table 2 in [8].

simulations and compares them to the results from [8]. In all of the simulations, the robot managed to reach the goal state, thereby substantiating the ability of the proposed approach to escaping local minima. From Table 4 it is furthermore seen that the "AR" method is better than any of the problem-specific methods in all three environments when only considering the minimum traveled distance. However, when considering the average distance for the "V-shape" environment it is nearly twice that of the best method from [8], i.e., "Reflected Virtual Target". Considering the first column of Figure 6 it is clear that the robot generally can take the two paths indicated with green and yellow colors. We suspect that the better average performance of the "Reflected Virtual Target" method in the "V-shape" environment is caused by an initial condition that makes the problem-specific methods favor paths similar to the one marked with yellow in Figure 6. The better performance achieved by such preference would not necessarily lead to better performance in general environments/problems, and a more reasonable comparison would probably be obtained by some variations in the initial conditions and/or goal state. As such we do not consider this an inauspicious characteristic of the "AR" approach.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2022 doi:10.20944/preprints202211.0511.v1

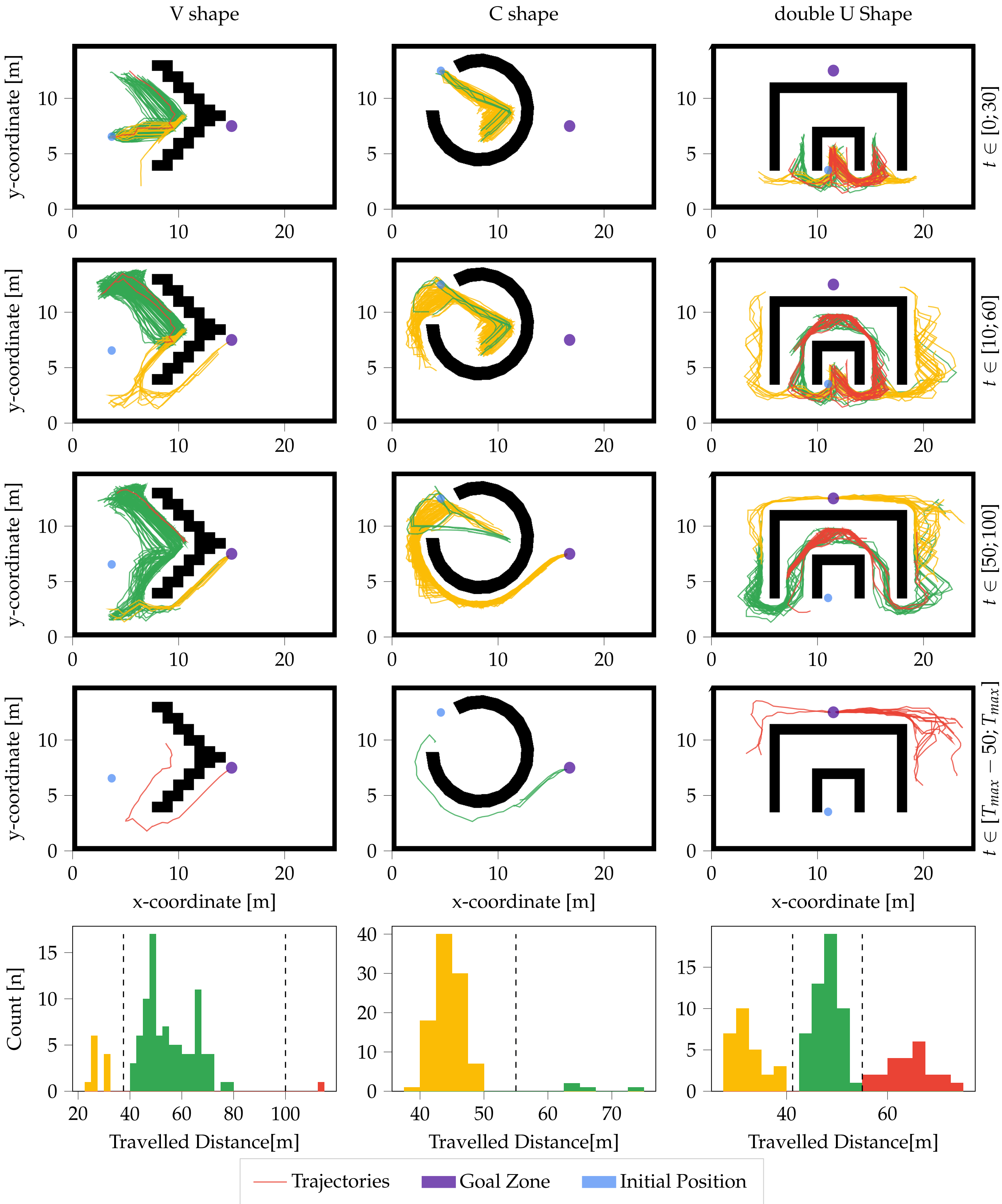
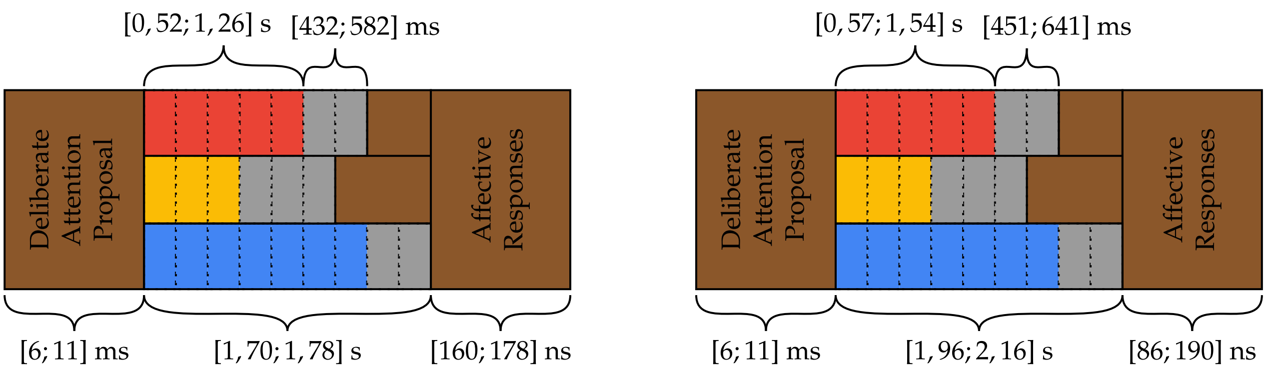


Figure 6. The robot trajectories in each of the 100 simulations for each of the three environments: "V shape", "C shape", and "double U shape". Each of the trajectories is color-coded according to the length of the trajectory. $T_{\max} = 308$, $T_{\max} = 191$, and $T_{\max} = 208$ for "V shape", "C shape", and "double U shape", respectively.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2022 doi:10.20944/preprints202211.0511.v1

7.3. Timings



(a) Apple M1 **(b)** Intel 12600k@5.1GHz

Figure 7. Timings for the main steps of the proposed solution running on two different CPUs, for the deliberate attention mechanism and the affective response mechanism presented in Section 6.2 and Section 6.3, respectively, and for the abstract method implementations specifically used for the simulations in Section 7.1 and Section 7.2.

One of the most critical features of any robotics system is the satisfaction of the real-time constraint, i.e., the ability of the system to make decisions on time scales appropriate to the expected behavior of the system. To investigate the computational time required for the proposed approach the average computation times were measured on two different CPUs. The results can be seen in Figure 7. Notice, that these timings are based on a relatively slow python implementation and are uniquely tied to the specific use-case presented in Section 7.1 and Section 7.2. As such, they should not be seen as the definitive timings that can be obtained utilizing the method, but rather as indicative of roughly what can be expected by the approach. Nevertheless, the timings given in Figure 7 would probably be too slow or jerky for most real-world robot applications. As should be clear from Figure 7 the most time-consuming part of the approach with the current implementation is the inference part of the *deliberate attention evaluations* step. As such, further optimization of this step would be needed to make the approach usable.

8. Discussion

The intention of the presented efforts was to implement general reflective mechanisms suitable for robotic applications with an outset in previous work. In Section 7.1 and Section 7.2 we demonstrate that the proposed method functionally improves upon our previous proposed probabilistic programming idiom and that it at least can perform as well as, if not better than, problem-specific methods. However, in Section 7.3 it was concluded that the current implementation would probably be too slow and jerky for real-world robot applications. The approach presented within this paper is supposed to be generally applicable and reusable, making it hard to assess how much the current implementation should be improved to be applicable to real-world robot applications since this would of course depend on each specific use case. One way to asses this anyway could be by comparing it to Allan Newell's analysis of the time scales of human cognition [25]. This is reasonable because the ultimate end goal of our efforts is to make robots as capable as humans.

In a single cycle of *deliberate attention proposal*, *deliberate attention evaluation* and *affective response*, access to parts of cognition distal to the decision process have to have occurred multiple times in order to infer the motor buffer posteriors. This places the proposed approach somewhere above the "biological band" of Newell's analysis said to be on the order of $\sim$ 10 ms. The next step up in Newell's analysis is to the "cognitive band" starting at the level of *deliberate acts* in the order of $\sim$ 100 ms. However, the proposed approach does not merely comprise deliberation, i.e., choosing one known operation over other known operators by bringing available knowledge to bear, since operators are constructed for the to-be-produced response based on the proposed deliberate attention mechanisms. Therefore, the proposed approach also belongs somewhere above the time scales of *deliberate acts*. At the other end of the "cognitive band", we have *unit tasks* in the order of ~ 10 s. At the time scale of *unit tasks*, operations should be composed to deal with tasks. By design, the specific affective response presented in Section 6.3 can only deliver simple responses in one decision cycle and not a plan of responses to solve complete tasks. This leaves us at the time scales of *elementary cognitive operations* or *immediate external cognitive behavior* at ~ 1 s. According to Newell's analysis, such elementary reactions often take $\sim 2-3$ s, however, with learning from experience, simplification, preparation, and carefully shaped anticipation, it can take less than $\sim 0,5$ s. By design, the specific affective response presented in Section 6.3 can deliver simple responses within 1 to 3 full cycles of *deliberate attention proposal*, *deliberate attention evaluation* and *affective response*. With the timings in Figure 7a a response thus takes anywhere from ~ 1.7 s up to at most ~ 5.34 s in the case of two impasses. Thus, to arrive at the upper end of elementary reactions, the computational times of the current implementation would have to be improved with a factor of $\sim 2-3$. Obtaining such improvements does not seem implausible via code optimizations, however, it brings us nowhere near the lower end of $\sim 0,5$ s. This begs the question: can the proposed approach support the necessary machinery to learn from experience in order to deliver responses at the lower end of $\sim 0,5$ s?

In Section 4 and Section 5 we described the use of stochastic variational inference, as the basis for inferring a parametric approximation of the posterior over the motor buffer, $q_{\phi}^{\{i\}}\left(Z_{\text{Mb}}^{\{t-1\}^+}\right)$. It was assumed that this inference process would have to be done from scratch in each decision cycle. However, this need not necessarily be the case. Instead, we could make use of amortized variational inference [28–31]. Thus, instead of making use of a variational distribution with free parameters, ϕ, we would make use of a variational distribution with parameters determined by a parametric function, $\phi = f_{\phi}^{\{i\}}\left(Z_{\text{WM}\setminus\text{b}}^{\{t\}^-}, Z_{\text{LTM}}\right)$, e.g., a neural network. When new situations are encountered we would not necessarily gain much by doing so, however, over time this would in principle allow the system to generate proper responses to situations similar to those that the system has previously encountered, without performing any inference. Thereby, removing the need for the most time-consuming step in the decision cycle. Again, when considering the timings in Figure 7a, reducing the inference step to near zero, would bring the total time of a single decision cycle down to around ~ 500 ms with the current implementation. Now if it is possible to improve the other steps with a factor of $\sim 2-3$ via code optimizations, it would indeed seem plausible to achieve *immediate external cognitive behavior* in around $\sim 0,5$ s after an initial learning period.

Further optimization might be achieved by considering when to stop the underlying inference algorithm. In the current implementation, the underlying inference algorithm uses a fixed number of iterations that has to be pre-defined. It might not be necessary with the same number of iterations in all situations, and thus time could be saved if a more clever mechanism for deciding the number of iterations could be implemented.

With these additions and optimizations of the approach and its implementation, we believe that the approach will be applicable to real-world robot applications, and thereby contribute to the goal of constructing autonomous robots that can safely and naturally interact with humans while solving different abstractly and/or vaguely defined tasks. As such, these optimizations will be the focus of our future work.

Author Contributions: Conceptualization, M.R.D.; Methodology, M.R.D.; Software, M.R.D.; Validation, M.R.D.; Formal Analysis, M.R.D.; Investigation, M.R.D.; Writing Original Draft, M.R.D.; Writing Review & Editing, M.R.D., R.P., and T.B.; Visualization, M.R.D.; Supervision, R.P., and T.B.

Funding: This research received no external funding

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2022 doi:10.20944/preprints202211.0511.v1

Data Availability Statement: The software used for the simulations is available at [26]. The Github repository also contains configuration files with the specific parameters and settings used for the experiments, as well as scripts to reproduce the two simulation experiments presented in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AKS	Active Knowledge Search
RGS	Reflective Goal Search

Appendix A. Affective Responses

Require: *Deliberate Attention Evaluations* of CA, SR(Z_s^*), SRP(Z_s^*), and SRE(Z_s^*)

```
DAMS ← {SR(Z_s^*), SRP(Z_s^*), SRE(Z_s^*)}
if ∄DAM ∈ DAMS : p(z_p^{DAM} = 1) > P_{z_p,lim} then                # No-change impasse
    Run Algorithm A2                                                 # Reflective Response
else
    repeat
        DAM* ← arg max_{DAM∈DAMS} p(z_d^{DAM} = 1)                   # pick the most desirable DAM
        if p(z_d^{DAM*} = 1) ≥ p(z_d^{SR} = 1) then
            # Only pick DAM* if it is more desirable than SR to avoid motoric
            # actions driven mainly by the progress or information gain appraisals
            if p(z_Ac^{DAM*} = 1) ≥ P_{z_c,lim} or p(z_Ac^{DAM*} = 1) ≥ p(z_Ac^{CA} = 1) then
                effectuate q_φ^{DAM*}(Z_Mb^{t})
            end if
        end if
        DAMS ← DAMS\DAM*
    until DAMS = ∅
    effectuate q_φ^{CA}(Z_Mb^{t})                                    # pick the constraint avoidance
                                                                     #   strategy as a backup
end if
```

Algorithm A1: Affective response for State reach

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2022 doi:10.20944/preprints202211.0511.v1

Require: *Deliberate Attention Evaluations* of CA, E, EP, and SR(Z_s^*) if $Z_s^* \neq None$

DAMS $\leftarrow \{\mathrm{E}, \mathrm{EP}\}$

DAM* $\leftarrow \arg\max_{\mathrm{DAM} \in \mathrm{DAMS}} p\left(z_{\mathrm{i}}^{\{\mathrm{DAM}\}} = 1\right)$ # pick the DAM yielding most information

if $\nexists \mathrm{DAM} \in \mathrm{DAMS}: \quad p\left(z_{\mathrm{p}}^{\{\mathrm{DAM}\}} = 1\right) > P_{z_{\mathrm{p}},\mathrm{lim}}$ # No-change impasse

  or $p\left(z_{\mathrm{i}}^{\{\mathrm{DAM^*}\}} = 1\right) \leq P_{z_{\mathrm{i}},\mathrm{lim}}$ **then** # No information gain possible

  $\mathbb{P}_{\mathrm{BT}} \leftarrow$ set_path_to_backtrack() # Generate the path to backtrack, $\mathbb{P}_{\mathrm{BT}}$, from the state tree

  Run Algorithm A3 # Reflective Response

else

  repeat

    DAM* $\leftarrow \arg\max_{\mathrm{DAM} \in \mathrm{DAMS}} p\left(z_{\mathrm{d}}^{\{\mathrm{DAM}\}} = 1\right)$ # pick the most desirable DAM

    if $Z_s^* \neq None$ **and** $p\left(z_{\mathrm{i}}^{\{\mathrm{SR}(Z_s^*)\}} = 1\right) \geq p\left(z_{\mathrm{i}}^{\{\mathrm{DAM^*}\}} = 1\right) \cdot (1 - P_{z_{\mathrm{i}},\Delta})$ # StateReach gives nearly as much information

      and $p\left(z_{\mathrm{p}}^{\{\mathrm{SR}(Z_s^*)\}} = 1\right) > P_{z_{\mathrm{p}},\mathrm{lim}}$ **then** # and sufficient progress

      DAM* $\leftarrow \mathrm{SR}(Z_s^*)$ # If SR does not violate constraints it will be effectuated and Algorithm A1 will be used in the next iteration.

    end if

    if $p\left(z_{\mathrm{Ac}}^{\{\mathrm{DAM^*}\}} = 1\right) \geq P_{z_{\mathrm{c}},\mathrm{lim}}$ **or** $p\left(z_{\mathrm{Ac}}^{\{\mathrm{DAM^*}\}} = 1\right) \geq p\left(z_{\mathrm{Ac}}^{\{\mathrm{CA}\}} = 1\right)$ **then** # No constraint violation?

      effectuate $q_\phi^{\{DAM*\}}\left(Z_{\mathrm{Mb}}^{\{t\}}\right)$

    else

      DAMS $\leftarrow$ DAMS$\backslash$DAM*

    end if

  until DAMS $\neq \emptyset$

  effectuate $q_\phi^{\{\mathrm{CA}\}}\left(Z_{\mathrm{Mb}}^{\{t\}}\right)$ # pick the constraint avoidance strategy as a backup

end if

Algorithm A2: Affective response for explore

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2022 doi:10.20944/preprints202211.0511.v1

Require: *Deliberate Attention Evaluations* of CA, E, EP, and $\mathrm{SR}(\mathbb{P}_{\mathrm{BT}}[\tau])$ for $\tau \in [1, ..., T_{\mathrm{BT}}]$

$\mathrm{DAM}^* \leftarrow \arg\max_{\mathrm{DAM} \in \{\mathrm{E,EP}\}} p\left(z_{\mathrm{i}}^{\{\mathrm{DAM}\}} = 1\right)$ # pick the DAM yielding most information
if $p\left(z_{\mathrm{i}}^{\{\mathrm{DAM}^*\}} = 1\right) > P_{z_{\mathrm{i}},\mathrm{lim}}$ **and** $p\left(z_{\mathrm{p}}^{\{\mathrm{DAM}^*\}} = 1\right) > P_{z_p,\mathrm{lim}}$ **then** # new information can be obtained
    create_new_state_branch() # add branch to state tree
else
    $\underline{\mathrm{DAMS}} \leftarrow \varnothing$
    for $\mathrm{DAM}^* \in \bigcup_{\tau=0}^{T_{\mathrm{BT}}} \{\mathrm{SR}(\mathbb{P}[T_{\mathrm{BT}} - \tau])\}$ **do** # SR for the first T_{BT} states in $\mathbb{P}_{\mathrm{BT}}$
        $\underline{\mathrm{DAMS}} \leftarrow \mathrm{DAM}^* \cup \underline{\mathrm{DAMS}}$
        if $p\left(z_{\mathrm{d}}^{\{\mathrm{DAM}^*\}} = 1\right) > P_{\mathrm{BT,min}}$ **then** # state can be reached
            if $p\left(z_{\mathrm{Ac}}^{\{\mathrm{DAM}^*\}} = 1\right) \geq P_{z_{\mathrm{c}},\mathrm{lim}}$ # without collision
              or $p\left(z_{\mathrm{Ac}}^{\{\mathrm{DAM}^*\}} = 1\right) \geq p\left(z_{\mathrm{Ac}}^{\{\mathrm{CA}\}} = 1\right)$ **then**
                $\mathbb{P}_{\mathrm{BT}} \leftarrow \mathbb{P}_{\mathrm{BT}} \backslash \underline{\mathrm{DAMS}}$ # remove the reachable states from the path currently being backtracked
                if $\mathbb{P}_{\mathrm{BT}} = \varnothing$ **then** # end of path has been reached
                    $\mathbb{P}_{\mathrm{BT}} \leftarrow$ set_path_to_backtrack() # generate new path to backtrack
                end if
                effectuate $q_{\phi}^{\{DAM*\}}\left(Z_{\mathrm{Mb}}^{\{t\}}\right)$
            end if
        end if
    end for
end if
if $p\left(z_{\mathrm{Ac}}^{\{\mathrm{DAM}^*\}} = 1\right) \geq P_{z_{\mathrm{c}},\mathrm{lim}}$ **or** $p\left(z_{\mathrm{Ac}}^{\{\mathrm{DAM}^*\}} = 1\right) \geq p\left(z_{\mathrm{Ac}}^{\{\mathrm{CA}\}} = 1\right)$ **then** # DAM* is less risky
    effectuate $q_{\phi}^{\{DAM*\}}\left(Z_{\mathrm{Mb}}^{\{t\}}\right)$ # than collision avoidance
else
    effectuate $q_{\phi}^{\{CA\}}\left(Z_{\mathrm{Mb}}^{\{t\}}\right)$
end if

Algorithm A3: Affective response for backtracking

References

1. Demir, K.A.; Döven, G.; Sezen, B. Industry 5.0 and Human-Robot Co-working. *Procedia Computer Science* **2019**, *158*, 688–695. 3rd WORLD CONFERENCE ON TECHNOLOGY, INNOVATION AND ENTREPRENEURSHIP"INDUSTRY 4.0 FOCUSED INNOVATION, TECHNOLOGY, ENTREPRENEURSHIP AND MANUFACTURE" June 21-23, 2019, https://doi.org/https://doi.org/10.1016/j.procs.2019.09.104.
2. Fang, B.; Guo, X.; Wang, Z.; Li, Y.; Elhoseny, M.; Yuan, X. Collaborative task assignment of interconnected, affective robots towards autonomous healthcare assistant. *Future Generation Computer Systems* **2019**, *92*, 241–251. https://doi.org/https://doi.org/10.1016/j.future.2018.09.069.
3. Farid, F.; Elkhodr, M.; Sabrina, F.; Ahamed, F.; Gide, E. A Smart Biometric Identity Management Framework for Personalised IoT and Cloud Computing-Based Healthcare Services. *Sensors* **2021**, *21*. https://doi.org/10.3390/s21020552.
4. Kaiser, M.S.; Al Mamun, S.; Mahmud, M.; Tania, M.H., Healthcare Robots to Combat COVID-19. In *COVID-19: Prediction, Decision-Making, and its Impacts*; Santosh, K.; Joshi, A., Eds.; Springer Singapore: Singapore, 2021; pp. 83–97. https://doi.org/10.1007/978-981-15-9682-7_10.
5. Damgaard, M.R.; Pedersen, R.; Bak, T. Toward an idiomatic framework for cognitive robotics. *Patterns* **2022**, *3*, 100533. https://doi.org/https://doi.org/10.1016/j.patter.2022.100533.
6. Laird, J.E.; Lebiere, C.; Rosenbloom, P.S. A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics. *AI Magazine* **2017**, *38*, 13–26. https://doi.org/10.1609/aimag.v38i4.2744.
7. Damgaard, M.R.; Pedersen, R.; Bak, T. A Probabilistic Programming Idiom for Active Knowledge Search. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), 2022, pp. 1–9. https://doi.org/10.1109/IJCNN55064.2022.9892094.
8. Tashtoush, Y.; Haj-Mahmoud, I.; Darwish, O.; Maabreh, M.; Alsinglawi, B.; Elkhodr, M.; Alsaedi, N. Enhancing Robots Navigation in Internet of Things Indoor Systems. *Computers* **2021**, *10*. https://doi.org/10.3390/computers10110153.
9. Grover, J.S.; Liu, C.; Sycara, K. Deadlock Analysis and Resolution for Multi-robot Systems. In Proceedings of the Algorithmic Foundations of Robotics XIV; LaValle, S.M.; Lin, M.; Ojala, T.; Shell, D.; Yu, J., Eds.; Springer International Publishing: Cham, 2021; pp. 294–312. https://doi.org/10.1007/978-3-030-66723-8_18.
10. Boldrer, M.; Andreetto, M.; Divan, S.; Palopoli, L.; Fontanelli, D. Socially-Aware Reactive Obstacle Avoidance Strategy Based on Limit Cycle. *IEEE Robotics and Automation Letters* **2020**, *5*, 3251–3258. https://doi.org/10.1109/LRA.2020.2976302.
11. Krishna, K.M.; Kalra, P.K. Solving the local minima problem for a mobile robot by classification of spatio-temporal sensory sequences. *Journal of Robotic Systems* **2000**, *17*, 549–564. https://doi.org/https://doi.org/10.1002/1097-4563(200010)17:10<549::AID-ROB3>3.0.CO;2-#.
12. Mohanty, P.K.; Kodapurath, A.A.; Singh, R.K. A Hybrid Artificial Immune System for Mobile Robot Navigation in Unknown Environments. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* **2020**, *44*, 1619–1631. https://doi.org/10.1007/s40998-020-00314-8.
13. Ordonez, C.; Collins, E.G.; Selekwa, M.F.; Dunlap, D.D. The virtual wall approach to limit cycle avoidance for unmanned ground vehicles. *Robotics and Autonomous Systems* **2008**, *56*, 645–657. https://doi.org/https://doi.org/10.1016/j.robot.2007.11.010.
14. Sanchez, G.M.; Giovanini, L.L. Autonomous navigation with deadlock detection and avoidance. *Inteligencia Artificial* **2014**, *17*, 13 23.
15. Alonso-Mora, J.; DeCastro, J.A.; Raman, V.; Rus, D.; Kress-Gazit, H. Reactive mission and motion planning with deadlock resolution avoiding dynamic obstacles. *Autonomous Robots* **2018**, *42*, 801–824. https://doi.org/10.1007/s10514-017-9665-6.
16. Laird, J.E. *The Soar Cognitive Architecture*; The MIT Press, 2012.
17. Rosenbloom, P.S.; Demski, A.; Ustun, V. The Sigma Cognitive Architecture and System: Towards Functionally Elegant Grand Unification. *Journal of Artificial General Intelligence* **2017**, *7*, 1–103. https://doi.org/doi:10.1515/jagi-2016-0001.
18. Newell, A.; Yost, G.R.; Laird, J.E.; Rosenbloom, P.S.; Altmann, E.G. Formulating the Problem Space Computational Model. *Carnegie Mellon Computer Science : A 25-Year Commemorative* **1991**, pp. 255–293.
19. Pynadath, D.V.; Rosenbloom, P.S.; Marsella, S.C.; Li, L. Modeling Two-Player Games in the Sigma Graphical Cognitive Architecture. In Proceedings of the Artificial General Intelligence; Kühnberger, K.U.; Rudolph, S.; Wang, P., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2013; pp. 98–108.
20. Scherer, K.; Schorr, A.; Johnstone, T. *Appraisal Processes in Emotion: Theory, Methods, Research*; Oup Usa, 2001.
21. Smith, C.A.; Kirby, L.D. Putting appraisal in context: Toward a relational model of appraisal and emotion. *Cognition and Emotion* **2009**, *23*, 1352–1372, [https://doi.org/10.1080/02699930902860386]. https://doi.org/10.1080/02699930902860386.
22. Rosenbloom, P.S.; Gratch, J.; Ustun, V. Towards Emotion in Sigma: From Appraisal to Attention. In Proceedings of the Artificial General Intelligence; Bieger, J.; Goertzel, B.; Potapov, A., Eds.; Springer International Publishing: Cham, 2015; pp. 142–151. https://doi.org/10.1007/978-3-319-21365-1_15.
23. Bingham, E.; Chen, J.P.; Jankowiak, M.; Obermeyer, F.; Pradhan, N.; Karaletsos, T.; Singh, R.; Szerlip, P.A.; Horsfall, P.; Goodman, N.D. Pyro: Deep Universal Probabilistic Programming. *J. Mach. Learn. Res.* **2019**, *20*, 28:1–28:6.
24. Li, T.; Ho, D.; Li, C.; Zhu, D.; Wang, C.; Meng, M.Q.H. HouseExpo: A Large-scale 2D Indoor Layout Dataset for Learning-based Algorithms on Mobile Robots. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 5839–5846. Article in proceedings, https://doi.org/10.1109/IROS45743.2020.9341284.

25. Newell, A. *Unified Theories of Cognition*; Harvard University Press, 1990.
26. Damgaard, M.R. ProbMind. https://github.com/damgaardmr/probMind/tree/ec996e295575c384879b3d72cfc7e64b8085b9a5 (Accessed on 3. November 2022), 2022.
27. Damgaard, M.R.; Pedersen, R.; Bak, T. Study of Variational Inference for Flexible Distributed Probabilistic Robotics. *Robotics* **2022**, *11*. https://doi.org/10.3390/robotics11020038.
28. Zhang, C.; Butepage, J.; Kjellstrom, H.; Mandt, S. Advances in Variational Inference, 2017. https://doi.org/10.48550/ARXIV.1711.05597.
29. Rezende, D.J.; Mohamed, S.; Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In Proceedings of the Proceedings of the 31st International Conference on Machine Learning; Xing, E.P.; Jebara, T., Eds.; PMLR: Bejing, China, 2014; Vol. 32, *Proceedings of Machine Learning Research*, pp. 1278–1286.
30. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings; Bengio, Y.; LeCun, Y., Eds., 2014.
31. Shu, R.; Bui, H.H.; Zhao, S.; Kochenderfer, M.J.; Ermon, S. Amortized Inference Regularization. In Proceedings of the Advances in Neural Information Processing Systems; Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; Garnett, R., Eds. Curran Associates, Inc., 2018, Vol. 31.