Learning Where to Park

Burak Ergul, Thijs van de Laar, Magnus Koudahl, Martin Roa-Villescas, and Bert de Vries

Eindhoven University of Technology, Electrical Eng. Dept.
Groene Loper 19, 5612 AP Eindhoven, NL
corr. bert.de.vries@tue.nl

Abstract. We consider active inference as a novel approach to the design of synthetic autonomous agents. In order to assess active inference's feasibility for real-world applications, we developed an agent that controls a ground-based robot. The agent contains a generative dynamic model for the robot's position and for performance appraisals by an observer of the robot. Our experiments show that the agent is capable of learning the target parking position from the observer's feedback and robustly steer the robot toward the learned target position.

Keywords: active inference · robotics · variational Bayesian learning.

1 Introduction

The idea of autonomously operating synthetic agents is an active research area in the machine learning community. Development of these agents involves a number of hard challenges, for instance the need for agents to be capable of adaptively updating their goals in dynamic real-world settings.

In this project we investigated a novel solution approach to the design of autonomous agents. We recognize that any "intelligent" autonomous agent needs to be minimally capable of realizing three tasks:

- Perception: online tracking of the state of the world.
- Learning: updating its world model in case real-world dynamics are poorly predicted.
- Decision making and control: executing purposeful behavior by taking advantage of its knowledge of the state of the world.

Active Inference (ActInf) is a powerful computational theory of how *biological* agents accomplish the above mentioned task palette. ActInf relies on formulating all tasks (perception, learning and control) as inference tasks in a biased generative model of the agent's sensory inputs [8].

In order to assess the feasibility and capabilities of active inference as a framework for the design of *synthetic* agents in a real-world setting, we develop here an agent for a ground-based robot that learns to navigate to an initially undisclosed location. The agent can only learn where to park through situated interactions with a human observer who is aware of the target location.

2 Problem Statement

In this design study, we are particularly interested in two issues:

1. Can the agent *learn* the correct target position from situated binary appraisals by a human observer?
2. Can the agent robustly *steer* the robot to the inferred target position?

3 Model Specification

Active inference, a corollary of the free energy principle, brings together perception, learning and control in a unifying theory [8]. Active inference agents comprise a biased generative model that encodes assumptions about the causes of the agent's sensory signals. The generative model is biased in the sense that the agent's goals are encoded as priors over future states or observations.

Following [11,12], the agent's model at time step t in this paper takes the form of a state-space model

$$p_t(o, s, u) \propto p(s_{t-1}) \prod_{k=t}^{t+T} \underbrace{p(o_k|s_k)}_{\text{observation}} \underbrace{p(s_k|s_{k-1}, u_k)}_{\text{state transition}} \underbrace{p(u_k)}_{\text{control}} \underbrace{p'(o_k)}_{\text{goal}}, \tag{1}$$

where o, s and u refer to the agent's observations, internal states and control signals respectively. Note that the model includes states and observations for T time steps in the future.

The agent's generative model consists of two interacting sub-models: a physical model for the robot's position and orientation and a target model for user appraisals, see Fig. 1. Initially, the physical model has no explicit goal priors. However, the agent's target model infers desired future locations from appraisals and relays this information to the physical model. Thus, as time progresses, the physical model acquires increasingly accurate information about desired future positions.

3.1 The Physical Model

The physical model is responsible for inferring the controls necessary for navigating the agent from any position a to position b. Observations are noisy samples of the robot's position and orientation. The inferred controls are translation and rotation velocities that are used in a differential steering scheme [7].

The states of the physical model are given by $s_k = (x_k, y_k, \phi_k)$ where (x_k, y_k) specify the (latent) position of the agent and ϕ_k the orientation. Controls are given by $u_k = (\Delta\phi_k, \Delta d_k)$, where $\Delta\phi_k$ specifies rotation velocity and Δd_k specifies translation velocity. The transition dynamics are specified as

$$p(s_k|s_{k-1}, u_k) = \mathcal{N}(s_k \,|\, g(s_{k-1}, u_k), 10^{-1}\mathrm{I}) \tag{2}$$

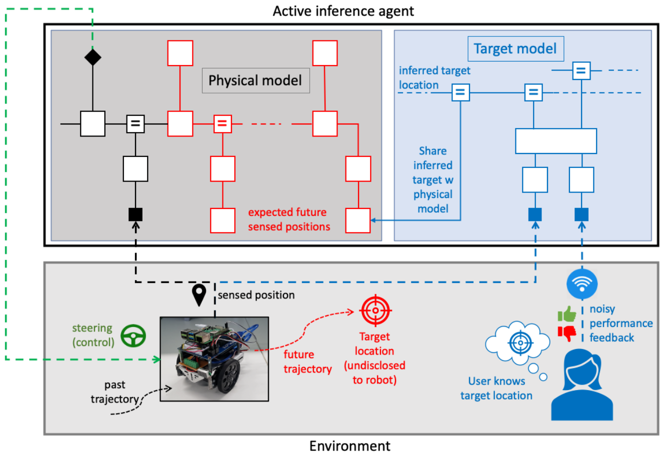


Fig. 1: The information processing architecture of the active inference agent and its environmental interactions. The environment consists of a robot and a human observer that (wirelessly) casts performance appraisals.

where

$$g(s_{k-1}, u_k) = \begin{pmatrix} \phi_{k-1} + \Delta\phi_k \\ x_{k-1} + \Delta d_k \cdot cos(\phi_k) \\ y_{k-1} + \Delta d_k \cdot sin(\phi_k) \end{pmatrix} . \tag{3}$$

In these expressions, $\mathcal{N}(\cdot|m, v)$ is a Normal distribution with mean m and variance v, and I denotes an identity matrix (of appropriate dimension). To couple the observations to internal states, we specify an observation model as

$$p(o_k|s_k) = \mathcal{N}(o_k \,|\, s_k, 10^{-1}\mathrm{I}) \tag{4}$$

We choose the following controls and state priors:

$$p(u_k) = \mathcal{N}(u_k \,|\, [0, 0], 10^{-2}\mathrm{I}) \tag{5a}$$

$$p(s_0) = \mathcal{N}(s_0 \,|\, [0, 0, \pi/2], 10^{-2}\mathrm{I}) \tag{5b}$$

Finally, the goal priors are specified as a prior on observations:

$$p'(o_k) = \mathcal{N}(o_k \,|\, \hat{o}, 10^{-2}\mathrm{I}) \, . \tag{6}$$

where we denote $\hat{o}$ as a "target parameter" of the physical model.

3.2 The Target Model

The target model is responsible for inferring beliefs about the intended target location $\hat{o}$ by observing user feedback. The inferred beliefs about the target location are subsequently used as a prior belief for the physical model's target parameter $\hat{o}$. The idea of learning a goal prior by a second generative model for additional sensory inputs is further explored in [11]. Technically, the target model is a generative model for user appraisals. In order to reason about the target location, the target model will also be aware of the robot's current and previous position.

Specifically, we use a target model at time step t given by

$$p(r_t, b_t, b_{t-1}, \lambda, \hat{o} \,|\, y_t, y_{t-1}) = \underbrace{p(r_t \,|\, b_t, b_{t-1}, \lambda, \hat{o})}_{\text{appraisal}} \cdot \underbrace{p(b_t \,|\, y_t) p(b_{t-1} | y_{t-1})}_{\text{position}} \cdot \underbrace{p(\lambda)}_{\text{precision}} \cdot \underbrace{p(\hat{o})}_{\text{target}} \tag{7}$$

where

$$p(r_t \,|\, \hat{o}, b_t, b_{t-1}, \lambda) = \text{Bernoulli}(r_t \,|\, \sigma(U(b_t, b_{t-1}, \hat{o}, \lambda))) \tag{8a}$$
$$p(b_t \,|\, y_t) = \mathcal{N}(b_t \,|\, y_t, 10\text{I}) \tag{8b}$$
$$p(b_{t-1} \,|\, y_{t-1}) = \mathcal{N}(b_{t-1} \,|\, y_{t-1}, 10\text{I}) \tag{8c}$$
$$p(\lambda) = \mathcal{N}(\lambda \,|\, [2, 2], 5\text{I}) \tag{8d}$$
$$p(\hat{o}) = \mathcal{N}(\hat{o} \,|\, o_0, 100\text{I}) \tag{8e}$$

The model for binary user appraisals uses a "utility" function

$$U(b_t, b_{t-1}, \hat{o}, \lambda) = f(y_t, \hat{o}, \lambda) - f(y_{t-1}, \hat{o}, \lambda) \tag{9}$$

with

$$f(y, \hat{o}, \lambda) = -\sqrt{(y - \hat{o})^T e^{\lambda} (y - \hat{o})} \tag{10}$$

to score the current position y_t to the previous position y_{t-1}, given the current belief over the target $\hat{o}$. λ is a precision parameter governing the width of the utility function. The utility is passed through a sigmoid $\sigma(x) = 1/(1 + e^{-x})$ to parameterize a Bernoulli distribution over binary user appraisals r_t. The user provides appraisals by observing the current and previous positions of the robot. The observed user appraisal is set to 1 if the user thinks that the current robot position is closer to the target than the previous assessment, and otherwise the appraisal is set to 0 (zero). The model was validated in a simulation environment first and later ported to the robot.

The physical model and the target model are linked by drawing a sample from the posterior belief about the intended target location in the target model. This sample is used to parameterize the goal prior of the physical model, i.e., $p'(o_k) = \mathcal{N}(o_k \,|\, \hat{o}^*, 10^{-2}\text{I})$ with $\hat{o}^*$ sampled from $q(\hat{o}|m_{\text{target}})$, see Fig. 2 for the factor graphs of both models.

4 Experimental Validation

4.1 Setup

In this study we design an active inference-based control agent for a two-wheeled robot made by Parallax, Inc. [14]. The actuators of the robot are two continuous-rotation servo motors (one for each wheel) and the robot's sensors include a

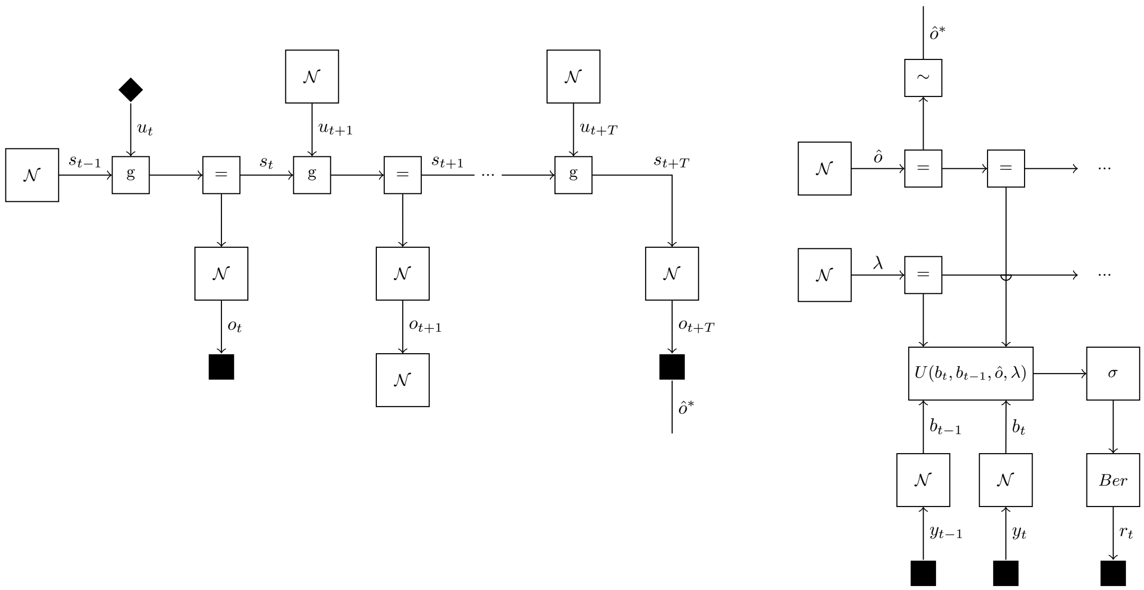


Fig. 2: (a) A Forney-style factor graph (FFG) of the physical model. (b) FFG of the Target model. Note that the mean of the future target position in the physical model ($\hat{o}^*$) is sampled from the posterior belief by the Target model about that position.

gyroscope and two angular position feedback sensors. The agent's control signals are independent (delta) velocity signals to the servo motors. While the gyroscope reports the current orientation of the robot, the angular position feedback sensors are used for determining how many degrees the wheels have rotated. The current position of the robot is calculated by dead reckoning. Dead reckoning is an infrastructure-free localization method where the current position of a mobile entity is calculated by advancing a previously known position using estimated speed over time and course [5].

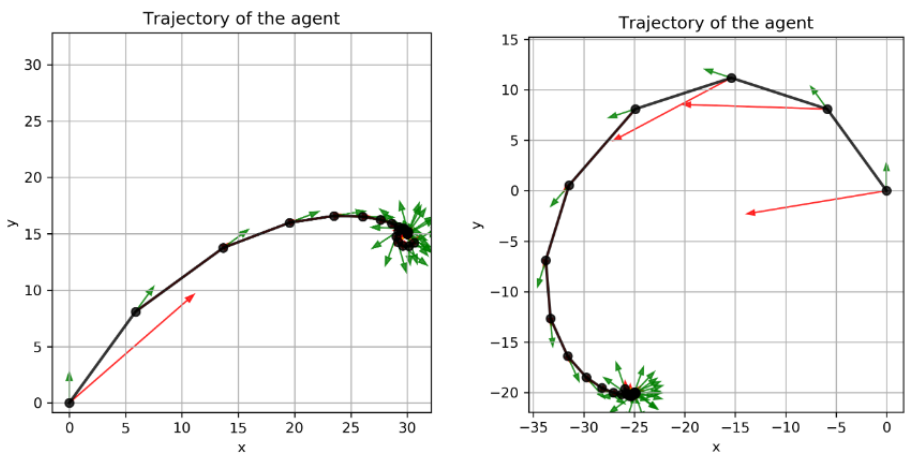
We employed a Raspberry Pi 4 [9] as a platform for executing free energy minimization (coded in Julia [3], running on Raspberry Pi's Linux variant) and an Arduino Uno [1] for gathering sensor readings and actuating the motors. The Raspberry Pi is wirelessly connected to a PC and user appraisals are provided using this wireless connection.

Inference algorithms were automatically generated using the probabilistic programming toolboxes ForneyLab [4] and Turing [10].

We use an *online* active inference simulation scheme that comprises three phases per time step: (1) act-execute-observe, (2) infer, (3) slide, as described in [15]. The simulation ran for 30 time steps with a horizon $T = 2$.

4.2 Results

Typical simulation results of the trajectory of the robot are shown in Fig. 3. The results show that the agent is capable of steering the robot to the intended target.



(a) Target position: (30,15)

(b) Target position: (-25,-20)

Fig. 3: Simulation results of the physical model. Green arrows show the orientation of the agent and the red arrows show the proposed motion for the next iteration.

Fig. 4 depicts a typical evolution of the agent's belief about the intended target location. The mean of the belief $\hat{o}$ comes within 2 cm of the target location in approximately 60 iterations.

We also tested the performance of the agent after interventions such as physically changing the orientation of the agent en route. The following video fragment demonstrates how the active inference agent immediately corrects a severe manual interruption and continues its path towards the target location: https://youtu.be/AJevoOmKMO8.

5 Related Work

Prior work on agent-based models within the active inference framework has mainly focused on simulated agents, with a few real-world implementations only recently emerging. In [2] a simulated photo-taxis agent is introduced with a focus on performance evaluation based on achieving goal-directed tasks rather than accurately describing world dynamics. In our work, we followed a similar approach. The physical model introduced in section 3.1 encodes information

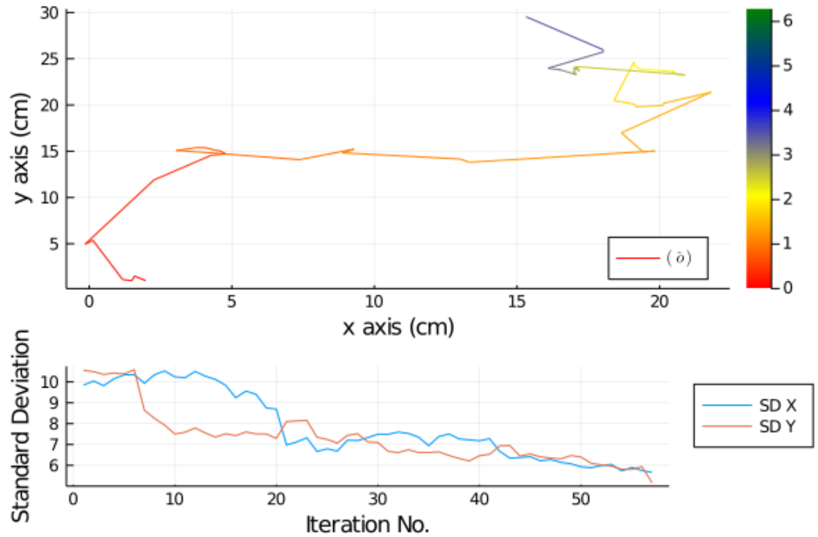


Fig. 4: Simulation results of the target model with a user in the loop. The agent converges to the target location on a 2D plane by observing binary user appraisals. The initial position of the agent is (0,0) and the target location specified by the user is (15,30). The user provides a binary appraisal in each time step.

about world dynamics. A major difference between [2] and this paper is the way goal-directed behavior is induced. In [2] a goal state is not explicitly specified, but rather is a consequence of how priors relating to observations and controls are implemented. In our formulation, a goal state is defined as a prior distribution over future observations.

More recent work, notably [13], addresses the gap between simulated agent implementations and real-world applications. In [13] an active inference model for body perception and actions in a humanoid robot is implemented with a comparison to classical inverse kinematics. Their results show improved accuracy without an increase in computational complexity providing further evidence for active inference's promise for real-world applications.

6 Conclusions

In order to assess active inference's feasibility for real-world applications, we developed an agent that controls a ground-based robot. The experiments provide support for the notion that active inference is a viable method for constructing synthetic agents that are capable of learning new goals in a dynamic world. More details about this project are available in [6].

Acknowledgements

This work was partly financed by research programmes ZERO and EDL with project numbers P15-06 and P16-25 respectively, which are both (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

References

1. Arduino.cc: Arduino uno rev3. https://store.arduino.cc/arduino-uno-rev3 (2020), accessed: 2020-04-08
2. Baltieri, M., Buckley, C.L.: An active inference implementation of phototaxis. Artificial Life Conference Proceedings (29), 36–43 (2017). https://doi.org/10.1162/isal_a_011, https://www.mitpressjournals.org/doi/abs/10.1162/isal_a_011
3. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A fresh approach to numerical computing. SIAM Review **59**(1), 65–98 (2017). https://doi.org/10.1137/141000671, https://doi.org/10.1137/141000671
4. Cox, M., van de Laar, T., de Vries, B.: A factor graph approach to automated design of Bayesian signal processing algorithms. International Journal of Approximate Reasoning **104**, 185–204 (Jan 2019). https://doi.org/10.1016/j.ijar.2018.11.002, http://www.sciencedirect.com/science/article/pii/S0888613X18304298
5. Edelkamp, S., Schroedl, S., Koenig, S.: Heuristic Search: Theory and Applications. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2010)
6. Ergul, B.: A Real-World Implementation of Active Inference. Master's thesis, Eindhoven University of Technology (04 2020), https://biaslab.github.io/pdf/msc/Ergul-2020-MSc-thesis-A-Real-World-Implementation-of-Active-Inference.pdf
7. Everett, H.R.: Sensors for Mobile Robots: Theory and Application. A. K. Peters, Ltd., USA (1995), ISBN: 1568810482
8. Friston, K.: The free-energy principle: a unified brain theory? Nature Reviews Neuroscience **11**(2), 127–138 (2010)
9. Gay, W.: Raspberry Pi Hardware Reference. Apress, USA, 1st edn. (2014). https://doi.org/10.1007/978-1-4842-0799-4
10. Ge, H., Xu, K., Ghahramani, Z.: Turing: a language for flexible probabilistic inference. In: International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain. pp. 1682–1690 (2018), http://proceedings.mlr.press/v84/ge18b.html
11. Koudahl, M.T., de Vries, B.: Batman: Bayesian target modelling for active inference. ICASSP-2020 conference, Barcelona (2020)
12. van de Laar, T.W., de Vries, B.: Simulating active inference processes by message passing. Frontiers in Robotics and AI **6**, 20 (2019). https://doi.org/10.3389/frobt.2019.00020, https://www.frontiersin.org/article/10.3389/frobt.2019.00020
13. Oliver, G., Lanillos, P., Cheng, G.: Active inference body perception and action for humanoid robots. CoRR **abs/1906.03022** (2019), http://arxiv.org/abs/1906.03022
14. Parallax: Robot shield with arduino. https://www.parallax.com/product/32335 (2020), accessed: 2020-04-08
15. van de Laar, T.: Automated design of Bayesian signal processing algorithms. Ph.D. thesis, Eindhoven University of Technology (2019)