Active inference for Robot control: A Factor Graph Approach

Mees Vanderbroeck*; Mohamed Baioumy; Daan van der Lans; Rens de Rooij; Tiis van der Werf
Delft University of Technology
*mees.vanderbroeck@gmail.com
All contributed equally.

ABSTRACT
Active Inference provides a framework for perception, action and learning, where the optimization is done by minimizing the Free-Energy of a system. This paper explores whether active inference can be used for closed-loop control of a 1 degree of freedom robot arm. This is done by implementing variational message passing on Forney-style factor graphs; a probabilistic programming framework. We show that an active inference controller with variational message passing can perform state estimation and control at the same time.

Keywords
Factor graphs, free-energy principle, active inference, closed-loop control, variational message passing

I. INTRODUCTION

The Free Energy Principle (FEP) has been proposed to provide a unified theory of the brain [1], providing an explanation for how cognitive functions such as perception, action and model learning are achieved [2]. It claims that in order for an intelligent agent to persist in a time-varying environment, it must minimize 'surprise' (the atypicality of an event). This is done by minimizing an upper bound called 'Free Energy', as organisms can not directly minimize surprisal.

Active Inference, corollary to the FEP, states that biological agents act to fulfill prior beliefs about preferred future observations. Desired behaviour is then achieved by minimizing Free Energy with respect to a generative model of the environment [3]. This generative model is an internal model of the environment that is used to produce actions and estimate posterior states.

There have been a few attempts of implementing active inference in robot control. In [4], a PR2 (Personal Robot2, a human-like robot) robot, simulated in Robot OperatingSystem (ROS, the open source robotic middleware), is con-trolled by open-loop Active Inference.Lanillos et al. [5] use concepts from Active Inference for sensor data fusion for an interactive robot.

The main contribution of this paper is to provide an *online* Active Inference implementation for control of a 1 degree of freedom (DoF) robot arm. The system is modeled as a stochastic State-Space Model implemented as Forney-Style Factor Graphs (FFG) [6]. A factor graph is a tool that is used in probability theory to enable efficient computations.The performance of this controller is tested for common control tasks. This contribution aims to clarify the potential of closed-loop robot control with active inference.

II. Background on the Free Energy principle and Active Inference

This section offers a short technical recap of Active Inference and Factor graphs. The concepts will be introduced with the application, provided in III, in mind: a torque controlled robot arm equipped with proprioceptive sensors.

Variational free energy

The Free Energy Principle postulates that well-adapted biological agents maintain a probabilistic model of their typical environment (including their bodies). In order to persist in a time-varying environment, these agents attempt to minimize the occurrence of events which are atypical in such an environment as estimated by their internal model. The atypicality of an event can be quantified by the negative logarithm of the probability of its sensory data $P(x)$, also known as 'surprise'. It is argued in the FEP that organisms can not minimize surprisal directly [7]. Instead they achieve this by minimizing an upper bound called the 'Free Energy'.

To see this into practice, let us consider a robot arm which receives sensor data x about the value of its state and it tries to infer its actual hidden state z. This could be formulated by Bayes' theorem as:

$$\boldsymbol{p(z|x)} = \frac{\boldsymbol{p(z)p(\phi|z)}}{\boldsymbol{p(\phi)}} \tag{1}$$

In equation 1, $p(x|z)$ is the sensory consequence of being in a physical state z and $p(z)$ is the prior belief of the environmental states. The marginalization of the likelihood over all the possible states is computationally tough. In this case, we can apply Variational Free Energy approximation [7]. The core idea is to minimize Kullback-Leibler (KL) divergence between a distribution $q(z)$ that is encoded in the agent and the true posterior $p(z|x)$. Meaning, the true posterior distribution ($p(z|x)$) is often intractable for all different values so it is approximated with a distribution $q(z)$ that has a standard shape (for example a normal distribution).

$$\begin{aligned} KL\big(q(z), p(z|x)\big) &= \int q(z) \ln \frac{q(z)}{p(z|x)} dz \\ &= \int q(z)(\ln q(z) \\ &\quad - \ln p(z,x))dz + \ln\ p(x) \\ &= F + \ln\ p(x) \end{aligned} \tag{2}$$

SRC 2019, December 5, 2019, The Netherlands.

The Kullback-Leibler is in essence a measure for dissimilarity between two distributions. Note that if *q(z)* and *p(z|x)* were identical their ratio would be 1 and its logarithm would be 0. This would mean the integral evaluates to 0 and thus there is no dissimilarity.

From equation (2) it is noted that minimizing *F* directly reduces KL since the term *ln p(x)* does not depend on *q(z)*. Instead of approximating *p(z|x)* with the whole *q(z)* distribution, the agent model is a delta distribution $\delta(z - s)$ that makes *s* the mean of the approximating density as done in section 3 of [3]. So *z* is the actual physical state of the robot and *s* is the estimation of this physical state which is encoded in the brain dynamics. Thus, we can remove the integrals simplifying the Variational Free Energy equation to:

$$F = -\ln p(s, x) = -\ln p(x|s) - \ln p(s) \qquad (3)$$

For further reading and full derivations see [3], [5] and [7].

Active inference

The minimization of VFE does not just account for perceptual inference but also for actions within the same framework. This means that while the beliefs are updated to better predict sensory data, actions are simultaneously executed on the environment to alter the sensory input and make these coincide with the sensory predictions [7]. This would mean that the true posterior *p(z|x)* is extended to *p(z, u|x)* where *u* are the actions. Subsequently, *q(z)* is extended to *q(z, u)*.

This process can be represented by the schematic in figure 1. The agent and the environment are two statistically separated nodes, shown in blue and red respectively. These can interact with each other through a Markov Blanket [8]. A Markov blanket only contains the variables that a node uses to interact with the outside world, isolating the dynamics within that node from the outside.

Instead of knowing the dynamics of the environment and its states exactly, the agent estimates the states and dynamics using an internal recognition model *q*. The agent furthermore observes the state output *y* from the environment imprecisely to some extent in the triangular node. Secondly, the control signal *u* and the environmental action *a* are linked in the square node. This control signal and the corresponding environmental action are generated by the agent, using observations of the states.

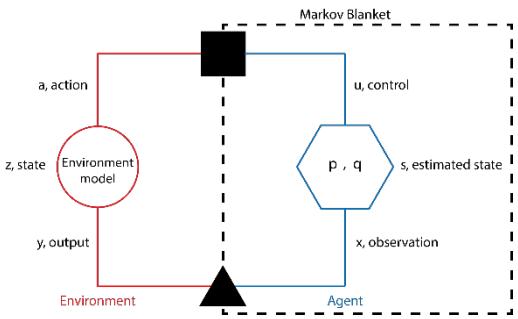

Fig. 1. Schematic of agent-environment interaction

III. METHOD

To research the feasibility of closed-loop AIC with FFG, a simulation of a 1 DoF robotic arm is set up. The robotic arm that is simulated can be seen in figure 2.

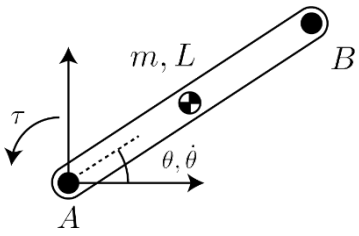

Fig. 2. Schematic of simulated robotic arm

The link can move relative to the 'fixed' world in point A. Point B marks the end-effector position. The state of the robotic arm can be changed by the motor in A with torque τ. The sensor used is a proprioceptive sensor in A.
The proprioceptive sensor outputs measurements of the angle and angular velocity between the robot arm and the horizontal axis $(\phi, \dot{\phi})$. $\phi, \dot{\phi}$ are sensor data that are captured in a probability function $p(\boldsymbol{x_t}|\boldsymbol{s_t})$. Be aware that ϕ and $\dot{\phi}$ are not necessarily identical to the real angle and angular velocity $\theta, \dot{\theta}$ as they include sensor noise.

Environment setup

The robotic arm operates in the horizontal plane, therefore the gravitational force has no influence on the motion dynamics. To simulate the saturation of the motor, the torque applied to the robot arm is limited to *10 Nm* in both directions, for all possible control signals by equation 4.

$$\tau(a_t) = 10 * \tanh(a_t) \qquad (4)$$

The environment is updated by Euler integration, as in equation 5. Equation 5b is the short vector notation of 5a.

$$\begin{bmatrix} \dot{\theta}_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ dt & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{\tau(a_t) * dt}{I_A} \\ 0 \end{bmatrix} \qquad (5a)$$

$$\boldsymbol{z_t} = \boldsymbol{g} * \boldsymbol{z_{t-1}} + \boldsymbol{h}(a_t) \qquad (5b)$$

Where I_A is the mass moment of inertia around point A.
To simulate the sensor data used in the AIC, Gaussian white noise is added to the real robot state values, with standard deviations $\sigma_\theta, \sigma_{\dot{\theta}}$. The output vector y_t will contain the real (hidden) angle and angular velocity values $\theta, \dot{\theta}$ similar to figure 1. The torque τ is given as a function of the action a_t. The control signal u_t from the agent generates these actions.

Internal (agent) model

In order to estimate the posterior states, the agent uses a stochastic state-space model. The execution of the generative model p_t is done with a factor graph represented in figure 3. The factorization of this factor graph is given by:

$$p_t(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{u}) \propto p(\boldsymbol{s_{t-1}}) \prod_{n=t}^{t+T} \overbrace{p(\boldsymbol{x_n}|\boldsymbol{s_n})}^{\text{observation}} \overbrace{p(\boldsymbol{s_n}|\boldsymbol{s_{n-1}}, u_n)}^{\text{state transition}} \overbrace{p(u_n)}^{\text{control}} \overbrace{\tilde{p}(\boldsymbol{x_n})}^{target} \qquad (6)$$

Where the probability functions represent the model for the observation, state transition, control and target; which will be explained one by one. Note that now 'n' is used to indicate a specific timestep, all variables that vary for each timestep will therefore be denoted with an 'n' as subscript.

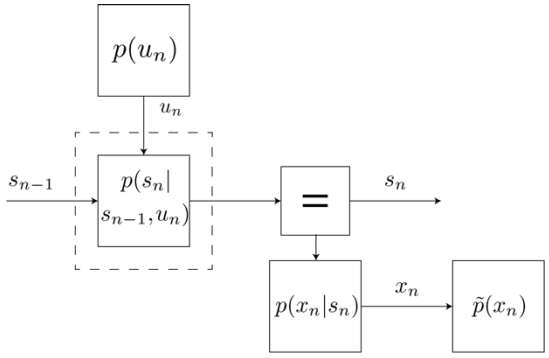

Fig. 3. Schematic of agent-environment interaction

The agent's target state is defined as $\boldsymbol{x_{ref}} = (\theta_{ref}, \dot{\theta}_{ref}) = (2.0, 0.0)$. T is the number of steps that the algorithm will look ahead ("lookahead" parameter), so the factor graph will be generated for T timesteps in the future on each iteration. The target prior is defined as a normal distribution, marked as $\tilde{p}(\boldsymbol{x_n})$, to clearly distinguish it from the observation model notation. For timesteps before the first lookahead, this target prior has a large variance of 10^{12} to make a change in state have almost no effect on the Free Energy, allowing for unrestricted state changes, because the controller is not yet expected to have reached $\boldsymbol{x_{ref}}$. For $t \geq Tdt$, The target prior is set to $\boldsymbol{x_{ref}}$, with a small variance, 10^{-4} in this case, to make sure the robot arm reaches this state with high precision (equation 7).

$$\tilde{\boldsymbol{p}}(\boldsymbol{x_n}) = \begin{cases} \boldsymbol{N(x_n|0, 10^{12})} & \boldsymbol{if\ t < Tdt} \\ \boldsymbol{N(x_n|x_{ref}, 10^{-4})} & \boldsymbol{otherwise} \end{cases} \quad (7)$$

Furthermore, the agent has a transition model that captures the state dynamics for the estimated robot states *s*. The AIC uses this transition model, as defined in equation 8 to infer the environmental dynamics.

$$p(\boldsymbol{s_n}|\boldsymbol{s_{n-1}}, u_n) = N(\boldsymbol{s_n}|p(\boldsymbol{s_{n-1}}, u_n), \Gamma) \quad (8)$$

Here Γ is the state transition variance. The value for Γ is chosen according to the belief of the accuracy of the generative model $p(\boldsymbol{s}, \boldsymbol{x}, u)$. For instance, if a non-linear system is approximated by a linear model, you would resort to a higher variance. For the presented results, Γ was chosen to be 10^{-4}.

The generative model is represented by $p(\boldsymbol{s}_{n-1}, u_n)$ and closely resembles the environmental dynamics. This is indicated in figure 5 within the dashed box. In this case $\boldsymbol{g}$ and $\boldsymbol{h}$ are the same functions as in 5b.

The internal model uses an observation model, defined in equation 9, to predict the sensory outputs.

$$p(\boldsymbol{x_n}|\boldsymbol{s_n}) = N(\boldsymbol{x_n}|\boldsymbol{s_n}, \alpha) \quad (9)$$

Here, α is the observation variance. The value of the observation variance is encoded in the agent, which resembles its belief about how noisy the observations of the environment are. For the results presented in this paper, α was set to 10^{-4}.

Finally, the internal model has a state prior and control prior as defined in equations 10a and 10b respectively.

$$p(s_0) = N(s_0|(0.0,0.0), \sigma_{s_0}^2) \quad (10a)$$
$$p(u_n) = N(u_n|0.0, 10^{12}) \quad (10b)$$

Where $\sigma_{s_0}^2$ is defined dependent on how well the initial state of the robot arm is known. In our experiments, the AIC is given the assumption that the brain state perfectly matches the initial hidden state (i.e. the initial state is known), so the variance is small: $\sigma_{s_0}^2 = 10^{-12}$. The control prior is given a large variance of 10^{12} to allow the controller to conduct a large range of control signals.

Control loop

The algorithm loops through the functions act, execute, observe, infer and slide respectively, similar to [9], assuming we start at step n and simulate T timesteps, as in fig. 4. These functions are defined as follows:

1. *Act:* The act-function simply converts the control signal *u*, generated by the agent, into the action *a*, making their values equivalent. This action *a* will be applied to the environment. This is illustrated in figure 5 by the label 1.
2. *Execute:* Next up, the action a is used to calculate a certain torque, which will move the robot arm. The angle and angular speed of the robot are updated as described in equation 5. This way, the action is used to change the environment. So Execute happens in the environment as labelled by 2 in figure 5.
3. *Observe:* As the environment changes in the execute step, the angle and angular speed of the robot arm change. These new angles will be observed by the sensors as $\boldsymbol{x}$. This value is clamped as indicated by 3 in figure 5.
4. *Infer:* In the infer-function, VMP is used to calculate a new control signal *u* by minimizing VFE. The input for the VMP is the action *a* and observed $\boldsymbol{x}$ generated in the steps before. Subsequently, the next state and control action are inferred. This is shown in figure 4 by 4.
5. *Slide:* The last part of the algorithm is to remove the first time step from the factor graph, and to add one time step at $n + t + 1$. This is shown in figure 4, where the first step (encircled) is removed and the last step is added.

IV. EXPERIMENTAL RESULTS

To evaluate the performance of the AIC, a number of experiments were conducted. As a proof of concept, the controller was instructed to go from an initial state to a target state, in this case from $\theta_{start} = 0\ rad$ to $\theta_{ref} = 2\ rad$. Secondly, noise was added to the sensors. In all experiments listed below, a timestep of 0.2 seconds and a lookahead of 5 timesteps are used.

No noise

Figure 6 show the smooth convergence towards its target state. The results show that the rise time and settling time of the controller are both approximately 15 seconds for an error band of 5%. This response is used as a benchmark to test the effect of noise and disturbances.

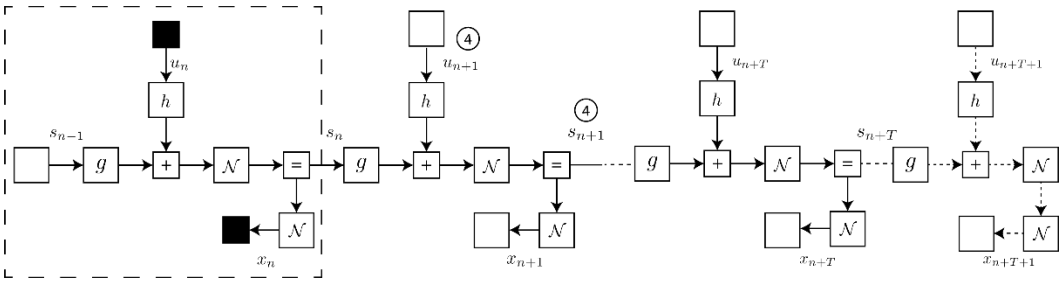

Fig. 4. Factor graph showing the slide function of the algorithm. The infer function is indicated with a 4.

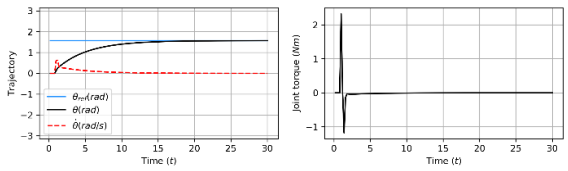

Fig. 5. Factor graph showing one timestep of the algorithm.

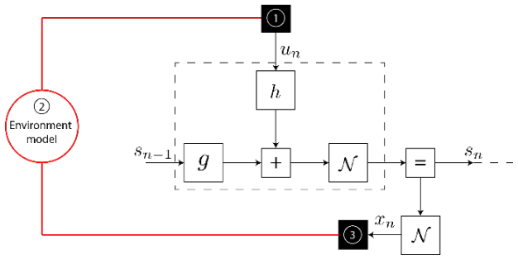

Fig. 6. Performance of the AIC with the standard settings and without any noise.

With noise

Figure 7 shows the controllers' response with Gaussian white noise on the sensors ($\sigma_\theta^2 = 0.02$ *rad*). The AIC's shows minor oscillations around the target state in both cases. The rise time and overshoot are comparable to the no-noise experiment. The required input torque is minimal. Finally, the estimated state is very smooth compared to the observations.

V. CONCLUSION

In this paper we have presented a closed-loop Active Inference controlled (AIC) robot simulation, using a Factor Graph implementation. This controller is able to simultaneously update its beliefs about its state and perform actions on this environment, based on sensory input. The AIC was used to control a 1DoF robot arm in the horizontal plane. The results have shown that the controller is able to steer the robot arm to an arbitrary target position. The controller shows the ability to reach and maintain the desired position when disturbances are applied. An implementation with more complicated disturbances such as gravitational forces has not yet been successful. Further research into the tuning methods for the AIC is required to fully understand and improve this control method. Altogether, we can conclude that the proposed Active Inference controller provides a way to control a robot that is robust to various disturbances and, with some extensions, shows promise for the future of robot control.

ROLE OF THE STUDENT

All authors are students who were under supervision of Carlos Hernandez Corbato. The work was equally divided under the students. The topic was proposed by the supervisor. The design of the controller, formulation of the results and the writing was done by the students.

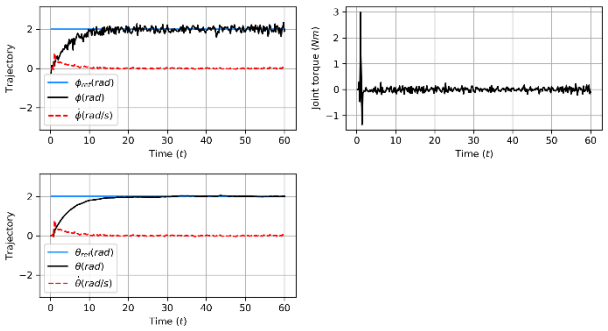

Fig. 7. Performance of the AIC with minor Gaussian white noise $\sigma_\theta^2 = 0.02$ on the sensor

ACKNOWLEDGMENTS

The authors thank Carlos Hernandez Corbato for the supervision, Thijs van der Laar for answering questions about *ForneyLab*.

REFERENCES

1. K. Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11:127 EP -, 01 2010.
2. K. Friston. The free-energy principle: a rough guide to the brain? *Trends in cognitive sciences*, 13:293–301, 07 2009.
3. R. Bogacz. A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology*, 76:198–211, 2017.
4. L. Pio-Lopez, A. Nizard, K. Friston, and G. Pezzulo. Active inference and robot control: a case study. *Journal of The Royal Society Interface*, 13(122):2016 0616, 2016
5. P. Lanillos and G. Cheng. Adaptive robot body learning and estimation through predictive coding. 05 2018.
6. H. Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, Jan 2004.
7. C. L. Buckley, Chang Sub Kim, Simon McGregor, and Anil K Seth. The free energy principle for action and perception: A mathematical review. *Journal of Mathematical Psychology*, 81:55–79, 2017.
8. J. Pearl. Probabilistic reasoning in intelligent systems: networks of plausible inference. *Elsevier*, 2014.
9. T. W. van de Laar and B. de Vries. Simulating active inference processes by message passing. *Frontiers in Robotics and AI*, 6(20), 2019