Adaptive Manipulator Control using Active Inference with Precision Learning

Mohamed Baioumy, Matias Mattamala, Paul Duckworth, Bruno Lacerda, and Nick Hawes
Oxford Robotics Institute
{mohamed, matias, pduckworth, bruno, nickh}@robots.ox.ac.uk

***Abstract*—Active inference provides a framework for decision-making where the optimization is achieved by minimizing free-energy. Previous work has used this framework for control and state-estimation of a robotic manipulator. This required manual definition of precision matrices which serve as controller gains. This paper provides an implementation for control and state-estimation where the precision matrices are tuned during execution-time (precision learning). Learning the precision matrices means automatically adjusting the controller's gains which decreases oscillations and overshoot.**

Supplementary Material

Code and further material is avaiable at: https://github.com/MoBaioumy/active_inference_panda_paper.
Video: https://youtu.be/Ii1Ig1Lt0Xk

I. INTRODUCTION

Modelling all time-varying dynamics for a robotic manipulator a priori is infeasible. Therefore, intelligent robotic manipulators require adaptive behaviour, e.g. to reject disturbances or to handle objects of unknown masses [1].

Recent approaches in robotics have taken inspiration from active inference [2], a theory of the brain prominent in neuroscience. Active inference provides a framework for understanding decision-making of biological agents. Under the active inference framework, optimal behavior arises from minimising variational free-energy: a measure of the fit between an internal model and (past) sensory observations [2]. Additionally, agents take actions that allow reaching preferred future observations specified a priori.

In [3], an active inference controller (AIC) for joint space control of robotic manipulators is presented, which outperforms the state-of-the-art Model Reference Adaptive Control (MRAC) [4]. This approach performs both state-estimation and control, and avoids scalability issues by requiring a fixed number of parameters, such as precision matrices (inverse covariance matrix). These precision matrices act as controller gains and thus affect performance properties such as oscillations, overshoot and rise-time.

In this paper, we present an approach for precision learning, i.e. learning the precision matrix. We show that by having an adaptation step to learn the precision matrices in execution time, which is derived from the same active inference principle, we are able to automatically adjust the controller's gains, henceforth decreasing oscillations and overshoot.

II. METHOD

Active Inference considers an agent in a dynamic environment that receives an observation $\mathbf{o}$ about a state $\mathbf{s}$. The agent then infers the posterior $p(\mathbf{s}|\mathbf{o})$ given a model of the agent's world. Instead of exactly calculating $p(\mathbf{s}|\mathbf{o})$, which could be computationally expensive, the agents approximates $p(\mathbf{s}|\mathbf{o})$ with a 'variational distribution' $Q(\mathbf{s})$ which we can define to have a standard form (Gaussian for instance). The goal is then to minimize the difference between the two distributions. This can be computed using the KL-divergence [5]:

$$\begin{aligned} KL(Q(\mathbf{s})||p(\mathbf{s}|\mathbf{o})) &= \int Q(\mathbf{s}) \ln \frac{Q(\mathbf{s})}{p(\mathbf{s},\mathbf{o})} d\mathbf{s} + \ln p(\mathbf{o}) \\ &= F + \ln p(\mathbf{o}). \end{aligned} \tag{1}$$

The quantity F is referred to as the (variational) free-energy -or Evidence lower bound- and minimizing F minimizes the KL-divergence. If we choose $Q(s)$ to be a Gaussian distribution with mean $\boldsymbol{\mu}$, and utilize the Laplace approximation [6], the free-energy expression simplifies to:

$$F \approx -\ln p(\boldsymbol{\mu}, \mathbf{o}). \tag{2}$$

Now the expression for variational free-energy is solely dependent on one parameter, $\boldsymbol{\mu}$, which is referred to as the 'belief state'. The objective is to find $\boldsymbol{\mu}$ which minimizes F; this results in the agent finding the best estimate of its state.

A. Problem statement

In this work we consider low-level joint control for a robotic manipulator. The state of the robot is given by its joint position, velocities, $\tilde{\boldsymbol{\mu}} = [\boldsymbol{\mu}, \boldsymbol{\mu}']$. Note that in the Active Inference framework the state given by position and higher derivatives is also known as *generalised motions* [7]. We also consider observations given by joint encoders, $\tilde{\mathbf{o}} = [\mathbf{o}, \mathbf{o}']$, which are the main source of information about the state.

As in [3], we aim to obtain a control law for the state $\tilde{\boldsymbol{\mu}}$ from the Active Inference framework as well as the robot joint state $\tilde{\boldsymbol{\mu}}$ given the observations $\tilde{\mathbf{o}}$. However, we also derive learning rules for the parameters of the distributions (precision matrices), which allow us to adapt the controller online.

B. Observation model and state transition model

Following [3], the joint probability from Equation (2) can be written as:

$$p(\tilde{\mathbf{o}}, \tilde{\boldsymbol{\mu}}) = p(\tilde{\mathbf{o}}|\tilde{\boldsymbol{\mu}})p(\tilde{\boldsymbol{\mu}}) = \underbrace{p(\mathbf{o}|\boldsymbol{\mu})p(\mathbf{o}'|\boldsymbol{\mu}')}_{\text{Observation model}}\underbrace{p(\boldsymbol{\mu}'|\boldsymbol{\mu})p(\boldsymbol{\mu}''|\boldsymbol{\mu}')}_{\text{Transition model}}, \tag{3}$$

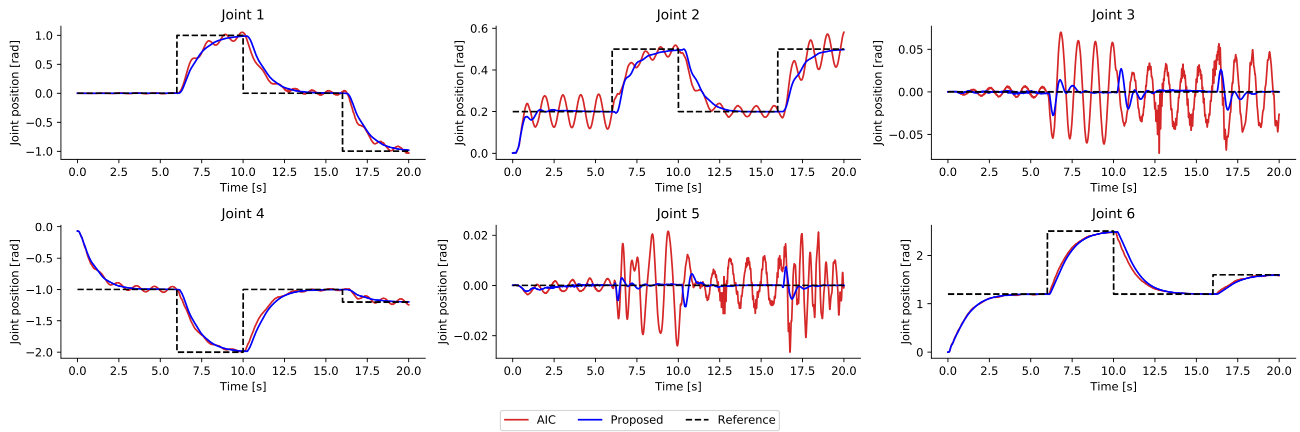

Fig. 1: Results comparing the active inference controller with and without precision learning.

where $p(\mathbf{o}|\boldsymbol{\mu})$ is the probability of receiving an observation $\mathbf{o}$ while in (belief) state μ, and $p(\boldsymbol{\mu'}|\boldsymbol{\mu})$ is the state transition model (also referred to as dynamic model or generative model). The state transition model predicts the state evolution given the current state. These distributions are assumed Gaussian according to:

$$\begin{aligned} p(\mathbf{o}|\boldsymbol{\mu}) &= \mathcal{N}(\boldsymbol{\mu}, \Sigma_o), \quad & p(\mathbf{o}'|\boldsymbol{\mu'}) &= \mathcal{N}(\boldsymbol{\mu'}, \Sigma_{o'}), \\ p(\boldsymbol{\mu'}|\boldsymbol{\mu}) &= \mathcal{N}(f(\boldsymbol{\mu}), \Sigma_\mu), \quad & p(\boldsymbol{\mu''}|\boldsymbol{\mu'}) &= \mathcal{N}(f'(\boldsymbol{\mu'}), \Sigma_{\mu'}), \end{aligned} \tag{4}$$

where the functions $f(\boldsymbol{\mu})$ and $f(\boldsymbol{\mu'})$ represent the evolution of the belief state over time. This encodes the agent's preference over future states (in this case the preferred future state is the target state, $\boldsymbol{\mu_d}$). We assume: $f(\boldsymbol{\mu}) = (\boldsymbol{\mu_d} - \boldsymbol{\mu})\tau^{-1}$ and $f'(\boldsymbol{\mu'}) = \tau^{-1}\boldsymbol{\mu'}$, where $\boldsymbol{\mu_d}$ is the desired state and τ is a temporal parameter.

Using the previous equations, we can expand F to:

$$\begin{aligned} F = \frac{1}{2}(&\boldsymbol{\varepsilon_o}^\top \Sigma_o^{-1} \boldsymbol{\varepsilon_o} + \boldsymbol{\varepsilon_{o'}}^\top \Sigma_{o'}^{-1} \boldsymbol{\varepsilon_{o'}} \\ &+ \boldsymbol{\varepsilon_\mu}^\top \Sigma_\mu^{-1} \boldsymbol{\varepsilon_\mu} + \boldsymbol{\varepsilon_{\mu'}}^\top \Sigma_{\mu'}^{-1} \boldsymbol{\varepsilon_{\mu'}} \\ &+ \ln|\Sigma_o| + \ln|\Sigma_{o'}| + \ln|\Sigma_\mu| + \ln|\Sigma_{\mu'}|) + C, \end{aligned} \tag{5}$$

where $\boldsymbol{\varepsilon_\mu} = \boldsymbol{\mu'} - (\boldsymbol{\mu_d} - \boldsymbol{\mu})\tau^{-1}$, $\boldsymbol{\varepsilon_{\mu'}} = \boldsymbol{\mu''} + \tau^{-1}\boldsymbol{\mu'}$, $\boldsymbol{\varepsilon_o} = \mathbf{o} - \boldsymbol{\mu}$ and $\boldsymbol{\varepsilon_{o'}} = \mathbf{o}' - \boldsymbol{\mu'}$ and C refers to constant terms.

C. Estimation and control

To achieve state estimation and control, we perform gradient descent on F using the following update rules:

$$\dot{\tilde{\boldsymbol{\mu}}} = D\tilde{\boldsymbol{\mu}} - \kappa_\mu \frac{\partial F}{\partial \tilde{\boldsymbol{\mu}}}, \tag{6}$$

$$\dot{\mathbf{a}} = -\kappa_a \frac{\partial F}{\partial \mathbf{a}} = -\kappa_a \frac{\partial F}{\partial \tilde{\mathbf{o}}} \frac{\partial \tilde{\mathbf{o}}}{\partial \mathbf{a}}, \tag{7}$$

where κ_a and κ_μ are tuning parameters depending on the desired behaviour, and D is the temporal derivative operator.

D. Precision learning

Precision learning refers to learning the inverse covariance matrices, also referred to as the precision matrices. This is done using one-step gradient descent [8] as:

$$\dot{\Sigma_o^{-1}} = -\kappa_\sigma \frac{\partial F}{\partial \Sigma_o^{-1}}, \quad \dot{\Sigma_{o'}^{-1}} = -\kappa_\sigma \frac{\partial F}{\partial \Sigma_{o'}^{-1}}. \tag{8}$$

Now, using Equations 6, 7 and 8, the manipulator can perform state-estimation, control and precision learning. Since in Equation 7, we use the chain rule with respect to the observations, Σ_o^{-1} and $\Sigma_{o'}^{-1}$ are the matrices that show up in the control law.

III. Results

We evaluate the presented approach and use the active inference controller (AIC) from [3] as a benchmark since the authors have shown their work outperforms MRAC.

Consider a manipulator moving between three different configurations (see supplementary video or section 5 of [3]). If the AIC is tuned properly ($\Sigma_o^{-1} = 1.5I$, $\Sigma_{o'}^{-1} = 0.5I$, $\Sigma_\mu^{-1} = 0.1I$ and $\Sigma_{\mu'}^{-1} = 0.5I$), this results in satisfactory behaviour. In this case, I refers to the 7x7 identity matrix. However, if we perturb all diagonal elements of Σ_μ^{-1} by random values between 0 and 0.2 during initialization, the controller suffers from oscillations. Our approach overcomes this issue by performing precision learning. We update Σ_o^{-1} and $\Sigma_{o'}^{-1}$ for the first two seconds during operation which allows the controller to 'self-tune'. Results are shown in Figure 1. This shows the performance of the AIC and our approach for a single run. Using our approach, all joints show almost no sign of oscillations.

IV. Conclusion

In this work we perform state estimation, control and precision learning under the active inference framework. The approach is tested (on the Panda Emika Franka) against previous work and shows the ability to damp oscillations when the precision matrices are not properly initialized.

REFERENCES

[1] Bin Wei. Adaptive control design and stability analysis of robotic manipulators. In *Actuators*, volume 7, page 89. Multidisciplinary Digital Publishing Institute, 2018.

[2] Karl Friston, Thomas FitzGerald, Francesco Rigoli, Philipp Schwartenbeck, and Giovanni Pezzulo. Active inference: a process theory. *Neural computation*, 29(1):1–49, 2017.

[3] Corrado Pezzato, Riccardo Ferrari, and Carlos Hernandez. A novel adaptive controller for robot manipulators based on active inference. *arXiv preprint arXiv:1909.12768*, 2019.

[4] Dan Zhang and Bin Wei. A review on model reference adaptive control of robotic manipulators. *Annual Reviews in Control*, 43:188–198, 2017.

[5] Charles W Fox and Stephen J Roberts. A tutorial on variational bayesian inference. *Artificial intelligence review*, 38(2):85–95, 2012.

[6] Karl Friston, Jérémie Mattout, Nelson Trujillo-Barreto, John Ashburner, and Will Penny. Variational free energy and the laplace approximation. *Neuroimage*, 34(1):220–234, 2007.

[7] Christopher 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] Rafal Bogacz. A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology*, 76:198–211, 2017.