Variational Inference for Predictive and Reactive Controllers

Mohamed Baioumy Matias Mattamala Nick Hawes

***Abstract*— Active inference is a general framework for decision-making prominent neuroscience that utilizes variational inference. Recent work in robotics adopted this framework for control and state-estimation; however, these approaches provide a form of 'reactive' control which fails to track fast-moving reference trajectories. In this work, we present a variational inference predictive controller. Given a reference trajectory, the controller uses its forward dynamic model to predict future states and chooses appropriate actions. Furthermore, we highlight the limitation of the reactive controller such as the dependency between estimation and control.**

I. INTRODUCTION

Recent approaches in robotics have taken inspiration from active inference [1], 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 act to fulfill prior beliefs about preferred future observations. This framework has been employed to explain and simulate a wide range of complex behaviors, including planning, abstract rule learning, reading, and social conformity (see Table 1 of [3] for references).

A handful of approaches have used active inference to control robotic systems. For example, in [4] an implementation of an active inference controller is presented for a 3 DoF humanoid robot. It was capable of performing reaching behaviors in the visual field under noisy observation. However, the control was performed using velocity commands rather than torque commands. Second, in [5], a method for joint space control of robotic manipulators is presented and compared to the state-of-the-art Model Reference Adaptive Control in adaptability. In [6], [7] this approach is extended to included hyperparameter learning.

In this work, we show how the approach in [4], [5] provides a form of 'reactive' control (the error occurs first, then the controller reacts to it). It can only track slow references and has a a significant time-delay.

Our main contribution is to present a predictive controller based on variational inference. Given a reference trajectory, the controller uses its forward dynamic model to predict future states and choose appropriate actions to reach the desired trajectory. Additionally, unlike the reactive approach, the state-estimation and control steps are separated which allows for faster response of the controller.

The authors are with the Oxford Robotics Institute, University of Oxford, UK. Use {mohamed, matias, nickh}@robots.ox.ac.uk for correspondence.

II. REACTIVE CONTROLLER BASED ON ACTIVE INFERENCE

Active Inference considers an agent in a dynamic environment that receives observations $\mathbf{o}$ about states $\mathbf{s}$. 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 which computed by the KL-divergence [8]:

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

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 [9], 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.

Generalised motions (GM) [10] are used to represent the (belief) states of a dynamical system, using increasingly higher order derivatives of the system state. This means that the n-dimensional state $\boldsymbol{\mu}$ and its higher order derivatives are combined in $\tilde{\boldsymbol{\mu}}$ ($\tilde{\boldsymbol{\mu}} = [\boldsymbol{\mu}, \boldsymbol{\mu'}...]$). The same can be done for the observations ($\tilde{\mathbf{o}} = [\mathbf{o}, \mathbf{o}' ...]$).

For instance, in the context of a robotic manipulator, $\mathbf{o}$ represents the sensory observation of a joint position, while $\mathbf{o}'$ represents the joint's velocity observation. We consider GM up to the second order.

A. Observation model and state transition model

Taking generalized motions into account, the joint probability from Equation (2) can be written as:

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

where $p(\mathbf{o}|\boldsymbol{\mu})$ is the probability of receiving an observation $\mathbf{o}$ while in (belief) state $\boldsymbol{\mu}$, and $p(\boldsymbol{\mu'}|\boldsymbol{\mu})$ is the state transition model (also referred to as the dynamic model or the

generative model). The state transition model predicts the state evolution given the current state. These distributions are assumed Gaussian according to:

$$p(\mathbf{o}|\boldsymbol{\mu}) = \mathcal{N}(\mathbf{o}; g(\boldsymbol{\mu}), \Sigma_o), \;\; p(\mathbf{o}'|\boldsymbol{\mu'}) = \mathcal{N}(\mathbf{o}'; g'(\boldsymbol{\mu'}), \Sigma_{o'}),$$
$$p(\boldsymbol{\mu'}|\boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu'}; f(\boldsymbol{\mu}), \Sigma_\mu), \;\; p(\boldsymbol{\mu''}|\boldsymbol{\mu'}) = \mathcal{N}(\boldsymbol{\mu''}; f'(\boldsymbol{\mu'}), \Sigma_{\mu'}), \tag{4}$$

where the functions $g(\boldsymbol{\mu})$ and $g'(\boldsymbol{\mu'})$ represent a mapping between observations and states. For many application the state is directly observable, thus $g(\boldsymbol{\mu}) = \boldsymbol{\mu}$ and $g'(\boldsymbol{\mu'}) = \boldsymbol{\mu'}$.

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 reference trajectory, $\boldsymbol{\mu_d}$). We assume: $f(\boldsymbol{\mu}) = (\boldsymbol{\mu_d} - \boldsymbol{\mu})\tau^{-1}$ and $f'(\boldsymbol{\mu'}) = (\boldsymbol{\mu'_d} - \boldsymbol{\mu'})\tau^{-1}$, where $\boldsymbol{\mu_d}$ is the desired trajectory and τ is a temporal parameter.

Given that all distributions are Gaussian, the expression for F become a sum of quadratic terms and the natural logarithms as:

$$F = \frac{1}{2}(\sum_i \boldsymbol{\varepsilon}_i^\top \Sigma_i^{-1} \boldsymbol{\varepsilon}_i + \ln|\Sigma_i|) + C, \tag{5}$$

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

B. Estimation and control

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

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

where κ_μ is the gradient descent step size and D is a temporal derivative operator.

We thus perform one gradient descent step for each iteration. To find suitable control actions, we would also also use one-step gradient descent; however, the expression for F does not include any actions and thus we resort to using the chain rule, assuming an implicit dependence between the actions performed by the agent and the measurements it acquires.

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

where κ_a is the gradient descent step size. The term $\frac{\partial \tilde{\mathbf{o}}}{\partial \mathbf{a}}$ is assumed linear, and equal to the identity matrix (multiplied by a constant) similar to existing work [5], [4].

C. Simultaneous state-estimation and control

Under this formulation, estimation and control are solved simultaneously and both processes are dependent. The observation terms (such as $\boldsymbol{\varepsilon_o}$) refine the current belief $\boldsymbol{\mu}$, whereas the reference terms $\boldsymbol{\varepsilon_\mu}$ bias the estimated state towards the target $\boldsymbol{\mu_d}$. In addition, if the parameters τ^{-1} and Σ_μ^{-1} are larger, the estimate $\boldsymbol{\mu}$ is biased more towards the target $\boldsymbol{\mu_d}$.

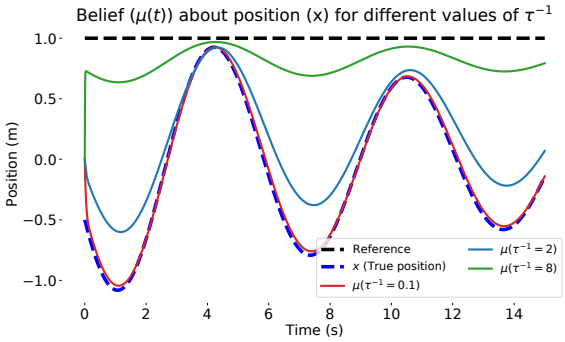

Fig. 1: State-estimation for different values of τ^{-1}. Higher values of τ^{-1} give more bias towards the target.

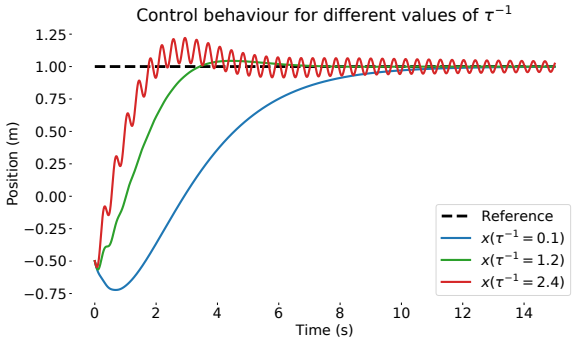

Fig. 2: Control behaviour for different values of τ^{-1}. Higher values of τ^{-1} potentially introduce overshoot oscillations.

To illustrate this, consider the mass damper system given by the equation: $\ddot{x} = a(t) - k_1 x - k_2 \dot{x}$, where x is the position of the mass, $a(t)$ the control action, k_1 the spring constant (set to $1N/m$), k_2 the damper coefficient (set to $0.1Ns/m$) and the system has unit mass. We simulated it with initial conditions $x(0) = -0.5m$, $\dot{x}(0) = -1m/s$ and $a(t) = 0N$. Figure 1 shows the results for the simulation when solving Equation 6 for different values of τ^{-1}.

It is clear that higher values of τ^{-1} give more bias towards the target. For $\tau^{-1} = 8$ (green line), the estimate is close to the target (black dashed line) and far away form the actual position (blue dashed line) as opposed to setting $\tau^{-1} = 0.1$ (red line). If $\tau^{-1} \to 0$, the estimation step reduces to a pure estimator, which would follow the trajectory without any bias towards the target.

The controller steers the system based on the observation error terms ($\boldsymbol{\varepsilon_o}$ and $\boldsymbol{\varepsilon_{o'}}$). Since larger values of τ^{-1} move the estimate more towards the target, the difference given by $\mathbf{o} - \boldsymbol{\mu}$ is larger and thus the controller is more aggressive. An illustration for the control behaviour given different values of τ^{-1} is shown in Figure 2.

D. Limitations of the reactive formulation

As explained in [6], the presented formulation has two extremes depending on the value of τ^{-1}. If $\tau^{-1} \to 0$, the estimation step has zero bias towards the target. As

Figure 1 has shown, for very small values of τ^{-1}, the estimator follows the real position without bias and thus only performing state estimation. The control action in this case will never steer the system towards the target. On the other hand, if If $\tau^{-1} \to \infty$ the system is completely biased towards the target. In this formulation the approach performs no state-estimation at all and the controller is equivalent to a PID controller. For any other value for τ^{-1}, there is a compromise between estimation and control.

Additionally, we also recognize that it is not intuitive to have an implicit notion of actions in the model. Recall that F does not include actions a, but its relationships is inferred from observations through the chain rule (Equation 7).

III. Predictive Controller

Given the previous limitations, we present a predictive controller which explicitly models actions and transitions to future states.

A. Predictive generative model

The model for the predictive controller includes the current state $\mathbf{s_t}$, the future state $\mathbf{s_{t+1}}$, the control action a and the observation o. The aim is to compute $p(\mathbf{s_t}, \mathbf{s_{t+1}}, \mathbf{a}|\mathbf{o})$. Similarly to the reactive case we approximate this distribution with a variational distribution, use the mean-field assumption and utilize the Laplace approximation. The posteriors over states $\mathbf{s_t}$ and $\mathbf{s_{t+1}}$ would have the means, $\tilde{\boldsymbol{\mu}}_s$ and $\tilde{\boldsymbol{\mu}}_{s+1}$ respectively. The distribution over actions is also assumed Gaussian with a mean of $\boldsymbol{\mu}_a$. This results in the following model:

$$\begin{aligned} p_t(\tilde{\boldsymbol{\mu}}_s, \tilde{\boldsymbol{\mu}}_{s+1}, \tilde{\mathbf{o}}, \boldsymbol{\mu}_a) \propto \quad & p(\tilde{\mathbf{o}}|\tilde{\boldsymbol{\mu}}_s) p(\tilde{\boldsymbol{\mu}}_s) p(\boldsymbol{\mu}_a) \\ & p(\tilde{\boldsymbol{\mu}}_{s+1}|f(\tilde{\boldsymbol{\mu}}_s, \boldsymbol{\mu}_a)) p(\tilde{\boldsymbol{\mu}}_{s+1}), \end{aligned} \tag{8}$$

where $p(\tilde{\mathbf{o}}|\tilde{\boldsymbol{\mu}}_s)$ is the observation model, similar to the reactive controller. The term $p(\tilde{\boldsymbol{\mu}}_s)$ is a prior on the current state $\tilde{\boldsymbol{\mu}}_s$ given the result from the previous time step. The last term, $p(\tilde{\boldsymbol{\mu}}_{s+1})$, is a prior on the next state which sets it to the desired target (from the reference trajectory). $p(\tilde{\boldsymbol{\mu}}_{s+1}|f(\tilde{\boldsymbol{\mu}}_s, \boldsymbol{\mu}_a))$, where $f(\tilde{\boldsymbol{\mu}}_s, \boldsymbol{\mu}_a)$ is a forward dynamic model that predicts future states. Finally, the prior over actions $p(\boldsymbol{\mu}_a)$ can embed any information about the dynamics, but set to a Gaussian with mean zero and a large variance ($\Sigma^{-1} \to 0$) otherwise.

Intuitively, we are not only using past information and current measurements to obtain a current state estimate $\tilde{\boldsymbol{\mu}}_s$, but also using information from dynamic models $f(\tilde{\boldsymbol{\mu}}_s, \boldsymbol{\mu}_a)$ to compute an action $\boldsymbol{\mu}_a$ that will enforce the next state $\tilde{\boldsymbol{\mu}}_{s+1}$ to be closer to the desired reference. This contrasts with a filtering scheme in which we are only concerned about the current state estimate, and also with a classic predictive controller in which we assume the current state is given.

B. Free-energy for a predictive controller

Again, given that all distributions are Gaussian, the expression for F becomes a sum of quadratic terms and the natural logarithm of the covariances as:

$$F = \frac{1}{2}(\sum_i \boldsymbol{\varepsilon}_i^\top \Sigma_i^{-1} \boldsymbol{\varepsilon}_i + \ln|\Sigma_i|) + C, \tag{9}$$

where $i \in \{\mathbf{o}, \mathbf{o}', \boldsymbol{\mu}, \boldsymbol{\mu}', \mathbf{p}, \mathbf{p}', \mathbf{d}, \mathbf{d}', \mathbf{a}\}$. There errors terms are defined as:

$$\begin{aligned} \boldsymbol{\varepsilon_o} = \mathbf{o} - \boldsymbol{\mu}_s, \quad \boldsymbol{\varepsilon_{o'}} = \mathbf{o}' - \boldsymbol{\mu}'_s, \\ \boldsymbol{\varepsilon_\mu} = \boldsymbol{\mu}_{s+1} - f(\mathbf{o}, \boldsymbol{\mu}_s), \quad \boldsymbol{\varepsilon_{\mu'}} = \boldsymbol{\mu}'_{s+1} - f(\mathbf{o}', \boldsymbol{\mu}'_s), \\ \boldsymbol{\varepsilon_p} = \boldsymbol{\mu}_s - \boldsymbol{\mu}_p, \quad \boldsymbol{\varepsilon_{p'}} = \boldsymbol{\mu}'_s - \boldsymbol{\mu}'_p, \\ \boldsymbol{\varepsilon_d} = \boldsymbol{\mu}_{s+1} - \boldsymbol{\mu}_d, \quad \boldsymbol{\varepsilon_{d'}} = \boldsymbol{\mu}'_{t+1} - \boldsymbol{\mu}'_d. \end{aligned}$$

Finally, if we consider a feedforward signal ($s(t)$),

$$\boldsymbol{\varepsilon_a} = \boldsymbol{\mu}_a - s(t),$$

otherwise $\boldsymbol{\varepsilon_a}$ is set to zero. The feedforward function is only dependant on time. This function has to be based on an accurate dynamic model since it does not account for any disturbances or imperfections in the model. Even when the feedforward function $s(t)$ is used, it should have a higher uncertainty in the optimization compared to the forward dynamic model $f(\tilde{\boldsymbol{\mu}}_s, \boldsymbol{\mu}_a)$.

C. Estimation and control

For the reactive case, every optimization was done by taking one step of gradient descent separately. The same is done in the predictive case. For every time-step t we run the following update rules:

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_s &\leftarrow \tilde{\boldsymbol{\mu}}_s - k_\mu \frac{\partial F}{\partial \tilde{\boldsymbol{\mu}}_s} \\ \tilde{\boldsymbol{\mu}}_{s+1} &\leftarrow \tilde{\boldsymbol{\mu}}_{s+1} - k_\mu \frac{\partial F}{\partial \tilde{\boldsymbol{\mu}}_{s+1}} \\ \boldsymbol{\mu}_a &\leftarrow \boldsymbol{\mu}_a - k_a \frac{\partial F}{\partial \boldsymbol{\mu}_a}, \end{aligned} \tag{10}$$

where k_a is the gradient descent step for actions and is set to a much larger value that k_μ. Subsequently , we move to the next time-step and $\tilde{\boldsymbol{\mu}}_s$ is set to the prior for the next time step $\tilde{\boldsymbol{\mu}}_p$.

IV. Results

In the results section, we evaluate the the reactive and predictive approaches on a slow trajectory ($\mu_d = sin(0.3t)$) and a faster trajectory ($\mu_d = sin(t)$).

A. Tracking a slow trajectory

The response to tracking $\mu_d = sin(0.3t)$ is given in Figure 3. The reactive controller can successfully track the reference; however, with a time delay. The predictive controller does not suffer from the time delay. In Table I and Fig. 4 the error is given. The error of the reactive controller is very large; however, this is partially due to the time delay. It is also shown how including the (accurate) feedforward signal $s(t)$ improves the performance of the predictive controller.

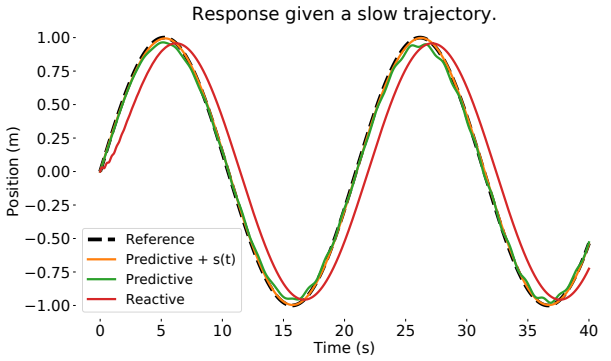

Fig. 3: Response given a slow trajectory $\mu_d = sin(0.3t)$.

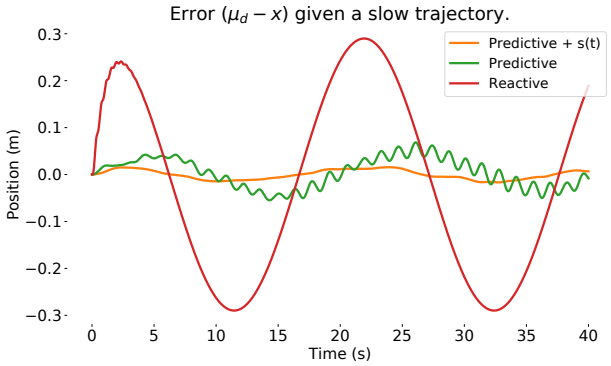

Fig. 4: Error given a slow trajectory $\mu_d = sin(0.3t)$.

	Reactive	Predictive	Predictive + $s(t)$
$\sin(0.3t)$	1.75×10^{-1}	5.34×10^{-2}	9.89×10^{-5}
$\sin(t)$	4.52×10^{-1}	1.21×10^{-2}	4.78×10^{-3}

TABLE I: The Mean Absolute Error (MAE) values associated with figures 3 and 5 are shown. As evident, the reactive controller has the highest error.

B. Tracking a fast trajectory

The response for tracking $\sin(t)$ is given in Figure 5. The reactive controller in this case struggles to track the reference. By re-tuning the reactive controller, the response time can be decreased; however, this introduces oscillations which get amplified over time. The predictive controller does not suffer from the time delay and has no issue tracking the fast trajectory. Again, in Table I the error is shown. For both trajectories, the predictive controller along with an accurate feedforward signal performs best.

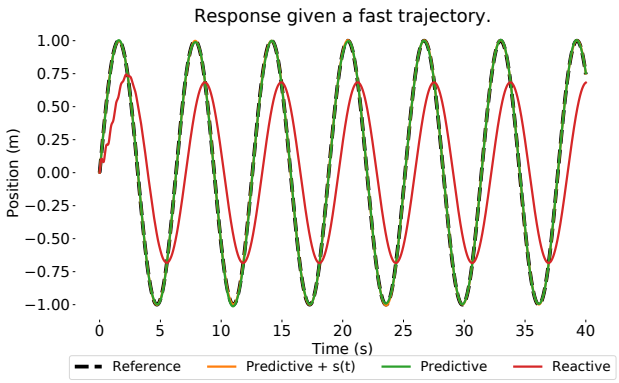

Fig. 5: Response given a fast trajectory $\mu_d = sin(t)$.

V. Conclusions and Future Work

In this work we presented a predictive controller based on variational inference. Given a reference trajectory, the controller uses its forward dynamic model to predict future states and choose appropriate actions to reach the desired trajectory. This overcomes the shortcomings of the reactive controller such as the compromise between estimation and control. For the predictive controller the actions are explicit in the free-energy expression.

In [6] the covariance and the temporal parameter τ (used in the reactive controller) were estimated during execution. Future work will focus on learning appropriate variances for the predictive controller. Additionally, model parameters can be estimated during the optimization process as well (the spring constant for instance).

Furthermore, the predictive model can be extended to include future H time-steps which would allow the agent to plan ahead. This is intimately related to work on planning as inference [11]. This extensions could be efficiently implemented using Probabilistic Graphical Models such as factor graphs [12].

References

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

[2] Karl Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11:127 EP –, 01 2010.

[3] Lancelot Da Costa, Thomas Parr, Noor Sajid, Sebastijan Veselic, Victorita Neacsu, and Karl Friston. Active inference on discrete state-spaces: a synthesis. *arXiv preprint arXiv:2001.07203*, 2020.

[4] Guillermo Oliver, Pablo Lanillos, and Gordon Cheng. Active inference body perception and action for humanoid robots. *arXiv preprint arXiv:1906.03022*, 2019.

[5] Corrado Pezzato, Riccardo MG Ferrari, and Carlos Hernandez. A novel adaptive controller for robot manipulators based on active inference. *IEEE Robotics and Automation Letters*, 2020.

[6] Mohamed Baioumy, Paul Duckworth, Bruno Lacerda, and Nick Hawes. Active inference for integrated state-estimation, control, and learning, 2020.

[7] Mohamed Baioumy, Matias Mattamala, Paul Duckworth, Bruno Lacerda, and Nick Hawes. Adaptive manipulator control using active inference with precision learning. In *UKRAS20 Conference: "Robots into the real world" Proceedings*, Lincoln, United Kingdom, May 2020.

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

[9] 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.

[10] Karl Friston. Hierarchical models in the brain. *PLoS computational biology*, 4(11):e1000211, 2008.

[11] Matthew Botvinick and Marc Toussaint. Planning as inference. *Trends in cognitive sciences*, 16(10):485–488, 2012.

[12] Mees Vanderbroeck, Mohamed Baioumy, Daan van der Lans, Rens de Rooij, and Tiis van der Werf. Active inference for robot control: A factor graph approach. *Student Undergraduate Research E-journal!*, 5:1–5, 2019.