TU Delft

Delft University of Technology

Fault-tolerant Control of Robot Manipulators with Sensory Faults using Unbiased Active Inference

Baioumy, Mohamed; Pezzato, C.; Ferrari, Riccardo ; Hernández Corbato, Carlos; Hawes, Nick

DOI
10.23919/ECC54610.2021.9654913

Publication date
2021

Document Version
Final published version

Published in
Proceedings of the European Control Conference (ECC 2021)

Citation (APA)
Baioumy, M., Pezzato, C., Ferrari, R., Hernández Corbato, C., & Hawes, N. (2021). Fault-tolerant Control of Robot Manipulators with Sensory Faults using Unbiased Active Inference. In *Proceedings of the European Control Conference (ECC 2021)* (pp. 1119-1125). IEEE. https://doi.org/10.23919/ECC54610.2021.9654913

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

2021 European Control Conference (ECC)
June 29 - July 2, 2021. Rotterdam, Netherlands

Fault-tolerant Control of Robot Manipulators with Sensory Faults using Unbiased Active Inference

Mohamed Baioumy[1*], Corrado Pezzato[2*], Riccardo Ferrari[2], Carlos Hernández Corbato[2] and Nick Hawes[1]

***Abstract*— This work presents a novel fault-tolerant control scheme based on active inference. Specifically, a new formulation of active inference which, unlike previous solutions, provides unbiased state estimation and simplifies the definition of probabilistically robust thresholds for fault-tolerant control of robotic systems using the free-energy. The proposed solution makes use of the sensory prediction errors in the free-energy for the generation of residuals and thresholds for fault detection and isolation of sensory faults, and it does not require additional controllers for fault recovery. Results validating the benefits in a simulated 2-DOF manipulator are presented, and future directions to improve the current fault recovery approach are discussed.**

***Index Terms*— Fault-tolerant control, fault recovery, active inference, free-energy principle, robotics, robot manipulator.**

I. Introduction

Fault tolerant (FT) control is of vital importance for many applications such robots working in production lines [1]. In order to guarantee high standards of reliability and security, the area of FT control of robot manipulators increasingly gathered importance in recent years [2]. Many approaches have been developed for faults in actuators and sensors [3]–[7]. Model based fault tolerant techniques are amongst the most advanced approaches to tackle the problem of fault detection and isolation (FDI) for dynamical systems [8]. These methods rely on monitoring system outputs using mathematical models to generate residual signals which are compared to a threshold. Faults are detected if the threshold is exceeded, while the actions for fault recovery are usually performed through controller re-design or by switching to a different controller [9].

Regarding FT control for robot manipulators, [6] performs fault diagnosis in image-based visual servoing. The residual signal is defined as the root mean squared estimation error (RMSE) of a Kalman filter, which predicts the values of a number of features in the camera image. The residual is compared against a user-defined static threshold and fault isolation is achieved via a decision table. The controller reconfiguration is not designed in the paper, and it would require an additional element on top of the FDI scheme. Visual information from a camera can also be used to compensate for lack of observability in case of sensory faults. In [7], three different second-order sliding mode observers are used for generating residuals using independent measurements from camera images. The detection thresholds were selected to minimize the probability of false or missed alarms on the basis of simulation and experimental data. However, the visual subsystem is assumed faultless.

FDI through static thresholds can be effective, but stochastic decision theory has been shown to outperform these methods. The actual distributions of residuals can be estimated such that a user can define thresholds based on their acceptable probability of false positives [10]–[13]. The work in [14] showed fault-tolerant steering control of an agricultural vehicle for bailing, detecting sensory faults and artefacts in images through statistical learning. The sensors configuration for control was then decided at run-time.

Generally speaking, model-based fault tolerant control schemes rely on additional supervision blocks that increase system complexity. FDI and fault recovery present two main challenges: a) The definition of robust yet sensitive pairs of residuals and thresholds; b) The design of specific additional supervisory systems to accommodate large faults.

The work presented in this paper extends the active inference controller (AIC) recently developed by the authors in [15], [16]. Active inference relies on approximate Bayesian inference [17], [18] for prediction errors minimization, and it allows state estimation and control using a single cost function. This scheme showed remarkable capabilities while dealing with missing sensory information or large unmodeled dynamics, see [15], [19] respectively. The authors in [20] initially presented a fault tolerant control scheme with an AIC which showed great promise. However, this presented a few fundamental limitations which will be thoroughly discussed in Sec. II-B. Most importantly, FDI in [20] can be affected by false-positives when target state changes, and it relies on a static threshold which might be over-conservative. In this work, we: improve the state estimation of a standard AIC; develop an unbiased AIC (u-AIC); reduce the probability of false-positives and allow to easily define a probabilistically robust threshold for fault detection.

The paper is organised as follow: Sec. II presents the background on active inference for fault-tolerant control highlighting the limitations of past work. Sec. III presents a new formulation of the AIC with unbiased state estimation, while Sec. IV explains how fault tolerant control is achieved with this new formulation. Results for a simulated 2-DOF robot arm are presented in Sec. V while Sec. VI reports a summary and future challenges.

*These authors contributed equally to this work.

[1] Authors are with the Oxford Robotics Institute, University of Oxford. For correspondence {mohamed, nickh}@robots.ox.ac.uk

[2] Authors are with Delft University of Technology. For correspondace {c.pezzato, c.h.corbato, r.ferrari}@tudelft.nl

This work was supported by Ahold Delhaize

Authorized licensed use limited to: TU Delft Library. Downloaded on January 10,2022 at 15:01:38 UTC from IEEE Xplore. Restrictions apply.

II. PRELIMINARIES

This section presents the background knowledge on active inference for robot control [15], [20] required to understand and justify the need for an unbiased AIC.

A. Active inference controller (standard AIC)

The AIC in [15], [20] is defined for torque control in joint space of a generic robot manipulator (Fig. 1). The robot arm is equipped with encoders and velocity sensors with relative sensory readings $\boldsymbol{y}_q,\ \boldsymbol{y}_{\dot{q}}$. In addition, the end-effector Cartesian position $\boldsymbol{y}_v$ is made available through a camera. The system's output is represented by $\boldsymbol{y} = [\boldsymbol{y}_q,\ \boldsymbol{y}_{\dot{q}},\ \boldsymbol{y}_v] \in \mathbb{R}^d$. The proprioceptive sensors and the camera are affected by zero mean Gaussian noise $\boldsymbol{\eta} = [\boldsymbol{\eta}_q, \boldsymbol{\eta}_{\dot{q}}, \boldsymbol{\eta}_v]$. The camera is affected by barrel distortion. The AIC can be used to control the robot arm to a desired configuration $\boldsymbol{\mu}_d$, providing the control input $\boldsymbol{u} \in \mathbb{R}^m$ as torques to the single joints. To this end, the control input is computed on the basis of the so-called *free-energy* and the *generalised motions* $\tilde{\boldsymbol{\mu}}$ [20].

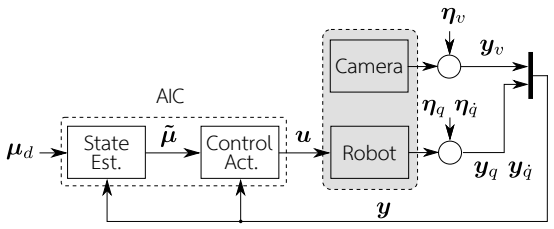

Fig. 1. General AIC control scheme for joint space torque control.

The free-energy principle relies on Bayesian inference [21] for state estimation. The goal is to find the posterior over states given observations, $p(\boldsymbol{x}|\boldsymbol{y}_v, \boldsymbol{y}_q, \boldsymbol{y}_{\dot{q}})$. This computation is in general intractable using Bayes' rule and in practice the true posterior distribution is approximated with a variational distribution $Q(\boldsymbol{x})$ [22]. The most probable state is computed by minimizing the Kullback-Leibler divergence (D_{KL}) between $p(\boldsymbol{x}|\boldsymbol{y})$ and $Q(\boldsymbol{x})$ [18]:

$$\begin{aligned} D_{KL}(Q(\boldsymbol{x})||p(\boldsymbol{x}|\boldsymbol{y})) &= \int Q(\boldsymbol{x}) \ln \frac{Q(\boldsymbol{x})}{p(\boldsymbol{x}|\boldsymbol{y})} d\boldsymbol{x} \\ &= F + \ln p(\boldsymbol{y}) \end{aligned} \tag{1}$$

where F represents the *free-energy*. By minimizing F, $Q(\boldsymbol{x})$ approaches the true posterior $p(\boldsymbol{x}|\boldsymbol{y})$. The probabilistic model $p(\boldsymbol{x}, \boldsymbol{y}_v, \boldsymbol{y}_q, \boldsymbol{y}_{\dot{q}})$ is factorized as:

$$p(\boldsymbol{x}, \boldsymbol{y}_v, \boldsymbol{y}_q, \boldsymbol{y}_{\dot{q}}) = \underbrace{p(\boldsymbol{y}_v|\boldsymbol{x}) p(\boldsymbol{y}_q|\boldsymbol{x}) p(\boldsymbol{y}_{\dot{q}}|\boldsymbol{x})}_{observation\ model} \underbrace{p(\boldsymbol{x})}_{prior} \tag{2}$$

If $Q(\boldsymbol{x})$ is a Gaussian with mean $\boldsymbol{\mu}$ and the Laplace approximation is utilized [23], then F reduces to

$$F = -\ln p(\boldsymbol{\mu}, \boldsymbol{y}_v, \boldsymbol{y}_q, \boldsymbol{y}_{\dot{q}}) + C\,, \tag{3}$$

where C is a constant. Higher order derivatives of the state $\boldsymbol{x} \in \mathbb{R}^n$ are combined in the term $\tilde{\boldsymbol{\mu}}$ (i.e. $\tilde{\boldsymbol{\mu}} = [\boldsymbol{\mu}, \boldsymbol{\mu}', \boldsymbol{\mu}'']$). This is referred to as *generalized motions* [18], [24]. The number of generalised motions considered will affect the update laws for state estimation and control.

In order to characterize the free-energy and compute it, one has to define two generative models, one for the state dynamics, and one for the observations. The latter is modeled according to [18] as $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{\mu}) + \boldsymbol{z}$, where $\boldsymbol{g}(\boldsymbol{\mu}) = [\boldsymbol{g}_q,\ \boldsymbol{g}_{\dot{q}},\ \boldsymbol{g}_v] : \mathbb{R}^d \mapsto \mathbb{R}^d$ represents the non-linear mapping between sensory data and states of the environment, and $\boldsymbol{z}$ is Gaussian noise $\boldsymbol{z} \sim (\boldsymbol{0}, \Sigma_y)$. The generative model of the state dynamics is defined as [18]:

$$\frac{d\boldsymbol{\mu}}{dt} = \boldsymbol{\mu}' = \boldsymbol{f}(\boldsymbol{\mu}) + \boldsymbol{w} \tag{4}$$

where $\boldsymbol{f}(\boldsymbol{\mu}) : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a generative function dependant on the belief about the states $\boldsymbol{\mu}$ and $\boldsymbol{w}$ is Gaussian noise $\boldsymbol{w} \sim (\boldsymbol{0}, \Sigma_\mu)$. As in previous work [16], we define $\boldsymbol{f}(\cdot)$ such that the robot will be steered to a desired joint configuration $\boldsymbol{\mu}_d$ following the dynamics of a first order linear system with time constant τ.

$$\boldsymbol{f}(\boldsymbol{\mu}) = (\boldsymbol{\mu}_d - \boldsymbol{\mu})\tau^{-1} \tag{5}$$

From equation (5) we can see that the desired state is encoded in the prior. The time constant τ influences the generative model of the state dynamics $\boldsymbol{f}(\cdot)$. As explained in [16], the AIC has two extremes depending on the value of τ^{-1}. If $\tau^{-1} \to 0$, the AIC only performs filtering and no control. On the other hand, if $\tau^{-1} \to \infty$ the AIC is equivalent to a PID controller [16], [25]. For any value in between, there is a compromise between estimation and control. The estimation and control are thus 'coupled' and state-estimation with the standard AIC results in a bias towards the desired target, see (7). If all distributions in equation (2) are assumed Gaussian, the expression for free-energy simplifies to (see [15] for complete derivations):

$$F = \frac{1}{2} \sum_i \left(\boldsymbol{\varepsilon}_i^\top P_i \boldsymbol{\varepsilon}_i - \ln |P_i| \right) + C, \tag{6}$$

where $i \in \{y_q,\ y_{\dot{q}},\ y_v,\ \mu,\ \mu'\}$, and P_i defines a precision (or inverse covariance) matrix. Note that we set $\tau = 1$ as in [15], [26]. The terms $\boldsymbol{\varepsilon}_i$ with $i \in \{y_q,\ y_{\dot{q}},\ y_v\}$ are the *Sensory Prediction Errors* (SPE), representing the difference between observed sensory input and expected one. In general, the SPE for a sensor s are defined as $(\boldsymbol{y}_s - \boldsymbol{g}_s(\boldsymbol{\mu}))$. The model prediction errors are instead defined considering the desired state dynamics as $\boldsymbol{\varepsilon}_\mu = (\boldsymbol{\mu}' - \boldsymbol{f}(\boldsymbol{\mu}))$ and $\boldsymbol{\varepsilon}_{\mu'} = (\boldsymbol{\mu}'' - \partial \boldsymbol{f}(\boldsymbol{\mu}) / \partial \boldsymbol{\mu} \boldsymbol{\mu}')$. In particular, for the robot arm used in this paper, $\boldsymbol{\varepsilon}_{y_q} = (\boldsymbol{y}_q - \boldsymbol{\mu})$, $\boldsymbol{\varepsilon}_{y_{\dot{q}}} = (\boldsymbol{y}_{\dot{q}} - \boldsymbol{\mu}')$, $\boldsymbol{\varepsilon}_v = (\boldsymbol{y}_v - \boldsymbol{g}_v(\boldsymbol{\mu}))$, and $\boldsymbol{\varepsilon}_\mu = (\boldsymbol{\mu}' + \boldsymbol{\mu} - \boldsymbol{\mu}_d)$, $\boldsymbol{\varepsilon}_{\mu'} = (\boldsymbol{\mu}'' + \boldsymbol{\mu}')$. For more details on the derivation of equation (6), refer to [15], [16], [18].

To achieve state-estimation a gradient descent on the free-energy is used. The variable $\tilde{\boldsymbol{\mu}}$ is updated as:

$$\dot{\tilde{\boldsymbol{\mu}}} = D\tilde{\boldsymbol{\mu}} - \kappa_\mu \frac{\partial F}{\partial \tilde{\boldsymbol{\mu}}}, \tag{7}$$

where κ_μ is the gradient descent step size, and D is a shifting operator with ones on the upper diagonal. This form of gradient descent is needed due to using generalized motions as mentioned earlier [17]. The control actions, are also computed through gradient descent; however, the expression

Authorized licensed use limited to: TU Delft Library. Downloaded on January 10,2022 at 15:01:38 UTC from IEEE Xplore. Restrictions apply.

for F does not include any actions explicitly. The chain rule is then used, assuming an implicit dependency between actions and sensory input.

$$\dot{\boldsymbol{u}} = -\kappa_u \frac{\partial F}{\partial \boldsymbol{u}} = -\kappa_u \frac{\partial F}{\partial \boldsymbol{y}} \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{u}}, \tag{8}$$

where κ_u is the gradient descent's step size. The term $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{u}}$ is often approximated as linear, similarly to [15], [26].

B. Limitations of fault-tolerant control with standard AIC

The AIC was used in [20] for fault tolerant control. In this subsection we aim at highlighting the main limitations of the AIC, which are addressed in Sec. III through the proposed unbiased active inference scheme. The main idea in [20] was that the SPE in the free-energy could be used as residuals for fault detection, removing the need for more advanced residual generators. Besides, the precision matrices in the free-energy could be used for fault accommodation, since they represent the trust associated to sensory readings. For a faulty sensor, the relative precision was decreased to zero to perform fault recovery. Specifically, the residuals in [20] are generated by considering the quadratic form of the SPE $\boldsymbol{\varepsilon}_s^\top P_{y_s} \boldsymbol{\varepsilon}_s$, where $\boldsymbol{\varepsilon}_s = \boldsymbol{y}_s - \boldsymbol{g}_s(\boldsymbol{\mu})$ with $s \in \{q,\ \dot{q},\ v\}$. A static fault detection threshold ψ_s was chosen as an upper bound on the quadratic terms.

The SPE depends on the belief $\boldsymbol{\mu}$, which is biased towards a given goal. This means that the SPE in standard AIC can increase for causes that are not necessarily related to a fault, i.e. the robot is stuck due to a collision or there is an abrupt change in the goal $\boldsymbol{\mu}_d$, for example in a pick-and-place application. The residuals are thus dependent on the goal, which is problematic. This was not an issue in [20] because of the over-conservative threshold used for fault detection, but it becomes apparent when the SPE are closely analysed. Fig. 2 reports the absolute value of the SPE for the joint of a robot arm controlled using an AIC, during a point to point motion with three via-points $\boldsymbol{\mu}_{d_1},\ \boldsymbol{\mu}_{d_2},\ \boldsymbol{\mu}_{d_3}$.

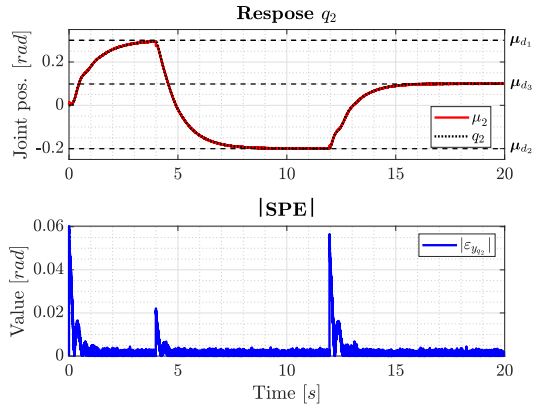

Fig. 2. Effect of abrupt change in goal on the SPE in standard AIC. The SPE increases sharply without the presence of any sensory faults.

Using a tighter threshold to detect faults of small entity by using the approach proposed in [20] would lead to many false positives. In this work we tackle the problem of biased sensory prediction errors and we define an *unbiased AIC* where the SPE are independent from the input, providing more accurate state estimation. With the u-AIC is also possible to use the sensory prediction error directly for fault detection using a probabilistically robust threshold instead of a static one as in [20].

III. Unbiased Active Inference

To tackle the problems discussed in the previous section, the unbiased AIC is developed (u-AIC).

A. Derivation of the u-AIC

Consider a probabilistic model with explicit actions $p(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}_v, \boldsymbol{y}_q, \boldsymbol{y}_{\dot{q}})$, where unlike the AIC, $\boldsymbol{x} = [\boldsymbol{q}\ \dot{\boldsymbol{q}}]^\top$. The distribution factorizes as:

$$p(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}_v, \boldsymbol{y}_q, \boldsymbol{y}_{\dot{q}}) = \underbrace{p(\boldsymbol{u}|\boldsymbol{x})}_{control} \underbrace{p(\boldsymbol{y}_v|\boldsymbol{x}) p(\boldsymbol{y}_q|\boldsymbol{x}) p(\boldsymbol{y}_{\dot{q}}|\boldsymbol{x})}_{observation\ model} \underbrace{p(\boldsymbol{x})}_{prior} \tag{9}$$

Given the sensory data, we then aim to find the posterior over states and actions $p(\boldsymbol{x}, \boldsymbol{u}|\boldsymbol{y}_v, \boldsymbol{y}_q)$. We approximate the posterior with a variational distribution $Q(\boldsymbol{x}, \boldsymbol{u})$. We utilize the mean-field assumption ($Q(\boldsymbol{x}, \boldsymbol{u}) = Q(\boldsymbol{x})Q(\boldsymbol{u})$) and the Laplace approximation. The posterior over the state $\boldsymbol{x}$ is assumed Gaussian with mean $\boldsymbol{\mu}_x$. The posterior over actions $\boldsymbol{u}$ is also assumed Gaussian with mean $\boldsymbol{\mu}_u$. This results in the following expression for variational free-energy:

$$F = -\ln p(\boldsymbol{\mu}_u, \boldsymbol{\mu}_x, \boldsymbol{y}_v, \boldsymbol{y}_q, \boldsymbol{y}_{\dot{q}}) + C \tag{10}$$

This expression can be factorized as in equation (9). Assuming Gaussian model F can be expanded to:

$$\begin{aligned} F = \frac{1}{2}(&\boldsymbol{\varepsilon}_{y_q}^\top P_{y_q} \boldsymbol{\varepsilon}_{y_q} + \boldsymbol{\varepsilon}_{y_{\dot{q}}}^\top P_{y_{\dot{q}}} \boldsymbol{\varepsilon}_{y_{\dot{q}}} + \boldsymbol{\varepsilon}_{y_v}^\top P_{y_v} \boldsymbol{\varepsilon}_{y_v} \\ &+ \boldsymbol{\varepsilon}_x^\top P_x \boldsymbol{\varepsilon}_x + \boldsymbol{\varepsilon}_u^\top P_u \boldsymbol{\varepsilon}_u - \ln |P_u P_{y_q} P_{y_{\dot{q}}} P_{y_v} P_x|) + C, \end{aligned} \tag{11}$$

where $\boldsymbol{\varepsilon}_{y_q}$, $\boldsymbol{\varepsilon}_{y_{\dot{q}}}$, $\boldsymbol{\varepsilon}_{y_v}$ are the sensory prediction errors of position encoder, velocity encoder, and the visual sensor respectively. Furthermore, $\boldsymbol{\varepsilon}_x$ and $\boldsymbol{\varepsilon}_u$ are the prediction errors for the prior on the state and the control action respectively.

The prediction error on the state prior $\boldsymbol{\varepsilon}_x$ is computed considering a prediction of the state $\hat{\boldsymbol{x}}$ at the current time-step, which is a deterministic value. It follows that $\boldsymbol{\varepsilon}_x = (\boldsymbol{\mu}_x - \hat{\boldsymbol{x}})$. The prediction $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{q}}\ \hat{\dot{\boldsymbol{q}}}]^\top$ can be computed in the same fashion as the prediction step of, for instance, a Kalman filter or of a Luenberger observer. While in such cases the prediction computation would require the knowledge of an accurate dynamic model, in this paper we assume we do not have access to that. Instead, we will propagate forward in time the current state belief using the following simplified discrete time model:

$$\hat{\boldsymbol{x}}_{k+1} = \begin{bmatrix} I & I\Delta t \\ 0 & I \end{bmatrix} \boldsymbol{\mu}_{x,k} \tag{12}$$

where I represents an unitary matrix of suitable size. This form assumes that the velocities of the joints will remain roughly constant and the position of each joint is thus

Authorized licensed use limited to: TU Delft Library. Downloaded on January 10,2022 at 15:01:38 UTC from IEEE Xplore. Restrictions apply.

computed as the discrete time integral of the velocity, using a first-order Euler scheme. As mentioned, if one has access to a better dynamic model, that can be used instead.

Finally, the state prior now does not encode information about the target $\boldsymbol{\mu}_d$ (unlike in the standard AIC). The information about the target is encoded in the distribution $p(\boldsymbol{u}|\boldsymbol{x})$. We choose this distribution to be Gaussian as well with a mean of $f^*(\boldsymbol{\mu}_x, \boldsymbol{\mu}_d)$, which is a function that steers the systems toward the target. This then results in $\boldsymbol{\varepsilon}_u = (\boldsymbol{\mu}_u - f^*(\boldsymbol{\mu}_x, \boldsymbol{\mu}_d))$. The function $f^*(\boldsymbol{\mu}_x, \boldsymbol{\mu}_d)$ can be *any* controller, for instance a P controller: $f^*(\boldsymbol{\mu}_x, \boldsymbol{\mu}_d)) = P(\boldsymbol{\mu}_d - \boldsymbol{\mu}_x)$. In this paper we choose the function such that it converges to a PID controller; however, other choices can be made without loss of generality. For state and actions update, first-order Euler integration is used to obtain discrete time equations.

B. Definition of the observation model

The sensory prediction errors are as in the AIC, that is $\boldsymbol{\varepsilon}_q = (\boldsymbol{y}_q - \boldsymbol{\mu})$, $\boldsymbol{\varepsilon}_{\dot{q}} = (\boldsymbol{y}_{\dot{q}} - \boldsymbol{\mu}')$ and $\boldsymbol{\varepsilon}_v = (\boldsymbol{y}_v - \boldsymbol{g_v}(\boldsymbol{\mu}))$ but $\boldsymbol{\mu}$ is not biased anymore towards the target $\boldsymbol{\mu}_d$. The relation between $\boldsymbol{\mu}$ and $\boldsymbol{y}$ is expressed through the generative model of the sensory input $\boldsymbol{g} = [\boldsymbol{g}_q,\ \boldsymbol{g}_{\dot{q}},\ \boldsymbol{g}_v]$. Position and velocity encoders directly measure the state. Thus $\boldsymbol{g}_q$ and $\boldsymbol{g}_{\dot{q}}$ are linear mappings between $\boldsymbol{\mu}$ and $\boldsymbol{y}$.

To define $\boldsymbol{g}_v(\boldsymbol{\mu})$, instead, we use a *Gaussian Process Regression* (GPR) as in [19]. This is particularly useful because we can model the noisy and distorted sensory input from the camera, and at the same time we can compute a closed form for the derivative of the process with respect to the beliefs $\boldsymbol{\mu}$, required for the state update laws. The training data is composed by a set of observations of the (planar) camera output $[\bar{\boldsymbol{y}}_{v_x},\ \bar{\boldsymbol{y}}_{v_z}]^\top$ in several robot configurations $\bar{\boldsymbol{y}}_q$. We use a squared exponential kernel k of the form:

$$\begin{aligned} k(\boldsymbol{y}_{q_i}, \boldsymbol{y}_{q_j}) \quad &= \sigma_f^2 \exp\left(-\tfrac{1}{2}(\boldsymbol{y}_{q_i} - \boldsymbol{y}_{q_j})^\top \Theta (\boldsymbol{y}_{q_i} - \boldsymbol{y}_{q_j})\right) \\ &\quad + \sigma_n^2 d_{ij} \end{aligned} \tag{13}$$

where $\boldsymbol{y}_{q_i},\ \boldsymbol{y}_{q_j} \in \bar{\boldsymbol{y}}_q$, and d_{ij} is the Kronecker delta function. Θ is a diagonal matrix of hyperparameters that are fit according to the training data to obtain accurate predictions. The hyperparameters are optimized in order to maximize the marginal likelihood, see [27]. The predictions are then [19]:

$$\begin{aligned} \boldsymbol{g}_v(\boldsymbol{y}_{\boldsymbol{q}_*}) &= \begin{bmatrix} k(\boldsymbol{y}_{\boldsymbol{q}_*}, \bar{\boldsymbol{y}}_q) K^{-1} \bar{\boldsymbol{y}}_{v_x} \\ k(\boldsymbol{y}_{\boldsymbol{q}_*}, \bar{\boldsymbol{y}}_q) K^{-1} \bar{\boldsymbol{y}}_{v_z} \end{bmatrix} \\ \boldsymbol{g}_v(\boldsymbol{y}_{\boldsymbol{q}_*})' &= \begin{bmatrix} -\Theta^{-1}(\boldsymbol{y}_{\boldsymbol{q}_*} - \bar{\boldsymbol{y}}_q)^\top [k(\boldsymbol{y}_{\boldsymbol{q}_*}, \bar{\boldsymbol{y}}_q)^\top \cdot \boldsymbol{\alpha}_x] \\ -\Theta^{-1}(\boldsymbol{y}_{\boldsymbol{q}_*} - \bar{\boldsymbol{y}}_q)^\top [k(\boldsymbol{y}_{\boldsymbol{q}_*}, \bar{\boldsymbol{y}}_q)^\top \cdot \boldsymbol{\alpha}_z] \end{bmatrix} \end{aligned} \tag{14}$$

where $\cdot$ means element-wise multiplication, K is the covariance matrix, $\boldsymbol{\alpha}_x = K^{-1}\bar{\boldsymbol{y}}_{v_x}$ and $\boldsymbol{\alpha}_z = K^{-1}\bar{\boldsymbol{y}}_{v_z}$.

C. Estimation and control using the u-AIC

Similar to the AIC, performing both state-estimation and control can be achieved using gradient descent on F. This is simpler in the case of the u-AIC since there is no need to use the chain rule when computing the control actions (see eq. (8)) or to consider generalized motion (7):

$$\dot{\boldsymbol{\mu}}_u = -\kappa_u \frac{\partial F}{\partial \boldsymbol{\mu_u}}, \quad \dot{\boldsymbol{\mu}}_x = -\kappa_\mu \frac{\partial F}{\partial \boldsymbol{\mu_x}}, \tag{15}$$

where κ_u and κ_μ are the gradient descent step sizes. The gradient on control can be computed as:

$$\frac{\partial F}{\partial \boldsymbol{\mu}_u} = P_u(\boldsymbol{\mu}_u - f^*(\boldsymbol{\mu}_x, \boldsymbol{\mu}_d)) \tag{16}$$

Setting this to zero results in the control action being equal to $f^*(\boldsymbol{\mu}_x, \boldsymbol{\mu}_d)$. If this function is chosen similar to a PID controller, then the control law of our system will converge towards that. This can be extended to richer control by modifying the probabilistic model in eq. (9). The only term depending on the control action now is $p(\boldsymbol{u}|\boldsymbol{x})$; however, a prior $p(\boldsymbol{u})$ can be also added. In that case the generative model would be:

$$\frac{1}{\alpha} p(\boldsymbol{u}) p(\boldsymbol{u}|\boldsymbol{x}) p(\boldsymbol{y}_v|\boldsymbol{x}) p(\boldsymbol{y}_q|\boldsymbol{x}) p(\boldsymbol{y}_{\dot{q}}|\boldsymbol{x}) p(\boldsymbol{x}) \tag{17}$$

where α is a normalization constant. This prior can then encode a feed-forward signal (open-loop control law). When computing the $\boldsymbol{u}$ that minimizes F in that case, it would be a combination of the feed-forward signal and $f^*(\boldsymbol{\mu}_x, \boldsymbol{\mu}_d)$ depending on the value of Σ_u. This is employed in [28].

Since $\boldsymbol{\mu}_u$ converges to $f^*(\boldsymbol{\mu}_x, \boldsymbol{\mu}_d)$, we can achieve unbiased state estimation. In fact, this would result in $\boldsymbol{\varepsilon}_u = 0$ and thus the current target would not affect the the beliefs about the states. In u-AIC, the belief about the states is only affected by the predicted values and the observations from various sensor, exactly as in a Kalman filter [29].

D. Key differences between AIC and u-AIC

The AIC considers the relationship between states and observations without explicitly modelling the control actions. The prior over state $p(\boldsymbol{x})$ in equation (2) thus encodes the target/goals state. In filters, such as the Kalman filter, the prior comes from a prediction step that relies on the previous state and action. In case of the AIC, the beliefs update law is essentially a filter where the prediction step is biased towards the goals state i.e. the agent normally predicts to move linearly towards the target (see eq. (5)).

In the u-AIC, actions are explicitly modelled and the target state is encoded in the term $p(\boldsymbol{u}|\boldsymbol{x})$. This has two implications. First, the state is not biased towards the target since its prior $p(\boldsymbol{x})$ is not. A prior encoding a prediction step will also improve the overall state-estimation accuracy. Secondly, explicit actions allow us to directly perform gradient descent on F with respect to $\boldsymbol{u}$. This is not possible in the general AIC case and the chain rule had to be utilized (see eq. (8)).

Finally, the u-AIC has guarantees regarding the stability of the controller while the AIC does not. This is out of scope of this paper and will be discussed in future work.

IV. Fault detection, isolation, and recovery

In this section we describe how the SPE from the u-AIC can be used as residual signals, and how to perform FDI. Without loss of generality, for these derivations we will consider only the proprioceptive sensors and we discretize the continuous dynamics with first order Euler integration. According to the approximated system's dynamics in (12),

Authorized licensed use limited to: TU Delft Library. Downloaded on January 10,2022 at 15:01:38 UTC from IEEE Xplore. Restrictions apply.

and expanding the second term in (15) where F is computed as in (11), the state estimation law can be rewritten as:

$$\boldsymbol{\mu}_{k+1} = \gamma(\boldsymbol{\mu}_k, \boldsymbol{u}_k) + \Lambda(\boldsymbol{g}(\boldsymbol{\mu}_k) - \boldsymbol{y}_k) \tag{18}$$

Equation (18) represents the dynamics of an estimator where stability results from the value that we obtain for diagonal matrix Λ. in [29] it has been shown how the free-energy principle can be used to derive stable state observers, for a linear case with coloured noise. It is important to note that γ and Λ are known expressions resulting from the partial derivatives of F. The term $(\boldsymbol{y}_k - \boldsymbol{g}(\boldsymbol{\mu}_k))$ represents the sensory prediction errors. In the u-AIC, the states of the system are steered towards the desired goal following the dynamics imposed by the controller. Therefore, the resulting system's dynamics can be represented in general as:

$$\begin{cases} \boldsymbol{x}_{k+1} = \gamma(\boldsymbol{x}_k, \boldsymbol{u}_k) + \phi(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\rho}_k) \\ \boldsymbol{y}_k = \boldsymbol{x}_k + \boldsymbol{z}_k \end{cases} \tag{19}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ and $\boldsymbol{u}_k \in \mathbb{R}^m$ are the state and input variables, while $\gamma(\boldsymbol{x}_k, \boldsymbol{u}_k) : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ represents the dynamics of the system in healthy conditions. $\boldsymbol{y}_k \in \mathbb{R}^n$ is the measurement of the full state which is affected by the presence of measurement noise $\boldsymbol{z}_k \in \mathbb{R}^n$. The expressions obtained so far for the system dynamics will allow us to cast easily a model-based fault diagnosis approach in the current framework, even if the knowledge of an a-priori model is not needed. The real system can be affected by faults which are represented by the fault function $\phi(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\rho}_k) : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \mapsto \mathbb{R}^n$. The unknown parameter $\boldsymbol{\rho}_k$ determines the fault amplitude and shall be such that $\phi(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{0}) = 0$.

A. Residual generation

Following [12], [30] we rely on two assumptions:

Assumption 1: No fault acts on the system before the fault time k_f, thus $\boldsymbol{\rho}_k = 0$ for $0 \leq k < k_f$. In addition, before and and after the occurrence of a fault the variables $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$ remain bounded, that is $\exists\ \mathcal{S} = \mathcal{S}^{\boldsymbol{x}} \times \mathcal{S}^{\boldsymbol{u}} \subset \mathbb{R}^n \times \mathbb{R}^m$ such that $(\boldsymbol{x}_k,\ \boldsymbol{u}_k) \in \mathcal{S}\ \forall k$.

Assumption 2: $\boldsymbol{z}_k$ is a random variable defined on some probability spaces $(\mathcal{Z}, \mathscr{B}, \mathbf{P}_{\mathcal{Z}})$, where $\mathcal{Z} \subseteq \mathbb{R}^n$. $\mathscr{B}(\cdot)$ indicates a Borel σ-algebra, and $\mathbf{P}_{\mathcal{Z}}$ is the probability measures defined over $\mathcal{Z}$. Furthermore, it is not correlated and are independent from $\boldsymbol{x}_k$, $\boldsymbol{u}_k$ and $\boldsymbol{\rho}_k\ \forall k$.

We define the residuals for fault detection as the sensory prediction errors $\boldsymbol{r}_k = \boldsymbol{y}_k - \boldsymbol{g}(\boldsymbol{\mu}_k)$. Thus, according to (19), (18) and following [12], [30], the residuals dynamics will be given by:

$$\boldsymbol{r}_{k+1} = \Lambda \boldsymbol{r}_k + \boldsymbol{\delta}_k + \phi(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\rho}_k) \tag{20}$$

$$\boldsymbol{\delta}_k = \gamma(\boldsymbol{y}_k - \boldsymbol{z}_k, \boldsymbol{u}_k) - \gamma(\boldsymbol{\mu}_k, \boldsymbol{u}_k) + \boldsymbol{z}_{k+1} \tag{21}$$

The stochastic process $\boldsymbol{\delta}_k$, is the random total uncertainty which affects the residuals generation. It follows that $\boldsymbol{\delta}_k$ is in the probability space $(\Delta_k, \mathscr{B}(\Delta_k), \mathbf{P}_{\boldsymbol{\delta}_k})$ where Δ_k is obtained by letting $\boldsymbol{z}_k$ and $\boldsymbol{z}_{k+1}$ vary over $\mathcal{Z}$. Apart from simple cases, it is not possible to obtain a closed form for this set, thus numerical approximations are used instead. The residual $\boldsymbol{r}_{k+1}$ can be seen as a random variable in the same probability space of $\boldsymbol{\delta}_k$ [12].

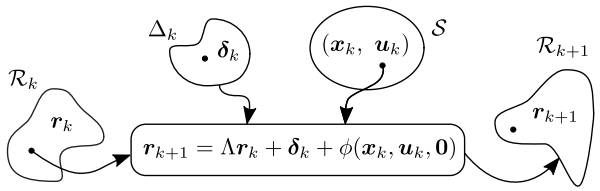

Fig. 3. Healthy residual set at time $k+1$ as image obtained from the output of (20).

The residual set $\mathcal{R}_{k+1}$ at the next time step $k+1$ can be seen as the image obtained by computing the output of (20), where the total uncertainty $\boldsymbol{\delta}_k$ varies over its domain Δ_k. Note that the healthy residuals can be characterized by setting $\boldsymbol{\rho}_k = 0$.

Remark 1: While the system dynamics γ and the fault function ϕ have been introduced for analysis purposes, they do not correspond to known models of the healthy and faulty behaviours. In particular, γ is obtained via algebraic manipulations from current free energy framework's update equations. In particular, the residual r for implementing the detection and isolation logic described next can be directly computed from the current belief and measurement. Indeed, the proposed fault diagnosis approach based on the u-AIC is completely model-free.

B. Threshold for fault detection

As in [30], we consider a probabilistically robust detection logic of the form:

$$d_M(\boldsymbol{r}_{k+1}) \leq \frac{n}{\alpha} \triangleq \overline{d_M} \tag{22}$$

where $\overline{d_M}$ is the detection threshold, n is the dimension of the residuals, α is a tuning parameter, and $d_M(\cdot)$ indicates the Mahalanobis distance of the residuals. In general:

$$d_M(\boldsymbol{r}) = \sqrt{(\boldsymbol{r} - \bar{\boldsymbol{r}})^\top C_r^{-1} (\boldsymbol{r} - \bar{\boldsymbol{r}})} \tag{23}$$

where $\bar{\boldsymbol{r}} \triangleq \mathbb{E}[\boldsymbol{r}] \in \mathbb{R}^n$ and $C_r \triangleq Cov[\boldsymbol{r}] \in \mathbb{R}^{n \times n}$ are the expected value and covariance matrix of the residuals. According to the Multivariate Chebyshev Inequality [31]:

$$Pr[d_M(\boldsymbol{r}_k) \geq \frac{n}{\alpha}] \leq \alpha, \qquad \forall \alpha \in [0,\ 1] \tag{24}$$

This means that, in healthy conditions, the probability of a false alarm, that is d_M exceeding the threshold $\bar{d}_M$, is upper bounded by α. The value α can be tuned to achieve the desired probabilistic robustness of the threshold. The present detection logic is equivalent to check if the residual at time step $k+1$ belongs to a time varying ellipsoid with mean $\bar{\boldsymbol{r}}_{k+1}$ and covariance $C_{r_{k+1}}$. In the general nonlinear case, these moments can be approximated by their sampled counterparts obtained in healthy conditions.

Authorized licensed use limited to: TU Delft Library. Downloaded on January 10,2022 at 15:01:38 UTC from IEEE Xplore. Restrictions apply.

C. Fault isolation

The detection policy in (22) results effective to label the system as healthy or faulty. However, it is not possible to use the same SPE generated through equation (11) for fault isolation. In fact, the free-energy is computed as a weighted sum of the squared prediction errors fusing different sensory sources. This means that when a fault occurs, its effects will propagate to all the SPE in equation (11). We define two additional free-energies and state estimation processes which rely on different sensory sources (i.e. either proprioceptive or visual).

$$F_p = \frac{1}{2}(\boldsymbol{\varepsilon}_{\boldsymbol{y_q}}^\top P_{y_q} \boldsymbol{\varepsilon}_{\boldsymbol{y_q}} + \boldsymbol{\varepsilon}_{\boldsymbol{y_{\dot{q}}}}^\top P_{y_{\dot{q}}} \boldsymbol{\varepsilon}_{\boldsymbol{y_{\dot{q}}}} + \boldsymbol{\varepsilon}_{\boldsymbol{x}}^\top P_x \boldsymbol{\varepsilon}_{\boldsymbol{x}} - \ln |P_{y_q} P_{y_{\dot{q}}} P_x|), \tag{25}$$

$$F_v = \frac{1}{2}(\boldsymbol{\varepsilon}_{\boldsymbol{y_v}}^\top P_{y_v} \boldsymbol{\varepsilon}_{\boldsymbol{y_v}} + \boldsymbol{\varepsilon}_{\boldsymbol{x}}^\top P_x \boldsymbol{\varepsilon}_{\boldsymbol{x}} - \ln |P_{y_v} P_x|) \tag{26}$$

Doing so, we can isolate encoder and camera faults since state estimation is now done using two independent measurement sources. F_p and F_v are only used for fault isolation and not control. The thresholds for the SPE from (25) and (26) are then computed as in Sec. IV-B.

D. Fault recovery

Fault recovery can be performed as in [20]. Once a fault is detected and isolated, fault recovery is triggered. If a sensor is marked as faulty, its corresponding precision matrix is set to zero. The controller can thus exploit the sensory redundancy to a) have a better a posteriori approximation of the states, and b) to compensate for missing or wrong sensory data. Once a fault is detected and isolated, the precision matrix of the faulty sensor P_{fs} is reduced to zero:

$$P_{fs} = \mathbf{0} \tag{27}$$

Interestingly, the active inference framework allows also for hyper-parameters (precision) learning [16], [32]. The bias in the standard AIC hindered hyper-parameters learning for fault recovery purposes, but learning meaningful precision matrices associated with sensory readings with u-AIC is now possible and will be investigated in future work.

V. Simulation Study

In this section we illustrate the fault detection, isolation, and recovery strategy on a MATLAB simulation using a 2-DOF robotic manipulator. The dynamical model of the robot is defined as [33]:

$$\boldsymbol{u} = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + D\dot{\boldsymbol{q}} + G(\boldsymbol{q}) \tag{28}$$

where $\boldsymbol{q} = [q_1,\ q_2]^\top$, $\boldsymbol{u}(t) = [u_1,\ u_2]^\top$, D is the friction coefficient matrix, M is the inertia matrix, C is the Coriolis matrix, and G models the effects of gravity. The dynamic model is only used to simulate the response to the commanded torques since the u-AIC provides a control law agnostic to the dynamical model. Thus, link's masses, friction coefficients, and other dynamical parameters are not part of the controller. The noisy sensory readings have zero-mean Gaussian noise. The standard deviation of the noise for encoders and velocity sensors is set to $\sigma_q = \sigma_{\dot{q}} = 0.001$, while the one for the camera to $\sigma_v = 0.01$.

The simulation consists of controlling the robot arm for a double point-to-point motion from the starting configuration $\boldsymbol{q} = [-\pi/2,\ 0]\ (rad)$ to $\boldsymbol{\mu}_{d1} = [-0.2,\ 0.5]\ (rad)$, and then $\boldsymbol{\mu}_{d2} = [-0.6,\ 0.2]\ (rad)$. A fault occurs during the trajectory from $\boldsymbol{\mu}_{d1}$ to $\boldsymbol{\mu}_{d2}$, i.e. we set $k_f = 8s$. The fault can affect either the encoders or the camera. The encoder fault is such that the output related to the first joint freezes so $\boldsymbol{y}_q(k) = [q_1(k_f),\ q_2(k)]^\top$ for $k \geq k_f$. The fault detection and recovery of such a fault, as well as the system's response, are reported below in Fig. 4 and Fig. 5:

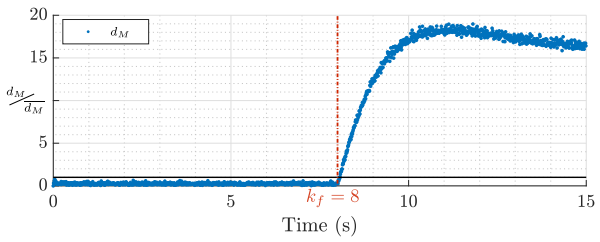

Fig. 4. Normalised d_M in case of encoder fault. A detection occur when the normalized value exceeds 1.

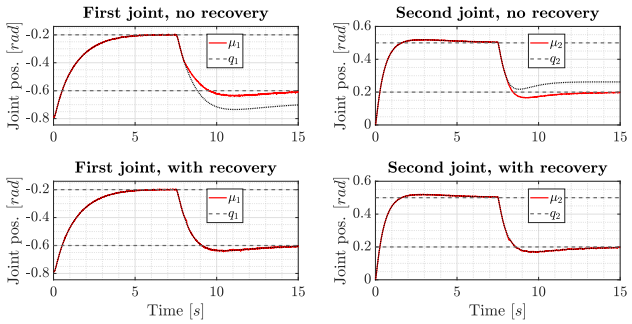

Fig. 5. System's response in case of encoder fault without recovery (fixed precision matrices) and with recovery (setting $P_{y_q} = \mathbf{0}$ after detection). The fault occurs after 8s.

As can be seen from Fig. 5, the system is not able to reach the set-point after the occurrence of the fault in case of no recovery. The robot arm reaches a different configuration to minimise the free-energy, which is built fusing the sensory information from the (faulty) encoders and the (healthy) camera. Recovery happens after 60 (ms), and after that the faulty reading is discarded by setting zero precision. The robot arm is then able to reach the final configuration.

The other situation that we consider in this work is a camera fault. This fault is supposed to be a misalignment, due for instance to a collision. This is simulated by injecting a bias i.e. $\boldsymbol{y}_v(k) = \boldsymbol{y}_v(k) + 0.04$ for $k \geq k_f$. Note that the entity of this fault is small and comparable with the camera noise, i.e. the zero mean Gaussian noise with $\sigma_v = 0.01\ (m)$. The fault detection is reported in Fig. 6. The system's response with and without recovery is similar to Fig. 5.

The simulation results will be extended to real experiments, leveraging the scalability and adaptability of the active inference controller [15]. The steady state errors (e_{ss}) and

Authorized licensed use limited to: TU Delft Library. Downloaded on January 10,2022 at 15:01:38 UTC from IEEE Xplore. Restrictions apply.

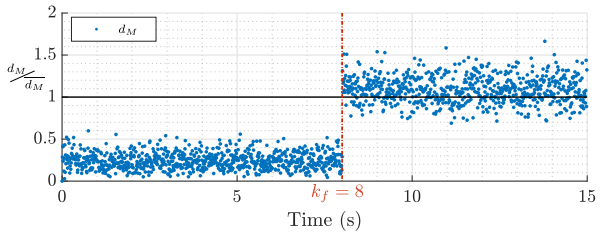

Fig. 6. Normalised d_M in case of camera fault.

the $RMSE$ between beliefs and actual joint positions for the simulations are reported in Table I.

TABLE I

SUMMARY OF THE RESULTS USING U-AIC

Metrics	Encoder Fault		Camera Fault	
	Recovery	No recovery	Recovery	No recovery
$e_{ss,\ q1}$	$-2.7e^{-3}$	0.1674	$-2.1e^{-4}$	0.0173
$e_{ss,\ q2}$	$-1.5e^{-3}$	-0.1176	$-2.5e^{-4}$	$7.0e^{-3}$
$RMSE_{q_1}$	$3.0e^{-3}$	0.0896	$5.7e^{-4}$	0.0112
$RMSE_{q_2}$	$2.5e^{-3}$	0.0636	$3.5e^{-4}$	$2.1e^{-3}$

VI. CONCLUSIONS

In this paper we presented a new formulation of an active inference controller where the free-energy depends explicitly on the control actions. This formulation brings two general advantages. First, it allows us to remove the bias from state-estimation which affected the standard active inference controller, leading to more accurate state estimation. Second, it simplifies the definition of the control actions since it removes the need for computing partial derivatives of the free-energy. Next to that, we showed how the new formulation simplifies the use of established fault detection techniques with probabilistic robustness, which would have caused many false positives using standard active inference. The effectiveness of the approach is showcased in a simulated 2-DOF arm subject to sensory faults. Future work will investigate fault-tolerant control with stochastic decision boundaries, a formal stability proof, and experiments on a real robot manipulator.

REFERENCES

[1] J. Krüger, T. Lien, and A. Verl, "Cooperation of human and machines in assembly lines," *CIRP Annals - Manufacturing Technology*, 2009.

[2] M. Visinsky, J. Cavallaro, and I. Walker, "Robotic fault detection and fault tolerance: A survey," *Reliability Engineering and System Safety*, vol. 46, pp. 139–158, 1994.

[3] W. Dixon, I. Walker, D. Dawson, and J. Hartranft, "Fault detection for robot manipulators with parametric uncertainty: A prediction-error-based approach," *IEEE Trans. on Robotics and Automation*, vol. 16(6), pp. 689–699, 2000.

[4] J. Shin and J. Lee, "Fault detection and robust fault recovery control for robot manipulators with actuator failures," in *Procs. of the IEEE Int. Conf. on Robotics and Automation*, 1999, pp. 861–866.

[5] G. Paviglianiti, F. Pierri, F. Caccavale, and M. Mattei, "Robust fault detection and isolation for proprioceptive sensors of robot manipulators," *Mechatronics*, vol. 20(1), pp. 162–170, 2010.

[6] M. Van, D. Wu, S. Ge, and H. Ren, "Fault diagnosis in image-based visual servoing with eye-in-hand configurations using Kalman filter," *IEEE Trans. Industrial Electronics*, vol. 12(6), pp. 1998–2007, 2016.

[7] M. Capisani, A. Ferrara, and P. Pisu, "Sliding mode observers for vision-based fault detection, isolation and identification in robot manipulators," in *American Control Conference*, 2010.

[8] J. Chen and R. J. Patton, *Robust model-based fault diagnosis for dynamic systems*. Springer Science & Business Media, LLC, 1999.

[9] K. S. Narendra and J. Balakrishnan, "Adaptive control using multiple models," *IEEE Transaction on Automatic Control*, 1997.

[10] S. Fang, M. Blanke, and B. J. Leira, "Mooring system diagnosis and structural reliability control for position moored vessels," *Control engineering Practice*, vol. 36, pp. 12–26, 2015.

[11] R. Ferrari, H. Dibowski, and S. Baldi, "A message passing algorithm for automatic synthesis of probabilistic fault detectors from building automation ontologies," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4184–4190, 2017.

[12] V. Rostampour, R. Ferrari, and T. Keviczky, "A set based probabilistic approach to threshold design for optimal fault detection," *Procs. of the American Control Conference*, pp. 5422–5429, 2017.

[13] V. Rostampour, R. M. Ferrari, A. M. Teixeira, and T. Keviczky, "Privatized distributed anomaly detection for large-scale nonlinear uncertain systems," *IEEE Transactions on Automatic Control*, 2020.

[14] M. R. Blas and M. Blanke, "Stereo vision with texture learning for fault-tolerant automatic baling," *Computers and electronics in agriculture*, vol. 75, no. 1, pp. 159–168, 2011.

[15] C. Pezzato, R. Ferrari, and C. H. Corbato, "A novel adaptive controller for robot manipulators based on active inference," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2973–2980, 2020.

[16] M. Baioumy, P. Duckworth, B. Lacerda, and N. Hawes, "Active inference for integrated state-estimation, control, and learning," arXiv preprint arXiv:2005.05894, 2020.

[17] K. J. Friston, "The free-energy principle: a unified brain theory?" *Nature Reviews Neuroscience*, vol. 11(2), pp. 27–138, 2010.

[18] C. L. Buckley, C. S. Kim, S. McGregor, and A. K. Seth, "The free energy principle for action and perception: A mathematical review," *Journal of Mathematical Psychology*, vol. 81, pp. 55–79, 2017.

[19] P. Lanillos and G. Cheng, "Adaptive robot body learning and estimation through predictive coding," in *IROS*, 2018.

[20] C. Pezzato, M. Baioumy, C. H. Corbato, N. Hawes, M. Wisse, and R. Ferrari, "Active inference for fault tolerant control of robot manipulators with sensory faults," in *1st Int. Workshop on Active Inference, ECML PKDD*, ser. Communications in Computer and Information Science, Springer, Ed., vol. 1326, 2020.

[21] D. Lindley, "Bayesian statistics, a review," *SIAM*, vol. 2, 1972.

[22] C. W. Fox and S. J. Roberts, "A tutorial on variational bayesian inference," *Artificial intelligence review*, vol. 38, no. 2, pp. 85–95, 2012.

[23] K. Friston, J. Mattout, N. Trujillo-Barreto, J. Ashburner, and W. Penny, "Variational free energy and the Laplace approximation," *Neuroimage*, vol. 34(1), pp. 220–234, 2007.

[24] K. J. Friston, J. Mattout, and J. Kilner, "Action understanding and active inference," *Biological cybernetics*, vol. 104(1-2), pp. 137–160, 2011.

[25] M. Baltieri and C. L. Buckley, "A probabilistic interpretation of PID controllers using active inference," in *Int. Conference on Simulation of Adaptive Behavior*. Springer, 2018, pp. 15–26.

[26] G. Oliver, P. Lanillos, and G. Cheng, "Active inference body perception and action for humanoid robots," arXiv preprint arXiv:1906.03022v2, 2019.

[27] C. Rasmussen and C. Williams, Eds., *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[28] M. Baioumy, M. Mattamala, and N. Hawes, "Variational inference for predictive and reactive controllers," in *ICRA 2020 Workshop on New advances in Brain-inspired Perception, Interaction and Learning*, 2020.

[29] A. Meera and M. Wisse, "Free energy principle based state and input observer design for linear systems with colored noise," in *2020 American Control Conference (ACC)*, 2020, pp. 5052–5058.

[30] V. Rostampour, R. Ferrari, A. M. Teixeira, and T. Keviczky, "Differentially-Private Distributed Fault Diagnosis for Large-Scale Nonlinear Uncertain Systems," *IFAC-PapersOnLine*, vol. 51, pp. 975–982, 2018.

[31] X. Chen, "A new generalization of Chebyshev inequality for random vectors," *arXiv preprint arXiv:0707.0805*, 2007.

[32] R. Bogacz, "A tutorial on the free-energy framework for modelling perception and learning," *Journal of mathematical psychology*, 2015.

[33] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer Science & Business Media, 2010.

Authorized licensed use limited to: TU Delft Library. Downloaded on January 10,2022 at 15:01:38 UTC from IEEE Xplore. Restrictions apply.