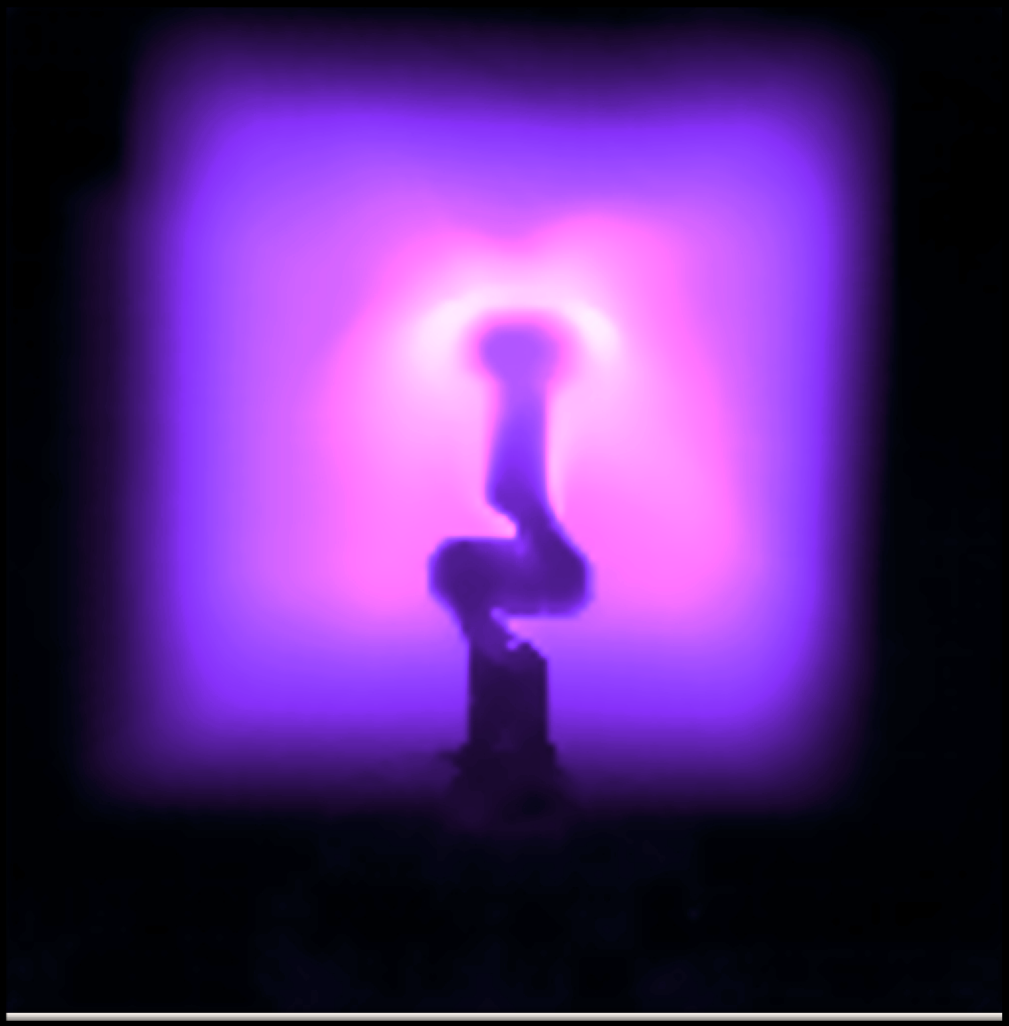
MAIC: Multimodal Active Inference Controller



Master's thesis

Cristian Meo

TU Delft Delft University of Technology

Department of Cognitive Robotics

MAIC: Multimodal Active Inference Controller

by

Cristian Meo

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended on Tuesday August 31th, 2021 at 11:30 AM

MSc. Mechanical Engineering
BMD – Biorobotics track

August 17, 2021

Student number: 5160901

Thesis committee:
Martijn Wisse, TU Delft, Main supervisor and Chairman
Pablo Lanillos, Radboud University, External advisor
Riccardo Ferrari, TUDelft, External examiner
Jens Kober, TUDelft, External examiner
Giovanni Franzese, TUDelft, Expert

Faculty of Mechanical, Maritime and Materials engineering (3mE)
Delft University of Technology

Preface

This Master thesis consists of a paper that has been done in collaboration with other PhD students at the Department of Cognitive robotics, Delft University of Technology. The project has been supervised by Prof Martijn Wisse and by Dr Lanillos Pablo, Donders Institute for Brain, Cognition and Behavior, Radboud University, the Netherlands. Here I illustrate the authors' contributions to this paper.

Authors contributions:

- Cristian Meo: Main author, I defined the research questions, developed the theoretical model, designed the performed experiments, processed the experimental results, wrote the article and managed the research team.
- Giovanni Franzese: He provided consultancies about Gaussian Processes and related topics. He tuned the impedance controller and performed the designed experiments.
- Corrado Pezzato: He tuned the standard and unbiased active inference controllers and performed the designed experiments.
- Max Spahn: He tuned the model predictive controller and performed the designed experiments.

Every experiment was performed under my supervision.

Cristian Meo
Delft, August 2021

MAIC: Multimodal Active Inference Controller

Cristian Meo[a], Giovanni Franzese[a], Corrado Pezzato[a], Max Spahn[a] and Pablo Lanillos[b]

***Abstract*—Active inference, a theoretical construct inspired by brain processing, is a promising approach to control artificial agents. Here we present a novel multimodal active inference torque controller for industrial arms that improves the adaptive characteristics of previous active inference approaches but also enables multimodal integration with any other sensor modality (e.g., raw images). We evaluated our model on a 7DoF Franka Emika Panda robot arm and systematically compared its behaviour with previous active inference baselines and classic controllers, analyzing both qualitatively and quantitatively adaptation capabilities and control accuracy. Results showed improved control accuracy in goal-directed reaching due to the increased representation power, high noise rejection due to multimodal filtering, and adaptability in changes on the environmental conditions and robot parameters without the need to relearn the generative models nor parameter retuning.**

I. Introduction

Real world complex systems, such as airplanes, cars and intelligent agents may need to process unstructured high-dimensional data coming from different sensors depending on the domain or task (e.g., LIDAR in cars, sonar in submarines and different sensors to measure the internal state of the robotic system). In this context, one of the biggest challenges is mapping this rich stream of multimodal information into a lower-dimensional space that integrates and compresses all modalities into a latent representation; the agent could then use this embedded latent representation that encodes the state of the robot and the world aiding the controller. Moreover, another critical challenge is the uncertainty, these environments may always present unmodeled behaviours, such as air turbulence in airplanes and unmodeled dynamics of water streams. In the last years, some proof-of-concept studies in robotics have shown that Active Inference (AIF) may be a powerful framework to address key challenges [16], such as adaptation [20, 24], robustness [1, 2] and multimodal state representation learning [15, 18]. Active Inference is prominent in neuroscientific literature as a biologically plausible mathematical construct of the brain based on the Free Energy Principle (FEP) [6]. According to this theory, the brain learns a generative model of the world/body that is used to perform state estimation (perception) as well as to execute control (actions), optimizing one single objective: Bayesian model evidence. This approach, which grounds on hierarchical variational inference and dynamical systems estimation [11], has strong connections with Bayesian filtering [25] and control as inference [19], as it both estimates the system state and computes the control commands as a result of the inference process.

[a]: Faculty of Mechanical Engineering, Department of Cognitive Robotics, Delft University of Technology, Delft, The Netherlands

[b]: Donders Institute for Brain, Cognition and behaviour, Department of Artificial Intelligence, Radboud University, Nijmegen, The Netherlands.

A. Related Works

Recently, a state estimation algorithm and an AIF-based reaching controller for humanoid robots were proposed in [14] and [20] respectively, showing robust sensory fusion (visual, proprioceptive and tactile) and adaptability to unexpected sensory changes. However, they could only handle low-dimensional inputs and did not implement low-level torque control. Latterly, adaptive active inference torque controllers [2, 23] showed better performances than a state-of-the-art model reference adaptive controller. However, they cannot handle high-dimensional inputs. Furthermore, an AIF planning algorithm was presented in [10], showing that the introduction of visual working memory and the variational inference mechanism significantly improve the performance in planning adequate goal-directed actions. Lastly, in a previous work we presented a Multimodal Variational Autoencoder Active Inference (MAIC-VAE) [18] torque controller, which integrates visual and joint sensory spaces. However, a clear and systematic comparison on adaptation between AIF and classic controllers is still missing. Furthermore, [18] does not present a generalized multimodal active inference control scheme, focusing mostly on the Multimodal VAE implementation.

B. Contribution

We propose a multimodal active inference torque controller (MAIC) which extends current active inference control approaches in the literature by allowing function learning [13] and multimodal state representation learning [17] while maintaining the adaptation capabilities of an active inference controller. To this end, we developed two versions of the proposed algorithm depending on the size of the sensory input: low-dimensional using Gaussian Processes (MAIC-GP), i.e., combining the joint space with the end-effector position and high-dimensional using a Variational Autoencoder (MAIC-VAE), i.e., combining the joint space with raw images as visual input. Finally, we experimentally evaluated the proposed algorithm on a 7DOF Franka Emika Panda arm under different conditions. We systematically compared the MAIC with state-of-the-art torque active inference controllers, such as the AIC [23] and the uAIC [2], and standard controllers, such as model predictive control (MPC, Appendix A) and joint impedance control (IC, Appendix B). We present both qualitative and quantitative analysis in different experiments, focusing on adaptation capability and control accuracy.

II. AIF general formulation and notation

Here we introduce the standard equations and concepts from the AIF literature [6], and the notation used in this paper, framed for estimation and control of robotic systems [20]. The aim of the robot is to infer its state (unobserved variable) by

means of noisy sensory inputs (observed). For that purpose, it can refine its state using the measurements or perform actions to fit the observed world to its internal model. This is dually computed by optimizing the variational free energy, a bound on the Bayesian model evidence [3].

System variables. State, observations, actions and their n-order time derivatives (generalized coordinates).

$$\begin{aligned}
\mathbf{x} &= [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_l] && \text{, sensors observations (l modalities)} \\
\mathbf{r} &= [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_l] && \text{, sensory noise (l modalities)} \\
\tilde{\mathbf{x}} &= [\mathbf{x}, \mathbf{x}', \mathbf{x}'', ..., \mathbf{x}^n] && \text{, generalized sensors} \\
\tilde{\mathbf{z}} &= [\mathbf{z}, \mathbf{z}', \mathbf{z}'', ..., \mathbf{z}^n] && \text{, multimodal system state} \\
\tilde{\boldsymbol{\mu}} &= [\boldsymbol{\mu}, \boldsymbol{\mu}', \boldsymbol{\mu}'', ..., \boldsymbol{\mu}^n] && \text{, proprioceptive state} \\
\tilde{\mathbf{r}} &= [\mathbf{r}, \mathbf{r}', \mathbf{r}'', ..., \mathbf{r}^n] && \text{, generalized sensory noise} \\
\tilde{\mathbf{w}} &= [\mathbf{w}, \mathbf{w}', \mathbf{w}'', ..., \mathbf{w}^n] && \text{, state fluctuations} \\
\mathbf{a} &= \{a_1, a_2, \ldots, a_p\} && \text{, actions (p actuators)} \\
\mathbf{m} &= \{m_1, m_2, \ldots, m_l\} && \text{, modalities}
\end{aligned}$$

Where $\mathbf{x}' = \frac{d\mathbf{x}}{dt}$. Depending on the formulation the action $\mathbf{a}$ can be force, torque, acceleration or velocity. In this work action refers to torque. We further define the time-derivative of the state vector $D\tilde{\mathbf{z}}$ as:

$$D\tilde{\mathbf{z}} = \frac{d}{dt}([\mathbf{z}, \mathbf{z}', \ldots, \mathbf{z}^n]) = [\mathbf{z}', \mathbf{z}'', \ldots, \mathbf{z}^{n+1}]$$

Generative models. Two generative models govern the robot: the mapping function between the robot's state and the sensory input $g(\tilde{\mathbf{z}})$ (e.g., forward kinematics) and the dynamics of the internal state $f(\tilde{\mathbf{z}})$ [3].

$$\tilde{\mathbf{x}} = g(\tilde{\mathbf{z}}) + \tilde{\mathbf{r}} \tag{1}$$

$$D\tilde{\mathbf{z}} = f(\tilde{\mathbf{z}}) + \tilde{\mathbf{w}} \tag{2}$$

where $\mathbf{r} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\tilde{\mathbf{x}}})$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\tilde{\mathbf{z}}})$ are the sensory and process noise respectively. $\Sigma_{\tilde{\mathbf{x}}}$ and $\Sigma_{\tilde{\mathbf{z}}}$ are the covariance matrices that represent the controller's confidence about each sensory input and about its dynamics respectively.

Variational Free Energy (VFE). The VFE is the optimization objective for both estimation and control. We use the definition of the $\mathcal{F}$ based on [7], where the action is implicit within the observation model $\mathbf{x}(a)$. Using the KL-divergence the VFE is:

$$\mathcal{F} = \mathrm{KL}\left[q(\tilde{\mathbf{z}}) || p(\tilde{\mathbf{z}}|\tilde{\mathbf{x}})\right] - \log p(\tilde{\mathbf{x}}) \tag{3}$$

where $q(\tilde{\mathbf{z}})$, $p(\tilde{\mathbf{z}}|\tilde{\mathbf{x}})$ and $p(\tilde{\mathbf{x}})$ are the variational density, posterior and prior distribution. As a result, $q(\tilde{\mathbf{z}})$ approximates $p(\tilde{\mathbf{z}}|\tilde{\mathbf{x}})$ when $\mathcal{F}$ is minimized.

State estimation using gradient optimization:

$$\dot{\tilde{\mathbf{z}}} = D\tilde{\mathbf{z}} - k_z \nabla_{\mathbf{z}} \mathcal{F}(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) \tag{4}$$

Control using gradient optimization:

$$\dot{\mathbf{a}} = -k_a \sum_{\tilde{\mathbf{x}}} \frac{d\tilde{\mathbf{x}}}{da} \cdot \nabla_{\tilde{\mathbf{x}}} \mathcal{F}(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) \tag{5}$$

where k_z and k_a are the gradient descent step sizes. The VFE has a closed form under the *Laplace and Mean-field approximations* [3, 20] and it is defined as:

$$\begin{aligned}
\mathcal{F}(\tilde{\mathbf{z}}, \tilde{\mathbf{x}}) &\simeq -\ln p(\tilde{\mathbf{z}}, \tilde{\mathbf{x}}) = p(\tilde{\mathbf{x}}|\tilde{\mathbf{z}}) p(\tilde{\mathbf{z}}) \\
&\simeq (\tilde{\mathbf{x}} - g(\tilde{\mathbf{z}}))^T \Sigma_{\tilde{\mathbf{x}}}^{-1} (\tilde{\mathbf{x}} - g(\tilde{\mathbf{z}})) \\
&\quad + (D\tilde{\mathbf{z}} - f(\tilde{\mathbf{z}}))^T \Sigma_{\tilde{\mathbf{z}}}^{-1} (D\tilde{\mathbf{z}} - f(\tilde{\mathbf{z}})) \\
&\quad + \frac{1}{2} \ln |\Sigma_{\tilde{\mathbf{x}}}| + \frac{1}{2} \ln |\Sigma_{\tilde{\mathbf{z}}}|
\end{aligned} \tag{6}$$

The first term of Eq. (6) is the sensor prediction error and the second term is the dynamics prediction error.

III. Architecture and Design: Multimodal Active Inference Controller

As long as we can learn the generative mapping of a certain sensory space, we can add any modality to Eq. (4), combining free energy optimization [7] with generative model learning and performing sensory integration. The online estimation and control problem is solved by optimizing the VFE through gradient optimization, computing Eq. (4) and (5). In this work we present two different versions of the same algorithm, in the first case we use the end-effector position $\mathbf{x_{ee}}$ (low dimensional sensory input), learning the generative mapping with Gaussian Processes (MAIC-GP), while in the second case we scale to the full raw image $\mathbf{x_v}$ (high dimensional sensory input), learning the mapping through a multimodal variational autoencoder (MAIC-VAE). We first introduce the required preliminaries. Consequently, we illustrate the multimodal active inference update equations and the full algorithm.

A. Multimodal Active Inference

As discussed in [3], Eq. (6) can be extended for different modalities. As a result, state estimation and control equations can be derived for the multimodal case as well. We define the sensory generative function $g(\tilde{\mathbf{z}})$ with multiple modalities as $\boldsymbol{g}(\tilde{\mathbf{z}}) = [g_{m_1}(\tilde{\mathbf{z}}), ..., g_{m_l}(\tilde{\mathbf{z}})]$. Moreover, as in [24] we define the system internal dynamics $f(\tilde{\mathbf{z}})$ as:

$$f(\tilde{\mathbf{z}}, \boldsymbol{\rho} = \mathbf{x}_d) = \frac{\partial \boldsymbol{g}(\tilde{\mathbf{z}})}{\partial \tilde{\mathbf{z}}} (\mathbf{x}_d - \boldsymbol{g}(\tilde{\mathbf{z}})) \tag{7}$$

where $\boldsymbol{\rho} = \mathbf{x}_d$ steers the system towards the desired target. Therefore, substituting Eq. (6) into Eq. (4) and (5) and rewriting it for the multimodal case, we can obtain the multimodal state estimation update law:

$$\begin{aligned}
\dot{\tilde{\mathbf{z}}} = D\tilde{\mathbf{z}} &+ \sum_{\boldsymbol{m}} \left(k_m \frac{\partial g_m}{\partial \tilde{\mathbf{z}}} \Sigma_m^{-1} (\mathbf{x}_m - g_m(\tilde{\mathbf{z}})) \right) \\
&+ k_{\mathbf{z}} \frac{\partial f(\tilde{\mathbf{z}}, \boldsymbol{\rho})}{\partial \tilde{\mathbf{z}}} \Sigma_{\tilde{\mathbf{z}}}^{-1} (\mathbf{x}_d - f(\tilde{\mathbf{z}}, \boldsymbol{\rho}))
\end{aligned} \tag{8}$$

and the control equation:

$$\dot{\mathbf{a}} = -\sum_{\boldsymbol{m}} k_{\mathbf{a}_m} \partial_{\mathbf{a}} \mathbf{x}_m \Sigma_m^{-1} (\mathbf{x}_m - g_m(\tilde{\mathbf{z}})) \tag{9}$$

where k_m and $k_{\mathbf{a}_m}$ are state estimation and control gradient descent step sizes related to modality m, and $\partial_{\mathbf{a}} \mathbf{x}_m = \frac{\partial \mathbf{x}_m}{\partial \mathbf{a}}$. Appendix F illustrates the derivation of Eq. (8) and Eq. (9). Algorithm 1 illustrates the general multimodal active inference controller scheme.

Algorithm 1 MAIC

Require: $\mathbf{x}_d = \{\mathbf{x}_{d_{m_1}}, \mathbf{x}_{d_{m_2}}, ..., \mathbf{x}_{d_{m_l}}\}$
  while $\neg goal\ reached$ **do**
    $\mathbf{x} = [\mathbf{x}_{m_1}, \mathbf{x}_{m_2}, ..., \mathbf{x}_{m_l}] \leftarrow Sensors(\boldsymbol{m})$
    State Estimation
    $\dot{\tilde{\mathbf{z}}} \leftarrow$ *multimodal state update law Eq.* (8)
    Control
    $\dot{\mathbf{a}} = -\sum_{\boldsymbol{m}} k_{\mathbf{a}_m} \partial_{\mathbf{a}} \mathbf{x}_m \Sigma_m^{-1}(\mathbf{x}_m - g_m(\tilde{\mathbf{z}}))$
    Euler integration
    $\tilde{\mathbf{z}} \mathrel{+}= \delta_t \dot{\tilde{\mathbf{z}}}$
    $\mathbf{a} \mathrel{+}= \delta_t \dot{\mathbf{a}}$
  end while

IV. Algorithm Implementations

A. MAIC-GP

Here we describe the multimodal active inference for low-dimensional inputs (e.g., end-effector position). We define the sensory generative functions as:

$$g_{\boldsymbol{q}}(\boldsymbol{\mu}) = \boldsymbol{\mu} \tag{10}$$

$$g_{\mathbf{ee}}(\boldsymbol{\mu}) = GP_{\mathbf{ee}}(\boldsymbol{\mu}) \tag{11}$$

where $g_{\boldsymbol{q}}(\boldsymbol{\mu})$, as in [23], is the proprioceptive generative sensory function (i.e., joint states), and $g_{\mathbf{ee}}(\boldsymbol{\mu})$ is the end-effector generative sensory function. As in [14], $g_{\mathbf{ee}}(\boldsymbol{\mu})$ is computed using a Gaussian Process Regressor (GPR) between proprioceptive sensory input and end-effector positions. This approach is particularly useful because we can compute a closed form for the derivative of the gaussian process with respect to the beliefs $\boldsymbol{\mu}$, which is required for the multimodal state update law, Eq. (8).

1) Learning: We train the model through guided self-supervised learning. This generated a dataset of 9261 pairs end-effector positions and joint values $(\mathbf{X_{ee}}, \mathbf{X_q})$. We use a squared exponential kernel k of the form:

$$k(\mathbf{x}_{\mathbf{q}_i}, \mathbf{x}_{\mathbf{q}_j}) = \sigma_f^2\, e^{\left(-\frac{1}{2}(\mathbf{x}_{\mathbf{q}_i} - \mathbf{x}_{\mathbf{q}_j})^T \boldsymbol{\Theta} (\mathbf{x}_{\mathbf{q}_i} - \mathbf{x}_{\mathbf{q}_j})\right)} + \sigma_n^2 d_{ij} \tag{12}$$

where $\mathbf{x}_{\mathbf{q}_i}, \mathbf{x}_{\mathbf{q}_j} \in \mathbf{X_q}$, d_{ij} is the Kronocker delta function and $\boldsymbol{\Theta}$ is the hyperparameters diagonal matrix. We can compute the end-effector location given any joint state configuration as:

$$g_{\mathbf{ee}}(\boldsymbol{\mu}) = k(\boldsymbol{\mu}, \mathbf{X_q}) \boldsymbol{K}^{-1} \mathbf{X_{ee}} \tag{13}$$

Finally, we can compute the derivative of $g_{\mathbf{ee}}(\boldsymbol{\mu})$ with respect to $\boldsymbol{\mu}$ as:

$$\frac{\partial g_{\mathbf{ee}}(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = -\boldsymbol{\Theta}^{-1}(\boldsymbol{\mu} - \mathbf{X_q})^T [k(\boldsymbol{\mu}, \mathbf{X_q})^T \cdot \boldsymbol{\alpha}] \tag{14}$$

where $\boldsymbol{K}$ is the covariance matrix, $\boldsymbol{\alpha} = \boldsymbol{K}^{-1}\mathbf{X_{ee}}$ and $\cdot$ represents element-wise multiplication. Appendix E illustrates the end-effector predictions accuracy using GPR.

2) State estimation and Control: Substituting Eq. (10) and (11) into Eq. (8) and (9), we can now write the state estimation update laws:

$$\dot{\boldsymbol{\mu}} = \boldsymbol{\mu'} + k_\mu \Sigma_{\boldsymbol{q}}^{-1} \boldsymbol{\epsilon}_{\mathbf{x_q}} + k_{ee} \Sigma_{\mathbf{ee}}^{-1} g_{\mathbf{ee}}(\boldsymbol{\mu})' \boldsymbol{\epsilon}_{\mathbf{x_{ee}}} - k_\mu \Sigma_{\boldsymbol{\mu}}^{-1} \boldsymbol{\epsilon}_{\boldsymbol{\mu}} \tag{15}$$

$$\dot{\boldsymbol{\mu}}' = \boldsymbol{\mu''} + k_\mu \Sigma_{\dot{\boldsymbol{q}}}^{-1} \boldsymbol{\epsilon}_{\dot{q}} - k_\mu \Sigma_{\boldsymbol{\mu}}^{-1} \boldsymbol{\epsilon}_{\boldsymbol{\mu}} - k_\mu \Sigma_{\boldsymbol{\mu'}}^{-1} \boldsymbol{\epsilon}_{\boldsymbol{\mu'}} \tag{16}$$

$$\dot{\boldsymbol{\mu}}'' = -k_\mu \Sigma_{\boldsymbol{\mu'}}^{-1} \boldsymbol{\epsilon}_{\boldsymbol{\mu'}} \tag{17}$$

where Σ_i^{-1}, with $i \in \{\mathbf{x_q}, \mathbf{x_{\dot{q}}}, \mathbf{x_{ee}}, \boldsymbol{\mu}, \boldsymbol{\mu'}\}$, are the inverse variance (precision) matrices related to state observations and internal state beliefs, and $\boldsymbol{\epsilon}_i$ are the Sensory Prediction Errors with $i \in \{\mathbf{x_q}, \mathbf{x_{\dot{q}}}, \mathbf{x_{ee}}, \boldsymbol{\mu}, \boldsymbol{\mu'}\}$, which represents the error between expected sensory input and observed one and are defined as: $\boldsymbol{\epsilon}_{\mathbf{x_q}} = \mathbf{x_q} - \boldsymbol{\mu}, \boldsymbol{\epsilon}_{\mathbf{x_{\dot{q}}}} = \mathbf{x_{\dot{q}}} - \boldsymbol{\mu'}, \boldsymbol{\epsilon}_{\mathbf{x_{ee}}} = \mathbf{x_{ee}} - g_{\mathbf{ee}}(\boldsymbol{\mu}), \boldsymbol{\epsilon}_{\boldsymbol{\mu}} = \boldsymbol{\mu'} + \boldsymbol{\mu} - \mathbf{x}_{\mathbf{q}_d}, \boldsymbol{\epsilon}_{\boldsymbol{\mu'}} = \boldsymbol{\mu'} + \boldsymbol{\mu''}$. Finally, we can rewrite the control equation as:

$$\dot{\mathbf{a}} = -k_a(\Sigma_{\boldsymbol{q}}^{-1} \boldsymbol{\epsilon}_{\mathbf{x_q}} + \Sigma_{\dot{\mathbf{q}}}^{-1} \boldsymbol{\epsilon}_{\mathbf{x_{\dot{q}}}} + g_{\mathbf{ee}}(\boldsymbol{\mu})' \Sigma_{\mathbf{ee}}^{-1} \boldsymbol{\epsilon}_{\mathbf{x_{ee}}}) \tag{18}$$

Note that, as in [23], in Eq. (18) the partial derivatives with respect to the action are set to identity matrices, encoding just the sign of the relation between actions and the change in the observations. Although we can compute the action inverse models $\partial_{\mathbf{a}}\boldsymbol{\mu}, \partial_{\mathbf{a}}\boldsymbol{\mu'}, \partial_{\mathbf{a}}\mathbf{x_{ee}}$ through online learning using regressors [13], we let the adaptive controller absorb the non-linearities. Thus, as described by [23] we just consider the sign of the derivatives.

B. MAIC-VAE

Here we describe the multimodal active inference controller for high-dimensional sensory inputs. We use the autoencoder architecture to compress the information into a common latent space $\mathbf{z}$ that represents the system internal state. We define the sensory generative functions as:

$$g_{\boldsymbol{q}}(\mathbf{z}) = decoder_{\mathbf{q}}(\mathbf{z}) \tag{19}$$

$$g_{\mathbf{v}}(\mathbf{z}) = decoder_{\mathbf{v}}(\mathbf{z}) \tag{20}$$

where $decoder_{\mathbf{q}}(\mathbf{z})$ and $decoder_{\mathbf{v}}(\mathbf{z})$ describe the mapping between $\mathbf{z}$ and the sensory spaces. The interested reader can find a detailed description of MAIC-VAE in [18].

1) Generative models learning: The multimodal variational autoencoder was trained through guided self-supervised learning. The dataset generated (50000 samples) consisted in pairs of images with size (128x128) and joint angles $(\mathbf{X_v}, \mathbf{X_q})$. In order to accelerate the training, we included a precision mask $\Pi_{\mathbf{x_v}} = \Sigma_{\mathbf{x_v}}^{-1}$, computed by the variance of all images and highlighting the pixels with more information. The augmented reconstruction loss employed was:

$$\mathcal{L} = \text{MSE}((1+\Pi_{\mathbf{x_v}}) g_{\mathbf{v}}(\mathbf{z}),\ \mathbf{x_v}) + \text{MSE}(g_{\mathbf{q}}(\mathbf{z}), \mathbf{x_q}) \tag{21}$$

where $\mathbf{x_q} \in \mathbf{X_q}$ and $\mathbf{x_v} \in \mathbf{X_v}$.

2) State Estimation and Control: As in MAIC-GP, substituting the defined generative mappings, Eq. (19) and (20), into Eq. (8) and (9), we can rewrite the **state estimation** update law:

$$\begin{aligned}\dot{\mathbf{z}} =& k_v \frac{\partial g_{\mathbf{v}}}{\partial \mathbf{z}} \Sigma_{\mathbf{x_v}}^{-1}(\mathbf{x_v} - g_{\mathbf{v}}(\mathbf{z})) + k_q \frac{\partial g_{\mathbf{q}}}{\partial \mathbf{z}} \Sigma_{\mathbf{q}}^{-1}(\mathbf{x_q} - g_{\mathbf{q}}(\mathbf{z})) \\ & - k_z \frac{\partial f}{\partial \mathbf{z}} \Sigma_f^{-1}(\mathbf{x}_d - f(\mathbf{z}, \boldsymbol{\rho}))\end{aligned} \tag{22}$$

As we do not have access to the high-order generalized coordinates of the latent space $\mathbf{z}', \mathbf{z}''$, we track both the multimodal shared latent space $\mathbf{z}$ and the higher orders of

the proprioceptive (joints) state $\boldsymbol{\mu}', \boldsymbol{\mu}''$. Thus, we update the proprioceptive state velocity and acceleration using Eq. (16) and Eq. (17), while the joint angles are predicted by the MVAE: $\boldsymbol{\mu} = g_{\mathbf{q}}(\mathbf{z})$. Finally, as before the **action** (torque) is computed by optimizing the VFE using Eq. (5). Here, since we cannot easily compute the partial derivative of $g_{\mathbf{v}}$ with respect to the action, we only consider the proprioceptive errors. Thus, the torque commands are updated with the following differential equation:

$$\dot{\mathbf{a}} = -k_a(\Sigma_{\mathbf{q}}^{-1}\boldsymbol{\epsilon}_{\mathbf{x_q}} + \Sigma_{\dot{\mathbf{q}}}^{-1}\boldsymbol{\epsilon}_{\mathbf{x_{\dot{q}}}}) \tag{23}$$

where even in this case we just consider the sign of the partial derivatives $\partial_{\mathbf{a}}\boldsymbol{\mu}, \partial_{\mathbf{a}}\boldsymbol{\mu}'$.

V. Results

A. Experiments and evaluation measures

We systematically evaluated our MAIC approach in a 7DOF Franka Emika Panda robot arm. We performed three different experimental analyses and compared the MAIC approach against two state-of-the-art torque active inference controllers (AIC[23] and uAIC[2]) and two classic controllers: model predictive control (MPC, Appendix A) and impedance control (IC, Appendix B).

1) **Qualitative analysis** in sequential reaching (Sec. V-C). We evaluated MAIC approaches qualitative behaviours, focusing on how multimodal filtering affects control accuracy on the presented controllers.
2) **Adaptation study** (Sec. V-D). We evaluated the response of the system to unmodeled dynamics and environment variations by altering dynamically the mass matrix (Inertial Experiment), by adding an elastic constraint (Constrain Experiment), by adding random human disturbances (Human disturbances experiment) and by adding random noise to the published joints values (Noisy Experiment).
3) **Ablation analysis** in sequential reaching (Sec. V-C). We evaluated the algorithm accuracy and behaviour removing the extra modality from the algorithm.

In order to evaluate the experiments, we used the following evaluation metrics:

- *Joints perception error*. It is the error between the inferred (belief) and the observed joint angle. The more accurate the predictions are, the lower will be the perception error.
- *Joints goal error*. It is the error between the current joint angles and the desired ones (goal).
- *Image reconstruction error*. It is the error between the predicted visual input and the observed image. It is computed as the Frobenius norm of the difference between current and goal images. It describes the accuracy of the visual generative model.
- *End-effector reconstruction error*. It is the Euclidean distance between the predicted end-effector positions and the ones computed through the forward kinematics of the observed joints.

To summarize, joints perception and image reconstruction errors measure how well the state is estimated, while joints goal errors give a measure of how well the control task is executed.

B. Experimental setup and parameters

Experiments were performed on the 7DOF Franka Panda robot arm using ROS [12] as the interface, Pytorch [21] for the MVAE and Sklearn [22] for the Gaussian Processes. An Intel Realsense D455 camera was used to acquire visual grey scaled images with size 128x128 pixels. The camera was centred in front of the robot arm with a distance of 0.8 m.

The tuning parameters for the MAIC controllers are:

- $\Sigma_{\mathbf{x_v}}$: Variance representing the confidence about visual sensory data was set as the variances of the training dataset (Appendix D).
- $\delta_t = 0.001$: Euler integration step;
- $\Sigma_{\mathbf{q}}$=3, $\Sigma_{\dot{\mathbf{q}}}$=3, $\Sigma_{\boldsymbol{\mu}}$=5, $\Sigma_{\boldsymbol{\mu}'}$=5, Σ_f=4, $\Sigma_{\mathbf{ee}}$=6: Variances representing the confidence of internal belief about the states;
- k_μ=18.67, k_q=1.5, k_v=0.2, k_{ee}=1.4, k_a=9: The learning rates for state update and control actions respectively were manually tuned in the ideal settings experiment.

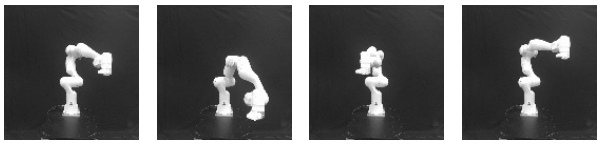
All experiments were executed on a computer with CPU: Intel core i7 8th Gen, GPU: Nvidia GeForce GTX 1050 Ti.

C. Qualitative analysis in a sequential reaching task

In order to analyse MAIC qualitative behaviour, we designed a sequential reaching task with four desired goals defined by the final joint angles $\{\mathbf{q}_{d_1}, \mathbf{q}_{d_2}, \mathbf{q}_{d_3}, \mathbf{q}_{d_4}\}$ expressed in radiants:

- $\boldsymbol{q_{d_1}} = [0.45,\ -0.38,\ 0.32,\ -2.45,\ 0.14,\ 2.06,\ 1.26\ ]$
- $\boldsymbol{q_{d_2}} = [0.70,\ -0.15,\ 0.10,\ -2.65,\ 0.31,\ 2.55,\ 1.23\ ]$
- $\boldsymbol{q_{d_3}} = [-0.03, -0.73, -0.25, -2.69, -0.18, 1.83, 0.79]$
- $\boldsymbol{q_{d_4}} = [0.31,\ -0.47,\ 0.38,\ -2.16,\ 0.14,\ 1.71,\ 1.28\ ]$

the desired end-effector positions $\{\mathbf{ee}_{d_1}, \mathbf{ee}_{d_2}, \mathbf{ee}_{d_3}, \mathbf{ee}_{d_4}\}$ and the desired visual input $\{\mathbf{I}_{d_1}, \mathbf{I}_{d_2}, \mathbf{I}_{d_3}, \mathbf{I}_{d_4}\}$ which show the related poses. In order to select unbiased desired goals, all the

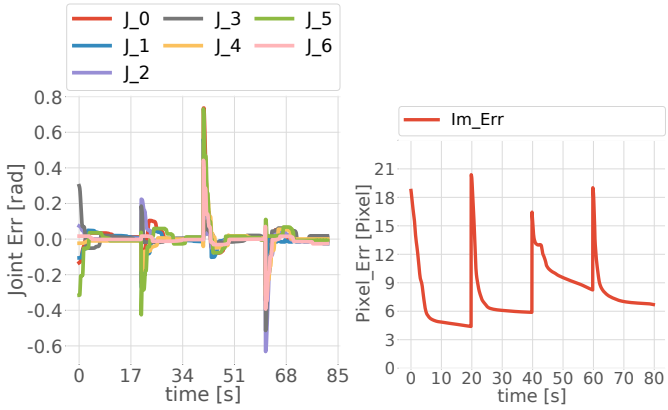


(a) $\mathbf{I}_{d_1}$ (b) $\mathbf{I}_{d_2}$ (c) $\mathbf{I}_{d_3}$ (d) $\mathbf{I}_{d_4}$

Fig. 1: Goal poses images.

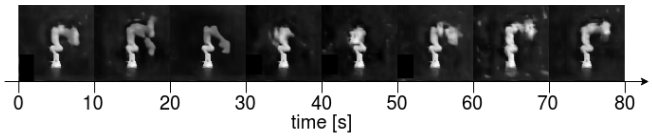
desired joint poses were randomly sampled from the dataset. In all experiments the robot starts in home position ($\mathbf{q}_{home}$= $\boldsymbol{q_{d_4}}$ rad).

1) MAIC-VAE qualitative behaviour: Figures 2a, 2b and 2c illustrate MAIC-VAE qualitative internal behaviour. It can be seen that both modalities are successfully estimated. However, Fig. 2a shows that joints reconstructions present overshoot, leading to a similar behaviour on the control task, as shown on Fig. 3. Moreover, the robot updates its internal belief by approximating the conditional density, maximizing the likelihood of the observed sensations and then generates an action that results in a new sensory state, which is consistent with the current internal representation. However, the visual decoder require much more computational time than the main

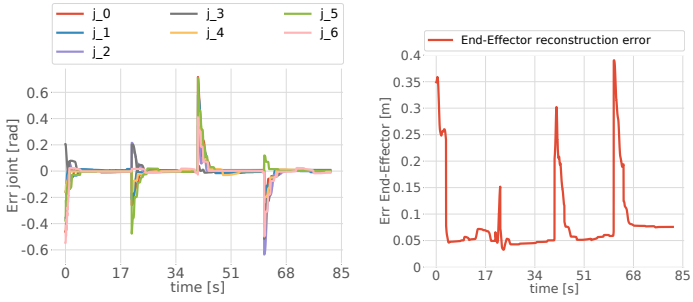


(a) MAIC-VAE: Joints perception error

(b) Image reconstruction error



(c) Sequence of some predicted visual input $g_v(\mathbf{z})$



(d) MAIC-GP: Joints perception error

(e) End-effector reconstruction error

Fig. 2: Qualitative analysis of the error measures in the sequential reaching of four goals. All errors present peaks when a new goal is set. (a-d) Each line represents the error between the i-th joint belief and the ground truth. (b) Image reconstruction error. (c) Sequence of the predicted images by the generative model along the trajectory. (e) End-effector Reconstruction error.

control loop, leading to the irregular behaviour showed on Fig. 2a. Although Fig. 2b shows that image reconstructions present different errors for different poses, Fig. 2c shows that the image reconstructions through the experiment are well reconstructed.

2) MAIC-GP qualitative behaviour: Figures 2d and 2e illustrate MAIC-GP qualitative internal behaviour. As in the previous case, both modalities are successfully estimated. Figure 2d shows that MAIC-GP joint estimations do not overshoot.

3) Vanilla Comparison: Figure 2 illustrates the qualitative behaviour of the compared controllers. From one goal to the next one the errors drop down. Although the joint belief errors (Fig. 2a) show synchronous convergence without significant steady-state errors, due to slow algorithmic frequency the MVAE-AIC behaviour is not smooth.

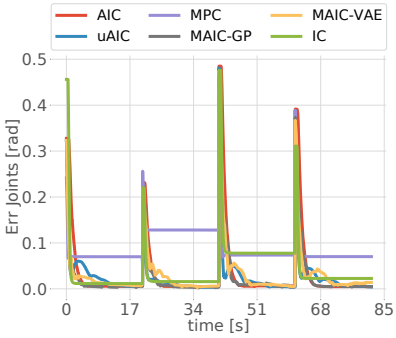
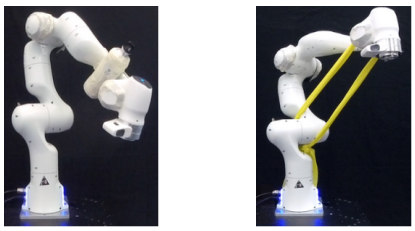


Fig. 3: Vanilla comparison. Lines represent the average of absolute joints goal errors. Peaks are present when the new goal is set.

Moreover, some goals can be better reconstructed than others, resulting in different steady-state errors. The reason is that different $\mathbf{z}$ solutions lead to similar images. Furthermore, due to dynamical model errors, MPC and IC present significant steady-state errors. Finally, MAIC-VAE and uAIC overshoot, while all the other present overdamped behaviours.

D. Adaptation Study

To investigate our approach adaptability to unmodeled dynamics and environment variations we systematically tested the controllers in four experiments. The first three experiments aim to evaluate the adaptability to unmodeled dynamics and the robustness against variations on inertial parameters. First, we attached a bottle half full of water to the 5th joint (Fig. 5a). As a result, due to water movements, the robot inertia changes dynamically. Second, we constrained the robot with an elastic band (Fig. 5b), connecting the first robot link to the last one and, therefore, introducing a substantial change in the robot dynamics. Third, we perturbed the robot along the experiment pushing it along random directions and, therefore, testing if they are able to recover from human random disturbances. Finally, we reevaluated the controllers in the presence of sensory noise, focusing on the robot behaviour. Again, we compared our algorithm implementations (MAIC-GP and MAIC-VAE) with AIC, uAIC, MPC and an IC. All controllers parameters were the same as in the previous experiments: no retuning was done. Table I reports the root-mean-square errors (RMSE)



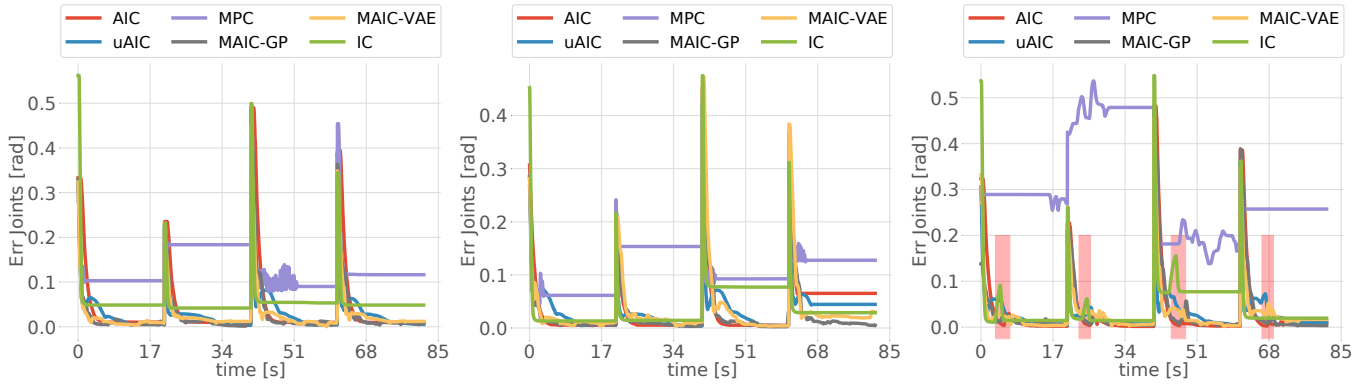
(a) Inertial Experiment

(b) Constraint Experiment

Fig. 4: Experimental setup. (a) Inertial experiment: a bottle half full of water is attached to the 5th joint. (b) Constraint Experiment setup: an elastic band links the first to the 5th joint.

	Controllers	Vanilla Experiment		Inertial Experiment		Constraint Experiment		Human disturbances Exp		Noisy Experiment	
		RMSE	std	RMSE	std	RMSE	std	RMSE	std	RMSE	std
Full Experiment	AIC	4.04E-03	4.85E-03	7.23E-03	3.05E-02	5.41E-03	1.42E-02	4.07E-03	1.21E-02	4.91E-03	3.33E-02
	uAIC	3.28E-03	1.32E-02	**3.38E-03**	1.16E-02	4.10E-03	8.88E-03	3.32E-03	9.56E-03	**3.03E-03**	2.20E-02
	MAIC-VAE	**3.18E-03**	1.78E-02	3.40E-03	1.45E-02	**3.65E-03**	2.26E-02	**3.62E-03**	1.44E-02	**2.38E-03**	1.81E-02
	MAIC-GP	**3.09E-03**	1.71E-02	**3.33E-03**	1.89E-02	**3.20E-03**	1.50E-02	**3.13E-03**	2.20E-02	3.40E-03	1.91E-02
	MPC	2.41E-02	6.81E-03	4.43E-02	1.77E-02	3.31E-02	7.84E-03	2.20E-01	5.00E-02	4.95E-02	1.32E-02
	IC	9.45E-03	2.07E-02	1.95E-02	1.87E-02	1.54E-02	1.23E-02	9.76E-03	2.04E-02	4.84E-03	2.13E-02
Transient (0-10s)	AIC	8.09E-03	3.97E-02	9.67E-03	4.18E-02	9.94E-03	1.97E-02	8.14E-03	1.68E-02	9.76E-03	4.22E-02
	uAIC	6.54E-03	1.85E-02	**6.75E-03**	1.62E-02	8.03E-03	1.24E-02	6.62E-03	1.33E-02	8.98E-03	2.50E-02
	MAIC-VAE	**6.36E-03**	2.48E-02	6.76E-03	2.02E-02	**7.26E-03**	3.15E-02	**6.48E-03**	2.01E-02	**6.63E-03**	2.71E-02
	MAIC-GP	**6.18E-03**	2.38E-02	**6.63E-03**	2.63E-02	**6.40E-03**	2.09E-02	**6.26E-03**	3.03E-02	**6.78E-03**	2.66E-02
	MPC	3.12E-02	9.45E-03	7.04E-02	2.47E-02	5.17E-02	1.09E-02	2.23E-01	4.96E-02	3.36E-02	1.82E-02
	IC	1.63E-02	2.89E-02	3.48E-02	2.62E-02	2.72E-02	1.72E-02	1.69E-02	2.86E-02	1.86E-02	2.98E-02
steady-state (10-20s)	AIC	**1.77E-06**	1.84E-06	4.88E-05	6.30E-07	8.70E-04	1.50E-03	**1.77E-06**	8.79E-05	8.33E-05	7.37E-04
	uAIC	**1.19E-05**	1.14E-05	**1.26E-05**	1.86E-05	1.69E-04	2.79E-04	3.201E-05	3.32E-05	5.89E-04	7.38E-03
	MAIC-VAE	3.29E-05	2.97E-05	3.50E-05	4.25E-05	**3.55E-05**	4.16E-05	3.31E-05	3.71E-05	**4.04E-05**	3.35E-04
	MAIC-GP	1.66E-05	2.02E-05	**1.77E-05**	2.47E-05	**1.54E-05**	8.67E-05	**1.69E-05**	3.21E-03	**7.15E-05**	4.90E-04
	MPC	1.70E-02	1.54E-03	1.81E-02	1.75E-03	1.44E-02	1.26E-03	1.18E-01	5.04E-02	1.81E-02	3.19E-03
	IC	2.61E-03	2.55E-03	4.32E-03	3.57E-04	3.64E-03	2.45E-03	2.62E-03	2.70E-03	2.91E-03	5.07E-03

TABLE I: Quantitative joints goal errors comparison. RMSE [rad] and std [rad] of the joints errors are presented, lowest errors are showed in **black bold** and second lowest in **blue bold**. Errors are computed for the full experiment, transient phase and steady-state.



(a) Inertial Experiment (b) Constraint Experiment (c) Human Disturbances Experiment

Fig. 5: Lines represent the average of absolute joints goal errors. Peaks coincide with the instants when a new goal is set. 5c) Red rectangles show the time intervals on which the disturbances are applied.

and the related standard deviations (std) which represent all the results collected during the experiments, the most accurate results are highlighted in black bold and the second most accurate in blue bold. In order to evaluate quantitatively both steady-state errors, transient behaviour and average errors we present both RMSE and std for each phase. On average MAIC-GP is the most robust against dynamic parameters change and the most adaptive to unmodeled dynamics, while MAIC-VAE is the best one on noise rejection. Only at the steady-state (after 10 seconds of execution) AIC has the lowest error on both Vanilla and Human disturbances experiments and uAIC at inertial experiment due to its integration term. Furthermore, at the steady-state MAIC-GP adapts better in the constraint experiment and MAIC-VAE is the best one on noise rejection. Finally, although both MPC and IC reported the worst performances in all experiments, they presented significant offsets already in the vanilla comparison. Therefore, we will focus just on their qualitative behaviours. We now present the details of each experiment:

1) Inertial experiment: A bottle half full of water has been attached to the 5th robot joint. The water moves along the experiment, changing the inertial characteristic of the object attached to the robot. Figure 5a illustrates the controllers' qualitative behaviours during the inertial experiment. It can be seen that, due to the unmodeled dynamics, IC and MPC show different offsets than the ones in the vanilla comparison. Moreover, MPC shows an unstable behaviour in one of the desired poses. Furthermore, since all the active inference controllers do not use any robot model, they are not affected by the change of dynamics. Table I shows that on average

the most accurate controllers are MAIC-GP (3.33E-03), uAIC (3.38E-03) and MAIC-VAE (3.40E-03).

2) Elastic constraint experiment: The experiment aims to drastically change the underlying dynamics of the system. Specifically, a rubber band was attached to the robot. To prevent the robot from entering to safety mode, we chose to link the first joint to the last one. As a result, we bounded the elastic tension to a sustainable value. Figure 5b shows that both classic and unimodal AIF controllers are significantly affected by the elastic tension, presenting remarkable offsets. By contrast, as recorded on Tab. I, MAIC-GP and MAIC-VAE present the highest control accuracy.

3) Human disturbances experiment: This experiment aims to evaluate compliance and controllers recovery ability after random disturbances. To do this, a human operator pushed the robot in random directions along the experiment. Red shaded areas on Fig. 5c indicate the periods on which the robot is disturbed. Apart from the MPC, which is not able to recover and perform the task, all the other ones fully recover from the disturbances, showing a safe behaviour in case of human disturbances.

4) Noise experiment: We reevaluated the controller behaviour in the presence of proprioceptive noise, focusing on the noise rejection capabilities of the six controllers. Proprioceptive noise was implemented as additive noise sampled from a Normal distribution $\mathbf{r_q} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{r_q}} = 0.1)$. Figure 6 shows that MAIC controllers were the most adaptive, presenting the smoothest behaviours. The reason is that multimodal filtering acts as a filter for the injected noise, reducing its effect and allowing a smooth control behaviour. All the other controllers oscillate significantly more along the experiment.

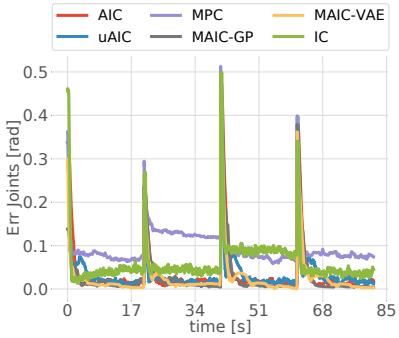


Fig. 6: Noisy experiment. Lines represent the average of absolute joints goal errors. Peaks coincide with the instants when a new goal is set.

E. Ablation Study

In order to evaluate the effect of the extra modalities, we performed an ablation study removing the extra modality from the algorithm scheme. Figure 7 shows that by removing the visual modality the behaviour becomes much smoother. Indeed, the control loop frequency increase from 120Hz to 1000Hz. However, the control accuracy does not change significantly.

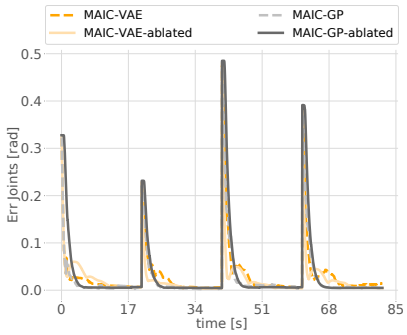


Fig. 7: Ablation study. Lines represent the average of absolute joints goal errors. Peaks are present when the new goal is set.

VI. Discussion

A. Limitations

Although the quantitative table comparison shows that on average MAIC implementations are more adaptive and accurate, they still have limitations. First of all, multimodal filtering requires more computational time, leading to irregular behaviours. Indeed, the ablation study clearly shows that when removing the extra modality, the control behaviour becomes significantly smoother. Using a faster GPU may solve this issue. Moreover, the multimodal state estimation depends on the accuracy of the learned generative mapping. Indeed, here we always use a black background to facilitate the image reconstruction. Furthermore, another limitation is that for goal-directed behaviours we need to provide the desired values for all the modalities, which may not be always available.

B. Future work

MAIC can also work in an imaginary regime by mentally simulating the expected behaviour (Appendix C), opening many opportunities for future research such as model predictive active inference controllers, where the controller predict N steps head. Moreover, the multimodal filtering scheme in principle could be integrated into other kinds of controllers, such as an IC. Although this work uses end-effector positions and images the proposed control scheme can be generalized to any other sensor modality and extended to M modalities. Future works will focus on integrating different modalities, studying the effect on adaptation and control accuracy.

VII. Conclusion

We presented MAIC, a multimodal extension of the standard AIF controller presented in [23]. Our approach makes use of the alleged adaptability and robustness of AIF, taking advantage of previous works and overcoming some related limitations. We solved state estimation by combining representation learning and multimodal filtering with free energy optimization, improving the representational power and adaptability. As a result, we derived a schema for online multimodal torque control, which does not require any dynamic or kinematic model of the robot at runtime, is less sensitive to unmodeled dynamics, and can also handle high-dimensional inputs.

Moreover, we performed a systematic comparison on different experiments of AIF and selected classic controllers, providing both qualitative and quantitative analysis. Results showed that our proposed algorithm is more adaptive than state-of-the-art torque AIF baselines and classical controllers (MPC and IC). Moreover, it was more accurate in the presence of sensory noise, showing the strongest noise rejection capability. Our MAIC was highly adaptive and robust to different contexts, such as changes in the robot dynamics (i.e., elastic constraint) and changes in the robot properties (i.e. inertial properties).

REFERENCES

[1] Mohamed Baioumy, Paul Duckworth, Bruno Lacerda, and Nick Hawes. Active inference for integrated state-estimation, control, and learning. *arXiv preprint arXiv:2005.05894*, 2020.

[2] Mohamed Baioumy, Corrado Pezzato, Riccardo Ferrari, Carlos Hernandez Corbato, and Nick Hawes. Fault-tolerant control of robot manipulators with sensory faults using unbiased active inference. In *European Control Conference, ECC*, 2021.

[3] 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.

[4] Alexander Domahidi and Juan Jerez. Forces professional. Embotech AG, url=https://embotech.com/FORCES-Pro, 2014–2019.

[5] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.

[6] Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138, 2010.

[7] Karl J Friston, Jean Daunizeau, James Kilner, and Stefan J Kiebel. Action and behavior: a free-energy formulation. *Biological cybernetics*, 102(3):227–260, 2010.

[8] Neville Hogan. Impedance control: An approach to manipulation: Part i—theory. 1985.

[9] Lill Maria Gjerde Johannessen, Mathias Hauan Arbo, and Jan Tommy Gravdahl. Robot dynamics with urdf & casadi. In *2019 7th (ICCMA)*. IEEE, 2019.

[10] Minju Jung, Takazumi Matsumoto, and Jun Tani. Goal-directed behavior under variational predictive coding: Dynamic organization of visual attention and working memory. *IROS*, 2019.

[11] Frinston K.J, Trujillo-Barreto N., and Daunizeau. Dem: a variational treatment of dynamic systems. *NeuroImage, 41, pp. 849-885*, 2008.

[12] Anis Koubaa. *Robot Operating System (ROS): The Complete Reference (Volume 2)*. Springer Publishing Company, Incorporated, 1st edition, 2017.

[13] Pablo Lanillos and Gordon Cheng. Active inference with function learning for robot body perception. In *Proc. Int. Workshop Continual Unsupervised Sensorimotor Learn.*, pages 1–5, 2018.

[14] Pablo Lanillos and Gordon Cheng. Adaptive robot body learning and estimation through predictive coding. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4083–4090. IEEE, 2018.

[15] Pablo Lanillos, Jordi Pages, and Gordon Cheng. Robot self/other distinction: active inference meets neural networks learning in a mirror. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI)*, pages 2410 – 2416, 2020.

[16] Pablo Lanillos and Marcel van Gerven. Neuroscience-inspired perception-action in robotics: applying active inference for state estimation, control and self-perception. *arXiv preprint arXiv:2105.04261*, 2021.

[17] Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 108:379–392, Dec 2018.

[18] Cristian Meo and Pablo Lanillos. Multimodal vae active inference controller, 2021.

[19] Beren Millidge, Alexander Tschantz, Anil K Seth, and Christopher L Buckley. On the relationship between active inference and control as inference. In *International Workshop on Active Inference*, pages 3–11. Springer, 2020.

[20] Guillermo Oliver, Pablo Lanillos, and Gordon Cheng. An empirical study of active inference on a humanoid robot. *IEEE Transactions on Cognitive and Developmental Systems*, 2021.

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, and Raison. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] Corrado Pezzato, Riccardo Ferrari, and Carlos Hernández Corbato. A novel adaptive controller for robot manipulators based on active inference. *IEEE Robotics and Automation Letters*, 5(2):2973–2980, 2020.

[24] Cansu Sancaktar, Marcel AJ van Gerven, and Pablo Lanillos. End-to-end pixel-based deep active inference for body perception and action. In *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 1–8. IEEE, 2020.

[25] Simo Särkkä. *Bayesian filtering and smoothing*. Number 3. Cambridge University Press, 2013.

[26] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari. Forces nlp: an efficient implementation of interior-point... methods for multistage nonlinear nonconvex programs. *International Journal of Control*, pages 1–17, 2017.

APPENDIX

A. Model Predictive Controller

The results are compared to a standard model predictive torque control (MPC) formulation.

1) Optimization problem: Neglecting external forces, the dynamics of the system are defined by the equation of motion as

$$\boldsymbol{\tau} = M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\mathbf{q} + \mathbf{g}(\mathbf{q}),$$

which is composed of the mass matrix M, the Coriolis matrix C and the gravitational forces $\mathbf{g}$ [5]. Various approaches to compute the forward dynamics have been proposed [9]. The forward dynamics can be discretized to obtain the transition function

$$\mathbf{z}_{k+1} = f(\mathbf{z}_k, \mathbf{a}_k),$$

where $\mathbf{z}$ is the concatenated vector of joint positions, velocities and accelerations.

The control problem can be formulated as an optimization problem as follows

$$J^\star = \min_{\mathbf{z}0:N, \mathbf{a}_{0:N}} \sum_{k=0}^{N} J(\mathbf{z}_k, \mathbf{a}_k), \tag{24}$$

$$\text{s.t.} \quad \mathbf{z}_{k+1} = f(\mathbf{z}_k, \mathbf{a}_k), \tag{25}$$

$$\mathbf{a}_k \in \mathcal{U}, \mathbf{z}_k \in \mathcal{Z}, \tag{26}$$

$$\mathbf{z}_0 = \mathbf{z}(0), \tag{27}$$

where J is the objective function, $\mathcal{U}$ and $\mathcal{Z}$ are the admissible sets of actions and states respectively and $\mathbf{z}_0$ is the initial condition. The objective function was formulated as follows

$$J(\mathbf{z}_k, \mathbf{a}_k) = (\mathbf{q}_k - \mathbf{q}_{goal})^T W_{goal} (\mathbf{q}_k - \mathbf{q}_{goal}) + \mathbf{a}_k^T W_{\mathbf{a}} \mathbf{a}_k, \tag{28}$$

where W_{goal} and W_τ are the weighting matrices for the goal configuration and the actions respectively.

2) Realization: In this work, we used the recursive Newton Euler algorithm to solve the forward dynamics and a second order explicit Runge-Kutta integrator. The parameter setting is summarized in Table II. In accordance to the time step the control frequency is 10Hz.

parameter	value
N	20
Δt	0.1s
W_{goal}	$400 I_7$
$W_{\mathbf{a}}$	$\text{diag}([1.75, 2, 2.5, 5, 20, 18.75, 62.5])$

TABLE II: Parameter setting for MPC

The optimization problem is solved using the nonlinear solver proposed in [26] and the corresponding implementation [4]. The forward dynamics are computed using [9].

B. Impedance Controller

The presented impedance controller [8] is based on the following dynamic equation:

$$\boldsymbol{\tau} = K(\boldsymbol{q}_{goal} - \boldsymbol{q}) + D(-\dot{\boldsymbol{q}}) + C(\mathbf{q}, \dot{\mathbf{q}})\mathbf{q} + \mathbf{g}(\mathbf{q}),$$

where K is the set joint stiffness D is the corresponding critical damping, C is the Coriolis matrix, and g is the gravitational term. Considering that the dynamics of the robot are described by

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(q, \dot{\boldsymbol{q}}) + g(g) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \tag{29}$$

with the impedance controller the dynamics results in

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} = K(\boldsymbol{q}_{goal} - \boldsymbol{q}) + D(-\dot{\boldsymbol{q}}) + \boldsymbol{\tau}_{ext} \tag{30}$$

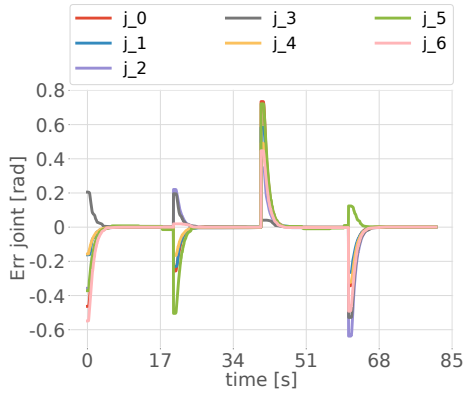
this translates in a second order critically damped dynamics of the robot in the the transition towards the desired goal.

C. Mental simulation

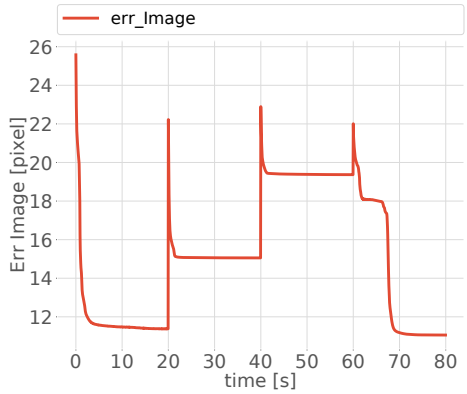
Unlike most of the AIF controllers present in literature, a great advantage of combining our approach with a multimodal VAE is the possibility to perform imagined simulations. In other words, given $\mathbf{x}_d$, the entire experiment can be simulated. Since sensory data are not available, the state update law becomes:

$$\dot{\mathbf{z}} = -k_z \frac{\partial f}{\partial \mathbf{z}} \Sigma_f^{-1} (\mathbf{x}_d - f(\mathbf{z}, \boldsymbol{\rho})) \tag{31}$$

As a result, performing the integration step of the new internal state and decoding it, the updated $\{\mathbf{x_v}, \mathbf{x_q}\}$ can be computed and the new errors can be back-propagated again, creating a loop that allows the system to do imaginary simulations.



(a) Imagined joints errors



(b) Imagined image reconstruction error

Fig. 8: Mental simulation of sequential reaching of four goals. The goal is updated on time steps where peaks are present. (a) Joints errors of an imagined simulation. Each line represents the error of the i-th joint. (b) Image reconstruction errors of an imagined simulation.

Fig. 8a and 8b show respectively imagined joints error and images reconstruction error through the entire simulation. These results show that the errors converge faster to zero than in the normal regime (Fig. 2a) as it does not need to accommodate the real dynamics of the robot.

D. Images Variance matrix

Fig. 9: Precision matrix

The visual precision matrix, $\Pi_{\mathbf{x_v}} = \Sigma^{-1}_{\mathbf{x_v}}$, which is used both as precision mask in Eq. (21) and Eq. (22), was defined computing the variance of all the images collected in the dataset. Figure 9 shows a plot of the visual precision matrix where it can be seen that the brighter pixels points are the ones less certain, while the one which do not change are darker. As a result, when we introduce it in the loss function the most informative pixels are highlighted, while when we use it in the internal state update, the pixels which do not change at all are defined with less uncertainty and the other way around.

E. Gaussian Process Accuracy

Figure 10 illustrates a 3D scatter plot that shows a heatmap of the end-effector reconstruction errors. Moreover, the axes define the cartesian workspace we considered in our experiments, where the robot base is placed at $\mathbf{x}_{base} = \{0, 0, 0\}$ and is frontally directed toward the x-direction. It can be seen that on average the reconstruction error is roughly 0.010m. However, we can clearly see that reconstructions accuracy increases along the positive y-direction. This error may be due to the inverse kinematics we used to create the algorithm.

F. MAIC equations derivation

Extending Eq. (6) to the multimodal case we can rewrite it as:

$$\mathcal{F}(\tilde{\mathbf{z}}, \tilde{\mathbf{x}}) \simeq \sum_{\mathbf{m}} \left((\tilde{\mathbf{x}}_m - g_m(\tilde{\mathbf{z}}))^T \Sigma^{-1}_{\tilde{\mathbf{x}}_m} (\tilde{\mathbf{x}}_m - g_m(\tilde{\mathbf{z}})) + \frac{1}{2} \ln |\Sigma_{\tilde{\mathbf{x}}_m}| \right) + (D\tilde{\mathbf{z}} - f(\tilde{\mathbf{z}}))^T \Sigma^{-1}_{\tilde{\mathbf{z}}} (D\tilde{\mathbf{z}} - f(\tilde{\mathbf{z}})) + \frac{1}{2} \ln |\Sigma_{\tilde{\mathbf{z}}}| \tag{32}$$

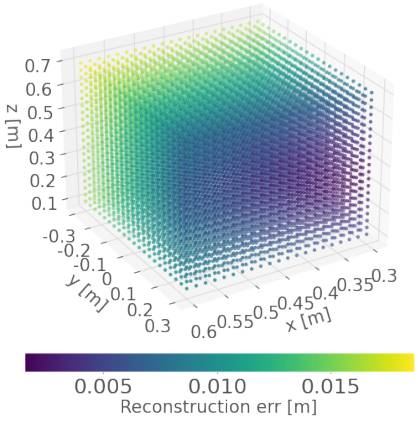


Fig. 10: End-effector reconstruction error.

As a result, substituting Eq. (32) into Eq. (4) and Eq. (5) we obtain respectively, the state estimation equation for the multimodal case:

$$\dot{\tilde{\mathbf{z}}} = D\tilde{\mathbf{z}} + \sum_{\boldsymbol{m}} \left(k_m \frac{\partial g_m}{\partial \tilde{\mathbf{z}}} \Sigma^{-1}_m (\mathbf{x}_m - g_m(\tilde{\mathbf{z}})) \right) - k_{\mathbf{z}} \frac{\partial f(\tilde{\mathbf{z}}, \boldsymbol{\rho})}{\partial \tilde{\mathbf{z}}} \Sigma^{-1}_{\tilde{\mathbf{z}}} (\mathbf{x}_d - f(\tilde{\mathbf{z}}, \boldsymbol{\rho})) \tag{33}$$

and the multimodal control equation:

$$\dot{\mathbf{a}} = -\sum_{\boldsymbol{m}} k_{\mathbf{a}_m} \frac{\partial \mathbf{x}_m}{\partial \mathbf{a}} \Sigma^{-1}_m (\mathbf{x}_m - g_m(\tilde{\mathbf{z}})) \tag{34}$$