

## Dynamic Expectation Maximization Algorithm for Estimation of Linear Systems with Colored Noise

Anil Meera, A.; Wisse, M.

**DOI**

[10.3390/e23101306](https://doi.org/10.3390/e23101306)

**Publication date**

2021

**Document Version**

Final published version

**Published in**

Entropy

**Citation (APA)**

Anil Meera, A., & Wisse, M. (2021). Dynamic Expectation Maximization Algorithm for Estimation of Linear Systems with Colored Noise. *Entropy*, 23(10), Article 1306. <https://doi.org/10.3390/e23101306>

**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.

## Article

# Dynamic Expectation Maximization Algorithm for Estimation of Linear Systems with Colored Noise

Ajith Anil Meera \* and Martijn Wisse

Department of Cognitive Robotics, Faculty of Mechanical, Maritime and Materials Engineering, Delft Institute of Technology, 2628 CN Delft, The Netherlands; m.wisse@tudelft.nl

\* Correspondence: ajitham1994@gmail.com

**Abstract:** The free energy principle from neuroscience has recently gained traction as one of the most prominent brain theories that can emulate the brain's perception and action in a bio-inspired manner. This renders the theory with the potential to hold the key for general artificial intelligence. Leveraging this potential, this paper aims to bridge the gap between neuroscience and robotics by reformulating an FEP-based inference scheme—Dynamic Expectation Maximization—into an algorithm that can perform simultaneous state, input, parameter, and noise hyperparameter estimation of any stable linear state space system subjected to colored noises. The resulting estimator was proved to be of the form of an augmented coupled linear estimator. Using this mathematical formulation, we proved that the estimation steps have theoretical guarantees of convergence. The algorithm was rigorously tested in simulation on a wide variety of linear systems with colored noises. The paper concludes by demonstrating the superior performance of DEM for parameter estimation under colored noise in simulation, when compared to the state-of-the-art estimators like Sub Space method, Prediction Error Minimization (PEM), and Expectation Maximization (EM) algorithm. These results contribute to the applicability of DEM as a robust learning algorithm for safe robotic applications.

**Keywords:** free energy principle; system identification; linear time-invariant (LTI) systems; colored noise; state space models; robotics



**Citation:** Anil Meera, A.; Wisse, M. Dynamic Expectation Maximization Algorithm for Estimation of Linear Systems with Colored Noise. *Entropy* **2021**, *23*, 1306. <https://doi.org/10.3390/e23101306>

Academic Editors: Thomas Parr, Manuel Baltieri, Thijs van de Laar, Kai Ueltzhöffer, Daniela Cialfi and Karl Friston

Received: 10 August 2021

Accepted: 1 October 2021

Published: 5 October 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The free energy principle (FEP) has emerged from neuroscience as a unifying theory of the brain [1] and has begun to guide the search for a brain-inspired learning algorithm for robots. Many attempts have been made in this direction, including the state and input observer [2,3], the adaptive controller for robot manipulators [4–6], the body perception and action scheme for humanoid robots [7], the robot navigation of ground robots [8] etc. However, the design of a parameter estimation algorithm for linear systems with colored noise remains unexplored. Since the design of an accurate parameter estimator for dynamic systems sits at the core of control systems and robotics, the reformulation of FEP into a brain-inspired estimation algorithm has an influential impact on the industry and applied robotics.

A wide range of estimators have been proposed in the literature for linear time-invariant (LTI) systems [9], including the blind system identification [10–12]. However, most of them assume the noises to be temporally uncorrelated (white noise), which is often violated in practice. This results in biased estimation for the least-square (LS)-based methods [13], and an inaccurate convergence for the iterative methods [14]. Although many attempts have been made to solve this problem, mainly through bias compensation methods [15,16], none of them perform state, input, parameter, and noise estimation for systems with colored noises [17]. The only method that does it is the Dynamic Expectation Maximization (DEM) [18] algorithm, which uses FEP to invert a highly nonlinear and hierarchical brain model from sensory data. However, the disconnect between neuroscience and control system literature hinders the wide use of this method for practical robotics

applications. Although FEP-based tools have already been applied to practical robotics applications [2–8,19,20], there is a gap in the literature on the applications of DEM owing to the mathematical formidability of the theory and its lack of formalism in the control systems domain. Therefore, it is important to reformulate DEM for the control systems audience. While DEM from computational neuroscience focuses on emulating the brain's perception through the hierarchical abstraction of a number of interacting non-linear dynamic systems, our work focuses on reformulating it into a blind system-identification algorithm for an LTI system with colored noise, which is a well-known challenge in robotics. In this attempt, we keep all the brain-related approximations intact, thereby aiming for a biologically plausible parameter estimation algorithm.

According to an FEP proposed by Karl Friston, the brain's inference mechanism is a gradient ascent over its free energy, where free energy is the information-theoretic measure that bounds the brain's sensory surprisal [21]. FEP emerges as a unified brain theory [22] by providing a mathematical description of brain functions [23]; unifying action and perception [24]; connecting physiological constructs like memory, attention, value, reinforcement, and salience [23]; and remaining consistent with Freudian ideas [25]. Similarities of FEP with reinforcement learning [26], neural networks [21,27], PID controller [28], Kalman Filter [2] and active learning [24] open up possibilities for biologically plausible parameter estimation algorithms. Although FEP emerged as a brain theory, the recent works have pushed the boundaries towards systems that survive over time, such as social and cultural dynamics. Notable works include the variational approach to culture [29], collective intelligence [30], cumulative cultural evolution [31], etc.

Numerous methods have been proposed based on the FEP framework. Predictive coding [32] models perception through a hierarchy of dynamical systems [27] with the brain's priors at the top, minimizing the prediction error at each level of the hierarchy. Bayesian message passing algorithms [33,34] use similar ideas for belief propagation. Active inference [24,35] uses FEP to model the brain's action and perception under one framework. On the perception side, there are two main type of methods to deal with dynamic systems: variational filtering and generalized filtering. Variational filtering [18,36] uses mean-field approximation (conditional independence between densities), whereas generalized filtering [37] does not. DEM [18] is a type of variational filtering that uses a Laplace approximation [38] (a fixed-form assumption for the conditional density of variables), whereas [36] uses ensemble dynamics to model the free form of the conditional densities. We focus on DEM for this work.

DEM is an FEP-based variational inference algorithm that models the brain's inference process as a maximization of its free energy for state, input, parameter, and noise estimation from data. Although the method shows high similarity to the variational inference [39], the key difference is in the use of generalized coordinates, which enables DEM to track the evolution of the trajectory of states instead of just the point estimates. This renders DEM with the capability of gracefully handling colored noises, a feature that conventional point estimators such as the Kalman Filter (KF) lacks. The modeling of noise color as analytic using generalized coordinates results in an improved state estimation under colored noise for LTI systems [2,3] and for nonlinear filtering [40], which directly improves the parameter estimation accuracy, making DEM a topic of interest [20]. This work directly impacts various sub-domains of robotics community: input estimation to the industrial automation community for fault detection systems, state estimation to the control systems community, parameter estimation to the system identification community, and hyperparameter estimation to the signal processing community.

With this paper, we aim to present DEM to the robotics audience as a robot learning algorithm for the blind system identification of LTI systems with colored noise. We elaborate on various components of DEM that are most relevant to the robotics community: 1) the derivation of the free energy objectives from Bayesian principles, 2) the modeling of colored noise using generalized coordinates, and 3) the simplification of update rules for LTI systems with colored noise. We reformulate DEM for LTI systems into a form that

is widely used in the robotics domain and use this mathematical formulations to prove that the estimation steps of DEM have theoretical guarantees of convergence [41]. In our prior work, we have discussed the stability conditions of our DEM-based linear state and input observer design [2]. This convergence guarantees and stability criteria are essential for robot safety while in operation and is of high relevance to the robotics community. Through extensive simulations on a range of random systems, we show that DEM is a competitive estimator when compared to other benchmarks in the control systems domain. The core contributions of the paper are:

1. Reformulating DEM into an estimation algorithm for LTI systems with colored noise (Section 12).
2. Proving that the estimator has theoretical guarantees of convergence for the estimation steps (Section 14).
3. Proving through rigorous simulation that DEM outperforms the state-of-the-art system identification methods for parameter estimation under colored noise (Section 16).

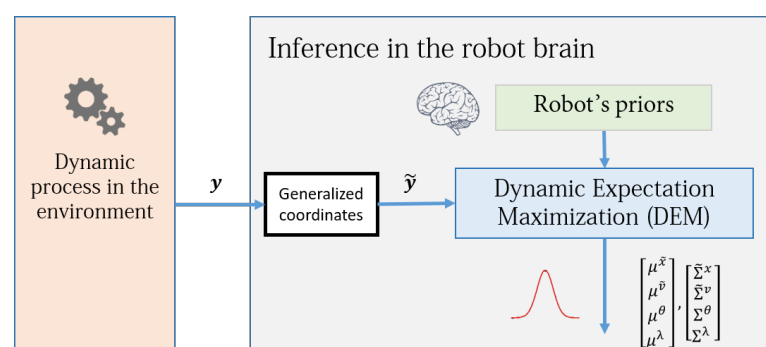
## 2. Problem Statement

Consider the linear plant dynamics given in Equation (1), where  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  are constant system matrices,  $\mathbf{x} \in \mathbb{R}^n$  is the hidden state,  $\mathbf{v} \in \mathbb{R}^r$  is the input and  $\mathbf{y} \in \mathbb{R}^m$  is the output.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{v} + \mathbf{w}, \quad \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{z}. \quad (1)$$

Here,  $\mathbf{w} \in \mathbb{R}^n$  and  $\mathbf{z} \in \mathbb{R}^m$  represent the process and measurement noise, respectively. The notations of the plant are denoted in boldface, whereas its estimates are denoted in nonboldface letters. The noises in this paper are generated through the convolution of white noise with a Gaussian kernel. The use of colored noise is motivated by the fact that in robotics, the unmodelled dynamics and the non-linearity errors can enter the plant dynamics through the noise terms, thereby violating the white noise assumption in practice [20].

The goal of this paper is to reformulate DEM for an LTI system such that given the output of the system  $\mathbf{y}$ , the estimator computes the associated states  $x$ , inputs  $v$ , parameter  $\theta$  containing the matrices  $A, B$  and  $C$ , hyperparameters  $\lambda$  that model the noise precision ( $\Pi = e^\lambda$ ), and the uncertainties of all its estimates ( $\Sigma^x, \Sigma^v, \Sigma^\theta, \Sigma^\lambda$ ), with the help of the prior (learned) knowledge encoded in the robot brain, such that the estimate best predicts the data. The schematic of the proposed robot brain's inference process is given in Figure 1.



**Figure 1.** A simple block diagram of the robot brain's inference process using DEM. It uses the measurement data  $\mathbf{y}$  generated from the environment (also called generative process). DEM enables the direct fusion of the prior information into the inference process. The concept of generalized coordinates will be detailed in Section 3.1.

### 3. Preliminaries

To reformulate DEM into an estimation algorithm for LTI systems, this section introduces the key concepts and terminologies that are familiar to the FEP audience.

#### 3.1. Generative Model

The generative model (plant model) is the robot brain’s estimate of the generative process in the environment that generated data. The robot brain infers this model via model evidencing from the measurement data. The key idea behind DEM to deal with colored noise is to model the trajectory of the time-varying components (states, for example) using generalized coordinates. The use of generalized coordinates is new to the control systems literature and is different from the familiar definition in classical mechanics. In mechanics, it is the minimum number of independent coordinates that define the configuration of a system, whereas in DEM, it is the vector defining the motion of a point using its higher derivatives. In DEM, the emphasis is on tracking the trajectories of states, inputs and outputs instead of their point estimates. The states are expressed in generalized coordinates using its higher-order derivatives, i.e.,  $\tilde{x} = [x \ x' \ x'' \ \dots]^T$ . The generalized state vector  $\tilde{x}$  with an order of generalized motion of  $p$  will have  $p + 1$  terms, with the copy of the state vector as the first term, followed by its  $p$  derivatives. The variables in generalized coordinates are denoted by a tilde, and its components (higher derivatives) are denoted by primes. The evolution of states is written as:

$$\begin{aligned} x' &= Ax + Bv + w & y &= Cx + z \\ x'' &= Ax' + Bv' + w' & \dot{y} &= Cx' + z' \\ \dots & & \dots & \end{aligned} \tag{2}$$

The colored noises are modeled such that the covariance of noise derivatives  $\tilde{z} = [z, z', z'', \dots]^T$  and  $\tilde{w} = [w, w', w'', \dots]^T$  are well defined (to be explained in Section 3.3). The generative model representing the system is compactly written as:

$$\dot{\tilde{x}} = D^x \tilde{x} = \tilde{A} \tilde{x} + \tilde{B} \tilde{v} + \tilde{w} \quad \tilde{y} = \tilde{C} \tilde{x} + \tilde{z} \tag{3}$$

where  $D^x = \begin{bmatrix} 0 & 1 & & \\ 0 & 0 & 1 & \\ & & \ddots & \ddots \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix}_{(p+1) \times (p+1)} \otimes I_{n \times n}$ , performs the block derivative operation, equivalent to shifting up all components in generalized coordinates by one block. A similar definition holds for  $D^v$  (appears later) with size  $r(d + 1) \times r(d + 1)$ , where  $p$  and  $d$  are the order of generalized motion of states and inputs, respectively. Here,  $\tilde{A} = I_{p+1} \otimes A$ ,  $\tilde{B} = I_{p+1} \otimes B$ , and  $\tilde{C} = I_{p+1} \otimes C$ , where  $\otimes$  is the Kronecker tensor product.

#### 3.2. Parameters and Hyperparameters

To simplify the parameter estimation steps of DEM for LTI systems (Section 13.2) and to facilitate the convergence proof (Section 14), we introduce an alternative generative model which is the direct reformulation of Equation (3) given by:

$$\dot{\tilde{x}} = M\theta + \tilde{w}, \quad \tilde{y} = N\theta + \tilde{z}, \quad \theta = \begin{bmatrix} \text{vec}(A^T) \\ \text{vec}(B^T) \\ \text{vec}(C^T) \end{bmatrix} \tag{4}$$

where

$$M = \begin{bmatrix} I_n \otimes x^T & I_n \otimes v^T & I_n \otimes O_{1 \times m} \\ I_n \otimes x'^T & I_n \otimes v'^T & I_n \otimes O_{1 \times m} \\ \dots & \dots & \dots \end{bmatrix}, \quad N = \begin{bmatrix} I_n \otimes O_{1 \times n} & I_n \otimes O_{1 \times r} & I_m \otimes x^T \\ I_n \otimes O_{1 \times n} & I_n \otimes O_{1 \times r} & I_m \otimes x'^T \\ \dots & \dots & \dots \end{bmatrix}. \tag{5}$$

Throughout the paper,  $\theta$  represents the set of parameters, whereas  $\lambda = [\lambda^w]$  represents the hyperparameters that define the precision matrices (inverse covariance matrices) of the observation noise and the process noise. For noise modelling, we parametrize the noise precisions using an exponential relation with the hyperparameters, given by [18]:

$$\Pi^w(\lambda^w) = e^{\lambda^w} \Omega^w, \Pi^z(\lambda^z) = e^{\lambda^z} \Omega^z, \tag{6}$$

where  $\Omega^w$  and  $\Omega^z$  represent constant matrices encoding how different noises are correlated. Here  $\Pi^w$  and  $\Pi^z$  are the inverse covariances  $\{(\Sigma^w)^{-1}, (\Sigma^z)^{-1}\}$  or precisions of the noises. This parameterization ensures the selection of positive definite noise precision matrices through hyperparameter updates. We assume zero cross-correlation between  $w$  and  $z$ . We also assume that  $\theta$  and  $\lambda$  are time-invariant.

### 3.3. Colored noise

The next step towards handling the colored noise is to embed the noise correlation between different components in the generalized noises  $\tilde{w}$  and  $\tilde{z}$  given in Equation (3). DEM uses generalized coordinates, which models a correlation between noise derivatives through the temporal precision matrix  $S$  (inverse of covariance matrix). The generalized noise correlation is assumed to be due to a Gaussian filter, where  $S$  can be calculated as [18]:

$$S(\sigma^2) = \begin{bmatrix} 1 & 0 & -\frac{1}{2\sigma^2} & \dots \\ 0 & \frac{1}{2\sigma^2} & 0 & \dots \\ -\frac{1}{2\sigma^2} & 0 & \frac{3}{4\sigma^4} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}^{-1}_{(p+1) \times (p+1)} \tag{7}$$

where  $\sigma$  (from the Gaussian kernel) denotes the noise smoothness level. While  $\sigma^2 \rightarrow 0$  denotes white noise, nonzero  $\sigma^2$  denotes colored noise. The covariance between noise derivatives increases exponentially with the order of noise derivatives. Simulations show that derivatives above 6 can be neglected [18]. The generalized noise precision matrices are given by  $\tilde{\Pi}^w = S(\sigma^2) \otimes \Pi^w$  and  $\tilde{\Pi}^z = S(\sigma^2) \otimes \Pi^z$ , where  $\Pi^w$  and  $\Pi^z$  are the inverse noise covariances.

### 3.4. Generalized Motion of the Outputs And Noises

The generalized motion of output  $\tilde{\mathbf{y}}$  is practically expensive and inaccessible to robotics systems. Moreover, most sensors such as encoders operate with discrete measurements. Therefore, as a pre-processing step for estimation (refer to Figure 1),  $\tilde{\mathbf{y}}$  should be computed from discrete measurements. The goal is to first express the discrete measurements as a function of output derivatives and then invert the function to compute the derivatives from the discrete measurements. Given the  $p$  temporal derivatives  $\tilde{\mathbf{y}}$  at time  $t$ , the  $p$  output sequence surrounding  $\mathbf{y}$  can be approximated using Taylor series as follows [18]:

$$\hat{\mathbf{y}} = \begin{bmatrix} \dots \\ \mathbf{y}(t - dt) \\ \mathbf{y}(t) \\ \mathbf{y}(t + dt) \\ \dots \end{bmatrix} = (Y \otimes I_m) \tilde{\mathbf{y}}, Y_{ij} = \frac{[(i - \text{ceil}(\frac{p+1}{2}))dt]^{j-1}}{(j-1)!}, \tag{8}$$

where  $i, j = 1, 2, \dots, p + 1$ ,  $\text{ceil}(\cdot)$  is the smallest integer function and  $\hat{\mathbf{y}}$  has the size  $m(p + 1) \times 1$ . Therefore, generalized motion of output  $\tilde{\mathbf{y}}$  at time  $t$  is:

$$\tilde{\mathbf{y}} = (Y^{-1} \otimes I_m) \hat{\mathbf{y}}. \tag{9}$$

Using  $\tilde{\mathbf{y}}$  embeds more temporal information about the plant into the data in the form of conditional trajectories, with the disadvantage of a time latency of  $\frac{p}{2}dt$  in estimation. For robotic systems with high sampling rates, this latency in estimation is negligible [3,20]. The

next section employs this generalized output along with the generative model for observer designs.

### 3.5. Notations and conventions

Throughout the paper, the superscript notation will be used to represent the quantity being referred to, and the subscript notation will be used to represent the derivative operation. For example,  $\tilde{\Pi}_\lambda^w$  represents the derivative of the generalized precision of the process noise  $w$  with respect to the hyperparameters  $\lambda$ . The tilde operator ( $\tilde{x}$ ) is used to represent a quantity in its generalized coordinates, whereas the bar operator ( $\bar{F}$ ) is used to represent the time integral of a quantity. All the probability distributions will be represented by the  $p(\cdot)$  notation, whereas its expectation will be represented by the  $\langle p(\cdot) \rangle$  notation.

## 4. Free Energy Objectives

With the preliminaries in place, we can build up towards the complete DEM algorithm. Eventually, in Sections 8 and 12, we will see that it is an optimization algorithm that finds the best estimates for states, inputs, parameters, and hyperparameters for given measurement data. This result is achieved by optimizing two objective functions, which are the core objectives under the entire Free Energy Principle: the free energy and the free action [1]. Here we derive and simplify this objective.

The derivation starts from the fundamentals of Variational Inference (VI) [39]. In VI, the posterior distribution  $p(\vartheta/y)$  of parameter  $\vartheta$ , given the measurement  $y$ , is expressed as:

$$p(\vartheta/y) = \frac{p(y/\vartheta)p(\vartheta)}{p(y)} = \frac{p(\vartheta, y)}{p(y)} = \frac{p(\vartheta, y)}{\int p(\vartheta, y)d\vartheta}. \quad (10)$$

However, the marginalization over  $\vartheta$  to calculate  $p(y)$  is often intractable because the search space of  $\vartheta$  is large. A widely used technique is to introduce a variational distribution  $q(\vartheta)$  known as the recognition density, which acts as an approximate representation of the posterior distribution with  $q(\vartheta) \approx p(\vartheta/y)$ . A common method used among variational Bayes algorithms is to minimize the Kullback–Leibler (KL) divergence between both the distributions, defined as:

$$KL(q(\vartheta)||p(\vartheta/y)) = \langle \ln q(\vartheta) \rangle_{q(\vartheta)} - \langle \ln p(\vartheta/y) \rangle_{q(\vartheta)}, \quad (11)$$

where  $\langle \cdot \rangle_{q(\vartheta)}$  represents the expectation over  $q(\vartheta)$ . Substituting Equation (10) in Equation (11) and using  $\int q(\vartheta)d\vartheta = 1$  yields:

$$KL(q(\vartheta)||p(\vartheta/y)) = \langle \ln q(\vartheta) \rangle_{q(\vartheta)} - \langle \ln p(\vartheta, y) \rangle_{q(\vartheta)} + \ln p(y). \quad (12)$$

The rearrangement of terms yield:

$$\begin{aligned} \ln p(y) &= \langle \ln p(\vartheta, y) \rangle_{q(\vartheta)} - \langle \ln q(\vartheta) \rangle_{q(\vartheta)} + KL(q(\vartheta)||p(\vartheta/y)), \\ &= \underbrace{\langle U(\vartheta, y) \rangle_{q(\vartheta)} + H(\vartheta)_{q(\vartheta)}}_{\text{free energy}} + KL(q(\vartheta)||p(\vartheta/y)), \end{aligned} \quad (13)$$

where  $\ln p(y)$  is the log-evidence,  $U(\vartheta, y) = \ln p(\vartheta, y)$  is the internal energy and  $H(\vartheta)_{q(\vartheta)} = -\langle \ln q(\vartheta) \rangle_{q(\vartheta)}$  is the entropy of the density. The free energy term in Equation (13) is defined as the sum of an energy term and its entropy. It acts as the lower bound on the log-evidence because the KL divergence term  $KL(q(\vartheta)||p(\vartheta/y))$  is always positive. The maximization of free energy minimizes the divergence term in Equation (13) because the log-evidence is independent of  $q(\vartheta)$ , thereby rendering the variational density  $q(\vartheta)$  as a close approximation of  $p(\vartheta/y)$ . Therefore, the difficult evaluation of an intractable integral term in Equation (10) is converted into a much simpler optimization problem of maximizing the free energy. This reduces the problem of inference to a direct optimization

of its free energy objectives and is the fundamental idea behind variation inference. The free energy term in Equation (13) is equivalent to the Evidence Lower Bound (ELBO) [39]. It can be simplified as:

$$\begin{aligned} F &= \langle U(\vartheta, y) \rangle_{q(\vartheta)} + H(\vartheta)_{q(\vartheta)} = \int q(\vartheta) U(\vartheta, y) d\vartheta - \int q(\vartheta) \ln q(\vartheta) d\vartheta \\ &= \int q(\vartheta) \ln p(\vartheta, y) d\vartheta - \int q(\vartheta) \ln q(\vartheta) d\vartheta. \end{aligned} \quad (14)$$

However, when the parameter set to be estimated includes both the time-variant and the time-invariant parameters, the free action is used as the objective function to be maximized. The free action is defined as the time integral of the free energy and is given by:

$$\bar{F} = \bar{V}(\vartheta) + \bar{H}(\vartheta) = \int \langle U(\vartheta, y) \rangle_{q(\vartheta)} dt + \int H(\vartheta)_{q(\vartheta)} dt, \quad (15)$$

where  $V(\vartheta) = \langle U(\vartheta, y) \rangle_{q(\vartheta)}$  is called the variational free energy (VFE).

The next sections will deal with two main assumptions used in DEM to simplify the free energy objectives, namely the Laplace approximation and the mean-field assumption. The aim is to derive the simplified free energy objectives for an LTI system under these assumptions.

## 5. Laplace Approximation

The first common approach to simplify the free energy objective is to assume the variational density  $q(\vartheta)$  to be Gaussian in nature with variational parameters  $\vartheta$  and  $\Sigma^\vartheta$  as its mode and covariance, respectively, [38]. Here, the inverse of  $\Sigma^\vartheta$  (denoted by  $\Pi^\vartheta$  and known as the conditional precision), represents the confidence in estimation. The recognition density takes the following form:

$$q(\vartheta) = \mathcal{N}(\vartheta : \mu^\vartheta, \Sigma^\vartheta) = \frac{1}{\sqrt{(2\pi)^n |\Sigma^\vartheta|}} e^{-\frac{1}{2}(\vartheta - \mu^\vartheta)^T \Pi^\vartheta (\vartheta - \mu^\vartheta)}. \quad (16)$$

There are two main advantages with this approximation:

1. It simplifies the internal energy expression  $U(\vartheta, y)$ ,
2. It facilitates an easy computation of the conditional precision  $\Pi^\vartheta$  (derived in Section 7.2) as the negative curvature of the internal energy at its mode  $\mu^\vartheta$ .

Therefore, the main aim of this section is to simplify the expression for internal energy  $U(\vartheta, y)$  using the Laplace approximation, for an LTI system with its states, inputs, and outputs expressed in generalized coordinates.

The internal energy in Equation (13) can be expressed as the sum of log-likelihood and prior terms as:

$$U(\vartheta, y) = \ln p(\vartheta, y) = \ln \underbrace{p(y/\vartheta)}_{\text{generative model}} + \ln \underbrace{p(\vartheta)}_{\text{prior}}. \quad (17)$$

The parameter set  $\vartheta$  includes two types of parameters:

1. the states and inputs, which are time-varying and therefore expressed in generalized coordinates,
2. the parameters and hyperparameters, which are time-invariant and not expressed in generalized coordinates.

Equation (17) can be simplified by assuming the conditional independence of  $\tilde{x}$  and  $\tilde{v}$  with  $\theta$  and  $\lambda$ . This factorization separates the deterministic quantities from the stochastic ones, thereby providing a separation of temporal scales. This is one of the core ideas behind



DEM, which will be detailed in Section 6. With the redefinition of  $\vartheta = \{\tilde{x}, \tilde{v}, \theta, \lambda\}$ , Equation (17) simplifies to:

$$\begin{aligned}
 U(\vartheta, y) &= \ln p(y/\tilde{x}, \tilde{v}, \theta, \lambda) + \ln p(\tilde{x}, \tilde{v}, \theta, \lambda) \\
 &= \ln p(y/\tilde{x}, \tilde{v}, \theta, \lambda) + \ln p(\tilde{x}/\tilde{v}, \theta, \lambda) + \ln p(\tilde{v}) + \ln p(\theta) + \ln p(\lambda).
 \end{aligned}
 \tag{18}$$

DEM combines the new sensory information  $y$  coming from the environment with the robot brain’s priors (refer to Figure 1) in a Bayesian fashion, through the internal energy  $U(\vartheta, y)$  expression given in Equation (18). The next sections will deal with simplifying  $U(\vartheta, y)$  and its action  $\bar{U}$  by first modeling the probability distributions for the generative model and the priors.

### 5.1. Generative Model

The probability distribution  $p(y/\tilde{x}, \tilde{v}, \theta, \lambda)$ , given in Equation (18), represents the generative model that predicts the output from the current parameter estimates. This probability can be assumed to be Gaussian-distributed, centered around the model’s output prediction  $\tilde{C}\tilde{x}$  (from Equation (3)), with the same uncertainty as that of the measurement noise  $\tilde{\Sigma}^z$ . The distribution  $p(y/\vartheta)$  becomes:

$$p(\tilde{y}/\vartheta) = \frac{1}{\sqrt{(2\pi)^{m(p+1)}|\tilde{\Sigma}^z|}} e^{-\frac{1}{2}(\tilde{y}-\tilde{C}\tilde{x})^T\tilde{\Gamma}^z(\tilde{y}-\tilde{C}\tilde{x})}.
 \tag{19}$$

Since the robot cannot directly measure  $\tilde{x}$  in Equation (3), we track the motion of the generalized states through the approximation  $\dot{\tilde{x}} = \tilde{x}' = D^x\tilde{x}$ . The prediction for motion is  $\tilde{A}\tilde{x} + \tilde{B}\tilde{v}$  with an uncertainty of  $\tilde{\Sigma}^w$ . The Gaussian distribution becomes:

$$p(\tilde{x}'/\tilde{x}, \tilde{v}, \theta, \lambda) = \frac{1}{\sqrt{(2\pi)^{n(p+1)}|\tilde{\Sigma}^w|}} e^{-\frac{1}{2}(D^x\tilde{x}-\tilde{A}\tilde{x}-\tilde{B}\tilde{v})^T\tilde{\Gamma}^w(D^x\tilde{x}-\tilde{A}\tilde{x}-\tilde{B}\tilde{v})}.
 \tag{20}$$

### 5.2. Prior Distributions

The remaining distributions  $p(\tilde{v})$ ,  $p(\theta)$  and  $p(\lambda)$  are the priors of the robot brain that can be transferred from prior (learned) experiences to the inference process. Similar to the previous section, a Gaussian prior is placed over the inputs  $p(\tilde{v}) = \mathcal{N}(\tilde{v} : \eta^{\tilde{v}}, L^{\tilde{v}})$  as well, with mean  $\eta^{\tilde{v}}$  and prior covariance  $L^{\tilde{v}} = (P^{\tilde{v}})^{-1}$ , as:

$$p(\tilde{v}) = \frac{1}{\sqrt{(2\pi)^{r(d+1)}|L^{\tilde{v}}|}} e^{-\frac{1}{2}(\tilde{v}-\eta^{\tilde{v}})^T P^{\tilde{v}}(\tilde{v}-\eta^{\tilde{v}})}.
 \tag{21}$$

The prior distribution of parameters  $\theta \in \mathbb{R}^l$  is assumed to be a Gaussian-centred around the prior parameter value  $\eta^\theta$ , with the prior covariance  $L^\theta = (P^\theta)^{-1}$ :

$$p(\theta) = \mathcal{N}(\theta : \eta^\theta, L^\theta) = \frac{1}{\sqrt{(2\pi)^l|L^\theta|}} e^{-\frac{1}{2}(\theta-\eta^\theta)^T P^\theta(\theta-\eta^\theta)}.
 \tag{22}$$

Similarly, a Gaussian prior is placed over the hyperparameters  $\lambda \in \mathbb{R}^2$ :

$$p(\lambda) = \mathcal{N}(\lambda : \eta^\lambda, L^\lambda) = \frac{1}{\sqrt{(2\pi)^2|L^\lambda|}} e^{-\frac{1}{2}(\lambda-\eta^\lambda)^T P^\lambda(\lambda-\eta^\lambda)}.
 \tag{23}$$

A higher value of  $P^{\tilde{v}}$ ,  $P^\theta$  and  $P^\lambda$  represents the robot’s higher confidence in its prior estimates  $\eta^{\tilde{v}}$ ,  $\eta^\theta$ , and  $\eta^\lambda$ , respectively.

### 5.3. Simplification of the Internal Energy Action $\bar{U}$

This section aims at using the distributions from the previous sections to simplify  $\bar{U}$ . The logarithm of a Gaussian prior after dropping constants takes the general form:

$$\ln p(\theta) = \ln \mathcal{N}(\theta : \eta^\theta, L^\theta) = -\frac{1}{2}(\theta - \eta^\theta)^T (L^\theta)^{-1}(\theta - \eta^\theta) - \frac{1}{2} \ln |L^\theta| \quad (24)$$

Therefore, substituting Equations (19)-(23) in Equation (18) and simplifying it using the prediction error terms  $\epsilon^{\tilde{x}} = D^x \tilde{x} - \tilde{A} \tilde{x} - \tilde{B} \tilde{v}$ ,  $\epsilon^{\tilde{y}} = \tilde{y} - \tilde{C} \tilde{x}$ ,  $\epsilon^{\tilde{v}} = \tilde{v} - \eta^{\tilde{v}} \epsilon^\theta = \theta - \eta^\theta$ , and  $\epsilon^\lambda = \lambda - \eta^\lambda$ , after dropping constants, yields:

$$U(\vartheta, y) = -\frac{1}{2} \epsilon^{\theta T} P^\theta \epsilon^\theta + \frac{1}{2} \ln |P^\theta| - \frac{1}{2} \epsilon^{\lambda T} P^\lambda \epsilon^\lambda + \frac{1}{2} \ln |P^\lambda| - \frac{1}{2} \tilde{\epsilon}^T \tilde{\Gamma} \tilde{\epsilon} + \frac{1}{2} \ln |\tilde{\Gamma}|, \quad (25)$$

where  $\tilde{\epsilon} = \begin{bmatrix} \epsilon^{\tilde{y}} \\ \epsilon^{\tilde{v}} \\ \epsilon^{\tilde{x}} \end{bmatrix} = \begin{bmatrix} \tilde{y} - \tilde{C} \tilde{x} \\ \tilde{v} - \eta^{\tilde{v}} \\ D^x \tilde{x} - \tilde{A} \tilde{x} - \tilde{B} \tilde{v} \end{bmatrix}$ , and  $\tilde{\Gamma} = \text{diag}(\tilde{\Gamma}^z, P^{\tilde{v}}, \tilde{\Gamma}^w)$ . Here  $\text{diag}(\cdot, \cdot)$  is the block diagonal operator. Grouping the internal energy terms of the temporal and nontemporal parameters yields:

$$U(\vartheta, y) = U(t) + U(\theta) + U(\lambda), \quad (26)$$

where

$$U(t) = -\frac{1}{2} \tilde{\epsilon}^T \tilde{\Gamma} \tilde{\epsilon} + \frac{1}{2} \ln |\tilde{\Gamma}|. \quad (27)$$

Summing up the internal energy of all the temporal parameters over time yields the action of internal energy as follows:

$$\begin{aligned} \bar{U} &= U(\theta) + U(\lambda) + \sum_t U(t) \\ &= -\frac{1}{2} \epsilon^{\theta T} P^\theta \epsilon^\theta + \frac{1}{2} \ln |P^\theta| - \frac{1}{2} \epsilon^{\lambda T} P^\lambda \epsilon^\lambda + \frac{1}{2} \ln |P^\lambda| - \sum_t \frac{1}{2} \tilde{\epsilon}^T \tilde{\Gamma} \tilde{\epsilon} + \frac{1}{2} \sum_t \ln |\tilde{\Gamma}|. \end{aligned} \quad (28)$$

It can be observed from Equation (28) that the robot's priors  $(\eta^{\tilde{v}}, P^{\tilde{v}}, \eta^\theta, P^\theta, \eta^\lambda, P^\lambda)$  enter the free energy objective through the internal energy term. Intuitively, this can be seen as the direct influence of the robot's prior beliefs on the inference process through the mismatches in the robot's predictions. These weighed prediction errors drive the robot's desire to maintain an equilibrium between its internal model and the generative process in the environment. A large mismatch between the robot's predictions and the data results in a large prediction error, which gets precision-weighted and enters the free energy objective through its internal energy.

## 6. Mean-Field Approximation

The second widely used assumption for the simplification of free energy objectives is the factorization of parameters into independent subdensities for the recognition density [18], given by:

$$q(\vartheta) = \prod_i q(\vartheta^i) = q(\tilde{x})q(\tilde{v})q(\theta)q(\lambda). \quad (29)$$

This approximation assumes the conditional independence between subdensities (states and parameters, for example). The subdensities are assumed to interact with each other only through the mean-field quantities. The strong biological plausibility of this approximation in terms of biological brain's inference process is delineated in [27]. The main advantage of this approximation is the simplification of  $\bar{V}$  in Equation (15). However, the mathematical proof for this simplification is missing from the DEM literature [18]. Therefore, this section aims to fill this gap by deriving these proofs by delineating all the intermediate assumptions.

### 6.1. Simplification of the Entropy Action $\bar{H}$

The entropy of the density in Equation (14), given by  $H(\vartheta)_{q(\vartheta)} = -\langle \ln q(\vartheta) \rangle_{q(\vartheta)}$ , can be simplified for all parameters as:

$$H(\vartheta)_{q(\vartheta)} = - \int \int \int \int q(\tilde{x}, \tilde{v}, \theta, \lambda) \ln q(\tilde{x}, \tilde{v}, \theta, \lambda) d\tilde{x} d\tilde{v} d\theta d\lambda \tag{30}$$

Substituting the mean-field assumption given in Equation (29) in Equation (30) and using the property of normalized recognition densities  $\int q(\theta^i) d\theta^i = 1$  yields:

$$H(\vartheta)_{q(\vartheta)} = H(\theta) + H(\lambda) + H(t), \tag{31}$$

where  $H(\theta) = -\langle \ln q(\theta) \rangle_{q(\theta)}$ ,  $H(\lambda) = -\langle \ln q(\lambda) \rangle_{q(\lambda)}$  and  $H(t) = -\langle \ln q(\tilde{x}) \rangle_{q(\tilde{x})} - \langle \ln q(\tilde{v}) \rangle_{q(\tilde{v})}$ . We place the Laplace approximation over the marginals of the recognition densities of all parameters as:

$$\begin{aligned} q(\theta) &= \mathcal{N}(\theta : \mu^\theta, \Sigma^\theta) = \frac{1}{\sqrt{(2\pi)^l |\Sigma^\theta|}} e^{-\frac{1}{2}(\theta - \mu^\theta)^T \Pi^\theta (\theta - \mu^\theta)} \\ q(\lambda) &= \mathcal{N}(\lambda : \mu^\lambda, \Sigma^\lambda) = \frac{1}{\sqrt{(2\pi)^2 |\Sigma^\lambda|}} e^{-\frac{1}{2}(\lambda - \mu^\lambda)^T \Pi^\lambda (\lambda - \mu^\lambda)} \\ q(\tilde{x}) &= \mathcal{N}(\tilde{x} : \mu^{\tilde{x}}, \Sigma^{\tilde{x}}) = \frac{1}{\sqrt{(2\pi)^{n(p+1)} |\Sigma^{\tilde{x}}|}} e^{-\frac{1}{2}(\tilde{x} - \mu^{\tilde{x}})^T \Pi^{\tilde{x}} (\tilde{x} - \mu^{\tilde{x}})} \\ q(\tilde{v}) &= \mathcal{N}(\tilde{v} : \mu^{\tilde{v}}, \Sigma^{\tilde{v}}) = \frac{1}{\sqrt{(2\pi)^{r(d+1)} |\Sigma^{\tilde{v}}|}} e^{-\frac{1}{2}(\tilde{v} - \mu^{\tilde{v}})^T \Pi^{\tilde{v}} (\tilde{v} - \mu^{\tilde{v}})}. \end{aligned} \tag{32}$$

The recognition densities given in Equation (32) might look similar to the priors distributions given in Equation (21)-(23), mainly due to the common Gaussian distribution. The prior distributions are centered around the prior mean and prior covariances, whereas the recognition density is centered around the mean  $\mu^i$  and conditional covariance  $\Sigma^i$  of the  $i$ -th parameter set.

The action of entropy can be calculated by substituting Equation (32) in Equation (31) and summing up the entropy terms of time dependent parameters with respect to time. Upon dropping the constant terms, it simplifies to:

$$\bar{H} = H(\theta) + H(\lambda) + \sum_t H(t) = \frac{1}{2} \ln |\Sigma^\theta| + \frac{1}{2} \ln |\Sigma^\lambda| + \frac{1}{2} \sum_t \ln |\Sigma^{\tilde{x}}| + \frac{1}{2} \sum_t \ln |\Sigma^{\tilde{v}}|, \tag{33}$$

Equation (33) shows how the uncertainty in the robot's inference directly enters the objective  $\bar{F}$ , through  $\bar{H}$ .

### 6.2. Mean-Field Terms

Given the simplified expressions for  $\bar{U}$  and  $\bar{H}$ , the next step towards finding  $\bar{F}$  in Equation (15) is to evaluate  $\bar{V}$  given by:

$$\bar{V} = \int \langle U(y, \vartheta) \rangle_{q(\vartheta)} dt \tag{34}$$

$U(y, \vartheta)$  can be simplified using the second-degree Taylor series expansion near the mean  $\mu^\vartheta = \{\mu^{\tilde{x}}, \mu^{\tilde{v}}, \mu^\theta, \mu^\lambda\}$  as:

$$U(y, \tilde{x}, \tilde{v}, \theta, \lambda) = U(y, \mu^\vartheta) + \sum_{i=1}^4 U(y, \mu^\vartheta)_{\theta^i} (\theta^i - \mu^i) + \sum_{i=1}^4 \sum_{j=1}^4 (\theta^i - \mu^i)^T U(y, \mu^\vartheta)_{\theta^i \theta^j} (\theta^j - \mu^j), \tag{35}$$

where we use the shorthand  $U(y, \mu^\theta) = U(y, \theta)|_{\theta=\mu^\theta}$  and  $U(y, \mu^\theta)_\theta = U(y, \theta)_\theta|_{\theta=\mu^\theta}$ . This approximation of the internal energy has nontrivial implications in terms of the biological brain’s decision-making process [21]. The second order approximation is justified because the Laplace and mean-field approximations entail an internal energy that is quadratic in  $\tilde{x}$ ,  $\tilde{v}$ , and  $\theta$ , as given in Equation (25), thereby reducing all its higher-order derivatives to zero. Moreover, the higher derivatives of  $U(y, \mu^\theta)$  with respect to  $\lambda$ , reduce to zero because of the assumptions made in Section 9. By differentiating Equation (25) at the mean  $\mu^i$ ,  $U(y, \mu^\theta)_{\theta^i}$  can be found to be all zeroes, which upon substitution in Equation (35) simplifies to:

$$U(y, \theta) = U(y, \mu^\theta) + \sum_{i=1}^4 \sum_{j=1}^4 (\theta^i - \mu^i)^T U(y, \mu^\theta)_{\theta^i \theta^j} (\theta^j - \mu^j). \tag{36}$$

Substituting it in Equation (34), upon simplification yields:

$$\bar{V} = \bar{U}(y, \mu^\theta) + \int W dt \tag{37}$$

where

$$\begin{aligned} W &= \frac{1}{2} \sum_{i,j=1}^4 \langle (\theta^i - \mu^i)^T U(y, \mu^\theta)_{\theta^i \theta^j} (\theta^j - \mu^j) \rangle_{q(\theta^i)q(\theta^j)} \\ &= \frac{1}{2} \sum_{i,j=1}^4 \langle \text{tr} \left( (\theta^i - \mu^i)^T U(y, \mu^\theta)_{\theta^i \theta^j} (\theta^j - \mu^j) \right) \rangle_{q(\theta^i)q(\theta^j)} \\ &= \frac{1}{2} \sum_{i,j=1}^4 \langle \text{tr} \left( (\theta^j - \mu^j) (\theta^i - \mu^i)^T U(y, \mu^\theta)_{\theta^i \theta^j} \right) \rangle_{q(\theta^i)q(\theta^j)} \\ &= \frac{1}{2} \sum_{i,j=1}^4 \text{tr} [\langle (\theta^j - \mu^j) (\theta^i - \mu^i)^T \rangle_{q(\theta^i)q(\theta^j)} U(y, \mu^\theta)_{\theta^i \theta^j}] = \frac{1}{2} \sum_{i,j=1}^4 \text{tr} [\Sigma^{ij} U(y, \mu^\theta)_{\theta^i \theta^j}] \end{aligned}$$

Since the parameters  $\theta^i$  are assumed to be independent of each other, the covariance between them drops to zero, resulting in:

$$\begin{aligned} \bar{V} &= \bar{U}(y, \mu^\theta) + \frac{1}{2} \int \sum_{i=1}^4 \text{tr} [\Sigma^i U(y, \mu^\theta)_{\theta^i \theta^i}] dt \\ &= \bar{U}(y, \mu^\theta) + \int [W^{\tilde{x}} + W^{\tilde{v}} + W^\theta + W^\lambda] dt, \end{aligned} \tag{38}$$

where

$$W^{\theta^i} = \frac{1}{2} \text{tr} [\Sigma^i U(y, \mu^\theta)_{\theta^i \theta^i}] \tag{39}$$

is defined as the mean-field term. Equation (38) shows that  $\bar{V}$  is the sum of internal energy action and the mean field terms. The simplification of  $\bar{V}$  is one of the main advantages of using mean-field approximation. However, this approximation can be relaxed to build Generalized Filtering [37,42], which is mainly relevant to nonlinear identification. It involves the modeling of parameters and hyperparameters in generalized coordinates (together with states) for online system identification. However, in this work, we take a simpler approach.

### 7. Simplified Free Energy Objectives

This section aims at simplifying the free energy objectives using the results from Sections 5 and 6.

### 7.1. Simplification of Free Action

$\bar{F}$  is simplified by substituting Equations (38), (28), and (33) into Equation (15), yielding:

$$\begin{aligned} \bar{F} = & -\frac{1}{2}(\epsilon^{\theta T} P^\theta \epsilon^\theta)|_{\theta=\mu^\theta} + \frac{1}{2} \ln |P^\theta| - \frac{1}{2}(\epsilon^{\lambda T} P^\lambda \epsilon^\lambda)|_{\theta=\mu^\theta} + \frac{1}{2} \ln |P^\lambda| - \sum_t \frac{1}{2}(\tilde{\epsilon}^T \tilde{\Pi} \tilde{\epsilon})|_{\theta=\mu^\theta} \\ & + \frac{1}{2} \sum_t (\ln |\tilde{\Pi}|)|_{\theta=\mu^\theta} + \frac{1}{2} \sum_t (\ln |\Sigma^{\tilde{x}}| + \ln |\Sigma^{\tilde{v}}|) + \frac{1}{2} \ln |\Sigma^\theta| + \frac{1}{2} \ln |\Sigma^\lambda| \\ & + \frac{1}{2} \sum_t \text{tr}[\Sigma^{\tilde{x}} U(y, \mu^\theta)_{\tilde{x}\tilde{x}} + \Sigma^{\tilde{v}} U(y, \mu^\theta)_{\tilde{v}\tilde{v}} + \Sigma^\theta U(y, \mu^\theta)_{\theta\theta} + \Sigma^\lambda U(y, \mu^\theta)_{\lambda\lambda}]. \end{aligned} \tag{40}$$

We highlight three important terms in Equation (40): the combined prediction error of (generalized) outputs, inputs, and states  $E = \frac{1}{2}(\tilde{\epsilon}^T \tilde{\Pi} \tilde{\epsilon})|_{\theta=\mu^\theta}$ , the log determinant of noise precision  $G = (\ln |\tilde{\Pi}|)|_{\theta=\mu^\theta}$ , and the mean field term  $W^{\theta^i} = \frac{1}{2} \text{tr}[\Sigma^i U(y, \mu^\theta)_{\theta^i \theta^i}]$ . These terms will be used rigorously in the rest of the document.

### 7.2. Simplification of the Parameter Precisions

One of the main advantages of Laplace approximation is the simple evaluation of the covariance associated with the parameter estimation. This is achieved by setting the gradient of the free action with respect to individual parameter covariance as zero. The free action gradients with respect to covariance of parameters  $\theta^i$  is given by:

$$\frac{\partial \bar{F}}{\partial \Sigma^i} = \frac{\partial}{\partial \Sigma^i} \left( \frac{1}{2} \ln |\Sigma^i| + \frac{1}{2} \sum_t \text{tr}(\Sigma^i U(y, \mu^\theta)_{\theta^i \theta^i}) \right) = \frac{1}{2} ((\Sigma^i)^{-1} + \bar{U}(y, \mu^\theta)_{\theta^i \theta^i}). \tag{41}$$

The optimal parameter covariance is the covariance that maximizes the free action with zero gradients. Assuming that the parameter covariances are time-invariant, and equating the gradients to zero yields  $\Pi^i = -\bar{U}(y, \mu^\theta)_{\theta^i \theta^i}$ , resulting in:

$$\Pi^{\tilde{x}} = -U(y, \mu^\theta)_{\tilde{x}\tilde{x}}, \quad \Pi^{\tilde{v}} = -U(y, \mu^\theta)_{\tilde{v}\tilde{v}}, \quad \Pi^\theta = -\bar{U}(y, \mu^\theta)_{\theta\theta}, \quad \Pi^\lambda = -\bar{U}(y, \mu^\theta)_{\lambda\lambda} \tag{42}$$

From Equation (42), it is clear that the precision of parameters can be estimated just by evaluating the negative curvature of the internal energy at the conditional mode. These precision values denote the confidence of the estimator in the parameter estimate. Ideally, the parameter precision increases as the estimation process proceeds. Intuitively, the robot is more confident about its estimates (higher precision) when its predictions on the sensory data show the least variance.

### 7.3. Free Action at Optimal Precision

Substituting Equation (42) into Equation (39) at optimal precisions results in constant mean field terms. Therefore, the mean field terms in the free action given in Equation (40),

reduces to a constant under the optimal precision given by Equation (42). Therefore, the free action at optimal precision for an LTI system reduces to:

$$\begin{aligned}
 \bar{F} = & -\frac{1}{2} \sum_t \left[ \underbrace{(\tilde{\mathbf{y}} - \tilde{\mathbf{C}}\tilde{\mathbf{x}})^T \tilde{\Gamma}^z (\tilde{\mathbf{y}} - \tilde{\mathbf{C}}\tilde{\mathbf{x}})}_{\text{prediction error of outputs}} \Big|_{\vartheta=\mu^\vartheta} + \underbrace{(\tilde{\mathbf{v}} - \eta^{\tilde{\mathbf{v}}})^T P^{\tilde{\mathbf{v}}} (\tilde{\mathbf{v}} - \eta^{\tilde{\mathbf{v}}})}_{\text{prediction error of inputs}} \Big|_{\vartheta=\mu^\vartheta} \right] \\
 & - \frac{1}{2} \underbrace{(\theta - \eta^\theta)^T P^\theta (\theta - \eta^\theta)}_{\text{prediction error of parameters}} \Big|_{\vartheta=\mu^\vartheta} - \frac{1}{2} \underbrace{(\lambda - \eta^\lambda)^T P^\lambda (\lambda - \eta^\lambda)}_{\text{prediction error of hyperparameters}} \Big|_{\vartheta=\mu^\vartheta} \\
 & - \frac{1}{2} \sum_t \underbrace{(D^x \tilde{\mathbf{x}} - \tilde{\mathbf{A}}\tilde{\mathbf{x}} - \tilde{\mathbf{B}}\tilde{\mathbf{v}})^T \tilde{\Gamma}^w (D^x \tilde{\mathbf{x}} - \tilde{\mathbf{A}}\tilde{\mathbf{x}} - \tilde{\mathbf{B}}\tilde{\mathbf{v}})}_{\text{prediction error of (generalized) states}} \Big|_{\vartheta=\mu^\vartheta} + \underbrace{\frac{1}{2} n_t \ln |\Sigma^X|}_{\text{state and input entropy}} \\
 & + \frac{1}{2} n^t \underbrace{[\ln |\tilde{\Gamma}^z| + \ln |P^{\tilde{\mathbf{v}}}| + \ln |\tilde{\Gamma}^w|]}_{\text{noise entropy}} \Big|_{\vartheta=\mu^\vartheta} + \underbrace{\frac{1}{2} \ln |\Sigma^\theta P^\theta|}_{\text{parameter entropy}} + \underbrace{\frac{1}{2} \ln |\Sigma^\lambda P^\lambda|}_{\text{hyperparameter entropy}}
 \end{aligned} \tag{43}$$

Note that the free action is not a function of the latent variables ( $\vartheta = \{\tilde{\mathbf{x}}, \tilde{\mathbf{v}}, \theta, \lambda\}$ ) but the sufficient statistics ( $\mu^\vartheta, \Sigma^\vartheta$ ) of the approximate posterior. For example, the weighted output prediction error term of  $\bar{F}$  is  $(\tilde{\mathbf{y}} - \tilde{\mathbf{C}}\tilde{\mathbf{x}})^T \tilde{\Gamma}^z (\tilde{\mathbf{y}} - \tilde{\mathbf{C}}\tilde{\mathbf{x}}) \Big|_{\vartheta=\mu^\vartheta} = (\tilde{\mathbf{y}} - \mu^{\tilde{\mathbf{C}}} \mu^{\tilde{\mathbf{x}}})^T \tilde{\Gamma}^z \Big|_{\lambda^z=\mu^{\lambda^z}} (\tilde{\mathbf{y}} - \mu^{\tilde{\mathbf{C}}} \mu^{\tilde{\mathbf{x}}})$ , where  $\Big|_{\vartheta=\mu^\vartheta}$  denotes the evaluation at  $\vartheta = \mu^\vartheta$ , ( $\tilde{\mathbf{x}} = \mu^{\tilde{\mathbf{x}}}, \tilde{\mathbf{v}} = \mu^{\tilde{\mathbf{v}}}, \theta = \mu^\theta, \lambda = \mu^\lambda$ ). We regroup the time-dependent components into one variable  $X = \begin{bmatrix} \mu^{\tilde{\mathbf{x}}} \\ \mu^{\tilde{\mathbf{v}}} \end{bmatrix}$ , and use capital letters for the mean estimates of time-independent components ( $\Theta = \mu^\theta, \Lambda = \mu^\lambda$ ). Note that  $X, \Theta$ , and  $\Lambda$  are part of the generative model and are the mean estimate of the components of plant dynamics.

#### 7.4. Equivalence with the EM Algorithm

One of the most popular approaches to solve the Expectation Maximization (EM) algorithm for state space models is to use the Maximum Likelihood Estimation (MLE), where the objective function is the log-likelihood of data, given by [43]:

$$\begin{aligned}
 \ln L = & -\frac{1}{2} \sum_t (x_t - Ax_{t-1} - Bu_{t-1})^T \Pi^w (x_t - Ax_{t-1} - Bu_{t-1}) - \frac{n_t}{2} \ln |\Sigma^w| \\
 & - \frac{n_t}{2} \ln |\Sigma^z| - \frac{1}{2} \ln |\Sigma^x| - \frac{1}{2} (x_0 - \mu)^T \Pi^x (x_0 - \mu) - \frac{1}{2} \sum_t (y_t - Cx_t)^T \Pi^z (y_t - Cx_t).
 \end{aligned} \tag{44}$$

Comparing the objective functions of EM and DEM given by Equation (44) and Equation (43), respectively, DEM is equivalent to EM when:

- the mean field terms are neglected,
- the generalized motion is not considered, and
- the robot's priors on  $\vartheta$  are not considered.

Therefore, DEM can be considered as a generalized version of the EM algorithm with additional capabilities.

The free action at optimal precision (Equation 43) is the sum of prediction errors (PE) and entropy of generalized states, parameters, noise, and hyperparameters and is independent of mean-field terms. Although the mean-field terms turns into a constant at optimal precision, their gradients do not. This property is leveraged in Section 8 to account for the uncertainties in the parameter estimation during state estimation and vice versa, through the gradient of mean-field terms.

### 8. Update Rules for Estimation

The Free Energy objectives (Section 4), together with the two simplifying approximations (Sections 5 and 6), will be combined with an optimization procedure to form the ultimate DEM algorithm. The optimization procedure itself consists of the following two sets of update rules:

1. a gradient ascent over its free action  $\bar{F}$  for the time invariant parameters  $\theta$  and  $\lambda$ ,
2. a gradient ascent over its free energy  $F$  for the time varying parameters  $X$ ,

where  $F$  and  $\bar{F}$  are related through  $F = \frac{\partial \bar{F}}{\partial t}$ . The core idea is that the time varying parameters  $\tilde{x}$  and  $\tilde{\nu}$  can be estimated online from the robot’s instantaneous free energy, whereas the time-invariant parameters  $\theta$  and  $\lambda$  can be estimated from its free action after observing a sequence of data. Accordingly, the update rules for both the gradient ascends are given by

$$\frac{\partial \mu^i}{\partial a} = k^i \frac{\partial \bar{F}}{\partial \mu^i}, \tag{45}$$

for the  $a^{th}$  parameter update of the time invariant parameters and

$$\frac{\partial \mu^i}{\partial t} = D\mu^i + k^i \frac{\partial F}{\partial \mu^i}, \tag{46}$$

for the time-varying parameters, where  $k^i$  is the learning rate. The presence of  $D\mu^i$  term in Equation (46) differentiates the update rule from the general gradient ascent equation used in machine learning. This is to accommodate the boundary condition that when  $F$  is maximized,  $\frac{\partial F}{\partial \mu^i} = 0$  and  $\dot{\mu}^i = D\mu^i$ . In other words, when the free energy is maximized, the motion of the generalized states becomes their generalized motion [44]. However, the update equations for the time-invariant parameters,  $\theta$  and  $\lambda$ , do not require the  $D\mu^i$  term. Therefore, the update equations at  $t^{th}$  time,  $a^{th}$  parameter update step, and  $b^{th}$  hyperparameter update step, after regrouping the (generalized) states and inputs as  $X = \begin{bmatrix} \mu^{\tilde{x}} \\ \mu^{\tilde{\nu}} \end{bmatrix}$ , is given by:

$$\frac{\partial X}{\partial t} = DX + k^X \frac{\partial F}{\partial X}, \quad \frac{\partial \Theta}{\partial a} = k^\Theta \frac{\partial \bar{F}}{\partial \Theta}, \quad \frac{\partial \Lambda}{\partial b} = k^\Lambda \frac{\partial \bar{F}}{\partial \Lambda}, \tag{47}$$

where  $D = \text{diag}(D^x, D^v)$ . Note that the gradient update rules are written not on the latent variables ( $\tilde{x}, \tilde{\nu}, \theta$  and  $\lambda$ ), but on their mean estimates ( $X, \Theta$  and  $\Lambda$ ). Since the update rules should be implemented in the discrete domain for robotics applications, Equation (47) is discretized under local linearization with the corresponding Jacobians as  $J^X = D + k^X \frac{\partial^2 F}{\partial X^2}$ ,  $J^\Theta = k^\Theta \frac{\partial^2 \bar{F}}{\partial \Theta^2}$ , and  $J^\Lambda = k^\Lambda \frac{\partial^2 \bar{F}}{\partial \Lambda^2}$ . The (generalized) state and input update at time  $t$ , parameter update at step  $a$  and hyperparameter update at step  $b$  are given by:

$$\begin{aligned} X_{t+\Delta t} &= X_t + (e^{J^X \Delta t} - I)(J^X)^{-1} \frac{\partial X}{\partial t}, \\ \Theta_{a+1} &= \Theta_a + (e^{J^\Theta} - I)(J^\Theta)^{-1} \frac{\partial \Theta}{\partial a}, \\ \Lambda_{b+1} &= \Lambda_b + (e^{J^\Lambda} - I)(J^\Lambda)^{-1} \frac{\partial \Lambda}{\partial b}. \end{aligned} \tag{48}$$

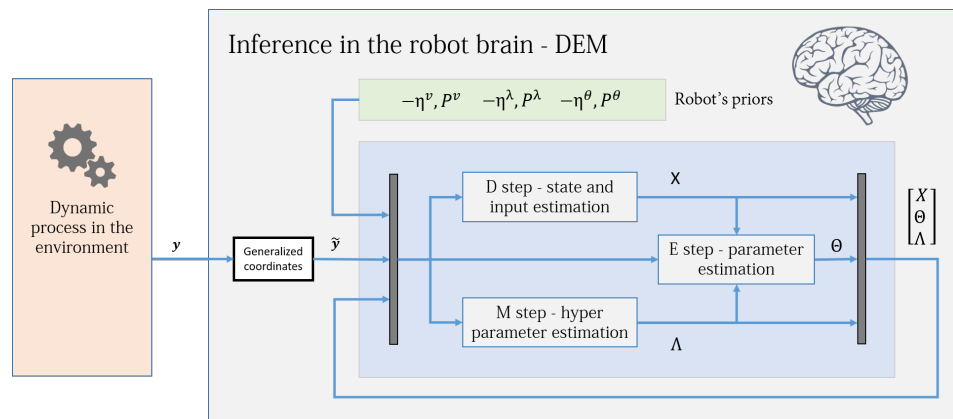
Equation (48) shows that the update rules are dependent only on the gradients and curvatures of the free energy objectives.

#### 8.1. The DEM Algorithm

The DEM algorithm is an iterative model inversion algorithm that uses Equations (48) and (42) to perform estimation on causal dynamic systems. It can be expressed using three steps:

1. D step: (generalized) state and input estimation,
2. E step: parameter estimation,
3. M step: noise hyperparameter estimation,

a nomenclature that is similar to the EM algorithm. Figure 2 shows an intuitive block diagram that demonstrates the inference process of DEM as a coupled dynamics between D, E, and M steps. The data from the environment and the robot brain’s prior distributions are used to infer the generative process. The pseudocode given in Algorithm 1 demonstrates how DEM performs estimation using only the gradient and curvatures of the free energy objectives. The next sections will focus on deriving the algebraic expressions for these quantities.



**Figure 2.** The DEM algorithm is represented using three coupled steps: D, E, and M steps. The algorithm combines the data from the environment with the robot’s prior beliefs to infer the states, inputs, parameters and hyperparameters of the system. For each parameter update in the E step, the D step updates the (generalized) states and inputs for all times instances, and the M step iterates until hyperparameter convergence, as demonstrated in Algorithm 1. The dynamic process is the generative model in Section 3.1, the priors are the distributions given in Section 5.2 and the generalized coordinates block is defined in Section 3.4. Section 13 will elaborate on the D, E, and M blocks.

### 8.2. Updated Equations for Estimation

The free energy gradients in Equation (47) can be evaluated by differentiating Equation (40) with  $\mu^i$ . Substituting the resulting expression in Equation (47), upon simplification yields:

$$\begin{aligned} \dot{X} &= DX + k^X \left[ -E_X + W_X^X + W_X^\Theta + W_X^\Lambda \right] \\ \frac{\partial \Theta}{\partial a} &= k^\Theta \left[ -P^\theta \epsilon^\theta \Big|_{\vartheta=\mu^\theta} + \sum_t \left( -E_\Theta + W_\Theta^X + W_\Theta^\Theta + W_\Theta^\Lambda \right) \right] \\ \frac{\partial \Lambda}{\partial b} &= k^\Lambda \left[ -P^\lambda \epsilon^\lambda \Big|_{\vartheta=\mu^\theta} + \sum_t \left( -E_\Lambda + W_\Lambda^X + W_\Lambda^\Theta + W_\Lambda^\Lambda + G_\Lambda \right) \right], \end{aligned} \tag{49}$$

where  $E_{\mu^i} = \frac{\partial E}{\partial \mu^i} = \frac{1}{2} \frac{\partial(\tilde{\epsilon}^T \tilde{\Gamma} \tilde{\epsilon})}{\partial \mu^i} \Big|_{\vartheta=\mu^\theta}$  is the gradient of the prediction error with respect to  $\mu^i$ ,  $W_{\mu^j}^{\mu^i} = \frac{\partial W_{\mu^j}^{\mu^i}}{\partial \mu^j} = \frac{1}{2} \frac{\partial}{\partial \mu^j} tr[\Sigma^i U(y, \mu^\theta)_{\mu^i \mu^i}]$  is the gradient of the mean field term of  $\mu^i$  with respect to  $\mu^j$ , and  $G_\Lambda = \frac{1}{2} \frac{\partial}{\partial \Lambda} \ln |\tilde{\Gamma}| \Big|_{\vartheta=\mu^\theta}$  is the gradient of the log-determinant of the



precision matrix with respect to  $\Lambda$ . From Equation (49), the Jacobians that are required for the updates given in Equation (48) can be evaluated as:

$$\begin{aligned}
 J^X &= D + k^X \left[ -E_{XX} + W_{XX}^X + W_{XX}^\Theta + W_{XX}^\Lambda \right] \\
 J^\Theta &= k^\Theta \left[ -P^\theta + \sum_t \left( -E_{\Theta\Theta} + W_{\Theta\Theta}^X + W_{\Theta\Theta}^\Theta + W_{\Theta\Theta}^\Lambda \right) \right] \\
 J^\Lambda &= k^\Lambda \left[ -P^\lambda + \sum_t \left( -E_{\Lambda\Lambda} + W_{\Lambda\Lambda}^X + W_{\Lambda\Lambda}^\Theta + W_{\Lambda\Lambda}^\Lambda + G_{\Lambda\Lambda} \right) \right]
 \end{aligned}
 \tag{50}$$

The Jacobians are employed in different loops corresponding to the D, E, and M steps in the Algorithm 1.

---

**Algorithm 1:** Dynamics Expectation Maximization

---

**Data:** Time series data of output  $y$

**Result:**  $X, \Theta, \Lambda, \Pi^{\bar{x}, \bar{v}}, \Pi^\theta, \Pi^\lambda$

$a = 0$ ;

**while**  $\Theta$  not converged **do**

**for**  $t = 0:\Delta t:T$  **do**

    ▷ D step

$\frac{\partial X}{\partial t} = DX + k^X \frac{\partial F}{\partial X}$ ;

$J^X = D + k^X \frac{\partial^2 F}{\partial X^2}$ ;

$X_{t+\Delta t} = X_t + (e^{J^X \Delta t} - I)(J^X)^{-1} \frac{\partial X}{\partial t}$ ;

**end**

**while**  $\Lambda$  not converged **do**

    ▷ M step

$\frac{\partial \Lambda}{\partial b} = k^\Lambda \frac{\partial \bar{F}}{\partial \Lambda}$ ;

$J^\Lambda = k^\Lambda \frac{\partial^2 \bar{F}}{\partial \Lambda^2}$ ;

$\Lambda \leftarrow \Lambda + (e^{J^\Lambda} - I)(J^\Lambda)^{-1} \frac{\partial \Lambda}{\partial b}$ ;

**end**

$\bar{F}_a \leftarrow$  Equation (43)

    ▷  $\bar{F}$  at optimal precision

**if**  $\bar{F}_a > \bar{F}_{a-1}$  **then**

    ▷ update  $\Theta$  if  $\bar{F}$  increased

$\frac{\partial \Theta}{\partial a} = k^\Theta \frac{\partial \bar{F}}{\partial \Theta}$ ;

$J^\Theta = k^\Theta \frac{\partial^2 \bar{F}}{\partial \Theta^2}$ ;

$\Theta \leftarrow \Theta + (e^{J^\Theta} - I)(J^\Theta)^{-1} \frac{\partial \Theta}{\partial a}$ ;

    ▷ E step

**end**

$a++$ ;

$\Pi^{\bar{x}, \bar{v}} = -U(y, \mu^\theta) \begin{bmatrix} \bar{x} \\ \bar{v} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{v} \end{bmatrix}$ ;

    ▷ update generalized state precision

$\Pi^\theta = -\bar{U}(y, \mu^\theta)_{\theta\theta}$ ;

    ▷ parameter precision

$\Pi^\lambda = -\bar{U}(y, \mu^\theta)_{\lambda\lambda}$ ;

    ▷ hyperparameter precision

**end**

---

**8.3. Update Equation for Precision of Estimates**

The uncertainty in estimation is represented by the inverse of precision matrices  $\Pi^{\theta^j}$ . Differentiating Equations (25) and (28) twice and substituting it into Equation (42) upon simplification yield:

$$\begin{aligned}
 \Pi^{\bar{x}, \bar{v}} &= -U(y, \mu^\theta) \begin{bmatrix} \bar{x} \\ \bar{v} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{v} \end{bmatrix} = E_{XX} \\
 \Pi^\theta &= -\bar{U}(y, \mu^\theta)_{\theta\theta} = P^\theta + \sum_t E_{\Theta\Theta} \\
 \Pi^\lambda &= -\bar{U}(y, \mu^\theta)_{\lambda\lambda} = P^\lambda + \sum_t (E_{\Lambda\Lambda} - G_{\Lambda\Lambda})
 \end{aligned}
 \tag{51}$$

The only unknowns in the DEM update equations given by Equations (49), (50), and (51) are the gradients and curvatures of  $E$ ,  $W$ , and  $G$ . Sections 9–11 will deal with evaluating the simplified algebraic expressions for these gradients and curvatures. Using these simplifications, Section 12 will proceed towards expanding Algorithm 1.

### 9. Gradients of (Log Determinant of) Precision

This section aims at evaluating the gradients of the log determinant of noise precision ( $G_\Lambda, G_{\Lambda\Lambda}$ ) that are required for the hyperparameter update rules of the DEM algorithm. The precision matrix for hyperparameter estimation is modeled as:

$$\tilde{\Pi} = \text{diag}(\tilde{\Pi}^z, P^{\tilde{v}}, \tilde{\Pi}^w) = \text{diag}(e^{\lambda^z} (S \otimes \Omega^z), P^{\tilde{v}}, e^{\lambda^w} (S \otimes \Omega^w)), \tag{52}$$

where  $S$  is the noise smoothness matrix given by Equation (7) and  $\Omega^z, \Omega^w$  are the constant matrices given in Equation (6). Here the precision matrix is parametrized using  $\lambda = \begin{bmatrix} \lambda^z \\ \lambda^w \end{bmatrix}$ , which is the only unknown in Equation (52). Therefore, the log determinant of precision and its gradients can be written as:

$$\begin{aligned} G &= \ln |\tilde{\Pi}| \Big|_{\theta=\mu^\theta}, \quad G_\Lambda = \frac{1}{2} \begin{bmatrix} \text{tr}(\tilde{\Pi}_{\lambda^z} \tilde{\Pi}^{-1}) \\ \text{tr}(\tilde{\Pi}_{\lambda^w} \tilde{\Pi}^{-1}) \end{bmatrix} \Big|_{\theta=\mu^\theta}, \\ G_{\Lambda^i \Lambda^j} &= \frac{1}{2} \text{tr}(\tilde{\Pi}_{\lambda^i \lambda^j} \tilde{\Pi}^{-1} - \tilde{\Pi}_{\lambda^i} \tilde{\Pi}^{-1} \tilde{\Pi}_{\lambda^j} \tilde{\Pi}^{-1}) \Big|_{\theta=\mu^\theta}, \end{aligned} \tag{53}$$

where  $\lambda^i$  is the  $i^{\text{th}}$  element in  $\lambda$ . The gradients of precision in Equation (53) can be evaluated by differentiating Equation (52) as:

$$\begin{aligned} \tilde{\Pi}_{\lambda^z} &= \text{diag}(e^{\lambda^z} (S \otimes \Omega^z), O, O) = \text{diag}(\tilde{\Pi}^z, O, O), \\ \tilde{\Pi}_{\lambda^w} &= \text{diag}(O, O, e^{\lambda^w} (S \otimes \Omega^z)) = \text{diag}(O, O, \tilde{\Pi}^z), \\ \tilde{\Pi}_{\lambda^z \lambda^w} &= O, \quad \tilde{\Pi}_{\lambda^w \lambda^z} = O. \end{aligned} \tag{54}$$

Substituting Equations (54) and (52) in Equation (53) yields:

$$G_\Lambda = \frac{1}{2} \begin{bmatrix} \text{tr}(I_{n^{\tilde{\Pi}^z}}) \\ \text{tr}(I_{n^{\tilde{\Pi}^w}}) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} n^{\tilde{\Pi}^z} \\ n^{\tilde{\Pi}^w} \end{bmatrix}, \tag{55}$$

where  $I_{n^{\tilde{\Pi}^z}}$  is the identity matrix of size  $n^{\tilde{\Pi}^z}$ , which is the size of the  $\tilde{\Pi}^z$  matrix.  $\Pi^w$  and  $\Pi^z$  are modeled to have an exponential relation with  $\lambda$ , so that any updates on  $\lambda$  would result in positive semi-definite precision matrices. However, this relation entails an infinitely differentiable precision matrix with respect to  $\lambda$ , increasing the computational complexity of the algorithm. Therefore, an approximation is made by forcefully setting  $\tilde{\Pi}_{\lambda^z \lambda^z} = \tilde{\Pi}_{\lambda^w \lambda^w} = O$ , while maintaining the exponential relation between  $\tilde{\Pi}$  and  $\lambda$ , thereby ensuring that the optimization process proceeds along the correct gradients  $\tilde{\Pi}_\lambda$ . Together with Equation (54), this approximation results in  $\tilde{\Pi}_{\lambda\lambda} = O$ . This assumption has two direct consequences:

- It simplifies all the update rules given in Equations (49) and (50),
- It simplifies the precision update rule for hyperparameters given in Equation (51).

A direct consequence of this approximation is in the simplification of  $G_{\Lambda\Lambda}$  in Equation (53), expressed as:

$$G_{\Lambda\Lambda} = -\frac{1}{2} \begin{bmatrix} \text{tr}(\tilde{\Pi}_{\lambda^z} \tilde{\Pi}^{-1} \tilde{\Pi}_{\lambda^z} \tilde{\Pi}^{-1}) & \text{tr}(\tilde{\Pi}_{\lambda^z} \tilde{\Pi}^{-1} \tilde{\Pi}_{\lambda^w} \tilde{\Pi}^{-1}) \\ \text{tr}(\tilde{\Pi}_{\lambda^w} \tilde{\Pi}^{-1} \tilde{\Pi}_{\lambda^z} \tilde{\Pi}^{-1}) & \text{tr}(\tilde{\Pi}_{\lambda^w} \tilde{\Pi}^{-1} \tilde{\Pi}_{\lambda^w} \tilde{\Pi}^{-1}) \end{bmatrix} \Big|_{\theta=\mu^\theta}, \tag{56}$$

which upon substitution of Equations (52) and Equation (54) yields:

$$G_{\Lambda\Lambda} = -\frac{1}{2} \begin{bmatrix} \text{tr}(I_{n^{\tilde{\Gamma}^z}}) & O \\ O & \text{tr}(I_{n^{\tilde{\Gamma}^w}}) \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} n^{\tilde{\Gamma}^z} & O \\ O & n^{\tilde{\Gamma}^w} \end{bmatrix}, \tag{57}$$

where  $n^{\tilde{\Gamma}^z} = m(p + 1)$  and  $n^{\tilde{\Gamma}^w} = n(p + 1)$  are the sizes of  $\tilde{\Gamma}^z$  and  $\tilde{\Gamma}^w$ , respectively. From Equations (55) and (57),  $G_\Lambda$  and  $G_{\Lambda\Lambda}$  are constants, and can be pre-computed in Algorithm 1.

**10. Gradients of Prediction Error**

As opposed to the result above, the gradients of the prediction errors are *not* constant, as is shown in this section.

*10.1. Gradients of Prediction Error Along (Generalized) States*

The error in prediction of (generalized) outputs, inputs and states is represented together by  $\tilde{\epsilon}$ , which makes up the precision weighted prediction error defined by  $E = \frac{1}{2} \tilde{\epsilon}^T \tilde{\Pi} \tilde{\epsilon} \Big|_{\theta=\mu^\theta}$ , where  $\tilde{\Pi} = \text{diag}(\tilde{\Gamma}^z, P^{\tilde{v}}, \tilde{\Gamma}^w)$ . The error and its gradient are:

$$\tilde{\epsilon} = \begin{bmatrix} \tilde{\mathbf{y}} - \tilde{C}\tilde{x} \\ \tilde{v} - \eta^{\tilde{v}} \\ D^x \tilde{x} - \tilde{A}\tilde{x} - \tilde{B}\tilde{v} \end{bmatrix}, \tilde{\epsilon}_X \Big|_{\theta=\mu^\theta} = \begin{bmatrix} -\tilde{C} & O \\ O & I \\ D^x - \tilde{A} & -\tilde{B} \end{bmatrix}. \tag{58}$$

The gradient of prediction error with respect to  $X$  can be simplified as:

$$E_X = \tilde{\epsilon}_X^T \tilde{\Pi} \tilde{\epsilon} \Big|_{\theta=\mu^\theta} = A_1 X + B_1 \begin{bmatrix} \tilde{\mathbf{y}} \\ -\eta^{\tilde{v}} \end{bmatrix}, \tag{59}$$

where  $A_1 = \begin{bmatrix} \tilde{C}^T \tilde{\Gamma}^z \tilde{C} + (D^x - \tilde{A})^T \tilde{\Gamma}^w (D^x - \tilde{A}) & -(D^x - \tilde{A})^T \tilde{\Gamma}^w \tilde{B} \\ -\tilde{B}^T \tilde{\Gamma}^w (D^x - \tilde{A}) & P^{\tilde{v}} + \tilde{B}^T \tilde{\Gamma}^w \tilde{B} \end{bmatrix} \Big|_{\theta=\mu^\theta}$  and

$B_1 = \begin{bmatrix} -\tilde{C}^T \tilde{\Gamma}^z & O \\ O & P^{\tilde{v}} \end{bmatrix} \Big|_{\theta=\mu^\theta}$ . Since  $E_X$  is linear in  $X$ , differentiating Equation (59) with respect to  $X$  yields a simple expression for curvature  $E_{XX}$  as:

$$E_{XX} = \tilde{\epsilon}_X^T \tilde{\Pi} \tilde{\epsilon}_X \Big|_{\theta=\mu^\theta} = \begin{bmatrix} \tilde{C}^T \tilde{\Gamma}^z \tilde{C} + (D^x - \tilde{A})^T \tilde{\Gamma}^w (D^x - \tilde{A}) & -(D^x - \tilde{A})^T \tilde{\Gamma}^w \tilde{B} \\ -\tilde{B}^T \tilde{\Gamma}^w (D^x - \tilde{A}) & P^{\tilde{v}} + \tilde{B}^T \tilde{\Gamma}^w \tilde{B} \end{bmatrix} \Big|_{\theta=\mu^\theta}. \tag{60}$$

*10.2. Gradients of Prediction Error Along Parameters*

To evaluate the gradients of prediction error along the parameters  $\Theta$ , the reformulated definition of  $\tilde{\epsilon}$  is used:

$$\tilde{\epsilon} = \begin{bmatrix} \tilde{\mathbf{y}} - N\theta \\ \tilde{v} - \eta^{\tilde{v}} \\ D^x \tilde{x} - M\theta \end{bmatrix}, \tilde{\epsilon}_\Theta = (\tilde{\epsilon}_\theta) \Big|_{\theta=\mu^\theta} = \begin{bmatrix} -N \\ O \\ -M \end{bmatrix}, \tag{61}$$

where  $M$  and  $N$  are given in Equation (5). This is to ensure that the variable  $\Theta$  can be separated out of the expression for  $E_\Theta$  such that it is linear in  $\Theta$  as follows:

$$\begin{aligned}
 E_\Theta &= (\tilde{\epsilon}_\theta^T \tilde{\Gamma} \tilde{\epsilon}_\theta) \Big|_{\vartheta=\mu^\vartheta} = -[N^T \quad 0 \quad M^T] \tilde{\Gamma} \begin{bmatrix} \tilde{\mathbf{y}} - N\theta \\ \tilde{\nu} - \tilde{\eta} \\ D^x \tilde{\mathbf{x}} - M\theta \end{bmatrix} \Big|_{\vartheta=\mu^\vartheta} \\
 &= (N^T \tilde{\Gamma}^z N + M^T \tilde{\Gamma}^w M) \Big|_{\vartheta=\mu^\vartheta} \Theta - [N^T \tilde{\Gamma}^z \quad M^T \tilde{\Gamma}^w D^x] \Big|_{\vartheta=\mu^\vartheta} \begin{bmatrix} \tilde{\mathbf{y}} \\ \mu^{\tilde{\mathbf{x}}} \end{bmatrix} \\
 &= A_2 \Theta - B_2 \begin{bmatrix} \tilde{\mathbf{y}} \\ \mu^{\tilde{\mathbf{x}}} \end{bmatrix}.
 \end{aligned} \tag{62}$$

Since  $E_\Theta$  is linear in  $\Theta$ , differentiating Equation (62) with respect to  $\Theta$  yields a simple expression for  $E_{\Theta\Theta}$  as:

$$E_{\Theta\Theta} = (\tilde{\epsilon}_\theta^T \tilde{\Gamma} \tilde{\epsilon}_\theta) \Big|_{\vartheta=\mu^\vartheta} = [N^T \tilde{\Gamma}^z N + M^T \tilde{\Gamma}^w M] \Big|_{\vartheta=\mu^\vartheta} = A_2. \tag{63}$$

### 10.3. Gradients of Prediction Error Along Hyperparameters

The gradients of prediction error along the hyperparameters  $\Lambda$  is simpler, and is given by  $E_\Lambda = \frac{1}{2} \tilde{\epsilon}^T \tilde{\Gamma}_\lambda \tilde{\epsilon} \Big|_{\vartheta=\mu^\vartheta}$ ,  $E_{\Lambda\Lambda} = \frac{1}{2} \tilde{\epsilon}^T \tilde{\Gamma}_{\lambda\lambda} \tilde{\epsilon} \Big|_{\vartheta=\mu^\vartheta}$ , which upon using  $\tilde{\Gamma}_{\lambda\lambda} = 0$ , gives:

$$\begin{aligned}
 E_\Lambda &= \frac{1}{2} \text{tr}(\tilde{\Gamma}_\lambda \tilde{\epsilon} \tilde{\epsilon}^T) \Big|_{\vartheta=\mu^\vartheta} = \frac{1}{2} \left[ \begin{array}{c} \text{tr}(\tilde{\Gamma}^z (\tilde{\mathbf{y}} - \tilde{C}\tilde{\mathbf{x}})(\tilde{\mathbf{y}} - \tilde{C}\tilde{\mathbf{x}})^T) \\ \text{tr}(\tilde{\Gamma}^w (D^x \tilde{\mathbf{x}} - \tilde{A}\tilde{\mathbf{x}} - \tilde{B}\tilde{\mathbf{u}})(D^x \tilde{\mathbf{x}} - \tilde{A}\tilde{\mathbf{x}} - \tilde{B}\tilde{\mathbf{u}})^T) \end{array} \right] \Big|_{\vartheta=\mu^\vartheta}, \\
 E_{\Lambda\Lambda} &= \frac{1}{2} \tilde{\epsilon}^T \tilde{\Gamma}_{\lambda\lambda} \tilde{\epsilon} \Big|_{\vartheta=\mu^\vartheta} = 0.
 \end{aligned} \tag{64}$$

In summary, Equations (59), (62), and (64) represent the gradients of the prediction error term, whereas Equations (60), (63), and (64) represent its curvatures. The next section will deal with evaluating the analytic expressions for all the gradients and curvatures of the mean field term.

## 11. Gradients of Mean Field Terms

This section aims to derive the analytic expressions for the mean field terms and their gradients:  $W_{\mu^j}^{\mu^i}$  and  $W_{\mu^j \mu^k}^{\mu^i}$ ,  $\forall \mu \in \{X, \Theta, \Lambda\}$ .

### 11.1. Gradients of mean field terms along hyperparameters

In this section, we prove that all the gradients and curvatures of  $W^\Lambda$  (namely  $W_{\mu^i}^\Lambda$  and  $W_{\mu^i \mu^j}^\Lambda$ ) are zeroes. The mean field term for hyperparameters  $\Lambda$  can be expressed as:

$$W^\Lambda = \frac{1}{2} \text{tr}[\Sigma^\Lambda U(y, \mu^\vartheta)_{\lambda\lambda}]. \tag{65}$$

To compute the gradients of  $W^\Lambda$ , we need the curvature of internal energy with respect to  $\lambda$ . This can be evaluated by first differentiating Equation (25) with respect to  $\lambda$  and evaluating it at  $\vartheta = \mu^\vartheta$ , which yields:

$$U(y, \mu^\vartheta)_\lambda = U(y, \vartheta)_\lambda \Big|_{\vartheta=\mu^\vartheta} = -P^\lambda \epsilon^\lambda \Big|_{\vartheta=\mu^\vartheta} - \frac{1}{2} (\tilde{\epsilon}^T \tilde{\Gamma}_\lambda \tilde{\epsilon}) \Big|_{\vartheta=\mu^\vartheta} + G_\Lambda, \tag{66}$$

where  $G_\Lambda$  is given by Equation (55). Upon further differentiation, we get:

$$U(y, \mu^\vartheta)_{\lambda\lambda} = -P^\lambda - \frac{1}{2} (\tilde{\epsilon}^T \tilde{\Gamma}_{\lambda\lambda} \tilde{\epsilon}) \Big|_{\vartheta=\mu^\vartheta} + G_{\Lambda\Lambda}. \tag{67}$$

The assumption of  $\tilde{\Pi}_{\lambda\lambda} = 0$  applied to Equation (67) yields:

$$U(y, \mu^\theta)_{\lambda\lambda} = -P^\lambda + G_{\Lambda\Lambda} \tag{68}$$

which contains only constants. Therefore, the assumption of  $\tilde{\Pi}_{\lambda\lambda} = 0$  reduces all the gradients and curvatures of mean field terms of  $\Lambda$  to zeros:

$$W_{\mu^i}^\Lambda = \frac{1}{2} tr \left[ \Sigma^\Lambda U(y, \mu^\theta)_{\lambda\lambda\mu^i} \right] = 0, W_{\mu^i\mu^i}^\Lambda = 0. \tag{69}$$

Since the internal energy given in Equation (25) is quadratic in  $\theta^i$ , and since  $\tilde{\Pi}_{\lambda\lambda} = 0$ , all the gradients and curvatures of the mean field term of  $\theta^i$  with respect to itself are zeros:

$$W_{\mu^i}^{\mu^i} = \frac{1}{2} tr \left[ \Sigma^{\theta^i} U(y, \mu^\theta)_{\theta^i\theta^i\mu^i} \right] = 0, W_{\mu^i\mu^i}^{\mu^i} = 0. \tag{70}$$

### 11.2. Gradients of Mean Field Terms Along Generalized States

The mean-field term of the combined generalized states  $X$  can be expressed as:

$$W^X = \frac{1}{2} tr \left[ \Sigma^X U(y, \mu^\theta)_{XX} \right]. \tag{71}$$

The curvature of internal energy with respect to  $X$  can be calculated by differentiating Equation (25) with respect to  $X$  twice, resulting in  $U(y, \mu^\theta)_{XX} = -\frac{1}{2} \tilde{\epsilon}_X^T \tilde{\Pi} \tilde{\epsilon}_X$ . Substituting it in Equation (71) upon differentiation with  $\Theta$  yields:

$$\begin{aligned} W_{\Theta^i}^X &= -\frac{1}{2} tr \left( \Sigma^X \tilde{\epsilon}_{X\Theta^i}^T \tilde{\Pi} \tilde{\epsilon}_X \right) \Big|_{\theta=\mu^\theta} = -\frac{1}{2} tr \left( \Sigma^X \begin{bmatrix} -\tilde{C}_{\theta^i}^T & O & -\tilde{A}_{\theta^i}^T \\ O & O & -\tilde{B}_{\theta^i}^T \end{bmatrix} \tilde{\Pi} \begin{bmatrix} -\tilde{C} & O \\ O & I \\ D^x - \tilde{A} & -\tilde{B} \end{bmatrix} \right) \Big|_{\theta=\mu^\theta} \\ &= -\frac{1}{2} tr \left[ \Sigma^X \begin{bmatrix} \tilde{C}_{\theta^i}^T \tilde{\Pi}^z \tilde{C} - \tilde{A}_{\theta^i}^T \tilde{\Pi}^w (D^x - \tilde{A}) & \tilde{A}_{\theta^i}^T \tilde{\Pi}^w \tilde{B} \\ -\tilde{B}_{\theta^i}^T \tilde{\Pi}^w (D^x - \tilde{A}) & \tilde{B}_{\theta^i}^T \tilde{\Pi}^w \tilde{B} \end{bmatrix} \right] \Big|_{\theta=\mu^\theta}, \end{aligned} \tag{72}$$

where the gradient of the mean field with respect to  $\Theta$  is given by  $W_{\Theta}^X = \left[ W_{\Theta^1}^X \quad W_{\Theta^2}^X \quad \dots \quad W_{\Theta^i}^X \right]^T$ . Similarly, the elements of the curvature matrix of the mean field term with respect to  $\Theta$  is given by:

$$W_{\Theta^i\Theta^j}^X = -\frac{1}{2} tr \left( \Sigma^X \tilde{\epsilon}_{X\Theta^i}^T \tilde{\Pi} \tilde{\epsilon}_{X\Theta^j} \right) \Big|_{\theta=\mu^\theta} = -\frac{1}{2} tr \left[ \Sigma^X \begin{bmatrix} \tilde{C}_{\theta^i}^T \tilde{\Pi}^z \tilde{C}_{\theta^j} + \tilde{A}_{\theta^i}^T \tilde{\Pi}^w \tilde{A}_{\theta^j} & \tilde{A}_{\theta^i}^T \tilde{\Pi}^w \tilde{B}_{\theta^j} \\ \tilde{B}_{\theta^i}^T \tilde{\Pi}^w \tilde{A}_{\theta^j} & \tilde{B}_{\theta^i}^T \tilde{\Pi}^w \tilde{B}_{\theta^j} \end{bmatrix} \right] \Big|_{\theta=\mu^\theta}. \tag{73}$$

The gradient and curvature of the mean field term of  $X$  with respect to  $\Lambda$  can be evaluated as:

$$\begin{aligned} W_\Lambda^X &= -\frac{1}{2} tr \left( \Sigma^X \tilde{\epsilon}_X^T \tilde{\Pi}_\lambda \tilde{\epsilon}_X \right) \Big|_{\theta=\mu^\theta} = -\frac{1}{2} tr \left( \tilde{\Pi}_\lambda \tilde{\epsilon}_X \Sigma^X \tilde{\epsilon}_X^T \right) \Big|_{\theta=\mu^\theta} \\ &= -\frac{1}{2} \left[ \begin{array}{c} tr \left( \tilde{\Pi}^z \tilde{C} \Sigma^X \tilde{C}^T \right) \\ tr \left( \tilde{\Pi}^w \left[ (D^x - \tilde{A}), -\tilde{B} \right] \Sigma^X \left[ (D^x - \tilde{A})^T \right] \right) \end{array} \right] \Big|_{\theta=\mu^\theta} \\ W_{\Lambda\Lambda}^X &= -\frac{1}{2} tr \left( \tilde{\Pi}_{\lambda\lambda} \tilde{\epsilon}_X \Sigma^X \tilde{\epsilon}_X^T \right) \Big|_{\theta=\mu^\theta} = 0, \end{aligned} \tag{74}$$

where  $\Sigma^x$  is a component of  $\Sigma^X = \left[ \begin{array}{cc} \Sigma^x & \Sigma^{x\theta} \\ \Sigma^{x\theta} & \Sigma^\theta \end{array} \right]$ . Here the curvature  $W_{\Lambda\Lambda}^X$  vanishes due to the assumption that  $\tilde{\Pi}_{\lambda\lambda} = 0$ .

### 11.3. Gradients of Mean Field Terms Along Parameters

The mean-field term of the parameters  $\Theta$  can be expressed as

$$W^\Theta = \frac{1}{2} \text{tr} [\Sigma^\theta U(y, \mu^\theta)_{\theta\theta}] = \frac{1}{2} \text{tr} [\Sigma^\theta (-P^\theta - \tilde{\epsilon}_\theta^T \tilde{\Gamma} \tilde{\epsilon}_\theta)] \Big|_{\theta=\mu^\theta} \tag{75}$$

Differentiating Equation (75) with  $X$  and substituting Equation (61) in it yields the gradient as:

$$W_X^\Theta = -\frac{1}{2} \left[ \begin{array}{c} \text{tr} \left( \Sigma^\theta \left( N_{X1}^T \tilde{\Gamma}^z N + M_{X1}^T \tilde{\Gamma}^w M \right) \right) \\ \text{tr} \left( \Sigma^\theta \left( N_{X2}^T \tilde{\Gamma}^z N + M_{X2}^T \tilde{\Gamma}^w M \right) \right) \\ \dots \end{array} \right] \Big|_{\theta=\mu^\theta} \tag{76}$$

and the elements of the curvature matrix as:

$$W_{X^i X^j}^\Theta = -\frac{1}{2} \text{tr} \left( \Sigma^\theta \left( N_{X^i}^T \tilde{\Gamma}^z N_{X^j} + M_{X^i}^T \tilde{\Gamma}^w M_{X^j}^T \right) \right) \Big|_{\theta=\mu^\theta} \tag{77}$$

Differentiating Equation (75) with respect to  $\lambda$  yields the gradient and curvature as:

$$\begin{aligned} W_\Lambda^\Theta &= -\frac{1}{2} \text{tr} (\Sigma^\theta \tilde{\epsilon}_\theta^T \tilde{\Gamma} \tilde{\Lambda} \tilde{\epsilon}_\theta) \Big|_{\theta=\mu^\theta} = -\frac{1}{2} \text{tr} (\tilde{\Gamma} \tilde{\Lambda} \tilde{\epsilon}_\theta \Sigma^\theta \tilde{\epsilon}_\theta^T) \Big|_{\theta=\mu^\theta}, \\ &= -\frac{1}{2} \left[ \frac{\text{tr} (\tilde{\Gamma} \tilde{\Lambda}^z \tilde{\epsilon}_\theta \Sigma^\theta \tilde{\epsilon}_\theta^T)}{\text{tr} (\tilde{\Gamma} \tilde{\Lambda}^w \tilde{\epsilon}_\theta \Sigma^\theta \tilde{\epsilon}_\theta^T)} \right] \Big|_{\theta=\mu^\theta} = -\frac{1}{2} \left[ \frac{\text{tr} (\tilde{\Gamma}^z N \Sigma^\theta N^T)}{\text{tr} (\tilde{\Gamma}^w M \Sigma^\theta M^T)} \right] \Big|_{\theta=\mu^\theta} \\ W_{\Lambda\Lambda}^\Theta &= -\frac{1}{2} \text{tr} (\tilde{\Gamma} \tilde{\Lambda} \tilde{\Lambda} \tilde{\epsilon}_\theta \Sigma^\theta \tilde{\epsilon}_\theta^T) \Big|_{\theta=\mu^\theta} = 0. \end{aligned} \tag{78}$$

Here,  $W_{\Lambda\Lambda}^\Theta$  vanishes due to the assumption that  $\tilde{\Gamma} \tilde{\Lambda} \tilde{\Lambda} = 0$ .

## 12. The Complete DEM Algorithm

By combining the gradients found from Sections 9, 10, and 11 with the Algorithm 1, we can finalize the full DEM algorithm so that it can iteratively compute the estimates and the associated precisions from data.

### 12.1. DEM Estimates

The main equations that are required to perform the update rules of DEM given in Equation (48) can be summarized as:

$$\begin{aligned} \dot{X} &= DX + k^X (-E_X + W_X^\Theta), \quad J^X = D + k^X (-E_{XX} + W_{XX}^\Theta) \\ \frac{\partial \Theta}{\partial a} &= k^\Theta \left[ -P^\theta \epsilon^\theta \Big|_{\theta=\mu^\theta} + \sum_t \left( -E_{\Theta} + W_{\Theta}^X \right) \right], \quad J^\Theta = k^\Theta \left[ -P^\theta + \sum_t \left( -E_{\Theta\Theta} + W_{\Theta\Theta}^X \right) \right] \\ \frac{\partial \Lambda}{\partial b} &= k^\Lambda \left[ -P^\lambda \epsilon^\lambda \Big|_{\theta=\mu^\theta} + \sum_t \left( -E_\Lambda + W_\Lambda^X + W_\Lambda^\Theta + G_\Lambda \right) \right], \quad J^\Lambda = k^\Lambda (-P^\lambda + n^t G_{\Lambda\Lambda}) \end{aligned} \tag{79}$$

where  $E_X, E_{XX}, E_\Theta, E_{\Theta\Theta}, E_\Lambda, W_X^\Theta, W_{XX}^\Theta, W_\Theta^X, W_{\Theta\Theta}^X, W_\Lambda^X, W_\Lambda^\Theta, G_\Lambda$ , and  $G_{\Lambda\Lambda}$  are given by Equations (59), (60), (62), (63), (64), (76), (77), (72), (73), (74), (78), (55), and (57), respectively. The hyperparameter update rule can be further simplified to reduce the computational complexity as:

$$\frac{\partial \Lambda}{\partial b} = -k^\Lambda P^\lambda \epsilon^\lambda \Big|_{\theta=\mu^\theta} + \frac{k^\Lambda n^t}{2} \left[ \frac{n^{\tilde{\Gamma}^z}}{n^{\tilde{\Gamma}^w}} \right] - \frac{k^\Lambda}{2} \left[ \frac{\text{tr} (\tilde{\Gamma}^z A_3)}{\text{tr} (\tilde{\Gamma}^w B_3)} \right] \Big|_{\theta=\mu^\theta} \tag{80}$$

where

$$A_3 = \sum_t ((\tilde{y} - \tilde{C}\tilde{x})(\tilde{y} - \tilde{C}\tilde{x})^T + N\Sigma^\theta N^T + \tilde{C}\Sigma^{\tilde{x}\tilde{x}}\tilde{C}^T)$$

$$B_3 = \sum_t ((D^x\tilde{x} - \tilde{A}\tilde{x} - \tilde{B}\tilde{u})(D^x\tilde{x} - \tilde{A}\tilde{x} - \tilde{B}\tilde{u})^T + M\Sigma^\theta M^T + [(D^x - \tilde{A}), -\tilde{B}] \Sigma^X \begin{bmatrix} (D^x - \tilde{A})^T \\ -\tilde{B}^T \end{bmatrix}).$$

Substituting Equation (57) to the expression for  $J^\Lambda$  in Equation (79) yields:

$$J^\Lambda = -k^\Lambda(P^\lambda + \frac{n^t}{2} \begin{bmatrix} n^{\tilde{\Gamma}^z} & O \\ O & n^{\tilde{\Gamma}^w} \end{bmatrix}), \tag{81}$$

which is independent of  $\Lambda$ . This reduces the algorithm’s computational complexity, as  $J^\Lambda$  can now be pre-computed.

### 12.2. Precision of Estimates

This section simplifies the precision for DEM’s estimates for an LTI system. The confidence in the estimate of (generalized) states and inputs can be simplified using Equations (51) and (60) as:

$$\Pi^{\tilde{x},\tilde{v}} = E_{XX} = \begin{bmatrix} \tilde{C}^T \tilde{\Gamma}^z \tilde{C} + (D^x - \tilde{A})^T \tilde{\Gamma}^w (D^x - \tilde{A}) & -(D^x - \tilde{A})^T \tilde{\Gamma}^w \tilde{B} \\ -\tilde{B}^T \tilde{\Gamma}^w (D^x - \tilde{A}) & P^{\tilde{v}} + \tilde{B}^T \tilde{\Gamma}^w \tilde{B} \end{bmatrix} \Big|_{\theta=\mu^\theta}. \tag{82}$$

From Equation (82), the precisions for state and input estimation are  $\Pi^{\tilde{x}\tilde{x}} = \tilde{C}^T \tilde{\Gamma}^z \tilde{C} + (D^x - \tilde{A})^T \tilde{\Gamma}^w (D^x - \tilde{A})$  and  $\Pi^{\tilde{v}\tilde{v}} = P^{\tilde{v}} + \tilde{B}^T \tilde{\Gamma}^w \tilde{B}$ , respectively. The cross-correlation between the (generalized) states and inputs are given by  $-\tilde{B}^T \tilde{\Gamma}^w (D^x - \tilde{A})$ . Since  $\Pi^{\tilde{x},\tilde{v}}$  is independent of  $X$ , it can be updated outside the D step.

Combining the results of Equations (51) and (63) yields the precision of parameter estimates  $\Pi^\theta$ , which is independent of  $\Theta$ , as:

$$\Pi^\theta = P^\theta + \sum_t E_{\Theta\Theta} = P^\theta + \sum_t [N^T \tilde{\Gamma}^z N + M^T \tilde{\Gamma}^w M] \Big|_{\theta=\mu^\theta}. \tag{83}$$

From Equations (51), (64), and (57), the precision of hyperparameter estimation is:

$$\Pi^\lambda = P^\lambda + \sum_t (E_{\Lambda\Lambda} - G_{\Lambda\Lambda}) = P^\lambda + \frac{n^t}{2} \text{diag}(n^{\tilde{\Gamma}^z}, n^{\tilde{\Gamma}^w}), \tag{84}$$

which is a constant and hence is never updated in the algorithm. In conclusion, the estimation using Equation (79), along with the precision of these estimates given by Equations (82), (83), and (84) completely define the DEM algorithm for an LTI system with colored noises. The complete DEM algorithm is given in Algorithm 2.

**Algorithm 2:** Dynamics Expectation Maximization

---

**Data:** Time series data of output  $\mathbf{y}$   
**Result:**  $(X, \Theta, \Lambda), (\Pi^{\tilde{x}, \tilde{v}}, \Pi^\theta, \Pi^\lambda)$   
 $a = 0$ ;  
 Evaluate  $\tilde{\mathbf{y}}$  from  $\mathbf{y}$  ▷ Equation (9)  
 Precompute gradients  $\tilde{A}_{\theta^i}, \tilde{B}_{\theta^i}, \tilde{C}_{\theta^i}, N_{X^i}, M_{X^i}$  ;  
 Set all priors:  $\eta^{\tilde{v}}, P^{\tilde{v}}, \eta^\theta, P^\theta, \eta^\lambda, P^\lambda$  ;  
 Initialize  $\Theta, \Lambda, \Pi^X, \Pi^\theta, \Pi^\lambda$  with their priors ;  
 $G_\Lambda, G_{\Lambda\Lambda} \leftarrow$  Equation (55) and (57) ;  
 $\Pi^\lambda \leftarrow P^\lambda - n^t G_{\Lambda\Lambda}$  ;  
 $J^\Lambda \leftarrow k^\Lambda (-P^\lambda + n^t G_{\Lambda\Lambda})$  ;  
**while**  $\Theta$  not converged **do** ▷ E step loop  
 Set  $E_\Theta, E_{\Theta\Theta}, E_1$  to zero ;  
 $W_{XX}^\Theta \leftarrow \text{find}W_{XX}^\Theta(N_X, M_X, \tilde{\Pi}, \Pi^\theta)$  ; ▷ Equation (77)  
 $E_{XX} \leftarrow \text{find}E_{XX}(A, B, C, \tilde{\Pi})$  ; ▷ Equation (60)  
 $J^X \leftarrow D + k^X (-E_{XX} + W_{XX}^\Theta)$  ;  
**for**  $t = 0:\Delta t:T$  **do** ▷ D step loop  
 // Perform D step ;  
 $W_X^\Theta \leftarrow$  Equation (76) ;  
 $E_X \leftarrow$  Equation (59) ;  
 $\frac{\partial X}{\partial t} \leftarrow DX + k^X (-E_X + W_X^\Theta)$  ;  
 $X_{t+\Delta t} \leftarrow X_t + (e^{J^X \Delta t} - I)(J^X)^{-1} \frac{\partial X}{\partial t}$  ;  
 // Accumulate terms for E step ;  
 $M, N \leftarrow$  Equation (5) ; ▷ use new X  
 $A_2, B_2 \leftarrow$  Equation (62) ;  
 $E_\Theta \leftarrow E_\Theta + A_2 \Theta - B_2 \left[ \frac{\tilde{\mathbf{y}}}{\mu^{\tilde{x}}} \right]$  ;  
 $E_{\Theta\Theta} \leftarrow E_{\Theta\Theta} + A_2$  ;  
 // Accumulate terms for M step ;  
 $\tilde{\epsilon}, \tilde{\epsilon}_X, \tilde{\epsilon}_\Theta \leftarrow$  Equation (58), (61) ;  
 $E_1 \leftarrow E_1 + \tilde{\epsilon} \tilde{\epsilon}^T + \tilde{\epsilon}_X \Sigma^X \tilde{\epsilon}_X^T + \tilde{\epsilon}_\Theta \Sigma^\theta \tilde{\epsilon}_\Theta^T$  ;  
**end**  
**while**  $\Lambda$  not converged **do** ▷ M step loop  
 $\frac{\partial \Lambda}{\partial b} \leftarrow k^\Lambda (-P^\lambda \epsilon^\lambda + G_\Lambda - \frac{1}{2} \left[ \frac{\text{tr}(\tilde{\Pi}_{\lambda z} E_1)}{\text{tr}(\tilde{\Pi}_{\lambda w} E_1)} \right])$  ;  
 $\Lambda \leftarrow \Lambda + (e^{J^\Lambda} - I)(J^\Lambda)^{-1} \frac{\partial \Lambda}{\partial b}$  ;  
 $\tilde{\Pi} \leftarrow$  Equation (52) ; ▷ update with new  $\Lambda$   
**end**  
 $\bar{F}_a \leftarrow \text{find}\bar{F}(X, \Pi^X, \Theta, \Pi^\theta, \Lambda, \Pi^\lambda)$  ; ▷ Equation (43)  
**if**  $\bar{F}_a > \bar{F}_{a-1}$  **then** ▷ update  $\Theta$  if  $\bar{F}$  increased  
 $W_{\Theta}^X, W_{\Theta\Theta}^X \leftarrow$  Equation (72), (73) ;  
 $\frac{\partial \Theta}{\partial a} \leftarrow k^\Theta [-P^\theta \epsilon^\Theta - E_\Theta + n^t W_{\Theta}^X]$  ;  
 $J^\Theta \leftarrow k^\Theta [-P^\theta - E_{\Theta\Theta} + n^t W_{\Theta\Theta}^X]$  ;  
 $\Theta \leftarrow \Theta + (e^{J^\Theta} - I)(J^\Theta)^{-1} \frac{\partial \Theta}{\partial a}$  ; ▷ E step  
 $A, B, C \leftarrow$  Equation (4) ; ▷ update with new  $\Theta$   
**end**  
 $a++$  ;  
 $\Pi^{\tilde{x}, \tilde{v}} = E_{XX}$  ; ▷ update generalized state precision  
 $\Pi^\theta = P^\theta + E_{\Theta\Theta}$  ; ▷ update parameter precision  
**end**

---



### 13. Translation into Simplified Mathematical Form

Although the pseudocode derived in the previous sections is sufficient to replicate the DEM algorithm for an LTI system with colored noise, it is not sufficient to analyze DEM using the standard control systems tools for stability checks, convergence, etc. Therefore, in this section we translate the algorithm into a simplified mathematical form that control engineers can easily analyze. The following subsections aim at converting the DEM updates into a coupled linear system.

#### 13.1. State and Input Estimation as a Linear Observer

This section deals with reformulating the D step of DEM for an LTI system as a (generalized) state and input observer. Substituting Equation (59) in Equation (79) with a learning rate of  $k^X = 1$  yields [2]:

$$\dot{X} = \begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{v}} \end{bmatrix} = (D - A_1)X - B_1 \begin{bmatrix} \tilde{y} \\ -\eta^{\hat{v}} \end{bmatrix} + W_X^{\ominus}. \tag{85}$$

We now aim to mathematically prove that the (generalized) state and input observer of DEM can be reduced into an augmented LTI system, for which an exact discretization can be performed. We proceed by simplifying the mean field terms in Equation (39) as:

$$\begin{aligned} W_X^{\ominus} &= -\frac{1}{2} \left[ \begin{array}{c} \text{tr}(\Sigma^{\theta} (N_{X1}^T \tilde{\Pi}^z N + M_{X1}^T \tilde{\Pi}^w M)) \\ \text{tr}(\Sigma^{\theta} (N_{X2}^T \tilde{\Pi}^z N + M_{X2}^T \tilde{\Pi}^w M)) \\ \dots \end{array} \right] \Big|_{\theta=\mu^{\theta}} \\ &= -\frac{1}{2} \left[ \begin{array}{c} \text{vec}(\Sigma^{\theta T})^T (I \otimes (N_{X1}^T \tilde{\Pi}^z) \text{vec}(N)) + \\ \text{vec}(\Sigma^{\theta T})^T (I \otimes (M_{X1}^T \tilde{\Pi}^w) \text{vec}(M)) \\ \dots \end{array} \right] \Big|_{\theta=\mu^{\theta}} \\ &= (Z_X^N \text{vec}(N) + Z_X^M \text{vec}(M)) \Big|_{\theta=\mu^{\theta}}, \end{aligned} \tag{86}$$

where,

$$Z_X^N = -\frac{1}{2} (I \otimes \text{vec}(\Sigma^{\theta T})^T) \begin{bmatrix} I \otimes (N_{X1}^T \tilde{\Pi}^z) \\ I \otimes (N_{X2}^T \tilde{\Pi}^z) \\ \dots \end{bmatrix}, \quad Z_X^M = -\frac{1}{2} (I \otimes \text{vec}(\Sigma^{\theta T})^T) \begin{bmatrix} I \otimes (M_{X1}^T \tilde{\Pi}^w) \\ I \otimes (M_{X2}^T \tilde{\Pi}^w) \\ \dots \end{bmatrix}. \tag{87}$$

Since  $M$  and  $N$  can be obtained from linear transformation of  $X$ ,  $\text{vec}(M)$  and  $\text{vec}(N)$  can be written as:

$$\text{vec}(M) = Z^M X \text{ and } \text{vec}(N) = Z^N X, \tag{88}$$

where  $Z^M$  and  $Z^N$  are matrices with elements 0 and 1. This leads to the mean field term being expressed as a linear transformation of  $X$ :

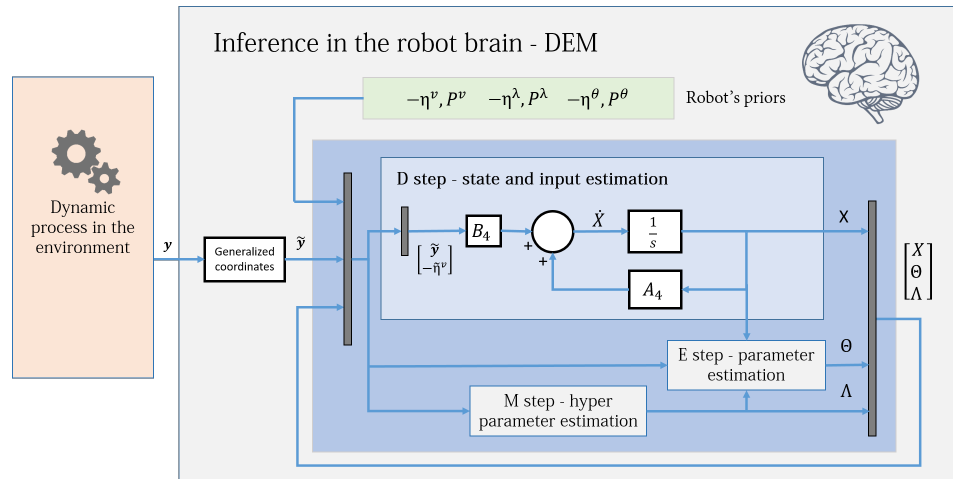
$$W_X^{\ominus} = (Z_X^N Z^N + Z_X^M Z^M) \Big|_{\theta=\mu^{\theta}} X. \tag{89}$$

Substituting Equation (89) into Equation (85) simplifies the observer as:

$$\begin{aligned} \dot{X} &= A_4 X + B_4 \begin{bmatrix} \tilde{y} \\ -\eta^{\hat{v}} \end{bmatrix}, \\ A_4 &= D - A_1 + (Z_X^N Z^N + Z_X^M Z^M) \Big|_{\theta=\mu^{\theta}} \text{ and } B_4 = -B_1. \end{aligned} \tag{90}$$

The (generalized) state and input observer given by Equation (90) is of the form of an augmented LTI system. Therefore, an exact discretization can be used to solve it without using the second order gradient  $J^X$  as given in Equation (48). This reduces the algorithm's computational complexity because  $E_{XX}$  and  $W_{XX}^{\ominus}$  for  $J^X$  calculation are no longer necessary. Figure 3 shows the simplified control diagram of the observer. The stability condition of

this observer (under known  $\theta$  and  $\lambda$ ) and its similarity with the Kalman Filter is discussed in our prior work [2]. To evaluate  $W_{\Theta}^X$ , one could either use Equation (89) or Equation (76). Equation (90) is derived mainly for simplification and exact discretization.



**Figure 3.** The DEM algorithm for an LTI system, with the D step simplified as an augmented LTI system given by Equation (90). The D-step block corresponds to the D-step loop in Algorithm 2 and operates at a different frequency from the E and M blocks.

### 13.2. Parameter Estimation—System Identification

This section aims to mathematically prove that the E step can be reduced to an augmented LTI system, for which an exact discretization can be performed. We proceed by first simplifying the parameter update equation given in Equation (79):

$$\frac{\partial \Theta}{\partial a} = -P^\theta (\Theta - \eta^\theta) \Big|_{\theta=\mu^\theta} + \sum_t (-E_\Theta + W_\Theta^X). \tag{91}$$

Grouping all  $W_{\Theta^i}^X$  using Equation (72) yields:

$$W_\Theta^X = -\frac{1}{2} \left( I \otimes \text{vec}(\Sigma^{XT})^T \right) \begin{bmatrix} I \otimes \tilde{\epsilon}_{X\theta^1}^T \tilde{\Gamma} \tilde{\Pi} \\ I \otimes \tilde{\epsilon}_{X\theta^2}^T \tilde{\Gamma} \tilde{\Pi} \\ \dots \end{bmatrix} \text{vec}(\tilde{\epsilon}_X) \Big|_{\theta=\mu^\theta} = Z_\theta^\epsilon \text{vec}(\tilde{\epsilon}_X) \Big|_{\theta=\mu^\theta} \tag{92}$$

where

$$\tilde{\epsilon}_{X\theta^i}^T \tilde{\Gamma} \tilde{\Pi} = - \begin{bmatrix} \tilde{C}_{\theta^i}^T \tilde{\Gamma} \tilde{\Pi}^z & O \\ O & O \end{bmatrix} \tilde{A}_{\theta^i}^T \tilde{\Gamma} \tilde{\Pi}^w, \text{vec}(\tilde{\epsilon}_X) = -\text{vec} \begin{bmatrix} \tilde{C} & O \\ O & \tilde{A} \\ \tilde{B} & \tilde{B} \end{bmatrix} + \text{vec} \begin{bmatrix} O & O \\ O & I \\ D^x & O \end{bmatrix} = -Z^\theta \theta + Z^I. \tag{93}$$

Here  $Z^\theta$  and  $Z^I$  are constant matrices with elements 0 and 1. Substituting  $\text{vec}(\tilde{\epsilon}_X)$  from Equation (93) in Equation (92) yields:

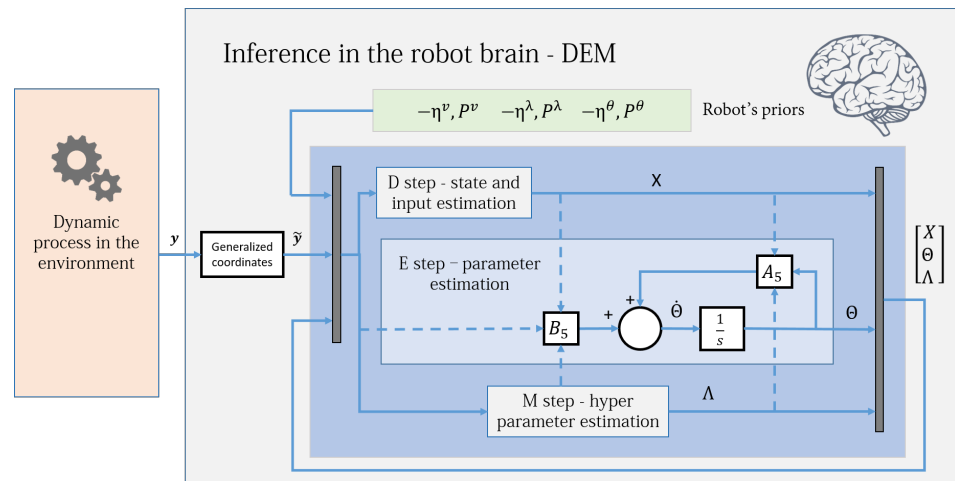
$$W_\Theta^X = -(Z_\theta^\epsilon Z^\theta) \Big|_{\theta=\mu^\theta} \Theta + (Z_\theta^\epsilon Z^I) \Big|_{\theta=\mu^\theta}. \tag{94}$$

Substituting Equations (94) and (62) in Equation (91), simplifies the parameter update equation to

$$\frac{\partial \Theta}{\partial a} = A_5 \Theta + B_5, \tag{95}$$

$$A_5 = -[P^\theta + n^t (Z_\theta^\epsilon Z^\theta) \Big|_{\theta=\mu^\theta} + \sum_t A_2], B_5 = P^\theta \eta^\theta + n^t (Z_\theta^\epsilon Z^I) \Big|_{\theta=\mu^\theta} + \sum_t B_2 \left[ \frac{\tilde{y}}{\mu^x} \right],$$

where  $A_2$  and  $B_2$  are given in Equation (62). Equation (95) is a linear differential equation in  $\Theta$  for which an exact discretization can be computed. For each  $\Theta$  update in Algorithm 2,  $A_2$  and  $B_2$  are also updated, consequently updating  $A_5$  and  $B_5$ . Therefore, Equation (95) is equivalent to a linear time-varying system. Figure 4 shows the simplified parameter estimation step of the robot brain. To evaluate  $W_{\Theta}^X$ , one could use either Equation (72) or Equation (94). Equation (94) was derived mainly for the exact discretization and for the convergence proof in Section 14.



**Figure 4.** The DEM algorithm for an LTI system, with the E step simplified as an augmented LTI system given by Equation (95). The E-step block corresponds to the E-step outer loop in Algorithm 2 and operates at a different frequency when compared to the D and M blocks. The dotted lines illustrate the flow of variables from other blocks and demonstrate the coupled nature of D, E, and M steps. This diagram is illustrative and should not be confused with a control diagram.

### 13.3. Hyperparameter Update

The update equation in Equation (80) can be simplified as:

$$\begin{aligned} \frac{\partial \Lambda}{\partial b} &= k^\Lambda \left[ -P^\Lambda e^\Lambda + \frac{n^t}{2} \begin{bmatrix} n^{\tilde{r}z} \\ n^{\tilde{r}w} \end{bmatrix} - \frac{1}{2} \begin{bmatrix} e^{\lambda z} \text{tr}((S \otimes \Omega^z) A_3) \\ e^{\lambda w} \text{tr}((S \otimes \Omega^w) B_3) \end{bmatrix} \right] \Big|_{\theta=\mu^\theta} \\ &= -\frac{k^\Lambda}{2} \begin{bmatrix} \text{tr}((S \otimes \Omega^z) A_3) & O \\ O & \text{tr}((S \otimes \Omega^w) B_3) \end{bmatrix} e^\Lambda - (k^\Lambda P^\Lambda) \Lambda + (k^\Lambda P^\Lambda \eta^\lambda + \frac{k^\Lambda n^t}{2} \begin{bmatrix} n^{\tilde{r}z} \\ n^{\tilde{r}w} \end{bmatrix}) \\ &= a_1 e^\Lambda + a_2 \Lambda + a_3, \end{aligned} \tag{96}$$

where  $a_1, a_2$ , and  $a_3$  are constants that are independent of  $\Lambda$ . Since Equation (96) is nonlinear in  $\Lambda$ , an approximate discretization like the conventional Gauss–Newton update scheme given in Equation (48) should be used for the M step. In summary, the D and E steps follow an exact discretization, whereas the M step follows an approximate discretization.

## 14. Convergence Proof for Parameter and Hyperparameter Estimation

In robotics, it is important that learning algorithms provide a stable solution, especially when robot safety during operation is a concern. Therefore, a proof of convergence for DEM is important for its widespread use in robotics as a learning algorithm. However, the DEM literature lacks any such mathematical proof of convergence for the estimator. Therefore, this section aims at providing one for the parameter and hyperparameter estimation step on LTI systems.

Since the update equation given by Equation (95) is a linear differential equation, proving that  $A_5 \prec O$  is sufficient to prove that  $\Theta$  converges to a stable solution. Substituting the expression for  $A_2$  from Equation (62) to the  $A_5$  in Equation (95), yields:

$$A_5 = -[P^\theta + n^t Z_\theta^\epsilon Z^\theta + \sum_t (N^T \tilde{\Pi}^z N + M^T \tilde{\Pi}^w M)] \Big|_{\theta=\mu^\theta} \tag{97}$$

Since the prior precision matrix can be chosen to be positive definite,  $P^\theta \succ O$ . It is straightforward to note from the expression for  $A_2$  in Equation (62) that  $\sum_t A_2 \succ O$ , because  $\tilde{\Pi}^z \succ O, \tilde{\Pi}^w \succ O \implies N^T \tilde{\Pi}^z N \succ O$ , and  $M^T \tilde{\Pi}^w M \succ O$ . Therefore, the proof of convergence is complete if we prove that  $Z_\theta^\epsilon Z^\theta \succ O$ . Simplifying the expressions for  $Z_\theta^\epsilon$  and  $Z^\theta$  from Equations (92) and (93), after some nontrivial linear algebra [41], yields:

$$Z_\theta^\epsilon Z^\theta = \frac{1}{2} \frac{\partial \tilde{\theta}^T}{\partial \theta} Z_1 Z_2 \frac{\partial \tilde{\theta}}{\partial \theta} \tag{98}$$

where

$$Z_1 = \text{diag}(\tilde{\Pi}^w \otimes I, \tilde{\Pi}^w \otimes I, \tilde{\Pi}^z \otimes I), \frac{\partial \tilde{\theta}}{\partial \theta} = \text{diag}(\text{vec} \tilde{A}_{\text{vec} A^T}^T, \text{vec} \tilde{B}_{\text{vec} B^T}^T, \text{vec} \tilde{C}_{\text{vec} C^T}^T),$$

$$Z_2 = \begin{bmatrix} I \otimes \Sigma^{\tilde{x}\tilde{x}^T} & I \otimes \Sigma^{\tilde{x}\tilde{v}^T} & O \\ I \otimes \Sigma^{\tilde{x}\tilde{v}^T} & I \otimes \Sigma^{\tilde{v}\tilde{v}^T} & O \\ O & O & I \otimes \Sigma^{\tilde{x}\tilde{x}^T} \end{bmatrix}.$$

It is straightforward from Equation (98) that  $Z_\theta^\epsilon Z^\theta \succ O$  because  $Z_1 \succ O$ , and  $Z_2 \succ O$ . Combining all the results from this section,  $P^\theta \succ O, Z_\theta^\epsilon Z^\theta \succ O$  and  $\sum_t A_2 \succ O, \implies A_5 \prec O$ . This completes the proof that the parameter estimation step of DEM converges for an LTI system. Similarly, from Equation (81),  $J^\Lambda \prec O$  proves the convergence of hyperparameter estimation step. For a detailed account of the linear algebra behind the proof of convergence, readers may refer to [41].

### 15. A Demonstrative Example

This section aims to provide the proof of concept for DEM through simulation for the estimation of an LTI system with colored noise. Since the algorithm can find an infinite number of solutions for a black box estimation of  $\tilde{x}, \tilde{v}, \theta$  and  $\lambda$  from  $\mathbf{y}$ , a black box estimation is not ideal as a demonstrative example. Therefore, we restrict this section to the joint estimation of  $x, A, B, \Pi^w$ , and  $\Pi^z$  from known  $\mathbf{y}$  and  $C$ .

#### 15.1. Generative Model

A stable LTI system of the form Equation (3) was selected, with randomly generated parameters  $\theta^i \in [-1, 1]$  having

$$A = \begin{bmatrix} 0.0484 & 0.7535 \\ -0.7617 & -0.2187 \end{bmatrix}, B = \begin{bmatrix} 0.3604 \\ 0.0776 \end{bmatrix}, C = \begin{bmatrix} 0.2265 & -0.4786 \\ 0.4066 & -0.2641 \\ 0.3871 & 0.3817 \\ -0.1630 & -0.9290 \end{bmatrix}.$$

A Gaussian bump input signal of  $v = e^{-0.25(t-12)^2}$  was centered around  $t = 12s$  and sampled at  $dt = 0.1s$  till  $T = 32s$  was used. The colored noise was generated with a smoothness value of  $\sigma = 0.5$  for the Gaussian kernel. The noise precisions were  $\Pi^w = e^8 I_2$  and  $\Pi^z = e^8 I_4$ , making  $\lambda^z = \lambda^w = 8$ . The embedding order of the generalized motion of states and inputs were  $p = 6$  and  $d = 2$ , respectively.

#### 15.2. Priors for Estimation

As discussed in Section 5.2, three prior distributions are necessary for the algorithm. Since the inputs are known, the input prior  $\eta^v$  is initialized with the known

input  $v$ , and a tight prior precision of  $P^v = e^{32}I_1$  is used to restrict any changes in  $v$ . Similarly, since the parameter  $C$  is known, the corresponding prior parameters in  $\eta^\theta$  are initialized with  $C$ , with tight priors of  $P^{\theta^i} = e^{32}$ . The prior parameters  $\eta^{\theta^i}$  for the unknown  $A$  and  $B$  matrices are randomly sampled from the range of  $[-2,2]$ , and a low prior precision of  $P^{\theta^i} = e^6$  is used to encourage exploratory behavior. In summary,  $\eta^\theta = [\text{vec}(\text{rand}(2,2)^T)^T \text{vec}(\text{rand}(2,1)^T)^T \text{vec}(C^T)^T]^T$  and  $P^\theta = \text{diag}(e^6I_4, e^6I_2, e^{32}I_8)$ . Since the hyperparameters are unknown, their priors were set to zero  $\lambda = [0 \ 0]^T$ , with a prior precision of  $P^\lambda = e^3I_2$  to encourage exploration.

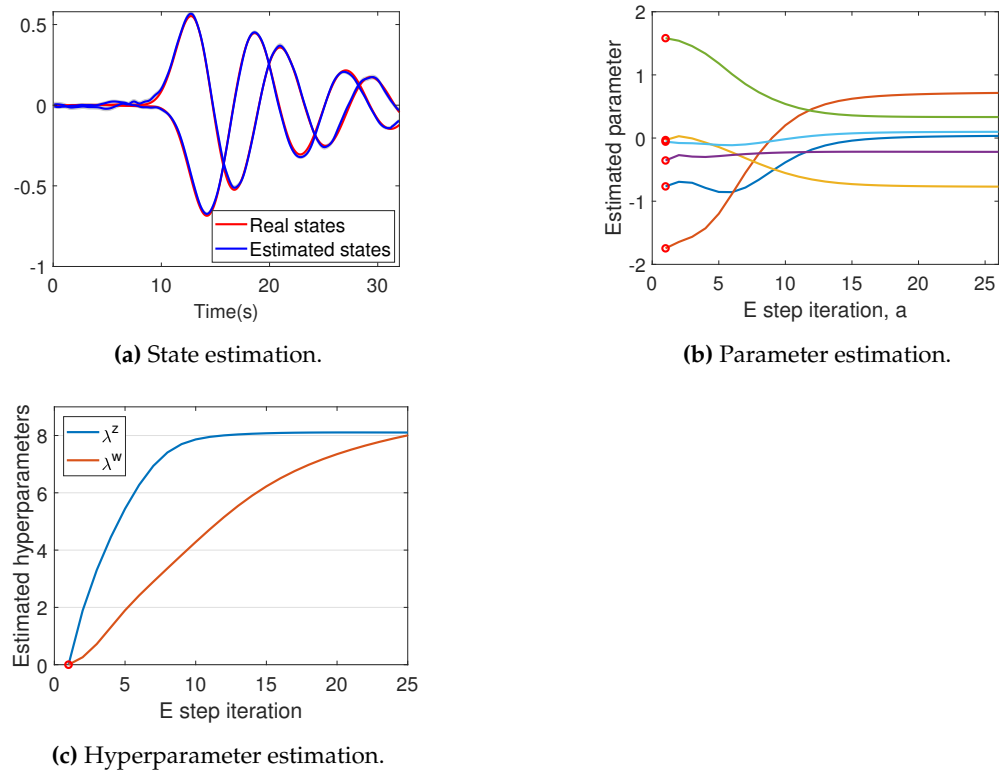
### 15.3. Results of Estimation

The data  $y$  generated from the system in Section 15.1 was used to run the DEM algorithm given in Algorithm 2. Figure 5a demonstrates the successful state estimation of the algorithm. The results of parameter estimation ( $A$  and  $B$ ) are shown in Figure 5b. The updates began from randomly selected priors  $\eta^\theta$ , marked by red circles, to finally converge. Table 1 shows that the DEM's estimate of  $A$  and  $B$  are close to the real values.

**Table 1.** DEM's estimate of  $A$  and  $B$  converges to real value.

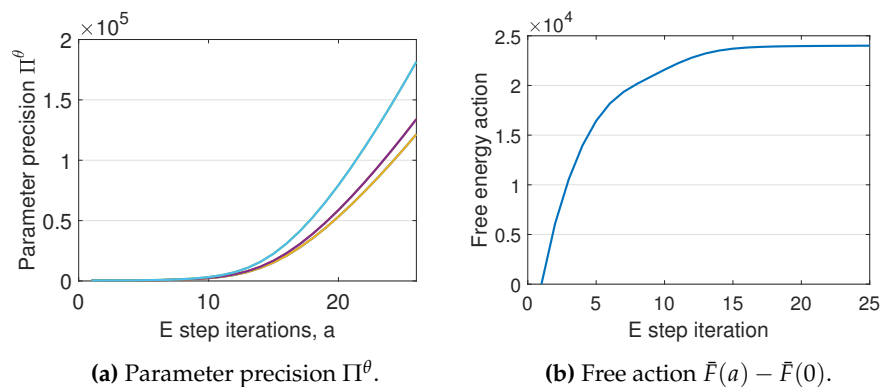
	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$
Real	0.048	0.753	−0.761	−0.218	0.360	0.077
Estimate	0.034	0.714	−0.769	−0.219	0.333	0.098

This confirms that the parameter estimation can converge close to the real parameters, even when  $\eta^\theta$  is randomly selected from the range  $\eta^\theta \in [-2,2]$  that is double the size of the real parameter range  $\theta^i \in [-1,1]$ . Figure 5c shows the successful hyperparameter convergence close to  $\lambda^z = \lambda^w = 8$ .



**Figure 5.** The results of DEM’s estimation process. (a) The estimated states in blue closely resembles the real states in red. (b) The parameter estimation starts from randomly selected  $\eta^\theta$ , marked by red circles and converges with each E step iteration  $a$ . (c) Both the hyperparameters start from  $\eta^\lambda = 0$ , and converge close to the correct value of 8.

DEM’s confidence on its estimates increase with the E step iterations, as can be seen from Figure 6a, which demonstrates an increase in parameter precision  $\Pi^\theta$ . A similar trend can be observed for  $\Pi^X$ . However,  $\Pi^\lambda$  remains a constant during the entire algorithm, as proved in Section 12.2. The key idea behind DEM’s inference is the maximization of free energy objectives. Read together, Figure 5 and 6 demonstrates that DEM successfully estimates  $\tilde{x}$ ,  $\theta$  and  $\lambda$ , with increasing confidence on its estimates as the estimation proceeds by maximizing  $\bar{F}$  from Equation (40). In summary, DEM can be used for the joint estimation of states, parameters and hyperparameters of an LTI system, subjected to colored noise.



**Figure 6.** Maximization of  $\bar{F}$  improves the confidence on estimates.

### 16. Benchmarking

This section deals with benchmarking DEM against the state-of-the-art parameter estimation methods such as Expectation Maximization (EM), Subspace method (SS), and Prediction Error Minimization (PEM), for black-box estimation (fully unknown  $x$ ,  $\theta$  and  $\lambda$ ).

### 16.1. Evaluation Metric for Parameter Estimation

For the black box identification, with completely unknown  $x$ ,  $\theta$ , and  $\lambda$ , there are infinite solutions with accurate input–output mapping. However, for LTI systems, there exists a unique transformation for identical systems. We use the companion canonical form to check the validity of parameter estimation by transforming both the real and the estimated parameters into their companion canonical form and then using the (square of) Euclidean distance between them as the sum of squared error (SSE) in parameter estimation. This evaluation metric will be used for parameter estimation in the next section.

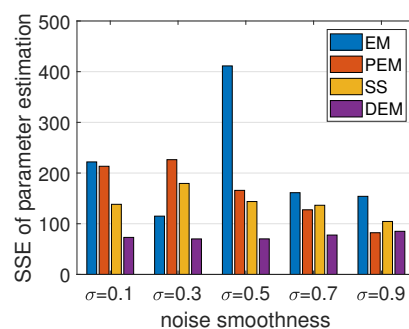
### 16.2. Simulation Setup

A total of 500 ( $5 \times 100$ ) different randomly generated stable systems were used with five different noise smoothness values for parameter estimation. All systems were selected with same number of parameters  $n^\theta = 14$  ( $n = 2, m = 4$  and  $r = 1$ ), with each  $\theta^i \in [-1, 1]$ , while ensuring that  $A$  matrix is stable. All the noises were generated with the precision of  $e^6$  ( $\Pi^w = e^6 I_{2 \times 2}, \Pi^z = e^6 I_{4 \times 4}$ ), with the embedding orders of states and inputs as  $p = 6$  and  $d = 2$ . A Gaussian bump of  $v = e^{-.25*(t-12)^2}$  was used as the input signal with  $dt = 0.5s$  and  $T = 32s$ . The prior parameter was randomly initialized such that all  $\eta^{\theta^i} \in [-2, 2]$  with a tight prior precision of  $P^\theta = e^4 I_{14 \times 14}$ . Both the hyperparameter priors  $\eta^{\lambda^i}$  were set to zero, with a prior precision of  $P^\lambda = e^{-4} I_{2 \times 2}$ .

The System Identification toolbox from MATLAB was used for SS (*n4sid()*) and PEM methods. The solution of SS was used to initialize PEM. An implementation of EM algorithm for state space models was written in MATLAB based on [45]. *n4sid()* is inherently designed to handle colored noise, whereas the implemented EM algorithm is not. The code for the DEM algorithm will be openly available at: [https://github.com/ajitham123/DEM\\_LTI](https://github.com/ajitham123/DEM_LTI).

### 16.3. Results

The results shown in Figure 7 demonstrate the superior performance of DEM in comparison with EM, PEM, and SS, with minimum SSE during parameter estimation across different noise smoothness. Additionally, EM and PEM exploded occasionally ( $< 5\%$  times), resulting in outliers in SSE, which were removed for better visualization. DEM demonstrated a consistent performance without generating any such outliers or exploding solutions, which could be explained by DEM's convergence guarantees for parameter estimation under colored noise [41], as proved in Section 14. In summary, DEM is a competitive parameter estimator for LTI systems with colored noise.



**Figure 7.** The sum of all SSE of  $\Theta$  for 100 random systems each, for 5 different noise smoothnesses. DEM outperforms EM, PEM, and SS with minimum SSE for parameter estimation under colored noise.

## 17. Discussion

The quest for a brain-inspired learning algorithm for robots has culminated in the free energy principle that postulates biological brain's perception as an optimization over its free energy objectives. FEP is of prime importance to robotics because of the use of generalized coordinates that enables it to gracefully handle colored noises. Colored noises

appear in real robotics systems through the unmodeled dynamics and the non-linearity errors in the model, thereby providing an advantage for DEM during estimation when compared to other estimators. An example could be the unmodeled wind disturbances acting on an unmanned aerial vehicle while in flight, or the non linearity errors in the dynamic model of a robotic manipulator arm involved in a pick and place operation. The scope of this work spans across the blind system identification of such linear dynamic systems with colored noise.

The fundamental difference between this work and the prior work is in the reformulation of DEM for an LTI system. While DEM from computational neuroscience focuses on emulating the biological brain's perception through the hierarchical abstraction of a number of non-linear dynamic systems that interact with each other, our work focuses on reducing this method into an algorithm for the system identification of an LTI system with colored noise, which is a well-known problem in robotics. This reformulation enables the standard analysis for convergence, stability and unbiased estimation, which is an essential analysis in practical robotics. It also enables DEM to be compared with other existing estimation algorithms in a control systems domain. The widespread use of DEM in robotics necessitates these mathematical analyses, especially when concerning the stable and safe operation of robots in industry and during human–robot interaction.

An algorithm with proved convergence for estimation is preferred for safe robotic applications. Therefore, one of the main contribution of this work was the reduction of the estimation algorithm into a coupled augmented system to prove the convergence of parameter and hyperparameter estimation steps. This work also demonstrated the successful applicability of DEM for the estimation of a randomly selected LTI system. Furthermore, we showed through rigorous simulations on a wide range of randomly generated LTI systems that DEM is a competitive algorithm for system identification under colored noise, thereby widening the scope of DEM to a large number of LTI systems in robotics.

One of the main drawbacks of the algorithm is its higher computational complexity when compared to the estimation algorithms that do not keep track of the trajectory of states. Therefore, future work can focus on the online estimation using DEM with reduced computational load. Future work can also focus on extending this algorithm for linear time varying systems to deal with robots with changing system parameters while in operation—a delivery drone dropping deliveries in mid-flight, for example. From a practical robotics point of view, DEM's parameter estimation module can be directly applied to a wide range of robots such as quadrotors, robotic arms, wheeled robots, etc. for black-box system identification, the input estimation module can be employed for fault-detection systems, and the hyperparameter estimation module can be used for online noise estimation for robust control. DEM can also be extended with a control loop for active inference to perform simultaneous perception and action on robots. This would result in the development of cognitive robots that can learn the generative model in the environment by interacting with it and actively seeking new information (active learning) for uncertainty resolution. This would influence multiple domains in robotics such as human–robot interaction for task learning, swarm robotics for collective learning and distributed control, informative path planning of aerial robots for environment monitoring, etc. The development of such brain-inspired autonomous agents sits at the core of cognitive robotics research. In summary, DEM has a huge potential to be the bioinspired learning algorithm for future robots.

## 18. Conclusion

The free energy principle from neuroscience has a great potential to be one of the most prominent frameworks for learning and control for the autonomous systems in future. Therefore, this paper converted the FEP-based inference scheme called DEM into a joint state, input, parameter, and hyperparameter estimation algorithm for LTI systems with colored noise. We derived the mathematical framework of DEM for LTI systems to prove that the resulting estimator is a combination of linear estimators that are coupled. We



provided the proof of convergence for the estimation steps. Through rigorous simulations on randomly generated linear systems with colored noise at varying smoothness levels, we demonstrated that the DEM algorithm outperforms EM, PEM, and SS methods for parameter estimation with minimal estimation error. In light of the potential for DEM to solve the parameter estimation problem, the future research will aim at applying DEM to a quadcopter flying in wind.

**Author Contributions:** A.A.M.: Conceptualization, methodology, software, validation, formal analysis, writing—original draft preparation, visualization. M.W.: Conceptualization, writing—review and editing, supervision. All authors have read and agreed to the published version of the manuscript

**Funding:** This research received no external funding.

**Data Availability Statement:** The MATLAB code for the DEM algorithm will be openly available at: [https://github.com/ajitham123/DEM\\_LTI](https://github.com/ajitham123/DEM_LTI)

**Acknowledgments:** The authors would like to thank Karl Friston for the thought-provoking discussions on the use of generalized coordinates within the FEP framework.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

LTI system	Linear time invariant system
DEM	Dynamic Expectation Maximization
FEP	Free energy principle
KF	Kalman Filter
KL divergence	Kullback–Leibler divergence

### References

1. Friston, K. The free-energy principle: a unified brain theory? *Nat. Rev. Neurosci.* **2010**, *11*, 127–138.
2. Anil Meera, A.; Wisse, M. Free Energy Principle Based State and Input Observer Design for Linear Systems with Colored Noise. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; pp. 5052–5058.
3. Bos, F.; Anil Meera, A.; Benders, D.; Wisse, M. Free Energy Principle for State and Input Estimation of a Quadcopter Flying in Wind. *arXiv* **2021**, arXiv:2109.12052.
4. Pezzato, C.; Ferrari, R.; Corbato, C.H. A novel adaptive controller for robot manipulators based on active inference. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2973–2980.
5. Pezzato, C.; Hernandez, C.; Wisse, M. Active Inference and Behavior Trees for Reactive Action Planning and Execution in Robotics. *arXiv* **2020**, arXiv:2011.09756.
6. Baioumy, M.; Duckworth, P.; Lacerda, B.; Hawes, N. Active inference for integrated state-estimation, control, and learning. *arXiv* **2020**, arXiv:2005.05894.
7. Oliver, G.; Lanillos, P.; Cheng, G. Active inference body perception and action for humanoid robots. *arXiv* **2019**, arXiv:1906.03022.
8. Çatal, O.; Verbelen, T.; Van de Maele, T.; Dhoedt, B.; Safron, A. Robot navigation as hierarchical active inference. *Neural Networks* **2021**, *142*, 192–204.
9. Lennart, L. *System Identification: Theory for the User*. PTR Prentice Hall: Upper Saddle River, NJ, USA, 1999; Volume 28.
10. Zhang, L. Q.; Cichocki, A.; Amari, S. Kalman filter and state-space approach to blind deconvolution. In Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop (Cat. No. 00TH8501), Sydney, Australia, 11–13 December 2000; pp. 425–434.
11. Han, R.; Bohn, C.; Bauer, G. Blind identification of state-space models in physical coordinates. *arXiv* **2021**, arXiv:2108.08498.
12. Abed-Meraim, K.; Qiu, W.; Hua, Y. Blind system identification. *Proceedings of the IEEE* **1997**, *85*, 1310–1322.
13. Zhang, Y. Unbiased identification of a class of multi-input single-output systems with correlated disturbances using bias compensation methods. *Math. Comput. Model.* **2011**, *53*, 1810–1819.
14. Liu, X.; Lu, J. Least squares based iterative identification for a class of multirate systems. *Automatica* **2010**, *46*, 549–554.
15. Zheng, W.X. On a least-squares-based algorithm for identification of stochastic linear systems. *IEEE Trans. Signal Process.* **1998**, *46*, 1631–1638.
16. Zhang, Y.; Cui, G. Bias compensation methods for stochastic systems with colored noise. *Appl. Math. Model.* **2011**, *35*, 1709–1716.
17. Cui, T.; Chen, F.; Ding, F.; Sheng, J. Combined estimation of the parameters and states for a multivariable state-space system in presence of colored noise. *Int. J. Adapt. Control. Signal Process.* **2020**, *34*, 590–613.
18. Friston, K.J.; Trujillo-Barreto, N.; Daunizeau, J. DEM: a variational treatment of dynamic systems. *Neuroimage* **2008**, *41*, 849–885.

19. van de Laar, T.; Özçelikkale, A.; Wymeersch, H. Application of the free energy principle to estimation and control. *arXiv* **2019**, arXiv:1910.09823.
20. Anil Meera, A.; Wisse, M. A Brain Inspired Learning Algorithm for the Perception of a Quadrotor in Wind. *arXiv* **2021**, arXiv:2109.11971.
21. Buckley, C.L.; Kim, C.S.; McGregor, S.; Seth, A.K. The free energy principle for action and perception: A mathematical review. *J. Math. Psychol.* **2017**, *81*, 55–79.
22. Hohwy, J. *Predict. Mind*; Oxford University Press: Oxford, UK, 2013.
23. Friston, K. The free-energy principle: a rough guide to the brain? *Trends Cogn. Sci.* **2009**, *13*, 293–301.
24. Friston, K.J.; Daunizeau, J.; Kilner, J.; Kiebel, S.J. Action and behavior: a free-energy formulation. *Biol. Cybern.* **2010**, *102*, 227–260.
25. Carhart-Harris, R.L.; Friston, K.J. The default-mode, ego-functions and free-energy: a neurobiological account of Freudian ideas. *Brain* **2010**, *133*, 1265–1283.
26. Friston, K.J.; Daunizeau, J.; Kiebel, S.J. Reinforcement learning or active inference? *PloS one* **2009**, *4*, e6421.
27. Friston, K. Hierarchical models in the brain. *PLoS Comput. Biol.* **2008**, *4*, e1000211.
28. Baltieri, M.; Buckley, C.L. PID control as a process of active inference with linear generative models. *Entropy* **2019**, *21*, 257.
29. Veissière, S.P.; Constant, A.; Ramstead, M.J.; Friston, K.J.; Kirmayer, L.J. Thinking through other minds: A variational approach to cognition and culture. *Behav. Brain Sci.* **2020**, *43*.
30. Kaufmann, R.; Gupta, P.; Taylor, J. An active inference model of collective intelligence. *arXiv* **2021**, arXiv:2104.01066
31. Miu, E.; Gulley, N.; Laland, K.N.; Rendell, L. Innovation and cumulative culture through tweaks and leaps in online programming contests. *Nat. Commun.* **2018**, *9*, 1–8.
32. Friston, K.; Kiebel, S. Predictive coding under the free-energy principle. *Philos. Trans. R. Soc. Biol. Sci.* **2009**, *364*, 1211–1221.
33. Parr, T.; Markovic, D.; Kiebel, S.J.; Friston, K.J. Neuronal message passing using Mean-field, Bethe, and Marginal approximations. *Sci. Rep.* **2019**, *9*, 1–18.
34. van de Laar, T.W.; de Vries, B. Simulating active inference processes by message passing. *Front. Robot. AI* **2019**, *6*, 20.
35. Friston, K.; FitzGerald, T.; Rigoli, F.; Schwartenbeck, P.; Pezzulo, G.; others. Active inference and learning. *Neurosci. Biobehav. Rev.* **2016**, *68*, 862–879.
36. Friston, K.J. Variational filtering. *NeuroImage* **2008**, *41*, 747–766.
37. Friston, K.; Stephan, K.; Li, B.; Daunizeau, J. Generalised filtering. *Math. Probl. Eng.* **2010**, *2010*.
38. Friston, K.; Mattout, J.; Trujillo-Barreto, N.; Ashburner, J.; Penny, W. Variational free energy and the Laplace approximation. *Neuroimage* **2007**, *34*, 220–234.
39. Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational inference: A review for statisticians. *J. Am. Stat. Assoc.* **2017**, *112*, 859–877.
40. Balaji, B.; Friston, K. Bayesian state estimation using generalized coordinates. In Proceedings of the Signal Processing, Sensor Fusion, and Target Recognition XX, Orlando, FL, USA, 5 May 2011; Volume 8050, p. 80501Y.
41. Anil Meera, A.; Wisse, M. On the convergence of DEM's linear parameter estimator. International Workshop on Active Inference, Springer, 2021. (Accepted)
42. Li, B.; Daunizeau, J.; Stephan, K.E.; Penny, W.; Hu, D.; Friston, K. Generalised filtering and stochastic DCM for fMRI. *Neuroimage* **2011**, *58*, 442–457.
43. Mader, W.; Linke, Y.; Mader, M.; Sommerlade, L.; Timmer, J.; Schelter, B. A numerically efficient implementation of the expectation maximization algorithm for state space models. *Appl. Math. Comput.* **2014**, *241*, 222–232.
44. Friston, K. A free energy principle for a particular physics. *arXiv* **2019**, arXiv:1906.10184.
45. Cara, F.J.; Juan, J.; Alarcón, E. Using the EM algorithm to estimate the state space model for OMAX. *In Pract.* **2014**, *1000*, 3.