Message Passing-based Bayesian Control of a Cart-Pole System

Sepideh Adamiat[1], Wouter M. Kouw[1], Bart van Erp[1,2], and Bert de Vries[1,2]

[1] Electrical Engineering Department, TU Eindhoven, Netherlands
[2] Lazy Dynamics B.V., Eindhoven, Netherlands
s.adamiat@tue.nl

Abstract. We describe a Bayesian controller for a cart-pole system, a well-known benchmark in control theory. The cart-pole system is characterized by its nonlinear and underactuated nature, and we further complicate the scenario by (1) assuming that the controller lacks knowledge of sensor noise variance, and (2) imposing bounds on the control signal. Traditional control algorithms often struggle to adapt to uncertainties and constraints. However, the Bayesian framework, particularly the active inference framework, smoothly accommodates these complexities. In the proposed controller, the entire computational process consists of online Bayesian inference. This process is streamlined through a toolbox for fast message passing-based inference in factor graphs. We describe the mechanics of message passing in factor graphs, addressing challenges such as non-linear factors, bounded control, and real-time parameter tracking. The primary objective of this paper is to demonstrate that, with the advancement of the active inference framework and the effectiveness of automated inference toolboxes, Bayesian control emerges as an appealing option for application engineers.

Keywords: Active inference · Bayesian control· Factor graphs· Message passing · NUV priors · Policy estimation

1 Introduction

The cart-pole problem (also known as the inverted pendulum) comprises a pole that is attached at one end to a movable cart with an associated goal of balancing the pole at the upright position by controlling the horizontal movements of the cart.This is a highly non-linear and underactuated system [3] that is widely used as a benchmark to illustrate the effectiveness of controllers [21]. Despite the challenges, a standard cart-pole system can be successfully controlled by classical control algorithms such as Model Predictive Control (MPC)[15].

In the real world, control systems inevitably have to deal with uncertainties that result from model inaccuracies and simplifications such as un-modeled external influences or sensor imperfections. In this paper, we add some complexity

[3] A cart-pole system has two degrees of freedom, namely the pendulum angle and the linear cart position, and only one actuator, the horizontal force on the cart.

to the cart-pole control task by assuming that (1) the variance of the sensor noise is unknown, and (2) the control signal is bounded.

Active inference is a mathematical framework designed for understanding biological agents, specifically the human brain [6]. This framework is grounded in a generative model that drives states, controls, and planning based on the principle of free energy minimizing [4, 19]. Recently, many studies have used active inference agents in robotic tasks [13, 3]. In this paper, due to the highly nonlinear and underactuated behavior of the cart pole system, we are interested in active inference agents with continuous control and state spaces [6, 12, 13].

Current (non-Bayesian) control algorithms have difficulties quantifying or processing such uncertainties appropriately [20]. In contrast, Bayesian control through probabilistic inference in a generative model provides a principled way to keep account of uncertainties and constraints in the system. Unfortunately, when trying to realize a Bayesian controller, exact inference quickly becomes computationally intractable, even for relatively simple models. Numerical solutions such as (Monte Carlo) sampling-based inference are often too computationally intensive or too slow for the application at hand.

This paper presents an approach based on casting both the control and parameter tracking problems as online inference tasks on a generative model of the system. To combat the computational issues surrounding probabilistic inference, we realize the inference tasks by message passing (MP) on a Forney-style factor graph (FFG) representation of the probabilistic model. Efficient probabilistic inference in the model is realized using automatable MP procedures that leverage the conditional independencies in the model. MP-based inference has a long history for efficient inference in signal processing and control systems [14].

To keep the control signals within physical bounds, we use a Normal-with-Unknown-variance (NUV) distribution as a prior distribution for the control signals. The NUV prior is a distribution that originated in the sparse Bayesian learning literature [22]. Recently, NUV priors were introduced as a sub-model to enforce domain constraints [10]. This sub-model has been successfully used in various MPC applications to impose constraints on the state trajectories [10], as well as in multi-agent trajectory estimation to prevent collisions [2].

In section 2, we introduce the cart-pole system formally and specify the control problem. The subsequent sections include our contributions:

- In section 3, we specify the controller, which comprises a probabilistic generative model for sensory observations from its environment. Crucially, the model can be used to predict both veridical and desired future observations.
- In section 4, we rehearse factor graphs and various message passing methods that can be used to automate the control-by-inference process.
- Finally, in section 5, we evaluate the proposed Bayesian control method for the cart-pole system in a simulated environment.

In short, the novelty of this paper lies not in the introduction of any specific technique, but rather this paper aims to demonstrate at a systems engineering level how to realize a complex Bayesian control system. We bring together various methods to show that Bayesian control with both uncertainties and constraints can be systematically realized through the specification of a biased generative model and a fully automatable inference process.

2 Problem Setting

2.1 The Cart-Pole System

In this paper, we simulate the cart-pole system as a state space model. In this model, the state variables are defined as a vector $z = [x, \theta, \dot{x}, \dot{\theta}]$, where $x \in \mathbb{R}$, $\theta \in \mathbb{R}$, $\dot{x} \in \mathbb{R}$, and $\dot{\theta} \in \mathbb{R}$ are the cart position, the pole angle, the cart velocity, and the angular velocity of the pole, respectively. Based on a Lagrangian mechanics approach, [21] derives the following equations of motion:

$$u = (m_c + m_p)\ddot{x} + m_p l\ddot{\theta}\cos\theta - m_p l\dot{\theta}^2 \sin\theta \tag{1a}$$

$$0 = m_p l\ddot{x}\cos\theta + m_p l^2\ddot{\theta} + m_p \mathrm{g} l \sin\theta\,, \tag{1b}$$

where m_c is the cart mass, m_p is the pendulum mass, g is the gravitational constant, and l is the pendulum length. These variables are assumed to be constant. Furthermore, $u \in \mathbb{R}$ is the horizontal force applied to the car, $\ddot{x} \in \mathbb{R}$ is the cart acceleration, and $\ddot{\theta} \in \mathbb{R}$ is the angular acceleration. For simulation purposes, based on Euler's method, we derive a discrete-time state space model with state variables $z_t = [x_t, \theta_t, v_t, \omega_t]$ where the x_t, θ_t, v_t and ω_t are the cart position, pole angle, cart velocity and the angular velocity of the pole at time t. This discrete-time state space model is then defined as

$$\underbrace{\begin{bmatrix} x_{t+1} \\ \theta_{t+1} \\ v_{t+1} \\ \omega_{t+1} \end{bmatrix}}_{z_{t+1}} = \overbrace{\underbrace{\begin{bmatrix} x_t \\ \theta_t \\ v_t \\ \omega_t \end{bmatrix}}_{z_t} + \begin{bmatrix} v_t \\ \omega_t \\ a_t \\ \alpha_t \end{bmatrix} \cdot \Delta t}^{g(z_t, u_t)} \tag{2}$$

where Δt is the interval between two time steps, and the a_t and α_t are the cart acceleration and the angular acceleration at time t. By using the equation of motion (1), the a_t and α_t at time t can be derived as

$$a_t = \frac{u_t + m_p \sin\theta_t (l\omega_t^2 + g\cos\theta_t)}{m_c + m_p \sin^2\theta_t} \tag{3a}$$

$$\alpha_t = \frac{\text{-}u_t \cos\theta_t - m_p l\omega_t^2 \cos\theta_t \sin\theta_t - (m_c + m_p)\mathrm{g}\sin\theta_t}{l(m_c + m_p \sin^2\theta_t)}. \tag{3b}$$

For more details, we refer to [21]. In the following, we also use the abbreviation $g(z_t, u_t)$ to indicate the right-hand side of (2).

The initial state of the cart-pole system is given by $z_0 = [0, -\pi, 0, 0]$, indicating that the pole is initially in a downward position and both the cart and the pole are at rest.

2.2 The Control Problem

We assume that the horizontal force signal u_t can be selected by a control agent. The agent interacts with its environment through a series of trials. The agent observes the cart-pole system's state through a sensor and executes actions u_t via an actuator that connects to the cart.

We assume that the agent partially observes its environment using a measurement matrix C with measurement noise v_t:

$$y_t = Cz_t + v_t\,, \tag{4}$$

with

$$C = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \end{bmatrix} \quad \text{and} \quad v_t \sim \mathcal{N}(0, R^*)\,, \tag{5}$$

where R^* is a fixed covariance matrix.

The goal of the agent is to guide the cart-pole system towards a desired goal state $z^* = [0, 0, 0, 0]$, where the pole is steady at the upright position. Given that direct control over the pole is not feasible (the controller can only apply a force to the cart), the control agent needs to *infer* a control sequence (policy) $u_1, u_2, u_3, \ldots$ to reach z^*.

We take a fully Bayesian inference approach to control [18]. In this framework, the controller is equipped with a probabilistic generative model for its observations. This model can be run forward in time to create predictions of future observations. As will be discussed in section 3.2, these predictions are biased toward desirable observations, and Bayes-optimal control signals can be obtained through online Bayesian inference.

3 The Probabilistic Controller Model

3.1 Specification of Generative (Veridical) Model

This subsection specifies the control agent for the simulated cart-pole system. The controller's model is a *generative* model, so we will specify a model with hidden dynamics that leads to probabilistic predictions of sensory inputs y_t.

We assume that the controller knows the dynamics of the cart-pole system, with uncertainty characterized by Gaussian process noise. Hence, the controller's state transition model is given by

$$p(s_{t+1} \mid s_t, u_t) = \mathcal{N}(s_{t+1} \mid g(s_t, u_t), Q)\,, \tag{6}$$

where s_t is controller's internal state, and Q is a fixed and known covariance matrix. In general, we run our simulation from an initial state $p(s_0) = \mathcal{N}(s_0 \,|\, z_0, Q)$. We also assume that the controller can predict its sensory observations with a "correct" observation model:

$$p(y_t \mid s_t, R_t) = \mathcal{N}(y_t \mid Cs_t, R_t)\,. \tag{7}$$

where C is defined as in (5) and R_t is a measurement noise covariance matrix at time step t. To complicate matters, we assume that the controller does not know the observation noise covariance matrix R_t, and therefore we will infer the appropriate value of R_t online, alongside inference for states s_t and controls u_t. The assumption is that the covariance matrix has a fixed unknown value, specified to be

$$p(R_t|R_{t-1}) = \delta(R_t - R_{t-1}) \tag{8a}$$

$$p(R_0) = \mathcal{W}^{-1}(R_0 \mid V_0, n_0), \tag{8b}$$

where $\mathcal{W}^{-1}$ is an inverse-Wishart distribution, which is a conjugate prior for the covariance matrix of a multivariate normal distribution.

To complete the generative model, we assume independent priors $p(u_t)$ over admissible actions at each time step. We will choose a Normal-with-Unknown Variance (NUV) prior, which effectively renders a "box constraint" of the form

$$p(u_t) \precapprox \exp\left\{-\gamma(|u_t - a| + |u_t - b|)\right\}, \tag{9}$$

where a and b are user-selectable control limits, γ is a parameter to control the softness of the constraint and $\precapprox$ means "is approximately proportional to". The exact form of the prior and subsequent inference procedure is discussed in section 4.5.

The set of equations (6), (7), (8) and (9) is called the agent's veridical generative model since it aims to predict the evolution of future sensory inputs according to the agent's knowledge about the environmental dynamics.

3.2 Specification of Target Model for Observations

The veridical model specification in section 3.1 can be used by the controller to predict future sensory inputs. In a Bayesian active inference framework, sensory targets are specified by extending the veridical model by *target* distributions [5]. These target distributions lead to goal-oriented behavior(such as guiding a cart-pole system to a target state) through Bayesian inference. We assume the following target distribution for observations:

$$p'(y_t) = \begin{cases} \mathcal{N}(y_t \mid Cz^*, 10^8 \cdot I) & \text{if } t \leq T \\ \mathcal{N}(y_t \mid Cz^*, 10^{-8} \cdot I) & \text{otherwise.} \end{cases} \tag{10}$$

The prime in $p'(\cdot)$ indicates that this distribution relates to desired or target beliefs, rather than veridical beliefs. Eq. (10) expresses that, for the first T time steps, the agent has essentially no preference for observations, but any time after T, the agent has a strong preference for receiving $y_t \approx Cz^*$.

The vague prior for $t \leq T$ allows the agent to infer actions that are most informative about the uncertainties in the model, such as the value of R_t. This is an explorative phase. For $t > T$, the tight target priors add an incentive to infer actions u_t that drive the cart-pole to its target state.

Taking the veridical and target beliefs together, the complete generative model for the controller can be represented as

$$\begin{aligned} p(y, s, u, R) &\propto p(s_0)p(R_0) \\ &\cdot \prod_{t>0} p'(y_t)p(y_t \mid s_t, R_t)p(s_t \mid s_{t-1}, u_t)p(u_t)p(R_t|R_{t-1}) \, . \end{aligned} \tag{11}$$

Note that, due to the extension with the target prior $p'(y_t)$, the controller holds a *biased* model of the future! When unrolling this model into the future, this model predicts desired future observations, in the context of given assumptions about how the world "really" works (as specified by the veridical model). This approach aligns with the active inference framework that is claimed to describe a biologically plausible approach to control in living systems [5].

4 Inference

4.1 Inference is the Only Ongoing Process

Since the controller's generative model is biased toward predicting target observations, the actual process to be executed by the controller is just continual inference over all latent variables as new data y_t keeps streaming in through its sensory channels. This online inference process will update beliefs over its internal states s_t, the latent control variables u_t, and the latent covariance matrix R_t.

For a complex non-linear system with some non-conjugate distribution pairings, such as this dynamic Cart-pole controller, it is not possible to derive closed-form analytical Bayesian inference solutions, and sampling-based inference methods are usually too slow for real-time systems. Therefore, in this paper, we automate the online inference process through efficient message-passing-based inference on a factor graph representation of the controller's model.

Fortunately, we do not need to derive all messages from scratch. The open-source Julia package RxInfer supports fast message passing-based inference for a large range of models [1]. Next, we rehearse factor graphs and message passing-based inference.

4.2 Forney-style Factor Graphs and the Sum-Product Rule

A Forney-style Factor Graph (FFG) is a graphical representation of a factorized probabilistic model [14]. In an FFG, edges represent random variables and nodes represent factors, which are functions that specify the relationships between the variables. An edge connects to a node if and only if the variable on that edge is an argument of the node's function. Figure 1 shows the FFG for the probabilistic model defined in (11). In an FFG, each edge can maximally connect to two nodes. If a variable is an argument in more than two factors, we introduce a

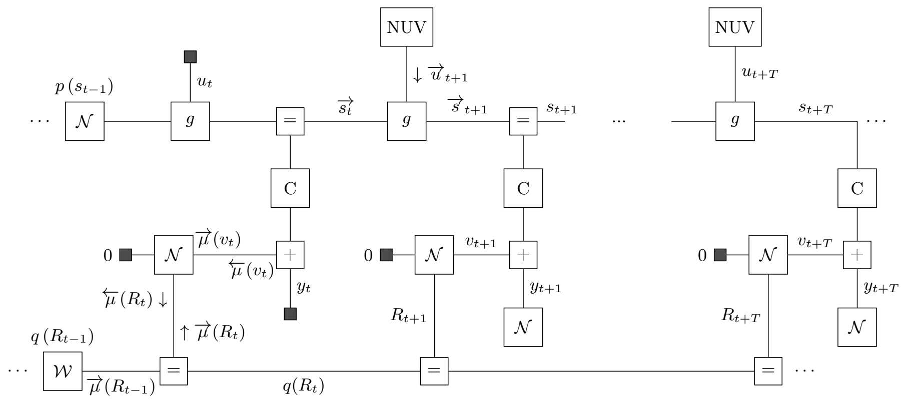


Fig. 1: A Forney-style factor graph representation of the probabilistic control model in (11). This snapshot illustrates the model at the time step t, with T future time horizon.

"branching" (also known as "equality") node that effectively copies the variable to an auxiliary variable with the same beliefs.

Aside from model visualization, FFGs support efficient message passing-based (MP) inference on the graph. MP-based inference is a highly efficient tool for performing probabilistic inference on sparsely connected generative models [14]. It scales well to large inference tasks and significantly speeds up Bayesian inference by effectively taking advantage of the distributive law ($ab+ac = a(b+c)$), which converts an (expensive) sum-of-products to a (cheaper) product-of-sums. We use $\overrightarrow{\mu}(\cdot)$ and $\overleftarrow{\mu}(\cdot)$ notations for the forward and backward messages respectively. Following the recipe above of moving factors over integrals (or summation signs), marginalization and Bayesian inference turn into a sequence of updating messages. These messages can be computed by the so-called sum-product rule [14].

In general, for any node $f(y, x_1, \ldots, x_n)$, the sum-product rule for an outgoing message over edge y is given by

$$\overrightarrow{\mu}(y) = \int \underbrace{\overrightarrow{\mu}(x_1)...\overrightarrow{\mu}(x_n)}_{\text{incoming messages}} \underbrace{f(y, x_1, ..., x_n)}_{\text{node function}} \mathrm{d}x_1...\mathrm{d}x_n \,. \tag{12}$$

Note that the MP algorithm minimizes the Bethe Free Energy (BFE)[23], which is known to lack epistemic (information-seeking) qualities. Consequently, agents using MP do not proactively seek informative states. Ongoing research seeks to address this limitation, for example, through the development of Constrained BFE (CBFE)[11]. The CBFE allows for inference that benefits from the MP algorithm's scaling advantages while retaining the epistemic qualities of expected free energy. Unfortunately, CBFE has thus far only been introduced for discrete active inference. Therefore, in this paper, we used the BFE.

To make message passing-based inference easy for the application engineer, there exist software toolboxes that have pre-computed message update rules for common factors and common distribution types [1, 16]. In principle, these toolboxes automate the inference process by calling pre-computed update rules.

Unfortunately, the sum-product rule is not always analytically solvable to a closed-form expression. In the next two sub-sections, we will discuss alternative message computation rules (Variational Message Passing and the Unscented Transform) that mesh seamlessly with the sum-product rule, leading to a *hybrid* message passing inference process. The interested reader is referred to [14] for a more in-depth explanation of message passing.

4.3 Variational message passing

The sum-product rule leads to closed-form outgoing messages if all incoming messages are Gaussian and the factor is a linear transformation. However, the graph for the controller's model contains a few factors where these conditions are not met. In those cases, Variational Message Passing (VMP) often resolves the issue since the VMP message computation rules lead to closed-form updates for all distributions in the exponential family as long as conjugacy is maintained [8]. VMP is a message-passing implementation of the more general variational approach to Bayesian inference [8]. Variational inference minimizes an upper bound (the variational free energy) on Bayesian evidence. In this way, the hard problem of evaluating an integral (needed for the Bayes rule) is replaced by an easier optimization problem.

Technically, for any node $f(y, x_1, \ldots, x_n)$, the VMP rule for an outgoing message over edge y is given by

$$\overrightarrow{\mu}(y) = \exp\Big(\int \underbrace{\overrightarrow{\mu}(x_1)..\overrightarrow{\mu}(x_n)}_{\text{incoming messages}} \underbrace{\ln f(y, x_1, .., x_n)}_{\text{log node function}} \mathrm{d}x_1..\mathrm{d}x_n\Big). \tag{13}$$

As an example of hybrid sum-product and VMP message passing-based inference, consider updating Bayesian beliefs about the measurement covariance matrix R_t, given a prior belief $q(R_{t-1})$ and a new observation y_t, see also Figure 1. Let the message

$$\overrightarrow{\mu}(R_{t-1}) = q(R_{t-1}) = \mathcal{W}^{-1}(R_{t-1} \mid \overrightarrow{n}_{t-1}, \overrightarrow{V}_{t-1}) \tag{14}$$

denote the posterior belief about R after observing $y_{1:t-1}$. This message will be used as the prior belief for time step t and is passed to the indicated equality node in Figure 1. The equality node also receives a message from the connected Gaussian node above. This message can be computed by the sum-product rule,

$$\overleftarrow{\mu}(R_t) = \int \overleftarrow{\mu}(v_t) \mathcal{N}(v_t \mid 0, R_t) \mathrm{d}v_t \propto \mathcal{W}^{-1}(R_t \mid \overleftarrow{n}_t, \overleftarrow{V}_t). \tag{15}$$

Note that the computation of $\overleftarrow{\mu}(R_t)$ uses an incoming message $\overleftarrow{\mu}(v_t)$ from the addition node. The equality node processes the two incoming messages to an updated posterior as follows:

$$q(R_t) = \int \overrightarrow{\mu}(R_{t-1}) \overleftarrow{\mu}(R_t) \overbrace{f_{=}(R_{t-1}, R_t)}^{\text{equality node}} \mathrm{d}R_{t-1} \propto \mathcal{W}^{-1}\big(R_t \mid n_t, V_t\big),$$

and the forward message from the measurement noise can be computed by a VMP update:

$$\overrightarrow{\mu}(v_t) \propto \exp\Big(\mathbb{E}_{q(R_t)}\left[\ln p(v_t \mid R_t)\right]\Big) \propto \mathcal{N}(v_t \mid 0, n_t V_t)\,. \tag{16}$$

4.4 Non-linear Dynamics and the Unscented Transform

In the controller's model, the transition function $g(s_t, u_t)$ is non-linear. When Gaussian messages are passed through a non-linear function, the outgoing message is non-Gaussian, both for the sum-product and VMP update rules. To keep going, we need to project the outgoing message in some way back to a Gaussian distribution.

Here, we discuss using the Unscented Transform (UT) to approximate outgoing messages with normal distributions [18]. As an example, consider the outgoing message $\overrightarrow{\mu}(s_{t+1})$ for the transition node with incoming messages

$$\overrightarrow{\mu}(u_{t+1}) = \mathcal{N}(u_{t+1} \mid \overrightarrow{m}^u_{t+1}, \overrightarrow{P}^u_{t+1})\,, \quad \overrightarrow{\mu}(s_t) = \mathcal{N}(s_t \mid \overrightarrow{m}^s_t, \overrightarrow{P}^s_t)\,, \tag{17}$$

as illustrated in Figure 1.

The Unscented Transform starts by selecting a set of "sigma points" $x_t^{(i)}$ and weights $\omega^{(i)}$ for $i = -M, \ldots, M$. For more details on the computation of sigma points and the weights, we refer to [7]. Next, the sigma points $x_t^{(i)}$ are processed through the nonlinear function as $\xi_t^{(i)} = g(x_t^{(i)})$. Then we compute the parameters of the outgoing (Gaussian) message $\overrightarrow{\mu}(s_{t+1}) = \mathcal{N}(s_{t+1} | \overrightarrow{m}^s_{t+1}, \overrightarrow{P}^s_{t+1})$ with mean and covariance matrix as

$$\overrightarrow{m}^s_{t+1} = \sum_{i=-M}^{M} \omega^{(i)} \xi_t^{(i)}\,,\ \overrightarrow{P}^s_{t+1} = \sum_{i=-M}^{M} \omega^{(i)} (\xi_t^{(i)} - \overrightarrow{m}^s_{t+1})(\xi_t^{(i)} - \overrightarrow{m}^s_{t+1})^{\top}. \tag{18}$$

4.5 Specification and Inference for the Control Prior

Real-world applications are often subject to environmental constraints, such as limits on engine power. In our application, We are interested in setting a prior constraint on the control signal in the form of $a < u_t < b$, where $a \in \mathbb{R}$ and $b \in \mathbb{R}$. While such a prior looks non-Gaussian, [10] describes an interesting way to implement these kinds of constraints efficiently by Gaussian message passing in a Normal distribution with Unknown Variance (NUV). A box-NUV prior is specified by a probabilistic sub-model, which contains two Normal distributions with means a and b, and unknown variances $\sigma_a^2 \in \mathbb{R}^+$ and $\sigma_b^2 \in \mathbb{R}^+$ with Gamma distribution priors. The box-NUV prior is specified as

$$p(u_t, \sigma_a^2, \sigma_b^2) = \mathcal{N}(u_t \mid a, \sigma_a^2)\mathcal{N}(u_t \mid b, \sigma_b^2)\Gamma\left(\sigma_a^2 \mid \frac{3}{2}, \frac{\gamma^2}{2}\right)\Gamma\left(\sigma_b^2 \mid \frac{3}{2}, \frac{\gamma^2}{2}\right), \quad (19)$$

where $\Gamma(\cdot|\alpha, \beta)$ represents a Gamma distribution with shape and rate parameters α and β, respectively. As it is shown in [9] the box-NUV prior can be obtained by:

$$\tilde{p}(u) = \sup_{\sigma_a^2, \sigma_b^2} p(u, \sigma_a^2, \sigma_b^2) \propto \exp\left\{-\gamma(|u-a| + |u-b|)\right\}. \quad (20)$$

In this paper for finding the σ_a^2, σ_b^2 we use the expectation maximization update rules according to [2].

5 Experiments

5.1 Experimental Setup

We evaluate the performance of the proposed controller. All experiments were simulated using the Julia programming language on a laptop with an Intel Core i9-12900HK processor and 32 GB of DDR4 RAM. To implement the controller, we used the open-source Julia package RxInfer [1], which supports (variational) Bayesian inference in models through hybrid message passing on an FFG. RxInfer implements many message passing techniques, including the needed methods that we discussed in this paper, namely the sum-product rule for Gaussian messages, VMP for conjugate distributions, the Unscented Transform for non-linear factors, and NUV priors for the control signal. In terms of trustworthiness, RxInfer comes with a large set of unit tests and has previously been used successfully in a wide range of applications, including audio processing devices [17] and control tasks [12]. In all simulations, we assume the Cart-Pole mechanical system described in (2), with parameter values $m_c = 1$ (kg), $m_p = 1$ (kg), $g = 9.81$ (m/s^2), $l = 0.5$ (m), and $\Delta t = 0.01$ (sec).

5.2 Controlling the Cart-Pole System

We validate the performance of the introduced model in section 3 for controlling the Cart-Pole system. For the controller, we used the model as described in (6)-(11). The parameters were set to $T = 100$, $a = -100$, $b = 100$, $Q = 10^{-8}$, $\gamma = 200$, $V_0 = 0.1 \cdot I$, and $n_0 = 10$. We set the controller time step size the same as the Cart-Pole system, i.e., $\Delta t = 0.01$ (sec).

We add noise with a variance $R^* = 0.01 \cdot I$ to the observation at each time step. The system starts in the state $s_0 = [0, -\pi, 0, 0]$ and the goal state is $s_T = [0, 0, 0, 0]$.

Figure 2 illustrates the evolution of the angle θ_t, position x_t, and estimated noise variance R_t during 800 time steps, totaling 8 seconds. It can be seen that the controller successfully guides the system to the target position. For R, we used the mode of $q(R)$, which for both dimensions converges to 0.015, a good approximation of the true added measurement noise.

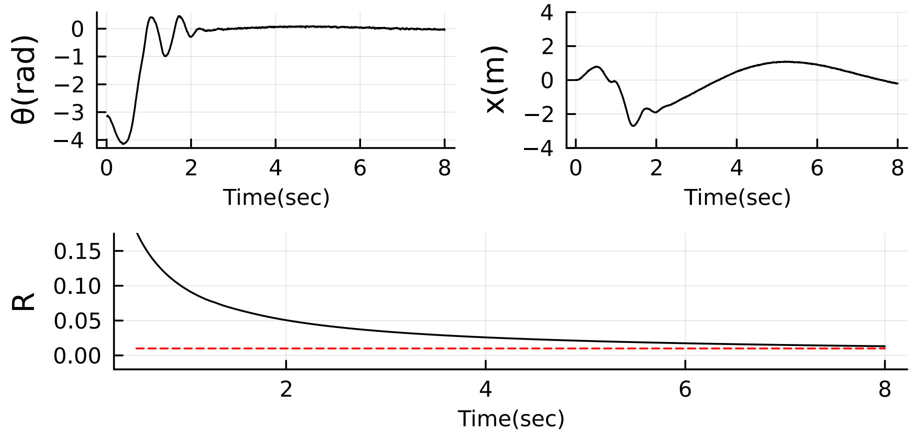


Fig. 2: Evolution of the pole angle (top-left), cart position (top-right), and inferred noise variance (bottom) over time. The dashed red line is the true added noise ($R^* = 0.01$).

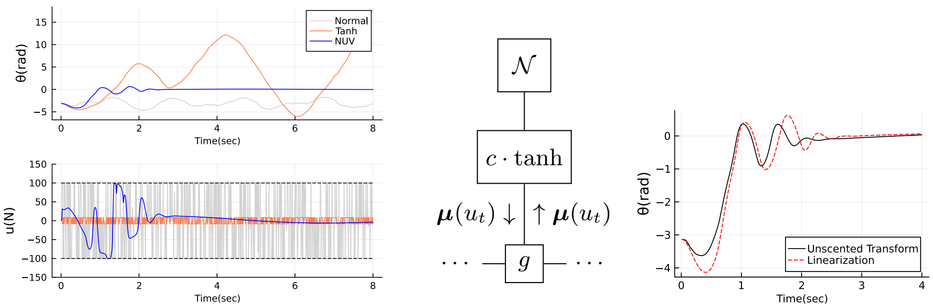
5.3 Alternative Control Limiters

In this experiment, we change the generative model of the controller and measure the effect on its performance. In particular, we compare using the box-NUV node with two alternative ways of limiting the control signal.

In the first alternative model, we set the action prior $p(u) = \mathcal{N}(0, c)$, where $c = 100$ is the control limit. This is the simplest model assumption. In the second alternative, we set the control prior as $p(u) = \mathcal{N}(0, c)$, and also we use a tanh-node according to the figure 3b. Using the tanh function is motivated by the fact that a scaled tanh function is a commonly used limiter, which, for instance, has been successfully used as an action constraint in [12]. But in our application, using the tanh function leads to a problem in the backward message. Since in the backward direction, the tanh-node acts like $\tanh^{-1}$ and its input should be mapped into $(-1, 1)$. This can be achieved by scaling the input as $(u_t - \delta)/c$, where δ is a small number used to prevent hitting the asymptotes 1 and -1 numerically. But, even with a tiny $\delta = 10^{-14}$ the $\tanh^{-1}$ range leads to around $[-18, 18]$ since $\tanh^{-1}((100 - 10^{-14})/100) = 18.7$. This dramatically reduces the controller's performance. The third approach is using the box-NUV node discussed in 4.5. The performance of the three controllers is shown in Figure 3a. Clearly, only the box-NUV node leads to the desired behavior of the controller.

5.4 Alternative Dynamics Approximations

We tested an alternative procedure to UT for passing Gaussian messages through the non-linear transition function $g(s_t, u_t)$. Linearization refers to a first-order Taylor approximation of the nonlinear dynamics around an operating point. We



(a) Comparison of the evolution of pole angle θ_t and inferred control u_t for three different control priors: Normal prior, tanh constraint, and box-NUV prior.

(b) The FFG of setting the limits on controls using the tanh transformation.

(c) Comparison of the evolution of the pole angle for the Unscented Transform and Linearization methods.

applied this method to approximate each of the outgoing messages of the g node. For each, the operating points were the means of the incoming messages. As illustrated in Figure 3c, the agent reaches the goal sooner. We may thus conclude that UT is a more useful approximation.

6 Conclusions

We introduced a fully Bayesian controller for a cart-pole system. While the Bayesian approach to control has a reputation for being both conceptually and computationally challenging, our findings demonstrate a viable path forward. By leveraging the active inference framework and employing a fast message passing-based inference toolbox, we showed how the role of the application engineer predominantly involves specifying a (biased) generative model for the controller. The clear separation of model specification from the inference process offers numerous benefits, notably streamlining the coding process, with the typical generative model requiring no more than half a page of code, even for complex controllers. Moreover, by automating the inference process, the application engineer can divert focus from computational efficiency issues, with this responsibility resting with the designers of the inference toolbox. In light of these advancements, we anticipate a growing prominence of Bayesian control applications.

Acknowledgments. This work was carried out in the context of the BayesBrain project. We gratefully acknowledge financial support from the Eindhoven Artificial Intelligence Systems Institute (EAISI) at TU Eindhoven.

Disclosure of Interests. The authors declare no conflict of interest.

References

1. Bagaev, D., de Vries, B.: Reactive message passing for scalable Bayesian inference. Scientific Programming (2023)
2. van Erp, B., Bagaev, D., Podusenko, A., İsmail, Ş., de Vries Bert: Multi-agent trajectory planning with NUV priors. American Control Conference (2024, in press)
3. Esaki, K., Matsumura, T., Minusa, S., Shao, Y., Yoshimura, C., Mizuno, H.: Dynamical perception-action loop formation with developmental embodiment for hierarchical active inference. In: International Workshop on Active Inference. pp. 14–28. Springer (2023)
4. Friston, K.: What is optimal about motor control? Neuron **72**(3), 488–498 (2011)
5. Friston, K., Samothrakis, S., Montague, R.: Active inference and agency: optimal control without cost functions. Biological Cybernetics **106**, 523–541 (2012)
6. Friston, K.J., Parr, T., de Vries, B.: The graphical brain: belief propagation and active inference. Network neuroscience **1**(4), 381–414 (2017)
7. Gustafsson, F., Hendeby, G.: Some relations between extended and unscented Kalman filters. IEEE Transactions on Signal Processing **60**(2), 545–555 (2011)
8. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Machine learning **37**, 183–233 (1999)
9. Keusch, R.: Composite NUV priors and applications. Doctoral Thesis, ETH Zurich (2022). https://doi.org/10.3929/ethz-b-000575227, iSBN: 9783866287686
10. Keusch, R., Loeliger, H.A.: Model-predictive control with new NUV priors. arXiv preprint arXiv:2303.15806 (2023)
11. van de Laar, T., Koudahl, M., van Erp, B., de Vries, B.: Active inference and epistemic value in graphical models. Frontiers in Robotics and AI **9**, 794464 (2022)
12. van de Laar, T.W., de Vries, B.: Simulating active inference processes by message passing. Frontiers in Robotics and AI **6**, 20 (2019)
13. Lanillos, P., Meo, C., Pezzato, C., Meera, A.A., Baioumy, M., Ohata, W., Tschantz, A., Millidge, B., Wisse, M., Buckley, C.L., et al.: Active inference in robotics and artificial agents: Survey and challenges. arXiv preprint arXiv:2112.01871 (2021)
14. Loeliger, H.A., Dauwels, J., Hu, J., Korl, S., Ping, L., Kschischang, F.R.: The factor graph approach to model-based signal processing. Proceedings of the IEEE **95**(6), 1295–1322 (2007)
15. Mills, A., Wills, A., Ninness, B.: Nonlinear model predictive control of an inverted pendulum. In: American control conference. pp. 2335–2340. IEEE (2009)
16. Minka, T., Winn, J., Guiver, J., Zaykov, Y., Fabian, D., Bronskill, J.: /Infer.NET 0.3 (2018), microsoft Research Cambridge. http://dotnet.github.io/infer
17. Podusenko, A., van Erp, B., Koudahl, M., de Vries, B.: AIDA: An active inference-based design agent for audio processing algorithms. Frontiers in Signal Processing **2**, 842477 (2022)
18. Särkkä, S., Svensson, L.: Bayesian filtering and smoothing, vol. 17. Cambridge University Press (2023)
19. Smith, R., Friston, K.J., Whyte, C.J.: A step-by-step tutorial on active inference and its application to empirical data. Journal of mathematical psychology **107**, 102632 (2022)
20. Stengel, R.F.: Optimal control and estimation. Courier Corporation (1994)
21. Tedrake, R.: Underactuated robotics: Learning, planning, and control for efficient and agile machines. Course notes for MIT **6**, 832 (2009)
22. Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research **1**, 211–244 (2001)

23. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Constructing free-energy approximations and generalized belief propagation algorithms. IEEE Transactions on information theory **51**(7), 2282–2312 (2005)