On deriving efficient information-seeking behaviour for intelligent autonomous systems

Applied Mathematics Seminar, McGill University October 20, 2025

Wouter Kouw
Assistant professor — Bayesian Intelligent Autonomous Systems lab

Department of Electrical Engineering

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Outline

- Problem: how to make robots more autonomous?

Outline

- Problem: how to make robots more autonomous?
- Technique: free energy minimization by message passing

Outline

- Problem: how to make robots more autonomous?
- Technique: free energy minimization by message passing
- Example: ARxI - Autoregressive Reactive Inference agent

Outline

- Problem: how to make robots more autonomous?
- Technique: free energy minimization by message passing
- Example: ARxI - Autoregressive Reactive Inference agent
- Discussion: effects of controlling information gain

Problem

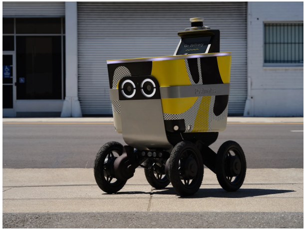
How can we make robots more autonomous?

(a) MTC, https://tinyurl.com/y45mksa6

Problem

How can we make robots more autonomous?

(a) MTC, https://tinyurl.com/y45mksa6

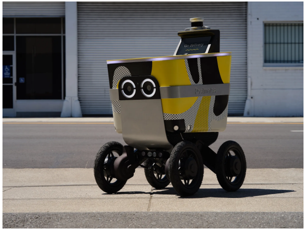


(b) Postmates, https://tinyurl.com/5442pph6

Problem

How can we make robots more autonomous?

(a) MTC, https://tinyurl.com/y45mksa6



(b) Postmates, https://tinyurl.com/5442pph6

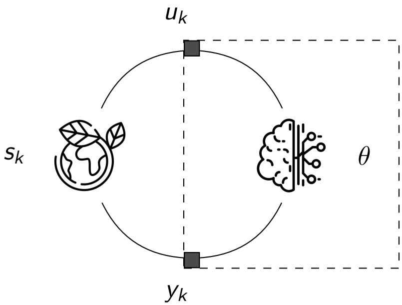
(c) MPI IS, https://tinyurl.com/5n8sd6ps

Problem

Let s_k be the state of a system at time t_k. An agent is an entity that models the system using parameters θ based on noisy partial observations y_k, and acts upon the world with u_k.

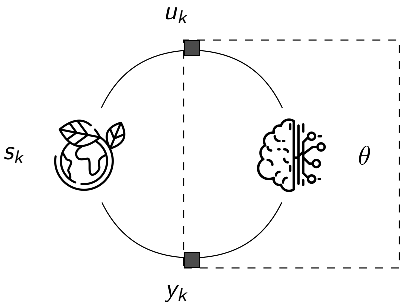
Problem

Let s_k be the state of a system at time t_k. An agent is an entity that models the system using parameters θ based on noisy partial observations y_k, and acts upon the world with u_k.



Problem

Let s_k be the state of a system at time t_k. An agent is an entity that models the system using parameters θ based on noisy partial observations y_k, and acts upon the world with u_k.



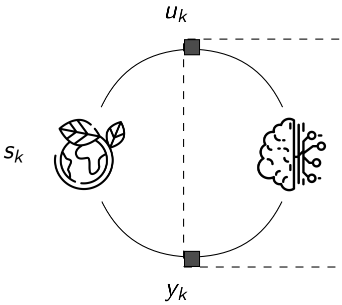
How can intelligent agents continually and efficiently adapt to uncertain environments?

Free Energy Principle

I approach this problem from a probabilistic perspective.

Free Energy Principle

I approach this problem from a probabilistic perspective.



$$\underbrace{p(y_k, \theta \mid u_k)}_{\text{generative model}} = \underbrace{p(y_k \mid u_k, \theta)}_{\text{likelihood}} \underbrace{p(\theta)}_{\text{prior}}$$

Free Energy Principle

We find intelligent agents in nature. Modern theories on neural information processing argue that agents pursue a single objective function: a free energy functional.

Friston (2010), The free-energy principle: a unified brain theory?, Nature.

Free Energy Principle

We find intelligent agents in nature. Modern theories on neural information processing argue that agents pursue a single objective function: a free energy functional.

Free energy functional

Given the generative model p and a variational model q, the free energy functional is

$$\mathcal{F}[q] = \int q(\theta) \ln \frac{q(\theta)}{p(y_k, \theta \mid u_k)} \mathrm{d}\theta \ .$$

Friston (2010), The free-energy principle: a unified brain theory?, Nature.

Free Energy Principle

We find intelligent agents in nature. Modern theories on neural information processing argue that agents pursue a single objective function: a free energy functional.

Free energy functional

Given the generative model p and a variational model q, the free energy functional is

$$\mathcal{F}[q] = \int q(\theta) \ln \frac{q(\theta)}{p(y_k, \theta \mid u_k)} \mathrm{d}\theta \ .$$

The variational model q represents a trade-off between accuracy and computational effort.

Friston (2010), The free-energy principle: a unified brain theory?, Nature.

Variational free energy minimization

Inference is the process of finding the optimal variational distribution:

$$q^*(\theta) = \underset{q \in Q}{\arg\min}\ \mathcal{F}[q]\,.$$

Variational free energy minimization

Inference is the process of finding the optimal variational distribution:

$$q^*(\theta) = \underset{q \in Q}{\arg\min}\ \mathcal{F}[q] \, .$$

This is done by minimizing the variational derivative over the free energy functional with a normalization constraint:

$$\mathcal{L}[q, \gamma] = \mathcal{F}[q] + \gamma\Big(\int q(\theta)\mathrm{d}\theta - 1\Big) \, .$$

Variational free energy minimization

Inference is the process of finding the optimal variational distribution:

$$q^*(\theta) = \underset{q \in Q}{\arg\min}\ \mathcal{F}[q]\,.$$

This is done by minimizing the variational derivative over the free energy functional with a normalization constraint:

$$\mathcal{L}[q, \gamma] = \mathcal{F}[q] + \gamma\Big(\int q(\theta)\mathrm{d}\theta - 1\Big)\,.$$

More complex models also have factorization, parametrization and/or chance constraints.

Senoz et al. (2021), Variational message passing and local constraint manipulation in factor graphs, Entropy.

Variational free energy minimization

Let $\delta q(\theta) = \epsilon\phi(\theta)$ be a variation with ϕ a continuous and differentiable test function. Then the variational derivative can be found with:

Variational free energy minimization

Let $\delta q(\theta) = \epsilon\phi(\theta)$ be a variation with ϕ a continuous and differentiable test function. Then the variational derivative can be found with:

$$\int \frac{\delta \mathcal{L}[q,\gamma]}{\delta q} \phi(\theta) \mathrm{d}\theta = \frac{d\mathcal{L}[q(\theta)+\epsilon\phi(\theta),\gamma]}{d\epsilon}\Big|_{\epsilon=0}$$

Variational free energy minimization

Let $\delta q(\theta) = \epsilon\phi(\theta)$ be a variation with ϕ a continuous and differentiable test function. Then the variational derivative can be found with:

$$\int \frac{\delta \mathcal{L}[q,\gamma]}{\delta q}\phi(\theta)\mathrm{d}\theta = \frac{d\mathcal{L}[q(\theta)+\epsilon\phi(\theta),\gamma]}{d\epsilon}\Big|_{\epsilon=0}$$

$$= \frac{d}{d\epsilon}\int \big(q(\theta)+\epsilon\phi(\theta)\big)\ln\frac{q(\theta)+\epsilon\phi(\theta)}{p(y_k,\theta \mid u_k)}\mathrm{d}\theta + \frac{d}{d\epsilon}\gamma\int (q(\theta)+\epsilon\phi(\theta))\mathrm{d}\theta\Big|_{\epsilon=0}$$

Variational free energy minimization

Let $\delta q(\theta) = \epsilon\phi(\theta)$ be a variation with ϕ a continuous and differentiable test function. Then the variational derivative can be found with:

$$\begin{aligned}
\int \frac{\delta \mathcal{L}[q,\gamma]}{\delta q}\phi(\theta)\mathrm{d}\theta &= \frac{d\mathcal{L}[q(\theta)+\epsilon\phi(\theta),\gamma]}{d\epsilon}\Big|_{\epsilon=0} \\
&= \frac{d}{d\epsilon}\int \big(q(\theta)+\epsilon\phi(\theta)\big)\ln\frac{q(\theta)+\epsilon\phi(\theta)}{p(y_k,\theta \mid u_k)}\mathrm{d}\theta + \frac{d}{d\epsilon}\gamma\int (q(\theta)+\epsilon\phi(\theta))\mathrm{d}\theta\Big|_{\epsilon=0} \\
&= \int \big(\phi(\theta)\ln\frac{q(\theta)}{p(y_k,\theta \mid u_k)} + \phi(\theta)\big)\mathrm{d}\theta + \gamma\int \phi(\theta)\mathrm{d}\theta
\end{aligned}$$

Variational free energy minimization

Let $\delta q(\theta) = \epsilon\phi(\theta)$ be a variation with ϕ a continuous and differentiable test function. Then the variational derivative can be found with:

$$\begin{aligned}
\int \frac{\delta \mathcal{L}[q,\gamma]}{\delta q}\phi(\theta)\mathrm{d}\theta &= \frac{d\mathcal{L}[q(\theta)+\epsilon\phi(\theta),\gamma]}{d\epsilon}\Big|_{\epsilon=0} \\
&= \frac{d}{d\epsilon}\int \big(q(\theta)+\epsilon\phi(\theta)\big)\ln\frac{q(\theta)+\epsilon\phi(\theta)}{p(y_k,\theta \mid u_k)}\mathrm{d}\theta + \frac{d}{d\epsilon}\gamma\int (q(\theta)+\epsilon\phi(\theta))\mathrm{d}\theta\Big|_{\epsilon=0} \\
&= \int \Big(\phi(\theta)\ln\frac{q(\theta)}{p(y_k,\theta \mid u_k)} + \phi(\theta)\Big)\mathrm{d}\theta + \gamma\int \phi(\theta)\mathrm{d}\theta \\
&= \int \Big(\ln\frac{q(\theta)}{p(y_k,\theta \mid u_k)} + 1 + \gamma\Big)\phi(\theta)\mathrm{d}\theta\,.
\end{aligned}$$

Variational free energy minimization

Setting the variational derivative to 0, yields

Variational free energy minimization

Setting the variational derivative to 0, yields

$$0 = \ln \frac{q(\theta)}{p(y_k, \theta \mid u_k)} + 1 + \gamma$$

Variational free energy minimization

Setting the variational derivative to 0, yields

$$\begin{aligned} 0 &= \ln \frac{q(\theta)}{p(y_k, \theta \mid u_k)} + 1 + \gamma \\ q(\theta) &= \exp(-\gamma - 1) p(y_k, \theta \mid u_k). \end{aligned}$$

Variational free energy minimization

Setting the variational derivative to 0, yields

$$
\begin{aligned}
0 &= \ln \frac{q(\theta)}{p(y_k, \theta \mid u_k)} + 1 + \gamma \\
q(\theta) &= \exp(-\gamma - 1) p(y_k, \theta \mid u_k).
\end{aligned}
$$

Plugging this into the constraint gives $\exp(-\gamma - 1) = 1 / \int p(y_k, \theta \mid u_k) \mathrm{d}\theta$, which means

$$
q(\theta) = \frac{p(y_k \mid u_k, \theta) p(\theta)}{\int p(y_k \mid u_k, \theta) p(\theta) \mathrm{d}\theta} .
$$

Factor graphs

Factor graphs

Using the factorization structure, the free energy functional can be decomposed to:

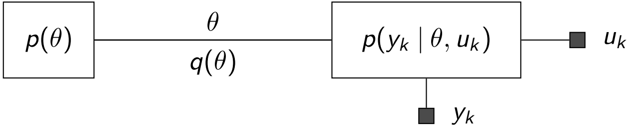
$$\mathcal{F}[q] = \int q(\theta)\big[-\ln p(y_k \mid u_k, \theta)\big]\mathrm{d}\theta + \int q(\theta) \ln p(\theta)\mathrm{d}\theta - \int q(\theta) \ln q(\theta)\mathrm{d}\theta\,.$$

Factor graphs

Using the factorization structure, the free energy functional can be decomposed to:

$$\mathcal{F}[q] = \int q(\theta)\big[-\ln p(y_k \mid u_k, \theta)\big]\mathrm{d}\theta + \int q(\theta)\ln p(\theta)\mathrm{d}\theta - \int q(\theta)\ln q(\theta)\mathrm{d}\theta\,.$$

This can be matched to a factor graph structure:

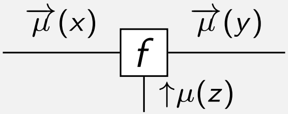


Loeliger (2004), An introduction to factor graphs, IEEE Signal Processing Magazine.

Message passing

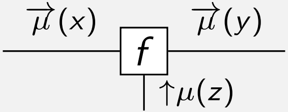
Message passing

Aggregating incoming messages:

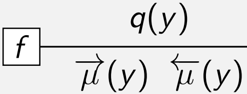


Message passing

Aggregating incoming messages:

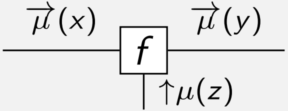


Updating marginals by product of messages:

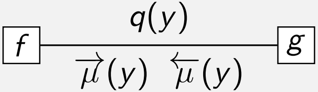


Message passing

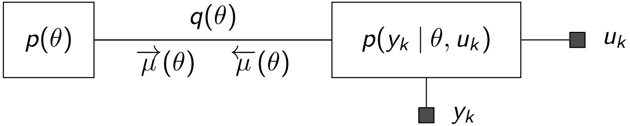
Aggregating incoming messages:



Updating marginals by product of messages:



For the example graph, this is:

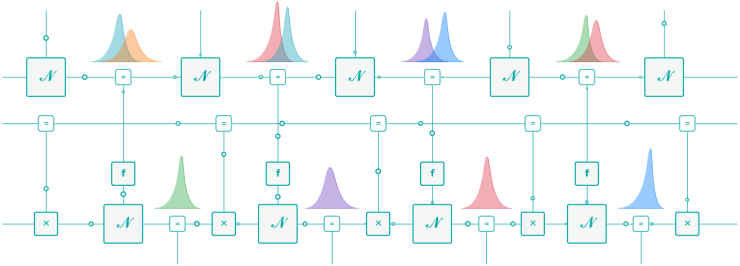


RxInfer.jl

At BIASlab in Eindhoven, we built an open-source state-of-the-art toolbox for this:

rxinfer

Automatic Bayesian Inference through Reactive Message Passing

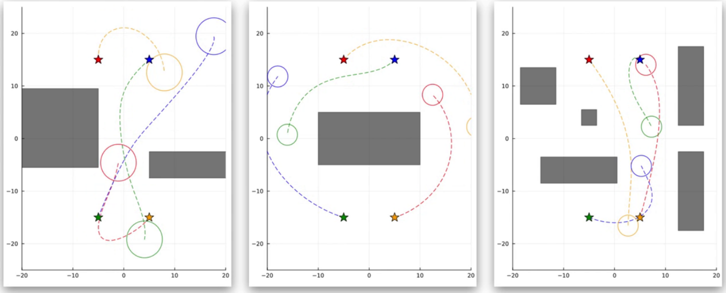


Free energy minimizing systems

Using these techniques, we construct all sorts of signal processing and control systems.

Free energy minimizing systems

Using these techniques, we construct all sorts of signal processing and control systems.



ARxI: Autoregressive Reactive Inference agent

ARxI is an example of an agent that minimizes free energy by message passing.

ARxI: Autoregressive Reactive Inference agent

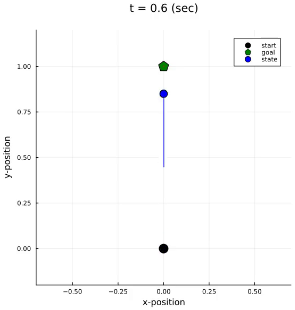
ARxI is an example of an agent that minimizes free energy by message passing.

I will show how ARxI plans and learns to navigate;

ARxI: Autoregressive Reactive Inference agent

ARxI is an example of an agent that minimizes free energy by message passing.

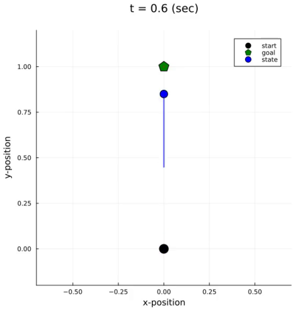
I will show how ARxI plans and learns to navigate;

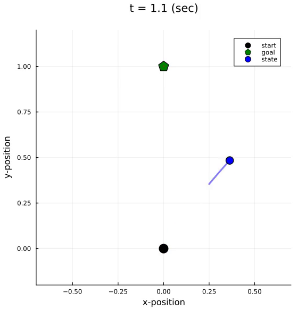


ARxI: Autoregressive Reactive Inference agent

ARxI is an example of an agent that minimizes free energy by message passing.

I will show how ARxI plans and learns to navigate;





ARxI: Generative model specification

Let the likelihood of $y_k \in \mathbb{R}^{D_y}$, given $u_k \in \mathbb{R}^{D_u}$, be Gaussian distributed,

Nisslbeck & Kouw (2025), Online Bayesian system identification in multivariate autoregressive models via message passing, ECC.

ARxI: Generative model specification

Let the likelihood of $y_k \in \mathbb{R}^{D_y}$, given $u_k \in \mathbb{R}^{D_u}$, be Gaussian distributed,

$$p(y_k \mid \Theta, u_k, \bar{u}_k, \bar{y}_k) = \mathcal{N}(y_k \mid A^\top \begin{bmatrix} \bar{y}_k \\ u_k \\ \bar{u}_k \end{bmatrix}, W^{-1}),$$

Nisslbeck & Kouw (2025), Online Bayesian system identification in multivariate autoregressive models via message passing, ECC.

ARxI: Generative model specification

Let the likelihood of $y_k \in \mathbb{R}^{D_y}$, given $u_k \in \mathbb{R}^{D_u}$, be Gaussian distributed,

$$p(y_k \mid \Theta, u_k, \bar{u}_k, \bar{y}_k) = \mathcal{N}(y_k \mid A^\top \begin{bmatrix} \bar{y}_k \\ u_k \\ \bar{u}_k \end{bmatrix}, W^{-1}),$$

where $\Theta = \{A, W : A \in \mathbb{R}^{D_u \times D_y}, W \in \mathbb{R}^{D_y \times D_y}\}$, $\bar{u}_k = \begin{bmatrix} u_{k-1} .. u_{k-n_u} \end{bmatrix}^\intercal$, $\bar{y}_k = \begin{bmatrix} y_{k-1} .. y_{k-n_y} \end{bmatrix}^\intercal$.

Nisslbeck & Kouw (2025), Online Bayesian system identification in multivariate autoregressive models via message passing, ECC.

ARxI: Generative model specification

Let the likelihood of $y_k \in \mathbb{R}^{D_y}$, given $u_k \in \mathbb{R}^{D_u}$, be Gaussian distributed,

$$p(y_k \mid \Theta, u_k, \bar{u}_k, \bar{y}_k) = \mathcal{N}\left(y_k \mid A^\top \begin{bmatrix} \bar{y}_k \\ u_k \\ \bar{u}_k \end{bmatrix}, W^{-1}\right),$$

where $\Theta = \{A, W : A \in \mathbb{R}^{D_u \times D_y}, W \in \mathbb{R}^{D_y \times D_y}\}$, $\bar{u}_k = \left[u_{k-1} \,..\, u_{k-n_u}\right]^\mathsf{T}$, $\bar{y}_k = \left[y_{k-1} \,..\, y_{k-n_y}\right]^\mathsf{T}$.

For the prior distribution, we select a matrix Normal Wishart distribution:

Nisslbeck & Kouw (2025), Online Bayesian system identification in multivariate autoregressive models via message passing, ECC.

ARxI: Generative model specification

Let the likelihood of $y_k \in \mathbb{R}^{D_y}$, given $u_k \in \mathbb{R}^{D_u}$, be Gaussian distributed,

$$p(y_k \mid \Theta, u_k, \bar{u}_k, \bar{y}_k) = \mathcal{N}\left(y_k \mid A^\top \begin{bmatrix} \bar{y}_k \\ u_k \\ \bar{u}_k \end{bmatrix}, W^{-1}\right),$$

where $\Theta = \{A, W : A \in \mathbb{R}^{D_u \times D_y}, W \in \mathbb{R}^{D_y \times D_y}\}$, $\bar{u}_k = \left[u_{k-1} \,..\, u_{k-n_u}\right]^\mathsf{T}$, $\bar{y}_k = \left[y_{k-1} \,..\, y_{k-n_y}\right]^\mathsf{T}$.

For the prior distribution, we select a matrix Normal Wishart distribution:

$$p(\Theta) = \mathcal{MNW}(A, W \mid M_0, \Lambda_0^{-1}, \Omega_0^{-1}, \nu_0)$$

Nisslbeck & Kouw (2025), Online Bayesian system identification in multivariate autoregressive models via message passing, ECC.

ARxI: Generative model specification

Let the likelihood of $y_k \in \mathbb{R}^{D_y}$, given $u_k \in \mathbb{R}^{D_u}$, be Gaussian distributed,

$$p(y_k \mid \Theta, u_k, \bar{u}_k, \bar{y}_k) = \mathcal{N}\left(y_k \mid A^\top \begin{bmatrix} \bar{y}_k \\ u_k \\ \bar{u}_k \end{bmatrix}, W^{-1}\right),$$

where $\Theta = \{A, W : A \in \mathbb{R}^{D_u \times D_y}, W \in \mathbb{R}^{D_y \times D_y}\}$, $\bar{u}_k = \begin{bmatrix} u_{k-1} \,..\, u_{k-n_u} \end{bmatrix}^\mathsf{T}$, $\bar{y}_k = \begin{bmatrix} y_{k-1} \,..\, y_{k-n_y} \end{bmatrix}^\mathsf{T}$.

For the prior distribution, we select a matrix Normal Wishart distribution:

$$\begin{aligned} p(\Theta) &= \mathcal{MNW}(A, W \mid M_0, \Lambda_0^{-1}, \Omega_0^{-1}, \nu_0) \\ &= \mathcal{MN}(A \mid M_0, \Lambda_0^{-1}, \Omega_0^{-1}) \mathcal{W}(W \mid \Omega_0, \nu_0) \end{aligned}$$

Nisslbeck & Kouw (2025), Online Bayesian system identification in multivariate autoregressive models via message passing, ECC.

ARxI: Inferring parameters

The free energy functional for this model is:

$$\mathcal{F}[q] = \int q(\Theta)\big[-\ln p(y_k \mid \Theta, u_k, \bar{u}_k, \bar{y}_k)\big]\mathrm{d}\Theta + \int q(\Theta) \ln p(\Theta)\mathrm{d}\Theta - \int q(\Theta) \ln q(\Theta)\mathrm{d}\Theta\,.$$

ARxI: Inferring parameters

The free energy functional for this model is:

$$\mathcal{F}[q] = \int q(\Theta)\big[- \ln p(y_k \mid \Theta, u_k, \bar{u}_k, \bar{y}_k)\big]\mathrm{d}\Theta + \int q(\Theta) \ln p(\Theta)\mathrm{d}\Theta - \int q(\Theta) \ln q(\Theta)\mathrm{d}\Theta \, .$$

The optimal variational distribution is:

$$q(\Theta) = \frac{p(y_k \mid \Theta, u_k, \bar{u}_k, \bar{y}_k)}{p(y_k \mid u_k, \mathcal{D}_{k-1})} p(\Theta)$$

ARxI: Inferring parameters

The free energy functional for this model is:

$$\mathcal{F}[q] = \int q(\Theta)\big[-\ln p(y_k \mid \Theta, u_k, \bar{u}_k, \bar{y}_k)\big]\mathrm{d}\Theta + \int q(\Theta) \ln p(\Theta)\mathrm{d}\Theta - \int q(\Theta) \ln q(\Theta)\mathrm{d}\Theta \,.$$

The optimal variational distribution is:

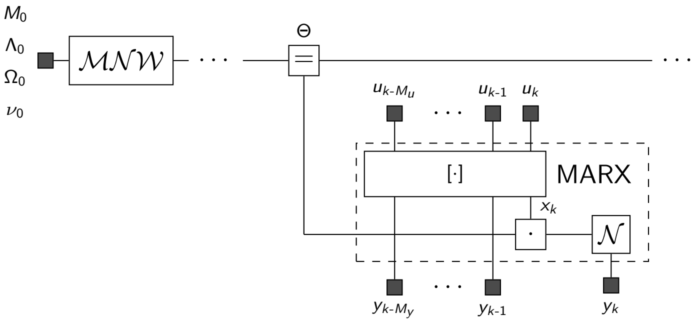
$$q(\Theta) = \frac{p(y_k \mid \Theta, u_k, \bar{u}_k, \bar{y}_k)}{p(y_k \mid u_k, \mathcal{D}_{k-1})} p(\Theta)$$

Here, $q(\Theta)$ recovers the true posterior distribution $p(\Theta \mid \mathcal{D}_k)$ exactly.

Särkkä, Svensson (2023). Bayesian filtering & Smoothing.

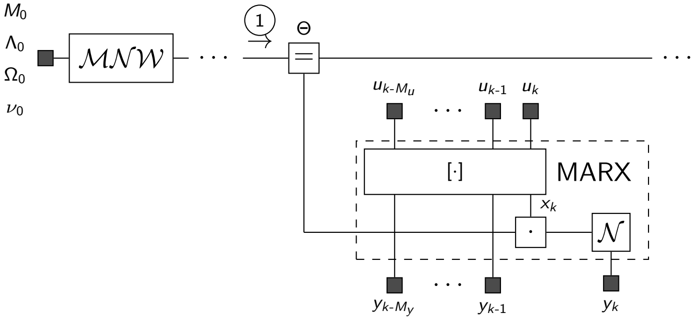
ARxI: Inferring parameters

In a factor graph, with messages, this is:



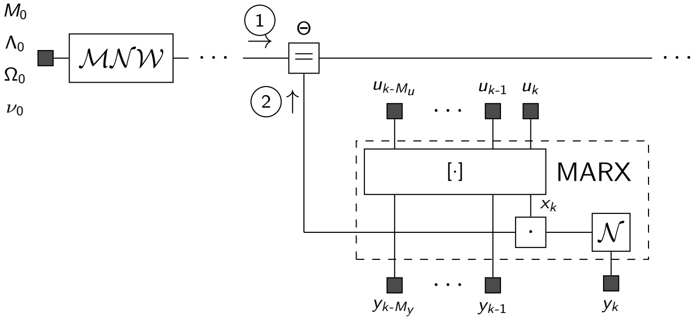
ARxI: Inferring parameters

In a factor graph, with messages, this is:



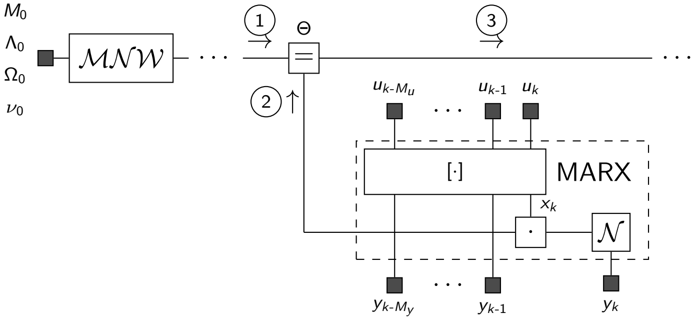
ARxI: Inferring parameters

In a factor graph, with messages, this is:



ARxI: Inferring parameters

In a factor graph, with messages, this is:



ARxI: Planning

For planning, we unroll the generative model into the future. For $t = k + 1$;

$$p(y_t, \Theta, u_t \mid \mathcal{D}_k) = p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k) p(u_t)$$

ARxI: Planning

For planning, we unroll the generative model into the future. For $t = k + 1$;

$$p(y_t, \Theta, u_t \mid \mathcal{D}_k) = p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k) p(u_t)$$

To incorporate a target, we first isolate the marginal distribution for the future output:

$$p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k) = p(y_t, \Theta \mid u_t, \mathcal{D}_k) = p(\Theta \mid y_t, u_t, \mathcal{D}_k) p(y_t) \, .$$

ARxI: Planning

For planning, we unroll the generative model into the future. For $t = k + 1$;

$$p(y_t, \Theta, u_t \mid \mathcal{D}_k) = p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k) p(u_t)$$

To incorporate a target, we first isolate the marginal distribution for the future output:

$$p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k) = p(y_t, \Theta \mid u_t, \mathcal{D}_k) = p(\Theta \mid y_t, u_t, \mathcal{D}_k) p(y_t) \, .$$

We intervene on the marginal, $p(y_t) \rightarrow p(y_t \mid y_*) = \mathcal{N}(y_t \mid m_*, S_*)$, and rewrite

$$p(\Theta \mid y_t, u_t, \mathcal{D}_k) = \frac{p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k)}{p(y_t \mid u_t, \mathcal{D}_k)} \, .$$

ARxI: Planning

For planning, we unroll the generative model into the future. For $t = k+1$;

$$p(y_t, \Theta, u_t \mid \mathcal{D}_k) = p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k) p(u_t)$$

To incorporate a target, we first isolate the marginal distribution for the future output:

$$p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k) = p(y_t, \Theta \mid u_t, \mathcal{D}_k) = p(\Theta \mid y_t, u_t, \mathcal{D}_k) p(y_t)\,.$$

We intervene on the marginal, $p(y_t) \rightarrow p(y_t \mid y_*) = \mathcal{N}(y_t \mid m_*, S_*)$, and rewrite

$$p(\Theta \mid y_t, u_t, \mathcal{D}_k) = \frac{p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k)}{p(y_t \mid u_t, \mathcal{D}_k)}\,.$$

Overall, the generative model for the future becomes:

$$p(y_t, \Theta, u_t \mid y_*, \mathcal{D}_k) = \frac{p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k)}{p(y_t \mid u_t, \mathcal{D}_k)} p(y_t \mid y_*) p(u_t)\,.$$

ARxI: Predictions

The planning model requires $p(y_t \mid u_t, \mathcal{D}_k)$, i.e., the posterior predictive distribution.

Murphy (2022). Probabilistic machine learning: an introduction.

ARxI: Predictions

The planning model requires $p(y_t \mid u_t, \mathcal{D}_k)$, i.e., the posterior predictive distribution.

Under the autoregressive model specification, we can solve the integral analytically:

Murphy (2022). Probabilistic machine learning: an introduction.

ARxI: Predictions

The planning model requires $p(y_t \mid u_t, \mathcal{D}_k)$, i.e., the posterior predictive distribution.

Under the autoregressive model specification, we can solve the integral analytically:

$$p(y_t \mid u_t, \mathcal{D}_k) = \int p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k) \mathrm{d}\Theta$$

Murphy (2022). Probabilistic machine learning: an introduction.

ARxI: Predictions

The planning model requires $p(y_t \mid u_t, \mathcal{D}_k)$, i.e., the posterior predictive distribution.

Under the autoregressive model specification, we can solve the integral analytically:

$$\begin{aligned} p(y_t \mid u_t, \mathcal{D}_k) &= \int p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k) \mathrm{d}\Theta \\ &= \mathcal{T}_{\eta_k}\big(y_t \mid \mu_k(u_t), \Sigma_k(u_t)\big) , \end{aligned}$$

Murphy (2022). Probabilistic machine learning: an introduction.

ARxI: Predictions

The planning model requires $p(y_t \mid u_t, \mathcal{D}_k)$, i.e., the posterior predictive distribution.

Under the autoregressive model specification, we can solve the integral analytically:

$$\begin{aligned} p(y_t \mid u_t, \mathcal{D}_k) &= \int p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k) \mathrm{d}\Theta \\ &= \mathcal{T}_{\eta_k}\big(y_t \mid \mu_k(u_t), \Sigma_k(u_t)\big) \, , \end{aligned}$$

where $\eta_k = \nu_k - D_y + 1$, $\mu_k(u_t) = M_k^\intercal \begin{bmatrix} u_k \\ \bar{u}_t \\ \bar{y}_t \end{bmatrix}$ and $\Sigma_k(u_t) = \eta_k \Omega_k \Big(1 + \begin{bmatrix} u_t \\ \bar{u}_t \\ \bar{y}_t \end{bmatrix}^\intercal \Lambda_k^{-1} \begin{bmatrix} u_t \\ \bar{u}_t \\ \bar{y}_t \end{bmatrix} \Big)$.

Murphy (2022). Probabilistic machine learning: an introduction.

ARxI: Inferring controls

To infer a variational posterior over controls, we form a free energy functional;

Kouw et al. (2025), *Message passing-based inference in an autoregressive active inference agent*, IWAI.

ARxI: Inferring controls

To infer a variational posterior over controls, we form a free energy functional;

$$\mathcal{F}_k[q] = \mathbb{E}_{q(y_t, \Theta, u_t)}\Big[\ln \frac{q(y_t, \Theta, u_t)}{p(y_t, \Theta, u_t \mid y_*, \mathcal{D}_k)} \Big] \, ,$$

Kouw et al. (2025), Message passing-based inference in an autoregressive active inference agent, IWAI.

ARxI: Inferring controls

To infer a variational posterior over controls, we form a free energy functional;

$$\mathcal{F}_k[q] = \mathbb{E}_{q(y_t,\Theta,u_t)}\left[\ln \frac{q(y_t,\Theta,u_t)}{p(y_t,\Theta,u_t \mid y_*,\mathcal{D}_k)}\right],$$

where $q(y_t,\Theta,u_t) = p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t)p(\Theta \mid \mathcal{D}_k)q(u_t)$.

Kouw et al. (2025), Message passing-based inference in an autoregressive active inference agent, IWAI.

ARxI: Inferring controls

To infer a variational posterior over controls, we form a free energy functional;

$$\mathcal{F}_k[q] = \mathbb{E}_{q(y_t, \Theta, u_t)}\Big[\ln \frac{q(y_t, \Theta, u_t)}{p(y_t, \Theta, u_t \mid y_*, \mathcal{D}_k)} \Big] ,$$

where $q(y_t, \Theta, u_t) = p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k) q(u_t)$.

The solution is proportional to the prior times an energy-based pseudo-likelihood:

Kouw et al. (2025), Message passing-based inference in an autoregressive active inference agent, IWAI.

ARxI: Inferring controls

To infer a variational posterior over controls, we form a free energy functional;

$$\mathcal{F}_k[q] = \mathbb{E}_{q(y_t, \Theta, u_t)}\Big[\ln \frac{q(y_t, \Theta, u_t)}{p(y_t, \Theta, u_t \mid y_*, \mathcal{D}_k)}\Big],$$

where $q(y_t, \Theta, u_t) = p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t)p(\Theta \mid \mathcal{D}_k)q(u_t)$.

The solution is proportional to the prior times an energy-based pseudo-likelihood:

$$q^*(u_t) = \underset{q \in Q}{\arg\min}\ \mathcal{F}_k[q]$$

Kouw et al. (2025), Message passing-based inference in an autoregressive active inference agent, IWAI.

ARxI: Inferring controls

To infer a variational posterior over controls, we form a free energy functional;

$$\mathcal{F}_k[q] = \mathbb{E}_{q(y_t,\Theta,u_t)}\Big[\ln \frac{q(y_t,\Theta,u_t)}{p(y_t,\Theta,u_t \mid y_*,\mathcal{D}_k)}\Big] ,$$

where $q(y_t,\Theta,u_t) = p(y_t \mid \Theta,u_t,\bar{u}_t,\bar{y}_t)p(\Theta \mid \mathcal{D}_k)q(u_t)$.

The solution is proportional to the prior times an energy-based pseudo-likelihood:

$$\begin{aligned} q^*(u_t) &= \underset{q \in Q}{\arg\min}\ \mathcal{F}_k[q] \\ &\propto p(u_t) \exp\Big(\underbrace{\mathbb{E}_{p(y_t \mid u_t,\mathcal{D}_k)}\big[-\ln p(y_t \mid u_t,\mathcal{D}_k)\big]}_{\text{predictive entropy}} - \underbrace{\mathbb{E}_{p(y_t \mid u_t,\mathcal{D}_k)}\big[-\ln p(y_t \mid y_*)\big]}_{\text{cross-entropy to goal}} \Big) . \end{aligned}$$

Kouw et al. (2025), Message passing-based inference in an autoregressive active inference agent, IWAI.

ARxI: information gain

Under the given model specification, we can derive an insightful interpretation:

ARxI: information gain

Under the given model specification, we can derive an insightful interpretation:

Theorem

For particular p, q, the entropy of the posterior predictive is equivalent, up to additive constants, to the information gain from future observations to parameters given actions:

$$\mathbb{E}_{p(y_t \mid u_t, \mathcal{D}_k)}\Big[-\ln p(y_t \mid u_t, \mathcal{D}_k)\Big] + C = \mathbb{E}_{p(y_t, \Theta \mid u_t, \mathcal{D}_k)}\Big[\ln \frac{p(y_t, \Theta \mid u_t, \mathcal{D}_k)}{p(y_t | u_t, \mathcal{D}_k) p(\Theta \mid \mathcal{D}_k)}\Big]$$

where the constants C are terms not involving u_t.

ARxI: information gain

Under the given model specification, we can derive an insightful interpretation:

Theorem

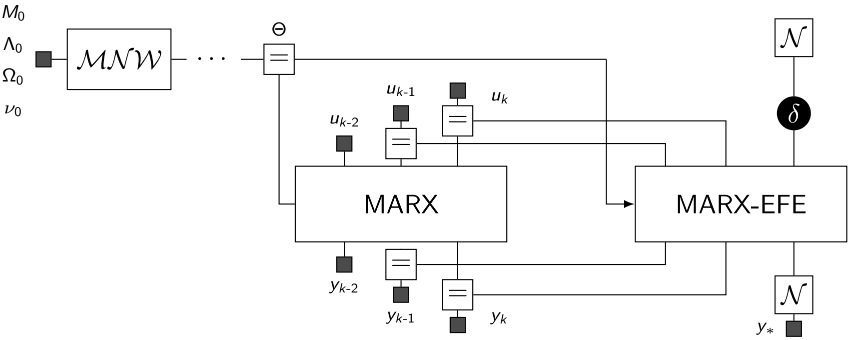
For particular p, q, the entropy of the posterior predictive is equivalent, up to additive constants, to the information gain from future observations to parameters given actions:

$$\mathbb{E}_{p(y_t \mid u_t, \mathcal{D}_k)}\big[-\ln p(y_t \mid u_t, \mathcal{D}_k)\big] + C = \mathbb{E}_{p(y_t, \Theta \mid u_t, \mathcal{D}_k)}\Big[\ln \frac{p(y_t, \Theta \mid u_t, \mathcal{D}_k)}{p(y_t|u_t, \mathcal{D}_k) p(\Theta \mid \mathcal{D}_k)} \Big]$$

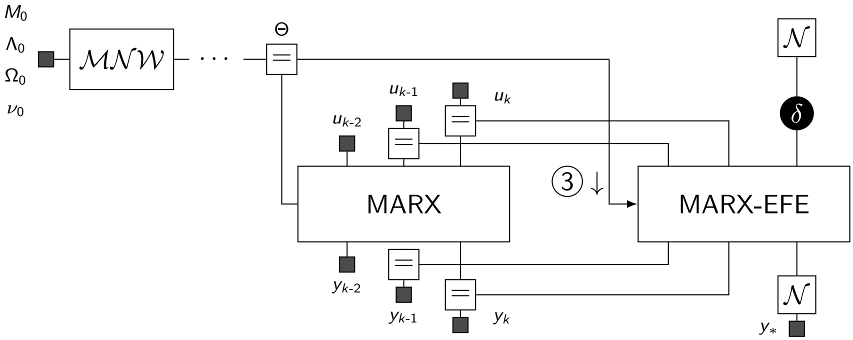
where the constants C are terms not involving u_t.

Here, maximizing predictive entropy, with respect to u_t, *is* maximizing information gain.

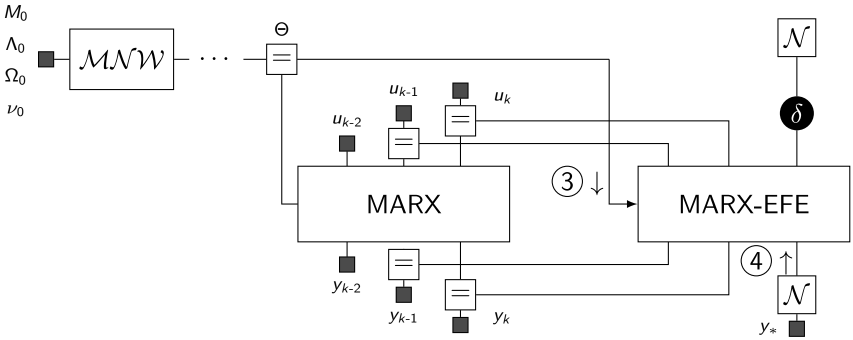
ARxI: 1-step ahead planning graph



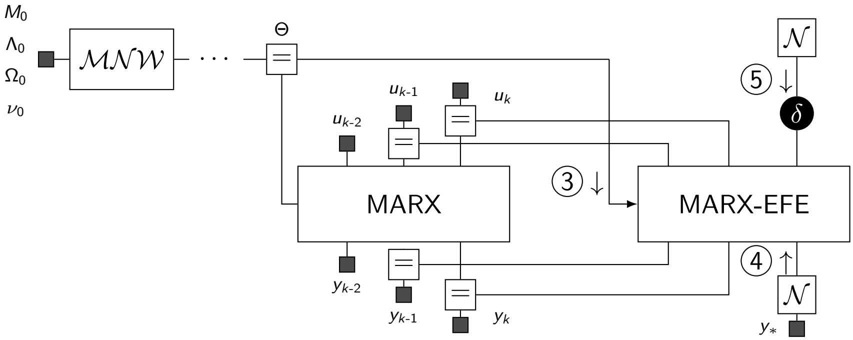
ARxI: 1-step ahead planning graph



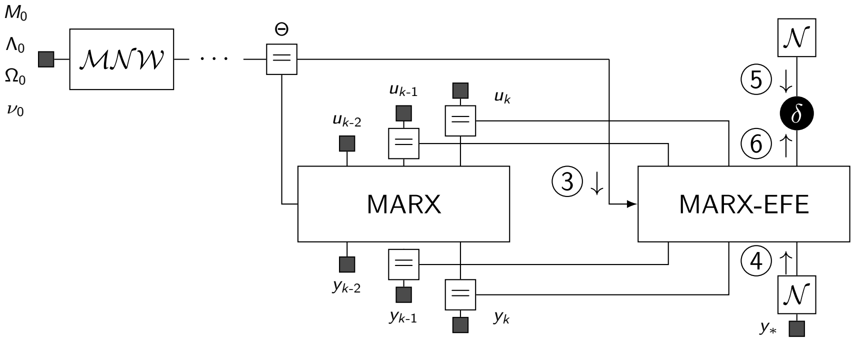
ARxI: 1-step ahead planning graph



ARxI: 1-step ahead planning graph



ARxI: 1-step ahead planning graph



ARxI: Extending the horizon

At $t \geq k + 2$, the buffers will contain latent instead of observed variables.

ARxI: Extending the horizon

At $t \geq k + 2$, the buffers will contain latent instead of observed variables.

Now, each node in the planning graph for $t \geq 2$ has to send backwards messages.

ARxI: Extending the horizon

At $t \geq k + 2$, the buffers will contain latent instead of observed variables.

Now, each node in the planning graph for $t \geq 2$ has to send backwards messages.

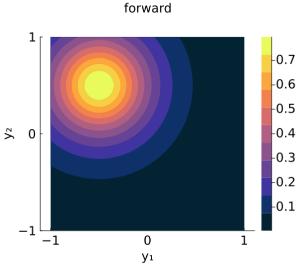
Forwards and backwards messages over y_t are combined to form marginals;

ARxI: Extending the horizon

At $t \geq k + 2$, the buffers will contain latent instead of observed variables.

Now, each node in the planning graph for $t \geq 2$ has to send backwards messages.

Forwards and backwards messages over y_t are combined to form marginals;

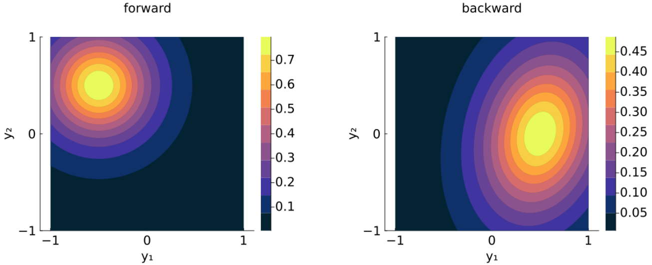


ARxI: Extending the horizon

At $t \geq k + 2$, the buffers will contain latent instead of observed variables.

Now, each node in the planning graph for $t \geq 2$ has to send backwards messages.

Forwards and backwards messages over y_t are combined to form marginals;

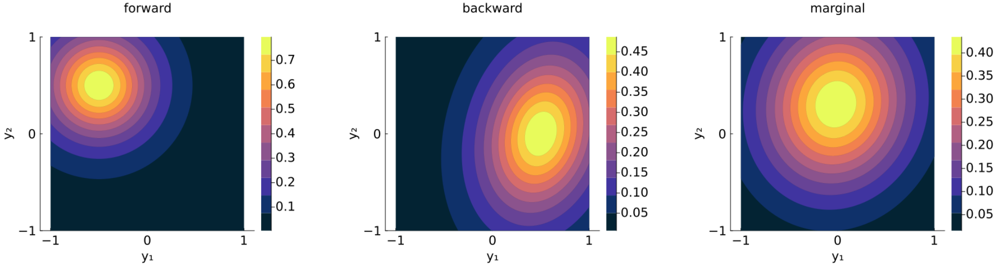


ARxI: Extending the horizon

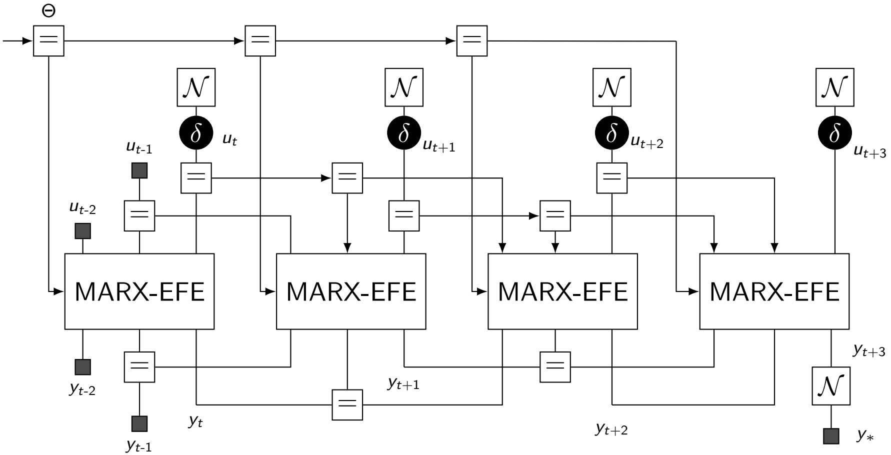
At $t \geq k+2$, the buffers will contain latent instead of observed variables.

Now, each node in the planning graph for $t \geq 2$ has to send backwards messages.

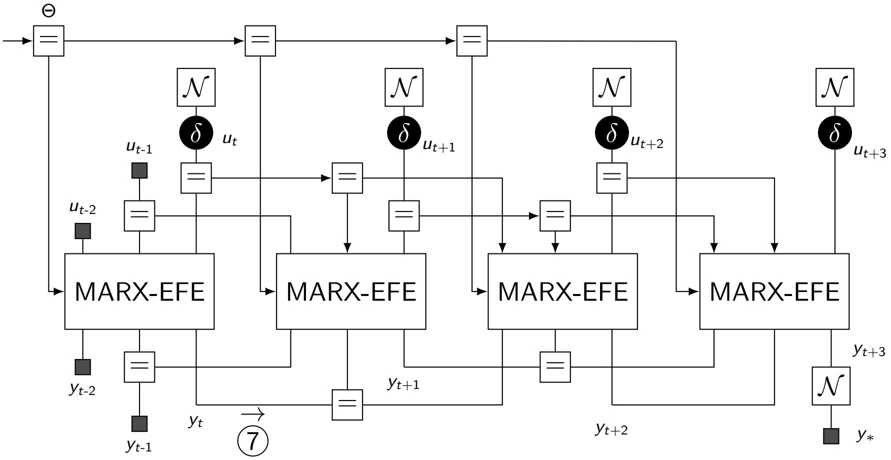
Forwards and backwards messages over y_t are combined to form marginals;



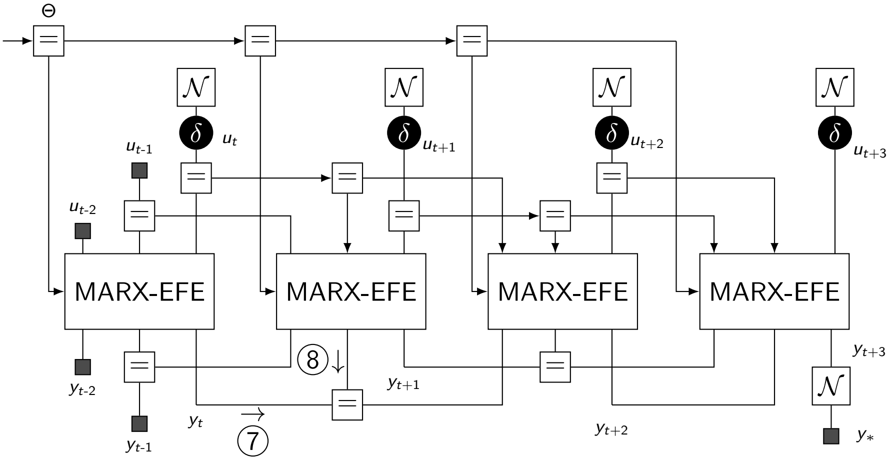
ARxI: T-step ahead planning



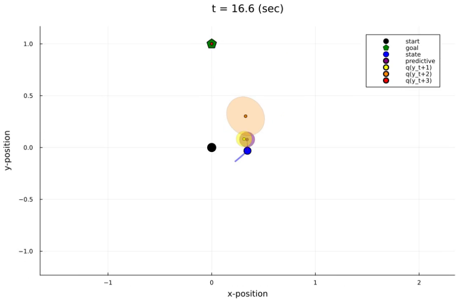
ARxI: T-step ahead planning

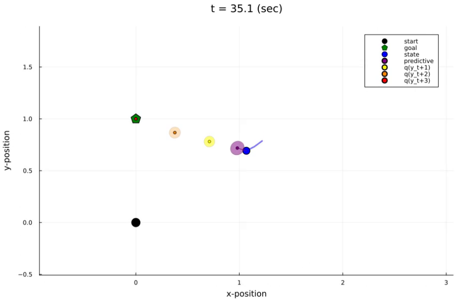


ARxI: T-step ahead planning



ARxI: Navigation trial





ARxI: T-step ahead planning

On the planning trajectory:

- The forward pass of predictions generates a trajectory, given prior actions.

ARxI: T-step ahead planning

On the planning trajectory:

- The forward pass of predictions generates a trajectory, given prior actions.
- The backwards pass from goal priors to previous time variables, gives direction.

ARxI: T-step ahead planning

On the planning trajectory:

- The forward pass of predictions generates a trajectory, given prior actions.
- The backwards pass from goal priors to previous time variables, gives direction.
- Together, the two passes generate a sequence of intermediate goal priors.

ARxI: T-step ahead planning

On the planning trajectory:

- The forward pass of predictions generates a trajectory, given prior actions.
- The backwards pass from goal priors to previous time variables, gives direction.
- Together, the two passes generate a sequence of intermediate goal priors.

ARxI: T-step ahead planning

On the planning trajectory:

- The forward pass of predictions generates a trajectory, given prior actions.
- The backwards pass from goal priors to previous time variables, gives direction.
- Together, the two passes generate a sequence of intermediate goal priors.

On the sequence of actions:

- Each planning node selects its own action based on balancing local info gain versus cross-entropy to target.

ARxI: T-step ahead planning

On the planning trajectory:

- The forward pass of predictions generates a trajectory, given prior actions.
- The backwards pass from goal priors to previous time variables, gives direction.
- Together, the two passes generate a sequence of intermediate goal priors.

On the sequence of actions:

- Each planning node selects its own action based on balancing local info gain versus cross-entropy to target.
- The ability to block backwards messages towards previous actions reduces computational cost, and improves scaling.

ARxI: T-step ahead planning

On the planning trajectory:

- The forward pass of predictions generates a trajectory, given prior actions.
- The backwards pass from goal priors to previous time variables, gives direction.
- Together, the two passes generate a sequence of intermediate goal priors.

On the sequence of actions:

- Each planning node selects its own action based on balancing local info gain versus cross-entropy to target.
- The ability to block backwards messages towards previous actions reduces computational cost, and improves scaling.
- Limitation: controls are not enforced to be smooth over time.

Comparing ARxI to MPC

Let's compare ARxI's controller with a standard model-predictive controller:

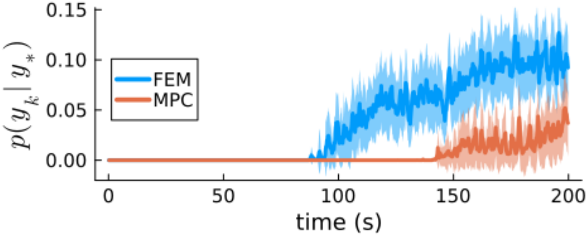
$$u_t^{\mathrm{MPC}} = \arg\min \; u_t^\intercal \Upsilon u_t + (\mu_t(u_t) - m_*)^\intercal S_*^{-1} (\mu_t(u_t) - m_*).$$

Comparing ARxI to MPC

Let's compare ARxI's controller with a standard model-predictive controller:

$$u_t^{\text{MPC}} = \arg\min \ u_t^\intercal \Upsilon u_t + (\mu_t(u_t) - m_*)^\intercal S_*^{-1} (\mu_t(u_t) - m_*).$$

ARxI learns its dynamics more quickly and is able to reach the goal sooner:



Mutual information

Working out the two entropic terms in the variational posterior gives:

Mutual information

Working out the two entropic terms in the variational posterior gives:

$$E(u_t) = \mathbb{E}_{p(y_t \mid u_t, \mathcal{D}_k)}\big[-\ln p(y_t \mid u_t, \mathcal{D}_k)\big] - \mathbb{E}_{p(y_t \mid u_t, \mathcal{D}_k)}\big[-\ln p(y_t \mid y_*)\big]$$

Mutual information

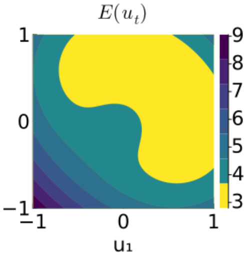
Working out the two entropic terms in the variational posterior gives:

$$
\begin{aligned}
E(u_t) &= \mathbb{E}_{p(y_t \mid u_t, \mathcal{D}_k)}\big[-\ln p(y_t \mid u_t, \mathcal{D}_k)\big] - \mathbb{E}_{p(y_t \mid u_t, \mathcal{D}_k)}\big[-\ln p(y_t \mid y_*)\big] \\
&= C + \underbrace{-\frac{1}{2}\ln|\Sigma_t(u_t)| + \frac{1}{2}\operatorname{Tr}\Big[S_*^{-1}\frac{\eta_t}{\eta_t - 2}\Sigma_t(u_t)\Big]}_{I(u_t)} + \underbrace{\frac{1}{2}(\mu_t(u_t) - m_*)^\intercal S_*^{-1}(\mu_t(u_t) - m_*)}_{G(u_t)} \ .
\end{aligned}
$$

Mutual information

Working out the two entropic terms in the variational posterior gives:

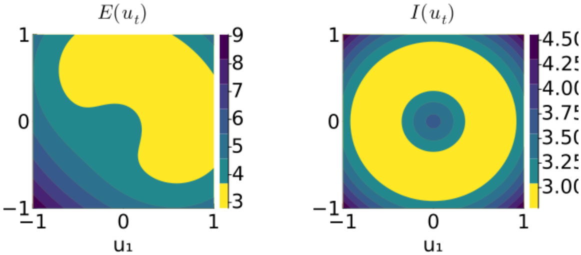
$$E(u_t) = \mathbb{E}_{p(y_t \mid u_t, \mathcal{D}_k)}\big[-\ln p(y_t \mid u_t, \mathcal{D}_k)\big] - \mathbb{E}_{p(y_t \mid u_t, \mathcal{D}_k)}\big[-\ln p(y_t \mid y_*)\big]$$
$$= C + \underbrace{-\frac{1}{2}\ln|\Sigma_t(u_t)| + \frac{1}{2}\mathrm{Tr}\Big[S_*^{-1}\frac{\eta_t}{\eta_t - 2}\Sigma_t(u_t)\Big]}_{I(u_t)} + \underbrace{\frac{1}{2}(\mu_t(u_t) - m_*)^\intercal S_*^{-1}(\mu_t(u_t) - m_*)}_{G(u_t)} \, .$$



Mutual information

Working out the two entropic terms in the variational posterior gives:

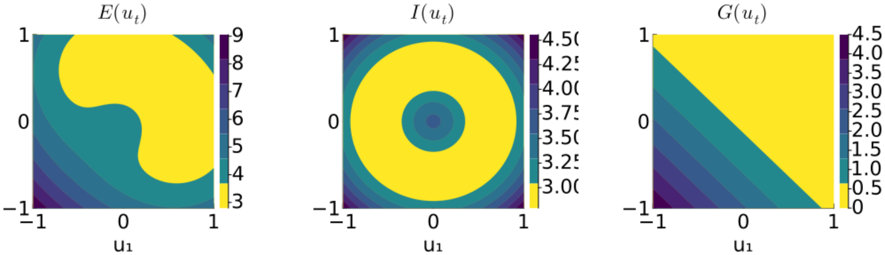
$$E(u_t) = \mathbb{E}_{p(y_t \mid u_t, \mathcal{D}_k)}\big[- \ln p(y_t \mid u_t, \mathcal{D}_k)\big] - \mathbb{E}_{p(y_t \mid u_t, \mathcal{D}_k)}\big[- \ln p(y_t \mid y_*)\big]$$
$$= C + \underbrace{-\frac{1}{2}\ln|\Sigma_t(u_t)| + \frac{1}{2}\mathrm{Tr}\Big[S_*^{-1}\frac{\eta_t}{\eta_t - 2}\Sigma_t(u_t)\Big]}_{I(u_t)} + \underbrace{\frac{1}{2}(\mu_t(u_t) - m_*)^\intercal S_*^{-1}(\mu_t(u_t) - m_*)}_{G(u_t)} \,.$$



Mutual information

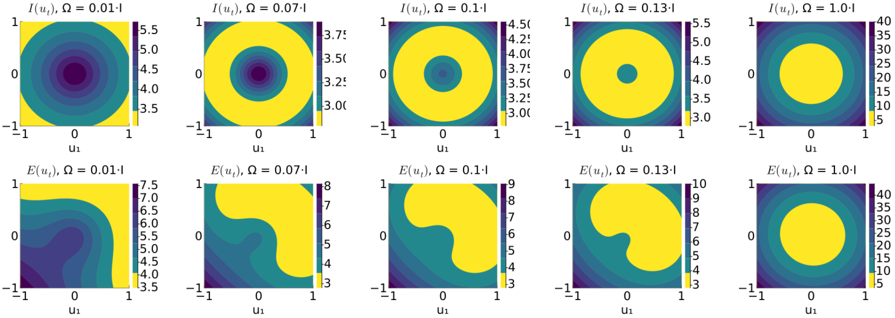
Working out the two entropic terms in the variational posterior gives:

$$
\begin{aligned}
E(u_t) &= \mathbb{E}_{p(y_t \mid u_t, \mathcal{D}_k)}\big[-\ln p(y_t \mid u_t, \mathcal{D}_k)\big] - \mathbb{E}_{p(y_t \mid u_t, \mathcal{D}_k)}\big[-\ln p(y_t \mid y_*)\big] \\
&= C + \underbrace{-\frac{1}{2}\ln|\Sigma_t(u_t)| + \frac{1}{2}\mathrm{Tr}\Big[S_*^{-1}\frac{\eta_t}{\eta_t - 2}\Sigma_t(u_t)\Big]}_{I(u_t)} + \underbrace{\frac{1}{2}(\mu_t(u_t) - m_*)^\intercal S_*^{-1}(\mu_t(u_t) - m_*)}_{G(u_t)} \, .
\end{aligned}
$$



Shape of information gain

Uncertainty affects the shape of the information gain, and the overall control posterior.



Controlling information gain

To be able to guide information gain, an agent must be able to *control* their predictive uncertainty, here $\Sigma_k(u_t)$.

Controlling information gain

To be able to guide information gain, an agent must be able to *control* their predictive uncertainty, here $\Sigma_k(u_t)$.

Perhaps obvious but note that some approximation techniques will drop controllable information gain.

Controlling information gain

To be able to guide information gain, an agent must be able to *control* their predictive uncertainty, here $\Sigma_k(u_t)$.

Perhaps obvious but note that some approximation techniques will drop controllable information gain.

For example, in deep reinforcement learning, one typically approximates expectations with sample averages, which leads to uncontrollable predictive entropy;

$$\int p(y_t \mid \Theta, u_t, \bar{u}_t, \bar{y}_t) p(\Theta \mid \mathcal{D}_k) \mathrm{d}\Theta \approx \frac{1}{n} \sum_{i=1}^{n} \mathcal{N}\Big(y_t \mid A_{(i)}^{\top} \begin{bmatrix} \bar{y}_t \\ u_t \\ \bar{u}_t \end{bmatrix}, W_{(i)}^{-1}\Big) \text{ for } A_{(i)}, W_{(i)} \sim p(\Theta \mid \mathcal{D}_k)\,.$$

Thanks for your attention

BIASlab

rxinfer

References

- Parr, Pezullo, Friston (2022), Active inference: the free energy principle in mind, brain, and behavior, MIT Press.
- Senoz et al. (2021), Variational message passing and local constraint manipulation in factor graphs, MDPI Entropy.
- Loeliger (2004), An introduction to factor graphs, IEEE Signal Processing Magazine.
- Nisslbeck & Kouw (2025), Online Bayesian system identification in multivariate autoregressive models via message passing, IEEE European Control Conference.
- Särkkä, Svensson (2023). Bayesian filtering & Smoothing. Cambridge University Press.
- Kouw et al. (2025), Message passing-based inference in an autoregressive active inference agent, International Workshop on Active Inference.
- Murphy (2022). Probabilistic machine learning: an introduction. MIT Press.

Tiny AI on robots