TECHNISCHE
UNIVERSITÄT
DARMSTADT

Approximate Input Inference as Stochastic Optimal Control

Joe Watson

Intelligent Autonomous Systems
TU Darmstadt
watson@ias.tu-darmstadt.de
@JoeMWatson, @ias_tudarmstadt

Motivation

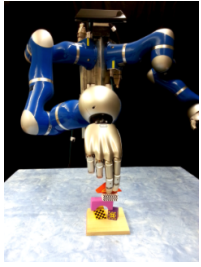


Figure 1: Object Manipulation

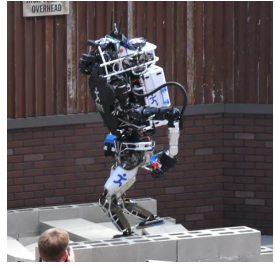


Figure 2: Full-body Locomotion

Motivation

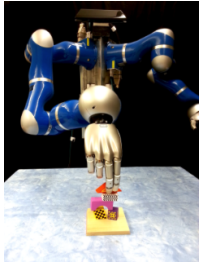


Figure 1: Object Manipulation

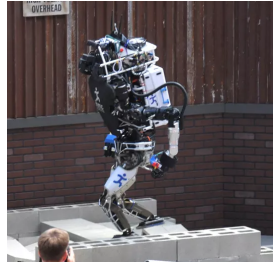


Figure 2: Full-body Locomotion

$$\min_{\mathbf{u}_{0:T-1}} C_T(\boldsymbol{x}_T) + \sum_{t=0}^{T-1} C(\boldsymbol{x}_t, \mathbf{u}_t)$$

$$\text{s.t. } \boldsymbol{x}_{t+1} = \mathbf{f}(\boldsymbol{x}_t, \mathbf{u}_t)$$

Motivation

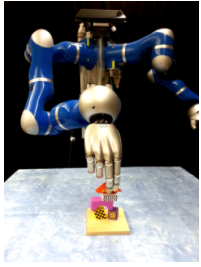


Figure 1: Object Manipulation

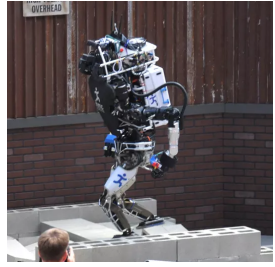


Figure 2: Full-body Locomotion

$$\min_{\mathbf{u}_{0:T-1}} C_T(\boldsymbol{x}_T) + \sum_{t=0}^{T-1} C(\boldsymbol{x}_t, \mathbf{u}_t)$$

$$\text{s.t. } \boldsymbol{x}_{t+1} = \mathbf{f}(\boldsymbol{x}_t, \mathbf{u}_t)$$

Motivation

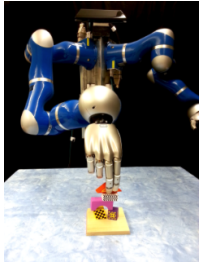


Figure 1: Object Manipulation

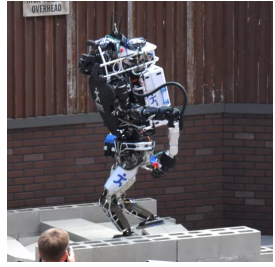


Figure 2: Full-body Locomotion

$$\min_{\mathbf{u}_{0:T-1}} C_T(\boldsymbol{x}_T) + \sum_{t=0}^{T-1} C(\boldsymbol{x}_t, \mathbf{u}_t)$$

$$\text{s.t. } \boldsymbol{x}_{t+1} \sim \mathbf{f}(\boldsymbol{x}_t, \mathbf{u}_t)$$

Motivation

Many successful Model-based Reinforcement Learning (MBRL) formulations combine probabilistic model learning with trajectory optimization:

Algorithm	Probabilistic Model	Optimizer	Weakness
PILCO[1]	GP	Backprop-through-time	Unstable gradients
AGP-iLQR[2]	GP	iLQR	Doesn't use uncertainty
Guided Policy Search[3]	GMM	Maximum Entropy iLQG	Heavy regularization
PETS[4]	NN Ensemble	CEM	Computational cost

[4] Marc Deisenroth and Carl E Rasmussen. "PILCO: A model-based and data-efficient approach to policy search". In: *Proceedings of the 28th International Conference on machine learning (ICML-11)*. 2011.

[4] Joschka Boedecker et al. "Approximate real-time optimal control based on sparse gaussian process models". In: *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. IEEE. 2014.

[4] Sergey Levine. "Motor skill learning with local trajectory methods". PhD thesis. Stanford University, 2014.

[4] Kurtland Chua et al. "Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models". In: *Advances in Neural Information Processing Systems 31*.

Probabilistic Numerics for Trajectory Optimization:

Probabilistic Numerics[5]

Numerical methods tackled as probabilistic inference.
The benefits are

- Uncertainty quantification
- Principled regularization
- Inclusion of prior knowledge through Priors

[5] Philipp Hennig, Michael A Osborne, and Mark Girolami. "Probabilistic numerics and uncertainty in computations". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* (2015).

Input Estimation

$$\max_{\mathbf{u}_{0:T-1},\boldsymbol{\theta}} P(\mathbf{u}_{0:T-1}, \boldsymbol{x}_{0:T}, \mathbf{y}_{0:T-1}, \boldsymbol{\theta}) = \max_{\mathbf{u}_{0:T-1},\boldsymbol{\theta}} P(\boldsymbol{x}_0) \prod_{t=0}^{T-1} \underbrace{P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta})}_{\text{Dynamics}} \prod_{t=0}^{T-1} \underbrace{P(\mathbf{y}_t|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta})}_{\text{Measurement}}$$

Input Estimation

$$\max_{\mathbf{u}_{0:T-1},\boldsymbol{\theta}} P(\mathbf{u}_{0:T-1}, \boldsymbol{x}_{0:T}, \mathbf{z}_{0:T-1}, \boldsymbol{\theta}) = \max_{\mathbf{u}_{0:T-1},\boldsymbol{\theta}} P(\boldsymbol{x}_0) \prod_{t=0}^{T-1} \underbrace{P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta})}_{\text{Dynamics}} \prod_{t=0}^{T-1} \underbrace{P(\mathbf{z}_t|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta})}_{\text{Control Cost}}$$

Input Inference for Control (`i2c`)

Find the inputs that maximize the expected likelihood of the goal states, given the dynamical system.

$$\max_{\mathbf{u}_{0:T-1}} P(\boldsymbol{x}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{z}_{0:T-1}, \boldsymbol{\theta}) = \max_{\mathbf{u}_{0:T-1}} P(\boldsymbol{x}_0) \prod_{t=0}^{T-1} P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta}) \prod_{t=0}^{T-1} P(\mathbf{z}_t|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta})$$

- Applies to finite horizon optimal control problems, moving from $\boldsymbol{x}_0$ to $\boldsymbol{x}_g$ over horizon T
- $\mathbf{z}$ represents the 'optimization states' and their desired trajectory (i.e. $\mathbf{z}_t = [\boldsymbol{x}_g \quad \mathbf{u}_g]^T$)
- The likelihood of $P(\mathbf{z}|\boldsymbol{x}, \mathbf{u})$ takes the role of the cost function in typical optimal control
- Conditional distribution $P(\mathbf{u}|\boldsymbol{x})$ becomes a stochastic control law

Input Inference for Control (`i2c`)

Find the inputs that maximize the expected likelihood of the goal states, given the dynamical system.

$$\max_{\mathbf{u}_{0:T-1}} P(\boldsymbol{x}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{z}_{0:T-1}, \boldsymbol{\theta}) = \max_{\mathbf{u}_{0:T-1}} P(\boldsymbol{x}_0) \prod_{t=0}^{T-1} P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta}) \prod_{t=0}^{T-1} P(\mathbf{z}_t|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta})$$

- Applies to finite horizon optimal control problems, moving from $\boldsymbol{x}_0$ to $\boldsymbol{x}_g$ over horizon T
- $\mathbf{z}$ represents the 'optimization states' and their desired trajectory (i.e. $\mathbf{z}_t = [\boldsymbol{x}_g \quad \mathbf{u}_g]^T$)
- The likelihood of $P(\mathbf{z}|\boldsymbol{x}, \mathbf{u})$ takes the role of the cost function in typical optimal control
- Conditional distribution $P(\mathbf{u}|\boldsymbol{x})$ becomes a stochastic control law

Input Inference for Control (`i2c`)

Find the inputs that maximize the expected likelihood of the goal states, given the dynamical system.

$$\max_{\mathbf{u}_{0:T-1}} P(\boldsymbol{x}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{z}_{0:T-1}, \boldsymbol{\theta}) = \max_{\mathbf{u}_{0:T-1}} P(\boldsymbol{x}_0) \prod_{t=0}^{T-1} P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta}) \prod_{t=0}^{T-1} P(\mathbf{z}_t|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta})$$

- Applies to finite horizon optimal control problems, moving from $\boldsymbol{x}_0$ to $\boldsymbol{x}_g$ over horizon T
- $\mathbf{z}$ represents the 'optimization states' and their desired trajectory (i.e. $\mathbf{z}_t = [\boldsymbol{x}_g \quad \mathbf{u}_g]^T$)
- The likelihood of $P(\mathbf{z}|\boldsymbol{x}, \mathbf{u})$ takes the role of the cost function in typical optimal control
- Conditional distribution $P(\mathbf{u}|\boldsymbol{x})$ becomes a stochastic control law

Input Inference for Control (`i2c`)

Find the inputs that maximize the expected likelihood of the goal states, given the dynamical system.

$$\max_{\mathbf{u}_{0:T-1}} P(\boldsymbol{x}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{z}_{0:T-1}, \boldsymbol{\theta}) = \max_{\mathbf{u}_{0:T-1}} P(\boldsymbol{x}_0) \prod_{t=0}^{T-1} P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta}) \prod_{t=0}^{T-1} P(\mathbf{z}_t|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta})$$

- Applies to finite horizon optimal control problems, moving from $\boldsymbol{x}_0$ to $\boldsymbol{x}_g$ over horizon T
- $\mathbf{z}$ represents the 'optimization states' and their desired trajectory (i.e. $\mathbf{z}_t = [\boldsymbol{x}_g \quad \mathbf{u}_g]^T$)
- The likelihood of $P(\mathbf{z}|\boldsymbol{x}, \mathbf{u})$ takes the role of the cost function in typical optimal control
- Conditional distribution $P(\mathbf{u}|\boldsymbol{x})$ becomes a stochastic control law

Input Inference for Control (`i2c`)

Find the inputs that maximize the expected likelihood of the goal states, given the dynamical system.

$$\max_{\mathbf{u}_{0:T-1}} P(\boldsymbol{x}_{0:T}, \mathbf{u}_{0:T-1}, \mathbf{z}_{0:T-1}, \boldsymbol{\theta}) = \max_{\mathbf{u}_{0:T-1}} P(\boldsymbol{x}_0) \prod_{t=0}^{T-1} P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta}) \prod_{t=0}^{T-1} P(\mathbf{z}_t|\boldsymbol{x}_t, \mathbf{u}_t, \boldsymbol{\theta})$$

- Applies to finite horizon optimal control problems, moving from $\boldsymbol{x}_0$ to $\boldsymbol{x}_g$ over horizon T
- $\mathbf{z}$ represents the 'optimization states' and their desired trajectory (i.e. $\mathbf{z}_t = [\boldsymbol{x}_g \quad \mathbf{u}_g]^T$)
- The likelihood of $P(\mathbf{z}|\boldsymbol{x}, \mathbf{u})$ takes the role of the cost function in typical optimal control
- Conditional distribution $P(\mathbf{u}|\boldsymbol{x})$ becomes a stochastic control law

Linear Gaussian Input Inference for Control

LQR :
$$C(\boldsymbol{x}_t, \mathbf{u}_t) = (\boldsymbol{x}_g - \boldsymbol{x}_t)^\intercal \mathbf{Q} (\boldsymbol{x}_g - \boldsymbol{x}_t) + (\mathbf{u}_g - \mathbf{u}_t)^\intercal \mathbf{R} (\mathbf{u}_g - \mathbf{u}_t) \tag{1}$$

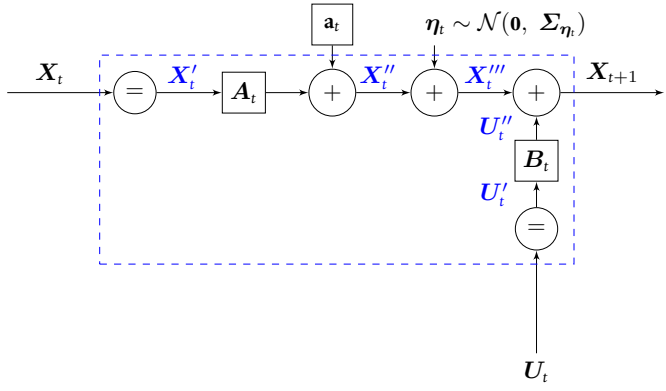
i2c :
$$\mathbf{z}_t = \mathbf{E}_t \boldsymbol{x}_t + \mathbf{F}_t \mathbf{u}_t + \mathbf{e}_t + \boldsymbol{\xi}_t \tag{2}$$

$$\boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0},\ \boldsymbol{\Sigma}_{\boldsymbol{\xi}}) \tag{3}$$

$$\mathbf{z}_t = \begin{bmatrix} \boldsymbol{x}_g \\ \mathbf{u}_g \end{bmatrix} \tag{4}$$

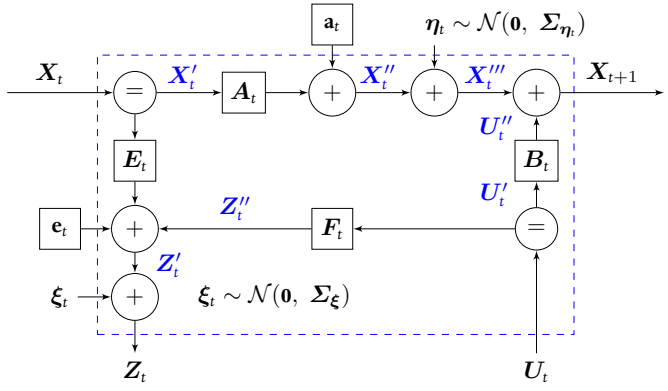
$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}} = \frac{1}{\alpha} \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}^{-1} = \frac{1}{\alpha} \boldsymbol{\Theta}^{-1} \tag{5}$$

The Graphical Model



Dynamics: $P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathbf{u}_t)$: $\boldsymbol{x}_{t+1} = \mathbf{A}_t\boldsymbol{x}_t + \mathbf{B}_t\mathbf{u}_t + \mathbf{a}_t + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0},\ \boldsymbol{\Sigma}_{\boldsymbol{\eta}_t}),$

The Graphical Model

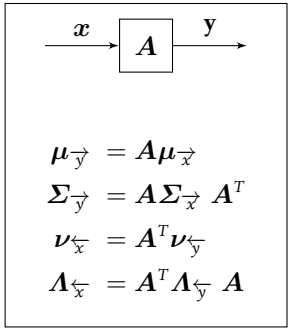


Dynamics: $P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mathbf{u}_t)$: $\boldsymbol{x}_{t+1} = \mathbf{A}_t\boldsymbol{x}_t + \mathbf{B}_t\mathbf{u}_t + \mathbf{a}_t + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0},\ \boldsymbol{\Sigma}_{\boldsymbol{\eta}_t}),$

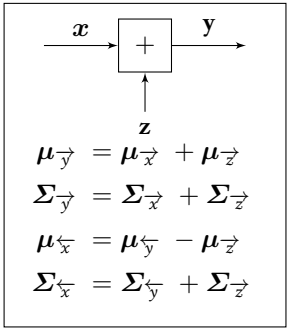
Cost: $P(\mathbf{z}_t|\boldsymbol{x}_t, \mathbf{u}_t)$: $\mathbf{z}_t = \mathbf{E}_t\boldsymbol{x}_t + \mathbf{F}_t\mathbf{u}_t + \mathbf{e}_t + \boldsymbol{\xi}_t, \quad \boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0},\ \boldsymbol{\Sigma}_{\boldsymbol{\xi}})$

Model-based Signal Processing through Message Passing

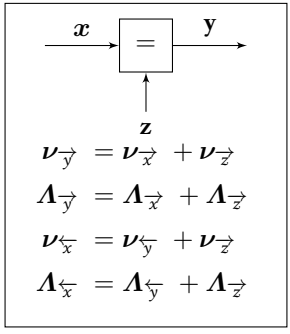
Linear Gaussian message passing rules are easily constructed from Factor Graph methods[6]:



(a) Linear Transform



(b) Addition



(c) Equality

Where $\boldsymbol{\nu}, \boldsymbol{\Lambda}$ denote the 'information' form: $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}, \ \boldsymbol{\nu} = \boldsymbol{\Lambda}\boldsymbol{\mu}$

[6] Hans-Andrea Loeliger et al. "The factor graph approach to model-based signal processing". In: *Proceedings of the IEEE* (2007).

Linear Gaussian Input Inference for Control

As with the LGDS, inference can be performed using EM:

E-Step: Linear Gaussian messsage passing to estimate $\boldsymbol{\mu}_{x_t}$, $\boldsymbol{\Sigma}_{x_t}$, $\boldsymbol{\mu}_{u_t}$, $\boldsymbol{\Sigma}_{u_t}$ from $\mathbf{z}_{0:T}$, $\boldsymbol{\mu}_{x_0}$ and $\boldsymbol{\Sigma}_{x_0}$.
Forward pass: construct priors, i.e $\boldsymbol{\mu}_{\overrightarrow{x}_t}$
Backward pass: improve likelihood, i.e. $\boldsymbol{\mu}_{\overleftarrow{x}_t}$
Marginalisation: calculuate posteriors, i.e. $\boldsymbol{\Sigma}_x = (\boldsymbol{\Lambda}_{\overrightarrow{x}} + \boldsymbol{\Lambda}_{\overleftarrow{x}})^{-1}$, $\boldsymbol{\mu}_x = \boldsymbol{\Sigma}_x (\boldsymbol{\nu}_{\overrightarrow{x}} + \boldsymbol{\nu}_{\overleftarrow{x}})$

M-Step: find $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$, therefore α, via the expected log-likelihood.
update priors on controls $\boldsymbol{\mu}_{\overrightarrow{u}_t}$, $\boldsymbol{\Sigma}_{\overrightarrow{u}_t}$

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} = \frac{1}{T} \sum_{t=0}^{T-1} \left[(\mathbf{z}_t - \mathbf{E}_t \boldsymbol{\mu}_{x_t} - \mathbf{F}_t \boldsymbol{\mu}_{u_t})(\mathbf{z}_t - \boldsymbol{E}_t \boldsymbol{\mu}_{x_t} - \mathbf{F}_t \boldsymbol{\mu}_{u_t})^\intercal + \boldsymbol{E}_t \boldsymbol{\Sigma}_{x_t} \mathbf{E}_t^\intercal + \mathbf{F}_t \boldsymbol{\Sigma}_{u_t} \mathbf{F}_t^\intercal \right] \tag{6}$$

$$\alpha = \frac{d_z}{\mathrm{tr}\{\boldsymbol{\Theta} \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\}} \tag{7}$$

As α increases as the trajectory gets closer to the goal state, the control cost becomes more important to the overall likelihood:

$$-\mathcal{L}(\boldsymbol{x}_{0:T-1}, \mathbf{u}_{0:T-1}, \alpha) = -\mathcal{L}_{\text{traj}}(\boldsymbol{x}_{0:T-1}, \mathbf{u}_{0:T-1}) + \frac{\alpha}{2} \sum_{t=0}^{T-1} C(\boldsymbol{x}_t, \mathbf{u}_t) + \ldots \tag{8}$$

Linear Gaussian Input Inference for Control

As with the LGDS, inference can be performed using EM:

E-Step: Linear Gaussian messsage passing to estimate $\boldsymbol{\mu}_{x_t}$, $\boldsymbol{\Sigma}_{x_t}$, $\boldsymbol{\mu}_{u_t}$, $\boldsymbol{\Sigma}_{u_t}$ from $\mathbf{z}_{0:T}$, $\boldsymbol{\mu}_{x_0}$ and $\boldsymbol{\Sigma}_{x_0}$.
Forward pass: construct priors, i.e $\boldsymbol{\mu}_{\overrightarrow{x}_t}$
Backward pass: improve likelihood, i.e. $\boldsymbol{\mu}_{\overleftarrow{x}_t}$
Marginalisation: calculuate posteriors, i.e. $\boldsymbol{\Sigma}_x = (\boldsymbol{\Lambda}_{\overrightarrow{x}} + \boldsymbol{\Lambda}_{\overleftarrow{x}})^{-1}$, $\boldsymbol{\mu}_x = \boldsymbol{\Sigma}_x (\boldsymbol{\nu}_{\overrightarrow{x}} + \boldsymbol{\nu}_{\overleftarrow{x}})$

M-Step: find $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$, therefore α, via the expected log-likelihood.
update priors on controls $\boldsymbol{\mu}_{\overrightarrow{u}_t}$, $\boldsymbol{\Sigma}_{\overrightarrow{u}_t}$

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} = \frac{1}{T}\sum_{t=0}^{T-1}\left[(\mathbf{z}_t - \mathbf{E}_t\boldsymbol{\mu}_{x_t} - \mathbf{F}_t\boldsymbol{\mu}_{u_t})(\mathbf{z}_t - \boldsymbol{E}_t\boldsymbol{\mu}_{x_t} - \mathbf{F}_t\boldsymbol{\mu}_{u_t})^\intercal + \boldsymbol{E}_t\boldsymbol{\Sigma}_{x_t}\mathbf{E}_t^\intercal + \mathbf{F}_t\boldsymbol{\Sigma}_{u_t}\mathbf{F}_t^\intercal\right] \tag{6}$$

$$\alpha = \frac{d_z}{\mathrm{tr}\{\boldsymbol{\Theta}\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\}} \tag{7}$$

As α increases as the trajectory gets closer to the goal state, the control cost becomes more important to the overall likelihood:

$$-\mathcal{L}(\boldsymbol{x}_{0:T-1}, \mathbf{u}_{0:T-1}, \alpha) = -\mathcal{L}_{\text{traj}}(\boldsymbol{x}_{0:T-1}, \mathbf{u}_{0:T-1}) + \frac{\alpha}{2}\sum_{t=0}^{T-1} C(\boldsymbol{x}_t, \mathbf{u}_t) + \dots \tag{8}$$

Linear Gaussian Input Inference for Control

As with the LGDS, inference can be performed using EM:

E-Step: Linear Gaussian messsage passing to estimate $\boldsymbol{\mu}_{x_t}$, $\boldsymbol{\Sigma}_{x_t}$, $\boldsymbol{\mu}_{u_t}$, $\boldsymbol{\Sigma}_{u_t}$ from $\mathbf{z}_{0:T}$, $\boldsymbol{\mu}_{x_0}$ and $\boldsymbol{\Sigma}_{x_0}$.
Forward pass: construct priors, i.e $\boldsymbol{\mu}_{\overrightarrow{x}_t}$
Backward pass: improve likelihood, i.e. $\boldsymbol{\mu}_{\overleftarrow{x}_t}$
Marginalisation: calculuate posteriors, i.e. $\boldsymbol{\Sigma}_x = (\boldsymbol{\Lambda}_{\overrightarrow{x}} + \boldsymbol{\Lambda}_{\overleftarrow{x}})^{-1}$, $\boldsymbol{\mu}_x = \boldsymbol{\Sigma}_x (\boldsymbol{\nu}_{\overrightarrow{x}} + \boldsymbol{\nu}_{\overleftarrow{x}})$

M-Step: find $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$, therefore α, via the expected log-likelihood.
update priors on controls $\boldsymbol{\mu}_{\overrightarrow{u}_t}$, $\boldsymbol{\Sigma}_{\overrightarrow{u}_t}$

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} = \frac{1}{T}\sum_{t=0}^{T-1}\left[(\mathbf{z}_t - \mathbf{E}_t\boldsymbol{\mu}_{x_t} - \mathbf{F}_t\boldsymbol{\mu}_{u_t})(\mathbf{z}_t - \boldsymbol{E}_t\boldsymbol{\mu}_{x_t} - \mathbf{F}_t\boldsymbol{\mu}_{u_t})^\intercal + \boldsymbol{E}_t\boldsymbol{\Sigma}_{x_t}\mathbf{E}_t^\intercal + \mathbf{F}_t\boldsymbol{\Sigma}_{u_t}\mathbf{F}_t^\intercal\right] \tag{6}$$

$$\alpha = \frac{d_z}{\mathrm{tr}\{\boldsymbol{\Theta}\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\}} \tag{7}$$

As α increases as the trajectory gets closer to the goal state, the control cost becomes more important to the overall likelihood:

$$-\mathcal{L}(\boldsymbol{x}_{0:T-1}, \mathbf{u}_{0:T-1}, \alpha) = -\mathcal{L}_{\text{traj}}(\boldsymbol{x}_{0:T-1}, \mathbf{u}_{0:T-1}) + \frac{\alpha}{2}\sum_{t=0}^{T-1} C(\boldsymbol{x}_t, \mathbf{u}_t) + \dots \tag{8}$$

Linear Gaussian Input Inference for Control

As with the LGDS, inference can be performed using EM:

E-Step: Linear Gaussian messsage passing to estimate $\boldsymbol{\mu}_{x_t}$, $\boldsymbol{\Sigma}_{x_t}$, $\boldsymbol{\mu}_{u_t}$, $\boldsymbol{\Sigma}_{u_t}$ from $\mathbf{z}_{0:T}$, $\boldsymbol{\mu}_{x_0}$ and $\boldsymbol{\Sigma}_{x_0}$.
Forward pass: construct priors, i.e $\boldsymbol{\mu}_{\overrightarrow{x}_t}$
Backward pass: improve likelihood, i.e. $\boldsymbol{\mu}_{\overleftarrow{x}_t}$
Marginalisation: calculuate posteriors, i.e. $\boldsymbol{\Sigma}_x = (\boldsymbol{\Lambda}_{\overrightarrow{x}} + \boldsymbol{\Lambda}_{\overleftarrow{x}})^{-1}$, $\boldsymbol{\mu}_x = \boldsymbol{\Sigma}_x (\boldsymbol{\nu}_{\overrightarrow{x}} + \boldsymbol{\nu}_{\overleftarrow{x}})$

M-Step: find $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$, therefore α, via the expected log-likelihood.
update priors on controls $\boldsymbol{\mu}_{\overrightarrow{u}_t}$, $\boldsymbol{\Sigma}_{\overrightarrow{u}_t}$

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} = \frac{1}{T}\sum_{t=0}^{T-1}\left[(\mathbf{z}_t - \mathbf{E}_t\boldsymbol{\mu}_{x_t} - \mathbf{F}_t\boldsymbol{\mu}_{u_t})(\mathbf{z}_t - \boldsymbol{E}_t\boldsymbol{\mu}_{x_t} - \mathbf{F}_t\boldsymbol{\mu}_{u_t})^\intercal + \boldsymbol{E}_t\boldsymbol{\Sigma}_{x_t}\mathbf{E}_t^\intercal + \mathbf{F}_t\boldsymbol{\Sigma}_{u_t}\mathbf{F}_t^\intercal\right] \tag{6}$$

$$\alpha = \frac{d_z}{\mathrm{tr}\{\boldsymbol{\Theta}\hat{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\}} \tag{7}$$

As α increases as the trajectory gets closer to the goal state, the control cost becomes more important to the overall likelihood:

$$-\mathcal{L}(\boldsymbol{x}_{0:T-1}, \mathbf{u}_{0:T-1}, \alpha) = -\mathcal{L}_{\text{traj}}(\boldsymbol{x}_{0:T-1}, \mathbf{u}_{0:T-1}) + \frac{\alpha}{2}\sum_{t=0}^{T-1} C(\boldsymbol{x}_t, \mathbf{u}_t) + \ldots \tag{8}$$

Experiment: Simple Linear System

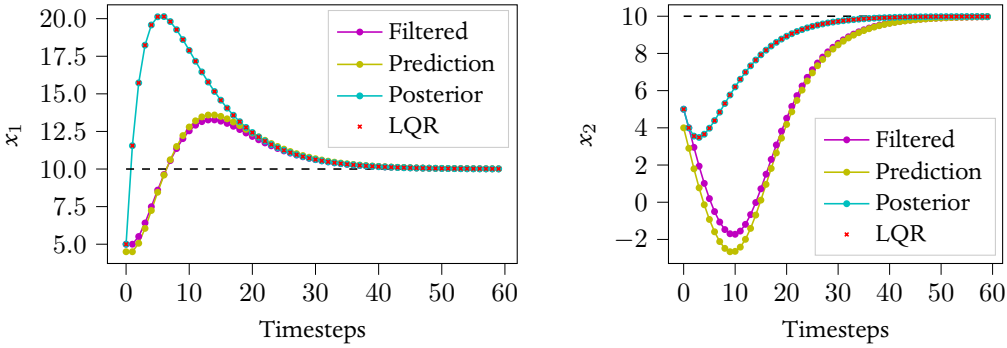


Figure 4: Demonstrating how `i2c` generalizes the Dynamic Programming Finite Horizon LQR trajectory. Note 'Filtered' and 'Prediction' correspond to $\boldsymbol{\mu}_{\overrightarrow{x}_t'}$ and $\boldsymbol{\mu}_{\overrightarrow{x}_{t+1}}$.

Linear Gaussian Assumption: DARE

Backward Propagation of value function weights in LQR

$$\boldsymbol{P}_t = \boldsymbol{Q} + \boldsymbol{A}^T \boldsymbol{P}_{t+1} \boldsymbol{A} - \boldsymbol{A}^T \boldsymbol{P}_{t+1} \boldsymbol{B} (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{A} \tag{9}$$

Backward Propagation of state uncertainty in `i2c`

$$\begin{aligned} \boldsymbol{\Lambda}_{\overleftarrow{x}_t} &= \boldsymbol{E}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathrm{F}_t \boldsymbol{\Sigma}_{\overrightarrow{u}_t} \mathrm{F}_t^\intercal)^{-1} \boldsymbol{E}_t + \boldsymbol{A}_t^\intercal \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{A}_t \\ &\quad - \boldsymbol{A}_t^\intercal \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} ((\boldsymbol{\Sigma}_{\boldsymbol{\eta}_t} + \boldsymbol{B}_t (\boldsymbol{\Lambda}_{\overrightarrow{u}_t} + \mathrm{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathrm{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathrm{E}_t^\intercal)^{-1} \mathrm{F}_t)^{-1} \boldsymbol{B}_t^\intercal)^{-1} + \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}})^{-1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{A}_t \end{aligned} \tag{10}$$

Linear Gaussian Assumption: DARE

Backward Propagation of value function weights in LQR

$$\boldsymbol{P}_t = \boldsymbol{Q} + \boldsymbol{A}^T \boldsymbol{P}_{t+1} \boldsymbol{A} - \boldsymbol{A}^T \boldsymbol{P}_{t+1} \boldsymbol{B} (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{A} \tag{9}$$

Backward Propagation of state uncertainty in `i2c`

$$\begin{aligned} \boldsymbol{\Lambda}_{\overleftarrow{x}_t} &= \boldsymbol{E}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{F}_t \boldsymbol{\Sigma}_{\overrightarrow{u}_t} \mathbf{F}_t^\intercal)^{-1} \boldsymbol{E}_t + \boldsymbol{A}_t^\intercal \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{A}_t \\ &\quad - \boldsymbol{A}_t^\intercal \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} ((\boldsymbol{\Sigma}_{\boldsymbol{\eta}_t} + \boldsymbol{B}_t (\boldsymbol{\Lambda}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} \mathbf{F}_t)^{-1} \boldsymbol{B}_t^\intercal)^{-1} + \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}})^{-1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{A}_t \end{aligned} \tag{10}$$

Linear Gaussian Assumption: DARE

Backward Propagation of value function weights in LQR

$$\boldsymbol{P}_t = \boldsymbol{Q} + \boldsymbol{A}^T \boldsymbol{P}_{t+1} \boldsymbol{A} - \boldsymbol{A}^T \boldsymbol{P}_{t+1} \boldsymbol{B} (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{A} \tag{9}$$

Backward Propagation of state uncertainty in `i2c`

$$\begin{aligned} \boldsymbol{\Lambda}_{\overleftarrow{x}_t} = & \boldsymbol{E}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{F}_t \boldsymbol{\Sigma}_{\overrightarrow{u}_t} \mathbf{F}_t^\intercal)^{-1} \boldsymbol{E}_t + \boldsymbol{A}_t^\intercal \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{A}_t \\ & - \boldsymbol{A}_t^\intercal \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} ((\boldsymbol{\Sigma}_{\boldsymbol{\eta}_t} + \boldsymbol{B}_t (\boldsymbol{\Lambda}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} \mathbf{F}_t)^{-1} \boldsymbol{B}_t^\intercal)^{-1} + \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}})^{-1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{A}_t \end{aligned} \tag{10}$$

Linear Gaussian Assumption: DARE

Backward Propagation of value function weights in LQR

$$\boldsymbol{P}_t = \boldsymbol{Q} + \boldsymbol{A}^T \boldsymbol{P}_{t+1} \boldsymbol{A} - \boldsymbol{A}^T \boldsymbol{P}_{t+1} \boldsymbol{B} (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{A} \tag{9}$$

Backward Propagation of state uncertainty in `i2c`

$$\begin{aligned} \boldsymbol{\Lambda}_{\overleftarrow{x}_t} &= \boldsymbol{E}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{F}_t \boldsymbol{\Sigma}_{\overrightarrow{u}_t} \mathbf{F}_t^\intercal)^{-1} \boldsymbol{E}_t + \boldsymbol{A}_t^\intercal \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{A}_t \\ &\quad - \boldsymbol{A}_t^\intercal \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} ((\boldsymbol{\Sigma}_{\boldsymbol{\eta}_t} + \boldsymbol{B}_t (\boldsymbol{\Lambda}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} \mathbf{F}_t)^{-1} \boldsymbol{B}_t^\intercal)^{-1} + \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}})^{-1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{A}_t \end{aligned} \tag{10}$$

Linear Gaussian Assumption: DARE

Backward Propagation of value function weights in LQR

$$\boldsymbol{P}_t = \boldsymbol{Q} + \boldsymbol{A}^T \boldsymbol{P}_{t+1} \boldsymbol{A} - \boldsymbol{A}^T \boldsymbol{P}_{t+1} \boldsymbol{B} (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{A} \tag{9}$$

Backward Propagation of state uncertainty in `i2c`

$$\begin{aligned} \boldsymbol{\Lambda}_{\overleftarrow{x}_t} &= \boldsymbol{E}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{F}_t \boldsymbol{\Sigma}_{\overrightarrow{u}_t} \mathbf{F}_t^\intercal)^{-1} \boldsymbol{E}_t + \boldsymbol{A}_t^\intercal \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{A}_t \\ &\quad - \boldsymbol{A}_t^\intercal \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} ((\boldsymbol{\Sigma}_{\boldsymbol{\eta}_t} + \boldsymbol{B}_t (\boldsymbol{\Lambda}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} \mathbf{F}_t)^{-1} \boldsymbol{B}_t^\intercal)^{-1} + \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}})^{-1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{A}_t \end{aligned} \tag{10}$$

Linear Gaussian Assumption: Controller

Like LQR, can `i2c` give us time-varying linear controllers?

The E-Step gave us the joint Normal distributions $P(\boldsymbol{x}_t, \mathbf{u}_t)$.
The conditional

$$P(\mathbf{u}_t|\boldsymbol{x}_t) : \boldsymbol{\mu}_{u_t|x_t} = \boldsymbol{\mu}_{u_t} + \boldsymbol{\Sigma}_{u_t x_t} \boldsymbol{\Sigma}^{-1}_{x_t x_t} (\boldsymbol{x}_t - \boldsymbol{\mu}_{x_t}), \tag{11}$$

has a mean linear is $\boldsymbol{x}$.

Therefore we get time-varying linear *Gaussian* controllers

$$P(\mathbf{u}_t|\boldsymbol{x}_t) = \mathcal{N}(\mathbf{K}_t \boldsymbol{x}_t + \mathbf{k}_t, \boldsymbol{\Sigma}_{k_t}) \tag{12}$$

Linear Gaussian Assumption: Controller

Maximum Entropy LQR:

$$\boldsymbol{K}_t = -(\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{A} \tag{13}$$

$$\mathbf{k}_t = (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} (\boldsymbol{B}^T (\boldsymbol{P}_{t+1} \mathbf{a}_t + \mathbf{p}_{t+1}) - \mathbf{R}\mathbf{u}_g) \tag{14}$$

$$\boldsymbol{\Sigma}_{k_t} = (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \tag{15}$$

Expressing $P(\mathbf{u}|\boldsymbol{x})$ through the messages:

$$\boldsymbol{K}_t = -\boldsymbol{\Sigma}_{u_t} \boldsymbol{B}_t \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{\Psi}_{t+1} \boldsymbol{A}_t, \tag{16}$$

$$\begin{aligned} \mathbf{k}_t = \boldsymbol{\Sigma}_{u_t} (\boldsymbol{\nu}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} (\mathbf{z}_t - \mathbf{E}_t \boldsymbol{\mu}_{\overrightarrow{x}_t} - \mathbf{e}_t) \\ + \boldsymbol{B}_t^\intercal (\boldsymbol{\Gamma}_{t+1} \boldsymbol{\nu}_{\overleftarrow{x}_{t+1}} + (\mathbf{I} - \boldsymbol{\Gamma}_{t+1}) \boldsymbol{\nu}_{\overrightarrow{x}''_t} - \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{\Psi}_{t+1} \mathbf{a}_t)), \end{aligned} \tag{17}$$

$$\boldsymbol{\Sigma}_{k_t} = \boldsymbol{\Sigma}_{u_t} = (\boldsymbol{\Lambda}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} \mathbf{F}_t + \boldsymbol{B}_t^\intercal \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{B}_t)^{-1} \tag{18}$$

Linear Gaussian Assumption: Controller

Maximum Entropy LQR:

$$\boldsymbol{K}_t = -(\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{A} \tag{13}$$

$$\mathbf{k}_t = (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} (\boldsymbol{B}^T (\boldsymbol{P}_{t+1} \mathbf{a}_t + \mathbf{p}_{t+1}) - \mathbf{R}\mathbf{u}_g) \tag{14}$$

$$\boldsymbol{\Sigma}_{k_t} = (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \tag{15}$$

Expressing $P(\mathbf{u}|\boldsymbol{x})$ through the messages:

$$\boldsymbol{K}_t = -\boldsymbol{\Sigma}_{u_t} \boldsymbol{B}_t \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{\Psi}_{t+1} \boldsymbol{A}_t, \tag{16}$$

$$\begin{aligned} \mathbf{k}_t = \boldsymbol{\Sigma}_{u_t} (& \boldsymbol{\nu}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} (\mathbf{z}_t - \mathbf{E}_t \boldsymbol{\mu}_{\overrightarrow{x}_t} - \mathbf{e}_t) \\ & + \boldsymbol{B}_t^\intercal (\boldsymbol{\Gamma}_{t+1} \boldsymbol{\nu}_{\overleftarrow{x}_{t+1}} + (\mathbf{I} - \boldsymbol{\Gamma}_{t+1}) \boldsymbol{\nu}_{\overrightarrow{x}''_t} - \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{\Psi}_{t+1} \mathbf{a}_t)), \end{aligned} \tag{17}$$

$$\boldsymbol{\Sigma}_{k_t} = \boldsymbol{\Sigma}_{u_t} = (\boldsymbol{\Lambda}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} \mathbf{F}_t + \boldsymbol{B}_t^\intercal \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{B}_t)^{-1} \tag{18}$$

Linear Gaussian Assumption: Controller

Maximum Entropy LQR:

$$\boldsymbol{K}_t = -(\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{A} \tag{13}$$

$$\mathbf{k}_t = (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} (\boldsymbol{B}^T (\boldsymbol{P}_{t+1} \mathbf{a}_t + \mathbf{p}_{t+1}) - \mathbf{R}\mathbf{u}_g) \tag{14}$$

$$\boldsymbol{\Sigma}_{k_t} = (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \tag{15}$$

Expressing $P(\mathbf{u}|\boldsymbol{x})$ through the messages:

$$\boldsymbol{K}_t = -\boldsymbol{\Sigma}_{u_t} \boldsymbol{B}_t \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{\Psi}_{t+1} \boldsymbol{A}_t, \tag{16}$$

$$\begin{aligned} \mathbf{k}_t = \boldsymbol{\Sigma}_{u_t} (& \boldsymbol{\nu}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} (\mathbf{z}_t - \mathbf{E}_t \boldsymbol{\mu}_{\overrightarrow{x}_t} - \mathbf{e}_t) \\ & + \boldsymbol{B}_t^\intercal (\boldsymbol{\Gamma}_{t+1} \boldsymbol{\nu}_{\overleftarrow{x}_{t+1}} + (\mathbf{I} - \boldsymbol{\Gamma}_{t+1}) \boldsymbol{\nu}_{\overrightarrow{x}''_t} - \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{\Psi}_{t+1} \mathbf{a}_t)), \end{aligned} \tag{17}$$

$$\boldsymbol{\Sigma}_{k_t} = \boldsymbol{\Sigma}_{u_t} = (\boldsymbol{\Lambda}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} \mathbf{F}_t + \boldsymbol{B}_t^\intercal \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{B}_t)^{-1} \tag{18}$$

Linear Gaussian Assumption: Controller

Maximum Entropy LQR:

$$\boldsymbol{K}_t = -(\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{A} \tag{13}$$

$$\mathbf{k}_t = (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} (\boldsymbol{B}^T (\boldsymbol{P}_{t+1} \mathbf{a}_t + \mathbf{p}_{t+1}) - \mathbf{R}\mathbf{u}_g) \tag{14}$$

$$\boldsymbol{\Sigma}_{k_t} = (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \tag{15}$$

Expressing $P(\mathbf{u}|\boldsymbol{x})$ through the messages:

$$\boldsymbol{K}_t = -\boldsymbol{\Sigma}_{u_t} \boldsymbol{B}_t \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{\Psi}_{t+1} \boldsymbol{A}_t, \tag{16}$$

$$\begin{aligned} \mathbf{k}_t = \boldsymbol{\Sigma}_{u_t} (& \boldsymbol{\nu}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} (\mathbf{z}_t - \mathbf{E}_t \boldsymbol{\mu}_{\overrightarrow{x}_t} - \mathbf{e}_t) \\ & + \boldsymbol{B}_t^\intercal (\boldsymbol{\Gamma}_{t+1} \boldsymbol{\nu}_{\overleftarrow{x}_{t+1}} + (\mathbf{I} - \boldsymbol{\Gamma}_{t+1}) \boldsymbol{\nu}_{\overrightarrow{x}''_t} - \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{\Psi}_{t+1} \mathbf{a}_t)), \end{aligned} \tag{17}$$

$$\boldsymbol{\Sigma}_{k_t} = \boldsymbol{\Sigma}_{u_t} = (\boldsymbol{\Lambda}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} \mathbf{F}_t + \boldsymbol{B}_t^\intercal \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{B}_t)^{-1} \tag{18}$$

Linear Gaussian Assumption: Controller

Maximum Entropy LQR:

$$\boldsymbol{K}_t = -(\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{A} \tag{13}$$

$$\mathbf{k}_t = (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} (\boldsymbol{B}^T (\boldsymbol{P}_{t+1} \mathbf{a}_t + \mathbf{p}_{t+1}) - \mathbf{R}\mathbf{u}_g) \tag{14}$$

$$\boldsymbol{\Sigma}_{k_t} = (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \tag{15}$$

Expressing $P(\mathbf{u}|\boldsymbol{x})$ through the messages:

$$\boldsymbol{K}_t = -\boldsymbol{\Sigma}_{u_t} \boldsymbol{B}_t \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{\Psi}_{t+1} \boldsymbol{A}_t, \tag{16}$$

$$\begin{aligned} \mathbf{k}_t = \boldsymbol{\Sigma}_{u_t} (\boldsymbol{\nu}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} (\mathbf{z}_t - \mathbf{E}_t \boldsymbol{\mu}_{\overrightarrow{x}_t} - \mathbf{e}_t) \\ + \boldsymbol{B}_t^\intercal (\boldsymbol{\Gamma}_{t+1} \boldsymbol{\nu}_{\overleftarrow{x}_{t+1}} + (\mathbf{I} - \boldsymbol{\Gamma}_{t+1}) \boldsymbol{\nu}_{\overrightarrow{x}_t''} - \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{\Psi}_{t+1} \mathbf{a}_t)), \end{aligned} \tag{17}$$

$$\boldsymbol{\Sigma}_{k_t} = \boldsymbol{\Sigma}_{u_t} = (\boldsymbol{\Lambda}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} \mathbf{F}_t + \boldsymbol{B}_t^\intercal \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{B}_t)^{-1} \tag{18}$$

Linear Gaussian Assumption: Controller

Maximum Entropy LQR:

$$\boldsymbol{K}_t = -(\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{A} \tag{13}$$

$$\mathbf{k}_t = (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} (\boldsymbol{B}^T (\boldsymbol{P}_{t+1} \mathbf{a}_t + \mathbf{p}_{t+1}) - \mathbf{R}\mathbf{u}_g) \tag{14}$$

$$\boldsymbol{\Sigma}_{k_t} = (\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{P}_{t+1} \boldsymbol{B})^{-1} \tag{15}$$

Expressing $P(\mathbf{u}|\boldsymbol{x})$ through the messages:

$$\boldsymbol{K}_t = -\boldsymbol{\Sigma}_{u_t} \boldsymbol{B}_t \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{\Psi}_{t+1} \boldsymbol{A}_t, \tag{16}$$

$$\begin{aligned} \mathbf{k}_t = \boldsymbol{\Sigma}_{u_t} (\boldsymbol{\nu}_{\overrightarrow{u}_t} &+ \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} (\mathbf{z}_t - \mathbf{E}_t \boldsymbol{\mu}_{\overrightarrow{x}_t} - \mathbf{e}_t) \\ &+ \boldsymbol{B}_t^\intercal (\boldsymbol{\Gamma}_{t+1} \boldsymbol{\nu}_{\overleftarrow{x}_{t+1}} + (\mathbf{I} - \boldsymbol{\Gamma}_{t+1}) \boldsymbol{\nu}_{\overrightarrow{x}_t''} - \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{\Psi}_{t+1} \mathbf{a}_t)), \end{aligned} \tag{17}$$

$$\boldsymbol{\Sigma}_{k_t} = \boldsymbol{\Sigma}_{u_t} = (\boldsymbol{\Lambda}_{\overrightarrow{u}_t} + \mathbf{F}_t^\intercal (\boldsymbol{\Sigma}_{\boldsymbol{\xi}} + \mathbf{E}_t \boldsymbol{\Sigma}_{\overrightarrow{x}_t} \mathbf{E}_t^\intercal)^{-1} \mathbf{F}_t + \boldsymbol{B}_t^\intercal \boldsymbol{\Gamma}_{t+1} \boldsymbol{\Lambda}_{\overleftarrow{x}_{t+1}} \boldsymbol{B}_t)^{-1} \tag{18}$$

Experiment: Simple Linear System

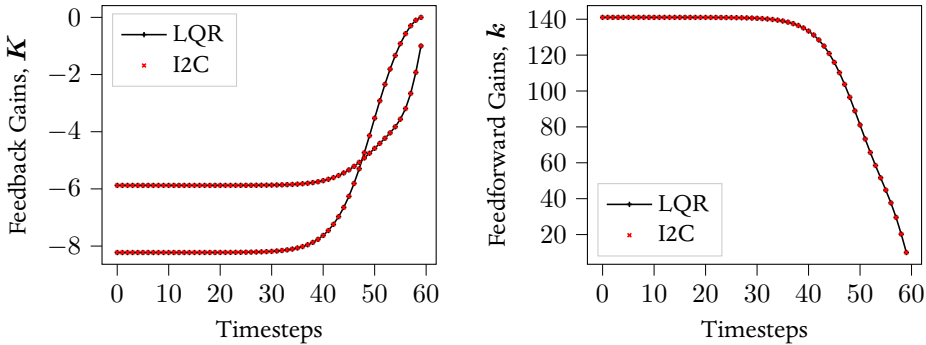
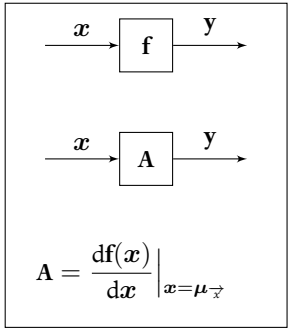


Figure 5: Demonstrating how `i2c` generalizes the Dynamic Programming Finite Horizon LQR controller.

i2c for Nonlinear Systems



Taking inspiration from the Extended Kalman Filter, we **linearize the dynamics** about the state action trajectory during the forward pass.

However, this local linearization assumption **reduces the amount of uncertainty in the trajectory we can tolerate**.

As the expected log-likelihood in the M-step is now approximate, we must regularize of the $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}(\alpha)$ update. Starting with a KL bound, this can be practically implemented as a bound of the update ratio:

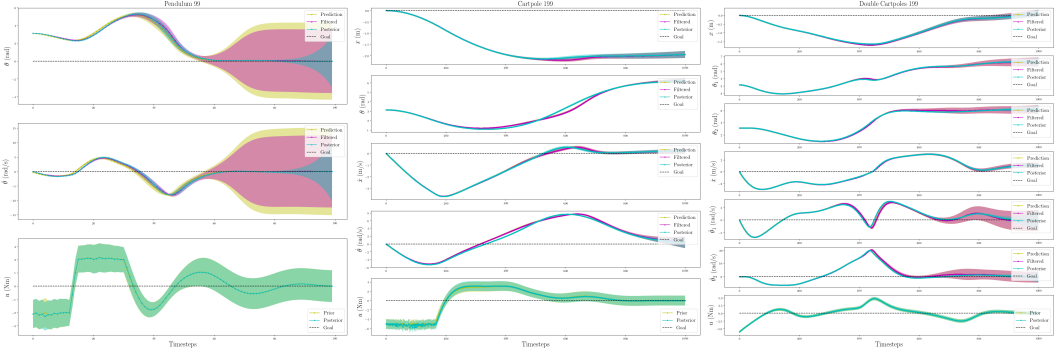
$$\frac{\alpha^i}{\alpha^{i+1}} \leq \delta_\alpha$$

The Algorithm

Data: Task:$\{T, \mathbf{z}_g, \boldsymbol{\Theta}, \boldsymbol{\mu}_{\overrightarrow{x}_0}, \mathbf{f}(\boldsymbol{x}, \mathbf{u}), \mathbf{g}(\boldsymbol{x}, \mathbf{u})\}$, Priors:$\{\boldsymbol{\Sigma}_{\overrightarrow{x}_0}, \boldsymbol{\mu}_{\overrightarrow{u}_t}, \boldsymbol{\Sigma}_{\overrightarrow{u}_t}\}$, Hyperparameters:$\{\alpha, \delta_\alpha\}$
Result: $\mathbf{K}_t, \mathbf{k}_t$, for $t = 0 : T\text{-}1$
while *not converged* **do**
  `// E-Step`
  for $i \leftarrow 0$ **to** $T - 1$ **do**
    Compute $\boldsymbol{\mu}_{\overrightarrow{x}_{t+1}}, \boldsymbol{\Sigma}_{\overrightarrow{x}_{t+1}}$ from forward messages, updating $\mathbf{A}_t$, $\mathbf{a}_t$,$\mathbf{B}_t$,$\mathbf{E}_t$, $\mathbf{e}_t$ and $\mathbf{F}_t$
  end
  for $i \leftarrow T$ **to** 1 **do**
    Compute $\boldsymbol{\mu}_{x_t}, \boldsymbol{\Sigma}_{x_t}$ $\boldsymbol{\mu}_{u_t}, \boldsymbol{\Sigma}_{u_t}$ from backward messages and marginalisation
  end
  `// M-Step`
  Update α with regularization, update priors $\boldsymbol{\mu}_{\overrightarrow{u}_{0:T}} = \boldsymbol{\mu}_{u_{0:T}}$, $\boldsymbol{\Sigma}_{\overrightarrow{u}_{0:T}} = \boldsymbol{\Sigma}_{u_{0:T}}$
end
`// Controller`
Computer linear Gaussian controller $\mathbf{K}_t, \mathbf{k}_t, \boldsymbol{\Sigma}_{k_t}$ for $t = 0{:}T\text{-}1$ from messages

Algorithm 1: EM for Linearized Gaussian `i2c`

Experiment: Simulated Environments



Experiment: Simulated Environments

We compared `i2c` against Iterative LQR (iLQR)[7] and Guided Policy Search (GPS)[8]

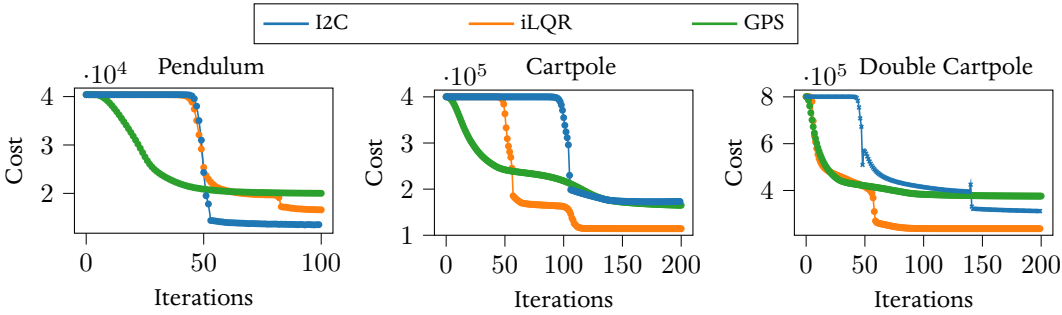


Figure 6: Comparison of the trajectory cost prediction over iterations during trajectory optimization.

[7] Weiwei Li and Emanuel Todorov. "Iterative linear quadratic regulator design for nonlinear biological movement systems.". In: *ICINCO (1)*. 2004.

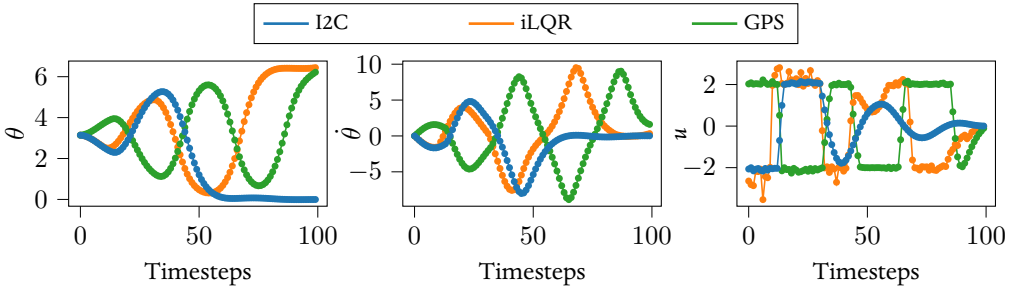
[8] Sergey Levine and Vladlen Koltun. "Guided Policy Search". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. 2013.

Experiment: Simulated Environments

Environment	Algorithm	Predicted Cost	Evaluated Cost
Pendulum	i2c	$\mathbf{1.35 \times 10^4}$	$\mathbf{1.37 \times 10^4 \pm 3.82}$
	iLQR	1.66×10^3	$1.11 \times 10^5 \pm 20.38$
	GPS	2.00×10^4	$7.01 \times 10^4 \pm 30.96$
Cartpole	i2c	1.73×10^5	$\mathbf{1.74 \times 10^5 \pm 0.14}$
	iLQR	$\mathbf{1.14 \times 10^5}$	$1.76 \times 10^7 \pm 88.63$
	GPS	1.65×10^5	$2.94 \times 10^6 \pm 17.60$
Double Cartpole	i2c	3.12×10^5	$\mathbf{3.21 \times 10^5 \pm 1.79}$
	iLQR	$\mathbf{2.37 \times 10^5}$	$1.76 \times 10^7 \pm 5.27 \times 10^5$
	GPS	3.76×10^5	$2.94 \times 10^6 \pm 44.39$

Table 1: Evaluating the optimized deterministic controller of each algorithm on the simulated stochastic environments. Predicted Cost refers to the converged value from Figure 6, Evaluated Cost shows the mean and standard deviation after 100 trials.

Experiment: Simulated Environments



Experiment: Simulated Environments

Animations

`i2c Pendulum`

`iLQR Pendulum`

Conclusion

- **LQR** can be framed as input inference of a **linear Gaussian dynamical system**
- `i2c` represents a practical **EM algorithm** for performing Bayesian nonlinear trajectory optimization through **message passing** and approximate inference, returning time-varying linear controllers
- Through use of Bayesian inference, the scheme naturally exhibits **uncertainty-derived regularization** that has been explicitly incorporated into (or lacking in) previous methods

Conclusion

- **LQR** can be framed as input inference of a **linear Gaussian dynamical system**
- `i2c` represents a practical **EM algorithm** for performing Bayesian nonlinear trajectory optimization through **message passing** and approximate inference, returning time-varying linear controllers
- Through use of Bayesian inference, the scheme naturally exhibits **uncertainty-derived regularization** that has been explicitly incorporated into (or lacking in) previous methods

Conclusion

- **LQR** can be framed as input inference of a **linear Gaussian dynamical system**
- `i2c` represents a practical **EM algorithm** for performing Bayesian nonlinear trajectory optimization through **message passing** and approximate inference, returning time-varying linear controllers
- Through use of Bayesian inference, the scheme naturally exhibits **uncertainty-derived regularization** that has been explicitly incorporated into (or lacking in) previous methods

Questions?

Boedecker, Joschka et al. "Approximate real-time optimal control based on sparse gaussian process models". In: *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. IEEE. 2014.

Chua, Kurtland et al. "Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models". In: *Advances in Neural Information Processing Systems 31*.

Deisenroth, Marc and Carl E Rasmussen. "PILCO: A model-based and data-efficient approach to policy search". In: *Proceedings of the 28th International Conference on machine learning (ICML-11)*. 2011.

Hennig, Philipp, Michael A Osborne, and Mark Girolami. "Probabilistic numerics and uncertainty in computations". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* (2015).

Hoffmann, Christian and Philipp Rostalski. "Linear Optimal Control on Factor Graphs - A Message Passing Perspective -". In: *IFAC (International Federation of Automatic Control)* (2017).

Levine, Sergey. "Motor skill learning with local trajectory methods". PhD thesis. Stanford University, 2014.

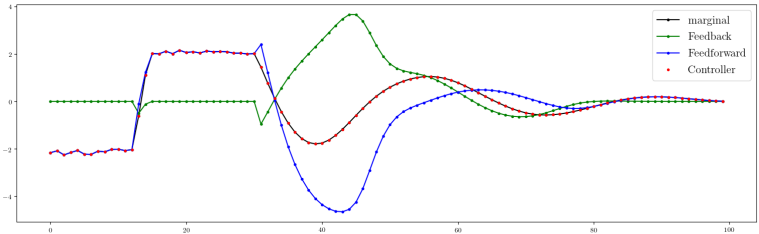
Levine, Sergey and Vladlen Koltun. "Guided Policy Search". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. 2013.

Li, Weiwei and Emanuel Todorov. "Iterative linear quadratic regulator design for nonlinear biological movement systems.". In: *ICINCO (1)*. 2004.

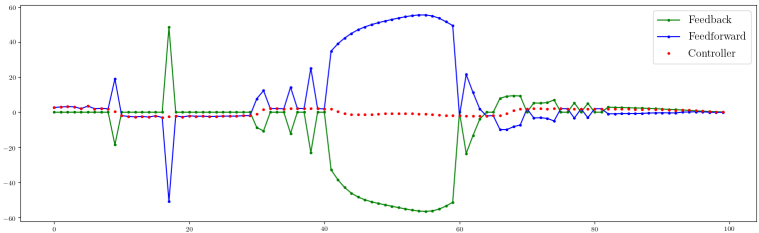
Loeliger, Hans-Andrea et al. "The factor graph approach to model-based signal processing". In: *Proceedings of the IEEE* (2007).

Toussaint, Marc. "Robot trajectory optimization using approximate inference". In: *Proceedings of the 26th annual international conference on machine learning*. ACM. 2009.

Experiment: Simulated Environments



(a) `i2c` Controller



(b) iLQR Controller

Previous Work

	Controller?	Adaptive Input Uncertainty?	Nonlinear?	Uncalibrated Cost?
AICO[9]	✗	✗	✓	✗
Hoffmann *et al*[10]	✓	✗	✗	✗
`i2c`	✓	✓	✓	✓

[9] Marc Toussaint. "Robot trajectory optimization using approximate inference". In: *Proceedings of the 26th annual international conference on machine learning*. ACM. 2009.

[10] Christian Hoffmann and Philipp Rostalski. "Linear Optimal Control on Factor Graphs - A Message Passing Perspective -". In: *IFAC (International Federation of Automatic Control)* (2017).

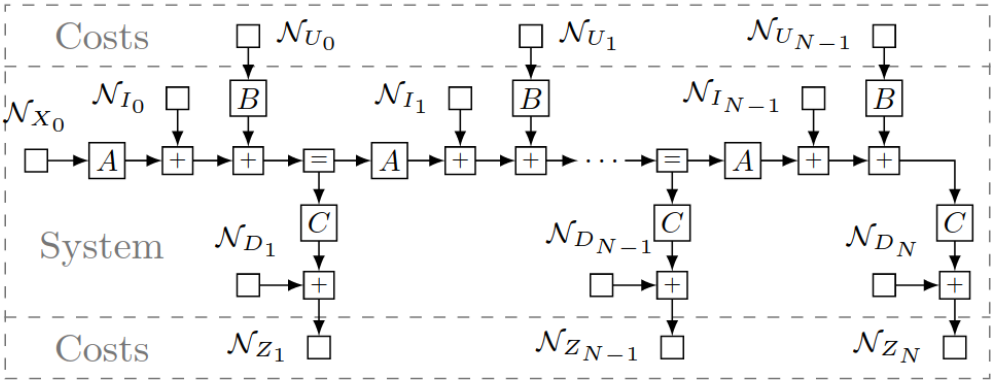


Fig. 3. Factor graph for LQG control.