Model-Based and Model-Free Imitation Learning

Hakan Girgin

25.09.2020

RESEARCH INSTITUTE

Learning from demonstration (LfD) in robotics

$\boldsymbol{x}_t$: state (e.g. position and velocity) of the robot at timestep t

$\boldsymbol{u}_t$: control command applied by the robot at timestep t

$\boldsymbol{s}$: context variable(s) such as initial/final positions, user preferences and environment properties.

Dataset: $\mathcal{D} = \{\boldsymbol{x}_t, \boldsymbol{u}_t\}_{t=0}^{T}, \boldsymbol{s}$

Trajectory level abstraction

$$p(\boldsymbol{\tau} | \boldsymbol{x}_0, \boldsymbol{s})$$

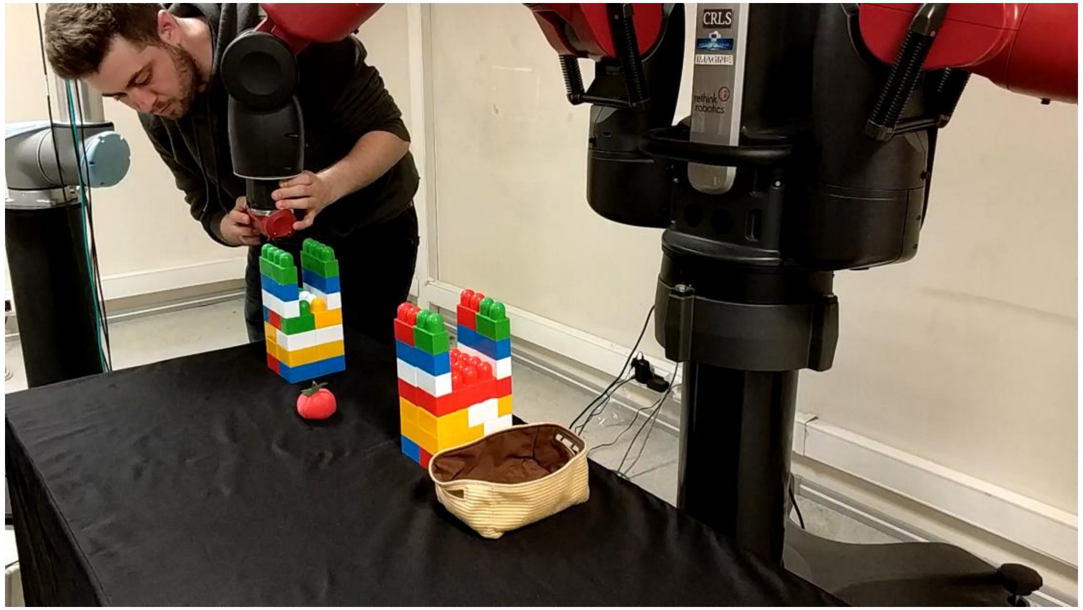
State level abstraction

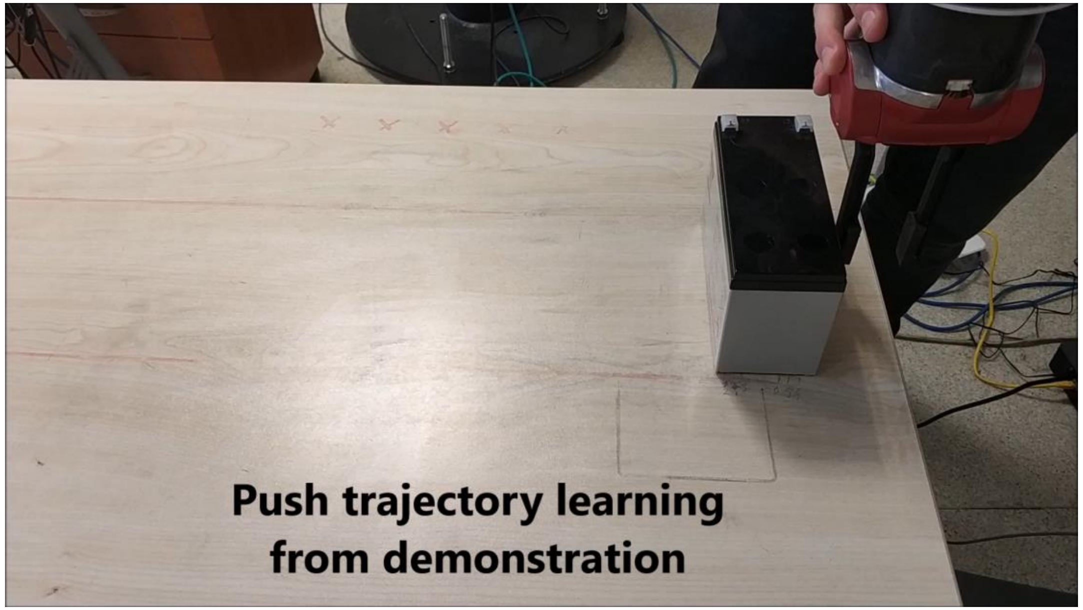
$$p(\boldsymbol{x}_t | t, \boldsymbol{s}), \quad p(\boldsymbol{x}_{t+1} | \boldsymbol{x}_t, \boldsymbol{s})$$

Action-State abstraction

$$p(\boldsymbol{u}_t | \boldsymbol{x}_t, \boldsymbol{s})$$

- DMP
- GMM-GMR
- HMM-LQR
- SEDS
- ProMP
- End-to-end deep visuomotor
- KMP
- BGMM





Learning from demonstration (LfD) in robotics

$\boldsymbol{x}_t$: state (e.g. position and velocity) of the robot at timestep t
$\boldsymbol{u}_t$: control command applied by the robot at timestep t
$\boldsymbol{s}$: context variable(s) such as initial/final positions, user preferences and environment properties.

Dataset: $\mathcal{D} = \{\boldsymbol{x}_t, \boldsymbol{u}_t\}_{t=0}^{T}, \boldsymbol{s}$

Trajectory level abstraction

$$p(\boldsymbol{\tau} | \boldsymbol{x}_0, \boldsymbol{s})$$

State level abstraction

$$p(\boldsymbol{x}_t | t, \boldsymbol{s}) \,, \quad p(\boldsymbol{x}_{t+1} | \boldsymbol{x}_t, \boldsymbol{s})$$

Action-State abstraction

$$p(\boldsymbol{u}_t | \boldsymbol{x}_t, \boldsymbol{s})$$

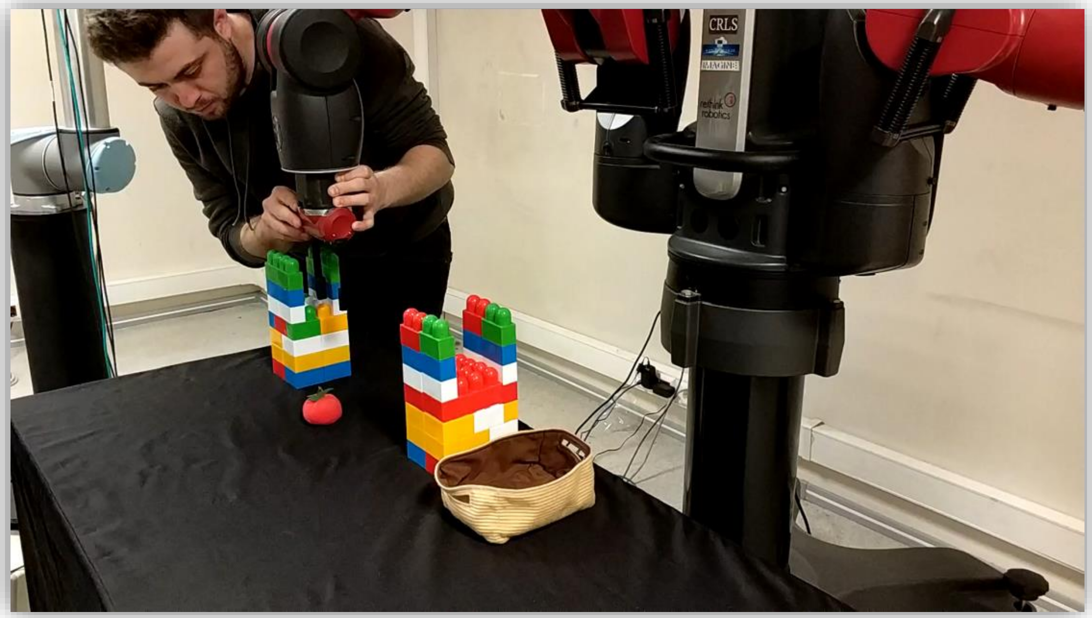
- DMP
- GMM-GMR
- HMM-LQR
- SEDS
- ProMP
- End-to-end deep visuomotor
- KMP
- BGMM

Model-free LfD

Note: They can still be combined with forward-models in the control level. But the learning is not model-aware.

Model-aware learning $p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)$

What are the problems that might arise if we do not care about the forward-model of the robot in learning movement primitives?



- …
- Generalization to infeasible trajectories
- If not kinesthetic teaching, correspondence problems
- Causality
- High gains in the controller

Model-aware learning:

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)p(\boldsymbol{u}_t|\boldsymbol{x}_t)d\boldsymbol{u}_t$$

$$p(\boldsymbol{x}_{t+1}) = \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)p(\boldsymbol{x}_t)d\boldsymbol{x}_t$$

$$p(\boldsymbol{\tau}) = p(\boldsymbol{x}_0)\prod_{t=1}^{T} p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)$$

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)$$

$$p(\boldsymbol{x}_t|t)$$

$$p(\boldsymbol{\tau}|\boldsymbol{x}_0)$$

$$p(\boldsymbol{u}_t|\boldsymbol{x}_t)$$

Probabilistic model-based imitation learning*

Matching :

- Probability density of the trajectory calculated from demonstration. $p_{\exp}(\boldsymbol{\tau})$
- Probability density of the trajectory parametrized by the unknown policy parameters $p_{\boldsymbol{\theta}}(\boldsymbol{\tau})$

$$\mathrm{KL}(p(x)||q(x)) = \int p(x)\log\frac{p(x)}{q(x)}dx$$

$$\min_{\boldsymbol{\theta}}\mathrm{KL}(p_{\exp}(\boldsymbol{\tau})||p_{\boldsymbol{\theta}}(\boldsymbol{\tau}))$$

$$\Rightarrow \min_{\boldsymbol{\theta}}\Sigma_{t=1}^{T}\mathrm{KL}(p_{\exp}(\boldsymbol{x}_t)||p_{\boldsymbol{\theta}}(\boldsymbol{x}_t))$$

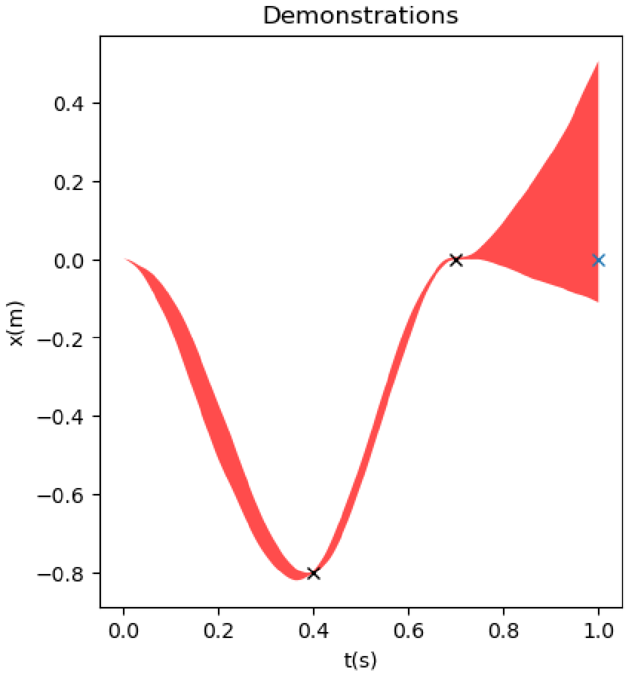
Assumption 1

$$p(\tau) = \prod_{t=1}^{T} p(\boldsymbol{x}_t)$$

*Englert, P., Paraschos, A., Deisenroth, M. P., & Peters, J. (2013). Probabilistic model-based imitation learning. *Adaptive Behavior*, *21*(5), 388–403.

Probabilistic model-based imitation learning*

How to calculate $p_{\exp}(\boldsymbol{\tau})$?

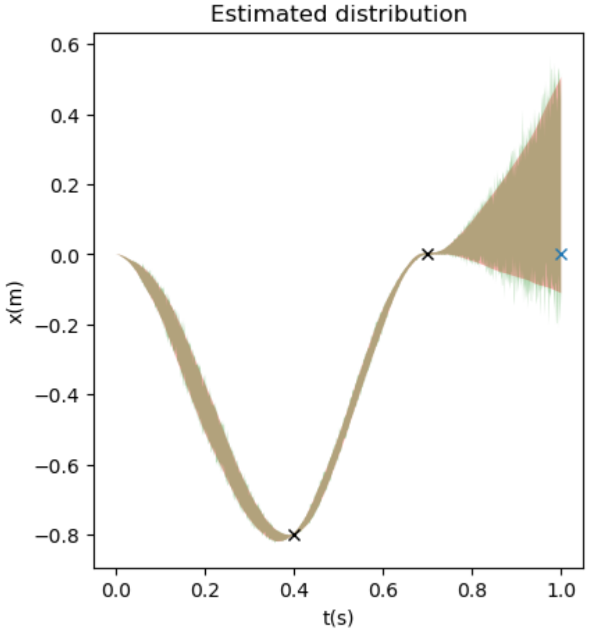


$$\hat{\mu}_t^{\exp} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_t^i$$

$$\hat{\Sigma}_t^{\exp} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_t^i - \hat{\mu}_t^{\exp})(\mathbf{x}_t^i - \hat{\mu}_t^{\exp})^\top$$

$$p(\boldsymbol{\tau}^{\exp}) = \mathcal{N}(\hat{\mu}^{\exp}, \widehat{\Sigma}^{\exp})$$

$$= \mathcal{N}\left(\begin{bmatrix}\hat{\mu}_1^{\exp}\\ \hat{\mu}_2^{\exp}\\ \vdots\\ \hat{\mu}_T^{\exp}\end{bmatrix}, \begin{bmatrix}\widehat{\Sigma}_1^{\exp} & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \widehat{\Sigma}_2^{\exp} & & \vdots\\ \vdots & & \ddots & \mathbf{0}\\ \mathbf{0} & \cdots & \mathbf{0} & \widehat{\Sigma}_T^{\exp}\end{bmatrix}\right)$$



$p_{\exp}(\boldsymbol{\tau})$

Structure is in line with Assumption 1.

*Englert, P., Paraschos, A., Deisenroth, M. P., & Peters, J. (2013). Probabilistic model-based imitation learning. *Adaptive Behavior*, *21*(5), 388–403.

Forward Models

How to calculate $p_{\boldsymbol{\theta}}(\boldsymbol{\tau})$?

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = \int \overbrace{p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)}^{\text{Forward model}} \overbrace{p(\boldsymbol{u}_t|\boldsymbol{x}_t)}^{\text{Control policy}} d\boldsymbol{u}_t$$

Forward models	
Linear	$x_{t+1} \sim \mathcal{N}(A_t \tilde{x}_t + b_t, \Sigma_t)$
Mixture of experts	$x_{t+1} \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(A_{kt} \tilde{x}_t + b_{kt}, \Sigma_{kt})$
RBF	$x_{t+1} = \sum_{k=1}^{K} \pi_k \mathcal{N}(\tilde{x}_t \| \mu_k, \Sigma)$
GP	$x_{t+1} = f(\tilde{x}_t) + \epsilon$ $f \sim \mathrm{GP}$ $\epsilon \sim \mathcal{N}(0, \Sigma_\epsilon)$ $\boxed{p(\mathbf{x}_{t+1} \| \mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1} \| \boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1})}$ with $\boldsymbol{\mu}_{t+1} = \mathbb{E}_f[f(\mathbf{x}_t, \mathbf{u}_t)] = m_f(\mathbf{x}_t, \mathbf{u}_t)$, $\boldsymbol{\Sigma}_{t+1} = \mathrm{var}_f[f(\mathbf{x}_t, \mathbf{u}_t)] = \sigma_f^2(\mathbf{x}_t, \mathbf{u}_t)$, $m_f(\tilde{\mathbf{x}}_\star) = \mathbf{k}_\star^\top \boldsymbol{K}^{-1} \mathbf{y}$, $\sigma_f^2(\tilde{\mathbf{x}}_\star) = \mathbf{k}_{\star\star} - \mathbf{k}_\star^\top \boldsymbol{K}^{-1} \mathbf{k}_\star$

Forward models	
NN	$x_{t+1} \sim \mathcal{N}(\mu_{\theta_\mu}(\tilde{x}_t), \Sigma_{\theta_\Sigma}(\tilde{x}_t))$
Koopman	$g(x_{t+1}) \sim \mathcal{N}(A_t g(x_t) + B_t u_t, \Sigma_t)$ $x_{t+1} = g^{-1}(g(x_{t+1}))$
More	?

Control Policies

How to calculate $p_{\boldsymbol{\theta}}(\boldsymbol{\tau})$?

Forward model

Control policy

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)p(\boldsymbol{u}_t|\boldsymbol{x}_t)d\boldsymbol{u}_t$$

Control Policies		Control Policies	
Linear	$u_t \sim \mathcal{N}(A_t x_t + b_t, \Sigma_t)$	**NN**	$u_t \sim \mathcal{N}(\mu_{\theta_\mu}(x_t), \Sigma_{\theta_\Sigma}(x_t))$
Mixture of experts	$u_t \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(A_{kt} x_t + b_{kt}, \Sigma_{kt})$	**Koopman**	$u_t \sim \mathcal{N}(A_t g(x_t), \Sigma_t)$ $x_t = g^{-1}(g(x_t))$
RBF	$u_t = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_t \mid \mu_k, \Sigma)$	**More**	?
GP	$u_t = f(x_t) + \epsilon$ $f \sim \mathrm{GP}$ $\epsilon \sim \mathcal{N}(0, \Sigma_\epsilon)$		

Calculating the trajectory distributions

How to calculate $p_{\boldsymbol{\theta}}(\boldsymbol{\tau})$?

Forward model

Control policy

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)p(\boldsymbol{u}_t|\boldsymbol{x}_t)d\boldsymbol{u}_t$$

For which forward model and control policy combinations, do we have an analytical solution for this integral?

$$p(\boldsymbol{x}_{t+1}) = \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)p(\boldsymbol{x}_t)d\boldsymbol{x}_t$$

Same question?

$$x_{t+1} \sim \mathcal{N}(A_t\tilde{x}_t + b_t, \Sigma_t)$$
$$u_t \sim \mathcal{N}(A_t x_t + b_t, \Sigma_t)$$

Linear

In PILCO and this paper*, they found out a way to use GP forward models with nonlinear control policies with a lot of approximation which seem to work well.

$$g(x_{t+1}) \sim \mathcal{N}(A_t g(x_t) + B_t u_t, \Sigma_t)$$
$$u_t \sim \mathcal{N}(A_t g(x_t), \Sigma_t)$$
$$x_t = g^{-1}(g(x_t))$$

Koopman

https://en.wikipedia.org/wiki/Conjugate_prior

*Englert, P., Paraschos, A., Deisenroth, M. P., & Peters, J. (2013). Probabilistic model-based imitation learning. *Adaptive Behavior*, *21*(5), 388–403.

Calculating the trajectory distributions

How to calculate $p_{\boldsymbol{\theta}}(\boldsymbol{\tau})$?

In PILCO and this paper*, they found out a way to use GP forward models with nonlinear control policies with a lot of approximation which seem to work well.

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = \int \underbrace{p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)}_{\text{Forward model}} \underbrace{p(\boldsymbol{u}_t|\boldsymbol{x}_t)}_{\text{Control policy}} d\boldsymbol{u}_t$$

and

$$p(\boldsymbol{x}_{t+1}) = \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) p(\boldsymbol{x}_t) d\boldsymbol{x}_t$$

gives

$$\Rightarrow \quad p(\boldsymbol{x}_{t+1}) = \int\int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t) p(\boldsymbol{u}_t|\boldsymbol{x}_t) p(\boldsymbol{x}_t) d\boldsymbol{u}_t d\boldsymbol{x}_t$$

$$\Rightarrow \quad p(\boldsymbol{x}_{t+1}) = \int\int p(\boldsymbol{x}_{t+1}|\tilde{\boldsymbol{x}}_t) p(\tilde{\boldsymbol{x}}_t) df\, d\tilde{\boldsymbol{x}}_t$$

with

$$x_{t+1} = f(\tilde{x}_t) + \epsilon$$
$$f \sim \mathrm{GP}$$
$$\epsilon \sim \mathcal{N}(0, \Sigma_\epsilon)$$

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = \mathcal{N}(\mathbf{x}_{t+1}|\mu_{t+1}, \Sigma_{t+1})$$
$$\text{with } \mu_{t+1} = \mathbb{E}_f[f(\mathbf{x}_t, \mathbf{u}_t)] = m_f(\mathbf{x}_t, \mathbf{u}_t),$$
$$\Sigma_{t+1} = \mathrm{var}_f[f(\mathbf{x}_t, \mathbf{u}_t)] = \sigma_f^2(\mathbf{x}_t, \mathbf{u}_t),$$

$$m_f(\tilde{\mathbf{x}}_\star) = \mathbf{k}_\star^\top K^{-1}\mathbf{y},$$
$$\sigma_f^2(\tilde{\mathbf{x}}_\star) = \mathbf{k}_{\star\star} - \mathbf{k}_\star^\top K^{-1}\mathbf{k}_\star$$

*Englert, P., Paraschos, A., Deisenroth, M. P., & Peters, J. (2013). Probabilistic model-based imitation learning. *Adaptive Behavior*, *21*(5), 388–403.

Calculating the trajectory distributions

How to calculate $p_{\boldsymbol{\theta}}(\boldsymbol{\tau})$?

In PILCO and this paper*, they found out a way to use GP forward models with nonlinear control policies with a lot of approximation which seem to work well.

Forward model

Control policy

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)p(\boldsymbol{u}_t|\boldsymbol{x}_t)d\boldsymbol{u}_t$$

and

$$p(\boldsymbol{x}_{t+1}) = \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)p(\boldsymbol{x}_t)d\boldsymbol{x}_t$$

gives

$$\Rightarrow \quad p(\boldsymbol{x}_{t+1}) = \int\int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)p(\boldsymbol{u}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t)d\boldsymbol{u}_t d\boldsymbol{x}_t$$

2. assumed Gaussian

$$\Rightarrow \quad p(\boldsymbol{x}_{t+1}) = \int\int p(\boldsymbol{x}_{t+1}|\tilde{\boldsymbol{x}}_t)p(\tilde{\boldsymbol{x}}_t)df\, d\tilde{\boldsymbol{x}}_t$$

3. Exact distribution not intractable. Approximate it by a Gaussian distribution.

Assumption 3

$$p(\boldsymbol{x}_{t+1}) \sim \mathcal{N}(\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1})$$

Assumption 2

$$p(\tilde{\boldsymbol{x}}_t) \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_t, \tilde{\boldsymbol{\Sigma}}_t)$$

*Englert, P., Paraschos, A., Deisenroth, M. P., & Peters, J. (2013). Probabilistic model-based imitation learning. *Adaptive Behavior*, *21*(5), 388–403.

Calculating the trajectory distributions for linear case

How to calculate $p_{\boldsymbol{\theta}}(\boldsymbol{\tau})$?

Forward model

Control policy

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)p(\boldsymbol{u}_t|\boldsymbol{x}_t)d\boldsymbol{u}_t$$

and

$$p(\boldsymbol{x}_{t+1}) = \int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)p(\boldsymbol{x}_t)d\boldsymbol{x}_t$$

gives

$$p(\boldsymbol{x}_{t+1}) = \int\int p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t)p(\boldsymbol{u}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t)d\boldsymbol{u}_t d\boldsymbol{x}_t$$

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{u}_t) = \mathcal{N}(\boldsymbol{A}_t\boldsymbol{x}_t + \boldsymbol{B}_t\boldsymbol{u}_t, \boldsymbol{\Sigma}_t^d)$$

$$p(\boldsymbol{u}_t|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{C}_t\boldsymbol{x}_t + \boldsymbol{d}_t, \boldsymbol{\Sigma}_t^u)$$

HINT

$$\mathbb{E}[\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}] = \boldsymbol{A}\mathbb{E}[\boldsymbol{x}] + \boldsymbol{b}$$

$$\mathrm{Var}[\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}] = \boldsymbol{A}\mathrm{Var}[\boldsymbol{x}]\boldsymbol{A}^T$$