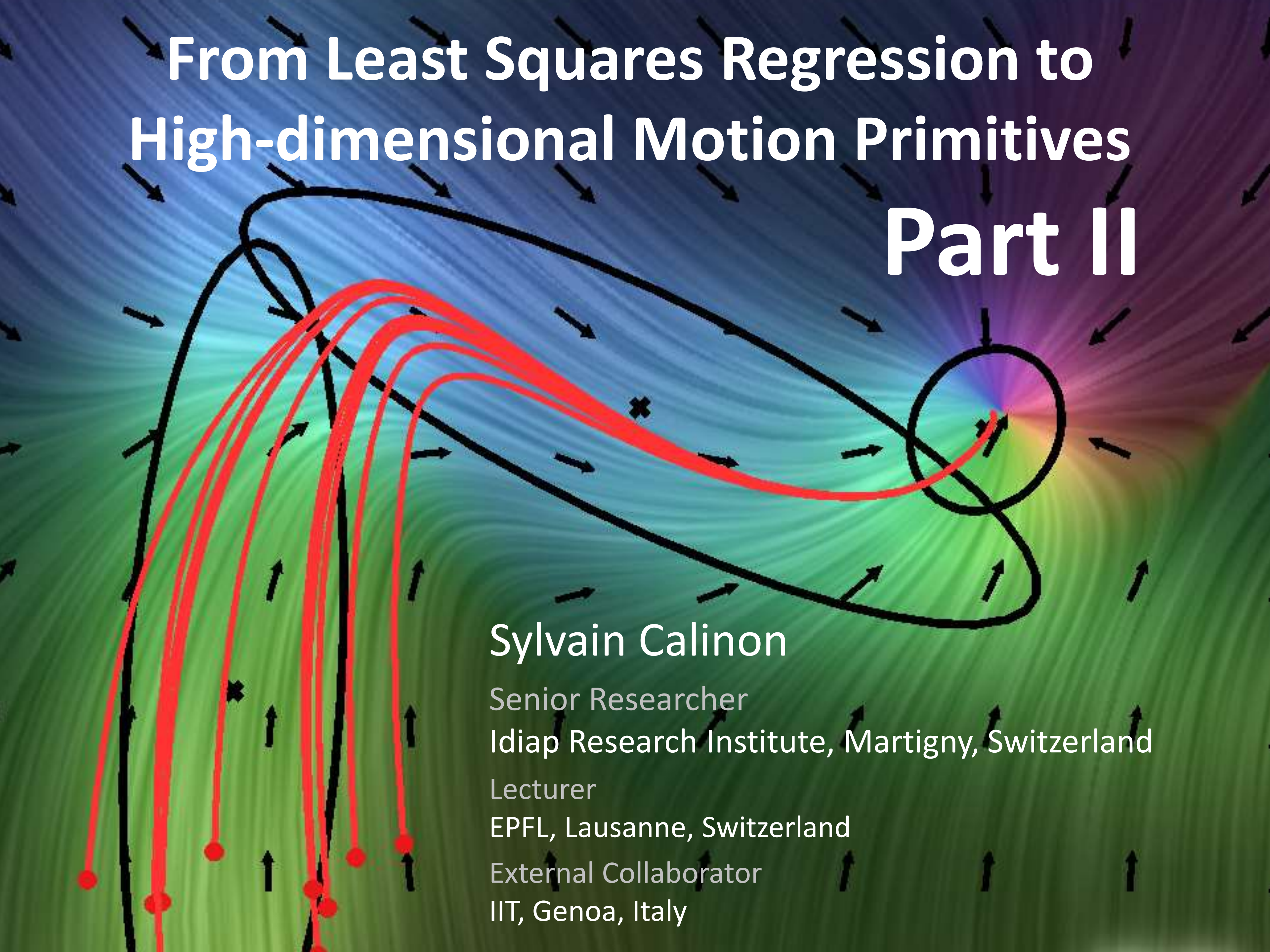
From Least Squares Regression to High-dimensional Motion Primitives

Part II



Sylvain Calinon

Senior Researcher

Idiap Research Institute, Martigny, Switzerland

Lecturer

EPFL, Lausanne, Switzerland

External Collaborator

IIT, Genoa, Italy

Superposition of movement primitives

VS

Fusion of movement primitives

Outline

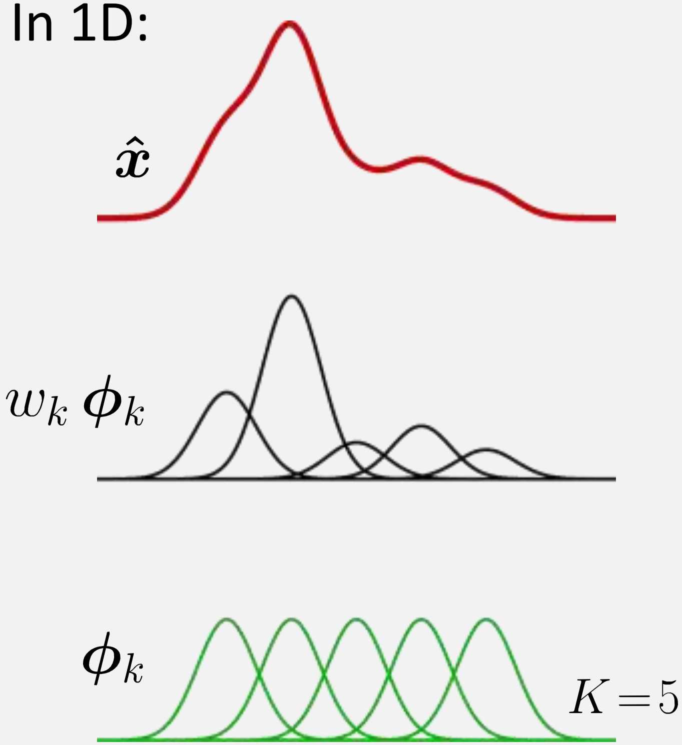
- Combination as fusion problem
- Application I:
  Ridge regression
- Application II:
  Model predictive control
- Application III:
  Task-parameterized models
- Examples of robot applications

Combination of primitives as a fusion problem

Superposition

$$\hat{\boldsymbol{x}} = \sum_{k=1}^{K} w_k \, \boldsymbol{\phi}_k$$

In 1D:



Fusion

$$\hat{\boldsymbol{x}} = \left(\sum_{k=1}^{K} \boldsymbol{W}_k \right)^{-1} \sum_{k=1}^{K} \boldsymbol{W}_k \, \boldsymbol{\phi}_k$$

$$= \arg\min_{\boldsymbol{x}} \sum_{k=1}^{K} \| \boldsymbol{\phi}_k - \boldsymbol{x} \|^2_{\boldsymbol{W}_k}$$

$$= \arg\min_{\boldsymbol{x}} \sum_{k=1}^{K} (\boldsymbol{\phi}_k - \boldsymbol{x})^{\top} \boldsymbol{W}_k (\boldsymbol{\phi}_k - \boldsymbol{x})$$

Choosing scalar weights or full weight matrices is not a detail…

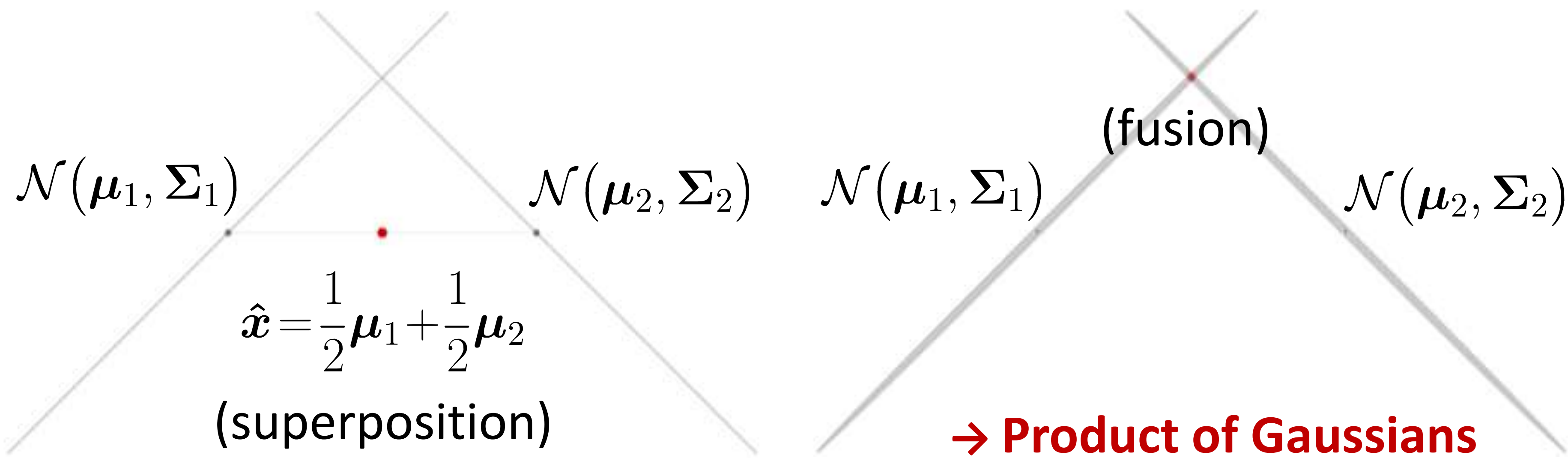
Motivating example:
A probabilistic view on segment crossing!

$\boldsymbol{\mu}_i$ center of the Gaussian

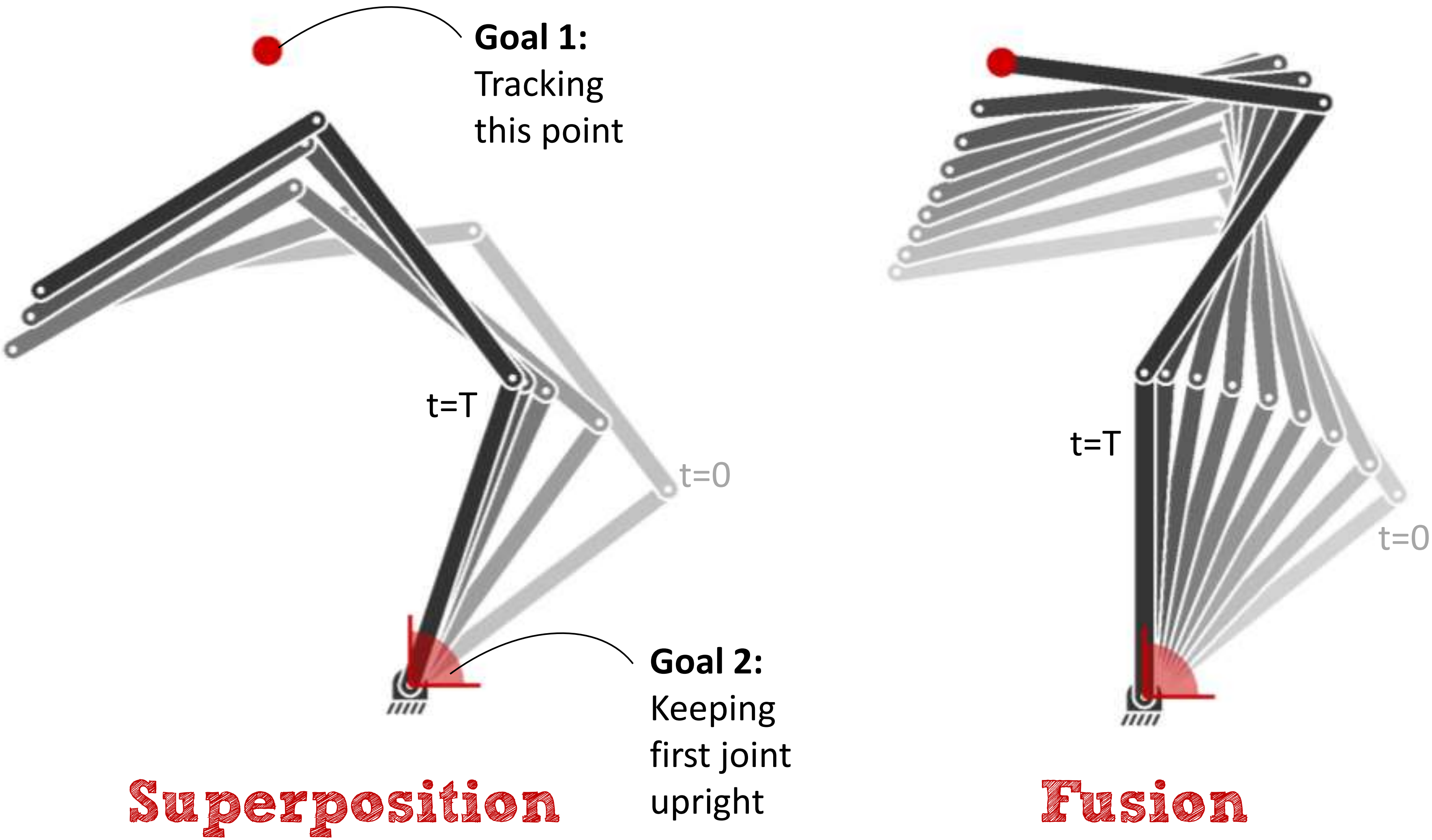
$\boldsymbol{\Sigma}_i$ covariance matrix

$\boldsymbol{W}_i$ precision matrix $\left(\boldsymbol{W}_i = \boldsymbol{\Sigma}_i^{-1}\right)$

$$
\begin{aligned}
\hat{\boldsymbol{x}} &= \arg\min_{\boldsymbol{x}} \left\|\boldsymbol{\mu}_1 - \boldsymbol{x}\right\|^2_{\boldsymbol{W}_1} + \left\|\boldsymbol{\mu}_2 - \boldsymbol{x}\right\|^2_{\boldsymbol{W}_2} \\
&= \left(\boldsymbol{W}_1 + \boldsymbol{W}_2\right)^{-1} \left(\boldsymbol{W}_1\boldsymbol{\mu}_1 + \boldsymbol{W}_2\boldsymbol{\mu}_2\right) \\
&= \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1} \left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right)
\end{aligned}
$$

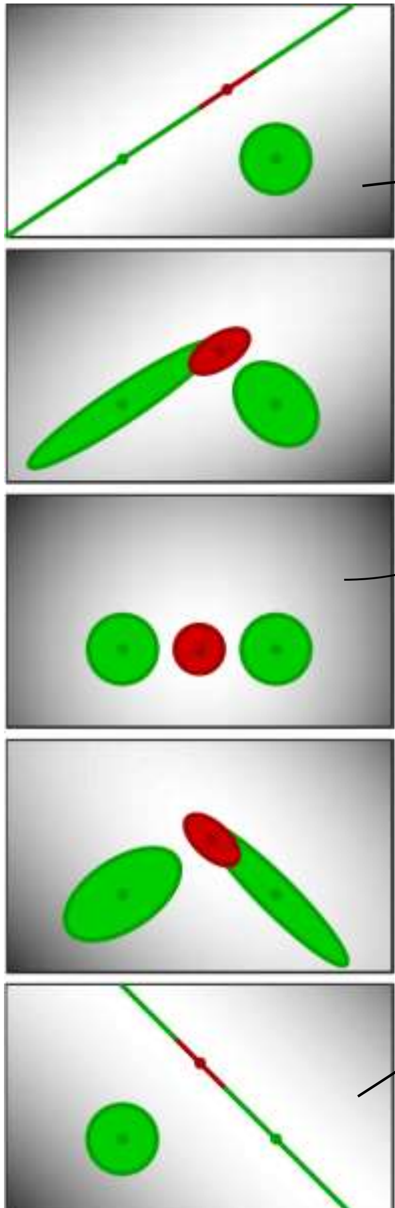


Motivating example:
Fusion of IK and joint angle controllers



Combination of primitives as a fusion problem

$$\mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \propto \mathcal{N}\left(\boldsymbol{\mu}^{(1)}, \boldsymbol{\Sigma}^{(1)}\right) \mathcal{N}\left(\boldsymbol{\mu}^{(2)}, \boldsymbol{\Sigma}^{(2)}\right)$$



Null space projection
(hierarchy constraints)

Scalar superposition

The full weight matrices approach covers both **scalar weights** (with isotropic diagonal matrix) and **null space projection** operations!

Application I

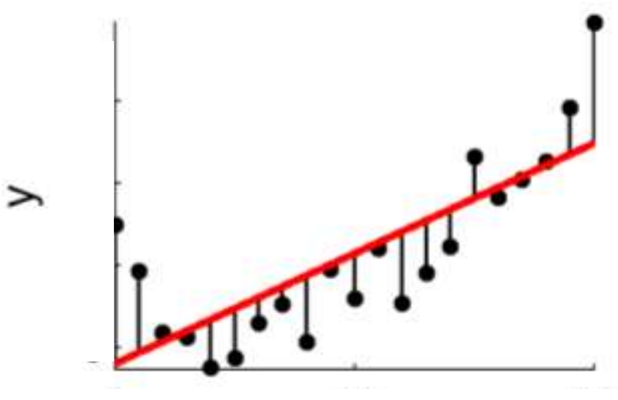
Bayesian linear regression

and its connection to ridge regression

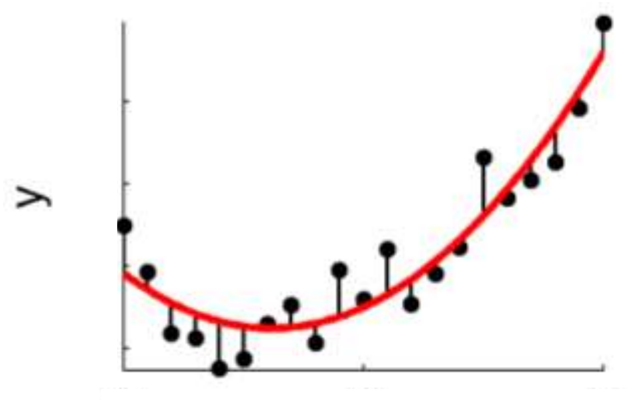
Ridge regression

Input data: $\boldsymbol{X} \in \mathbb{R}^{N \times D^{\mathcal{I}}}$
Output data: $\boldsymbol{Y} \in \mathbb{R}^{N \times D^{\mathcal{O}}}$
Goal: estimating $\boldsymbol{\beta} \in \mathbb{R}^{D^{\mathcal{I}} \times D^{\mathcal{O}}}$ to have $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta}$

Ridge regression:
$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \|\lambda\boldsymbol{\beta}\|^2 \\ &= \arg\min_{\boldsymbol{\beta}} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda^2\boldsymbol{\beta}^\top\boldsymbol{\beta} \\ &= (\boldsymbol{X}^\top\boldsymbol{X} + \lambda^2\boldsymbol{I})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}\end{aligned}$$



$\boldsymbol{X} = [\boldsymbol{x}, \boldsymbol{1}]$



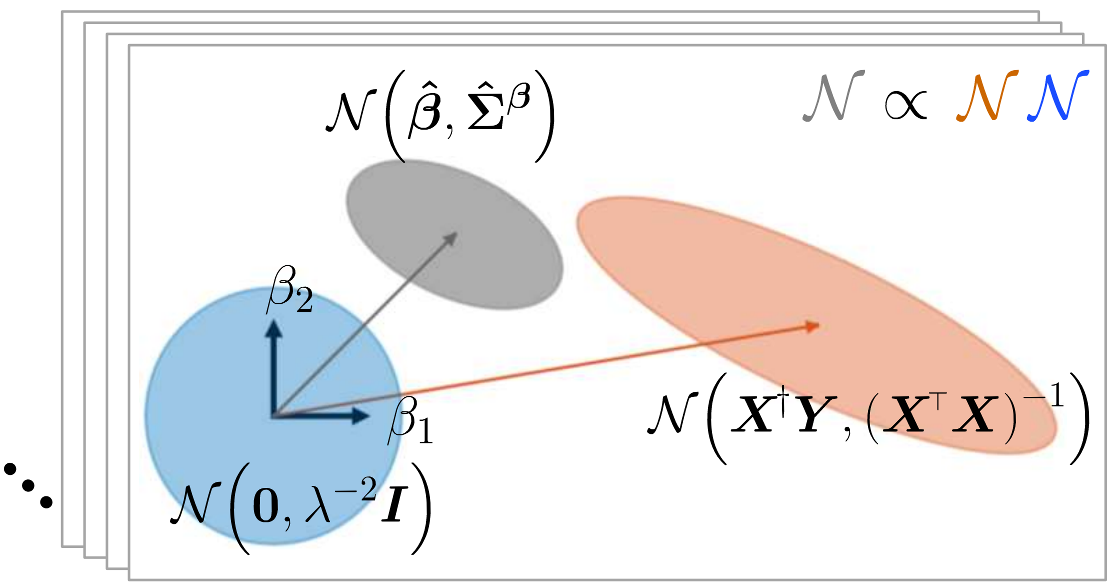
$\boldsymbol{X} = [\boldsymbol{x}^2, \boldsymbol{x}, \boldsymbol{1}]$

Bayesian linear regression

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \|\lambda\boldsymbol{\beta}\|^2$$

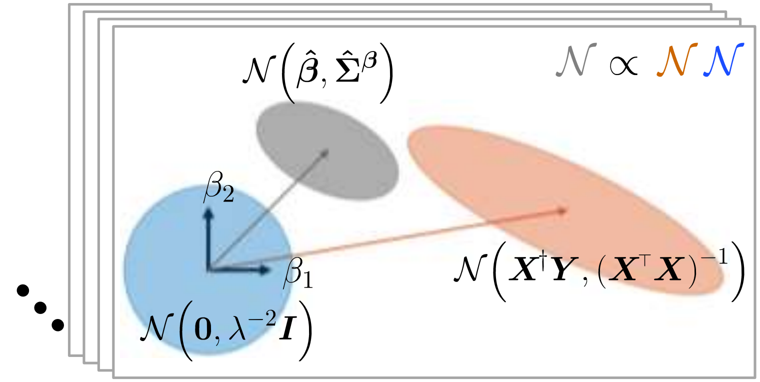
$$\boldsymbol{X}^\dagger = \boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{X}^\top)^{-1}$$

$$\begin{aligned}
c &= (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda^2\boldsymbol{\beta}^\top\boldsymbol{\beta} \\
&= (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^\top \, (\boldsymbol{X}\boldsymbol{X}^\dagger)^\top \boldsymbol{X}\boldsymbol{X}^\dagger \, (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda^2\boldsymbol{\beta}^\top\boldsymbol{\beta} \\
&= (\boldsymbol{X}^\dagger\boldsymbol{Y} - \boldsymbol{X}^\dagger\boldsymbol{X}\boldsymbol{\beta})^\top \, \boldsymbol{X}^\top\boldsymbol{X} \, (\boldsymbol{X}^\dagger\boldsymbol{Y} - \boldsymbol{X}^\dagger\boldsymbol{X}\boldsymbol{\beta}) + \lambda^2\boldsymbol{\beta}^\top\boldsymbol{\beta} \\
&= (\boldsymbol{X}^\dagger\boldsymbol{Y} - \boldsymbol{\beta})^\top \, \boldsymbol{X}^\top\boldsymbol{X} \, (\boldsymbol{X}^\dagger\boldsymbol{Y} - \boldsymbol{\beta}) + \lambda^2\boldsymbol{\beta}^\top\boldsymbol{\beta}
\end{aligned}$$



Least squares with a regularization term corresponds to a *maximum a posteriori (MAP)* estimate with a **Gaussian likelihood** and a **Gaussian prior**

Bayesian linear regression



Product of Gaussians:

$$\hat{\boldsymbol{x}} = \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1} \left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right)$$

Proposed Bayesian view of ridge regression:

$$\mathcal{N}\left(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\Sigma}}^{\boldsymbol{\beta}}\right) \propto \mathcal{N}\left(\boldsymbol{X}^{\dagger}\boldsymbol{Y}, (\boldsymbol{X}^{\top}\boldsymbol{X})^{-1}\right) \mathcal{N}\left(\boldsymbol{0}, \lambda^{-2}\boldsymbol{I}\right)$$

Verification:

$$\boldsymbol{X}^{\dagger} = \boldsymbol{X}^{\top}(\boldsymbol{X}\boldsymbol{X}^{\top})^{-1}$$

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^{\top}\boldsymbol{X} + \lambda^2\boldsymbol{I})^{-1}\boldsymbol{X}^{\top}\boldsymbol{X}\ \boldsymbol{X}^{\dagger}\boldsymbol{Y}$$
$$= (\boldsymbol{X}^{\top}\boldsymbol{X} + \lambda^2\boldsymbol{I})^{-1}\boldsymbol{X}^{\top}\boldsymbol{Y}$$

← ridge regression!

Model predictive control (MPC)

&

Linear quadratic tracking (LQT)

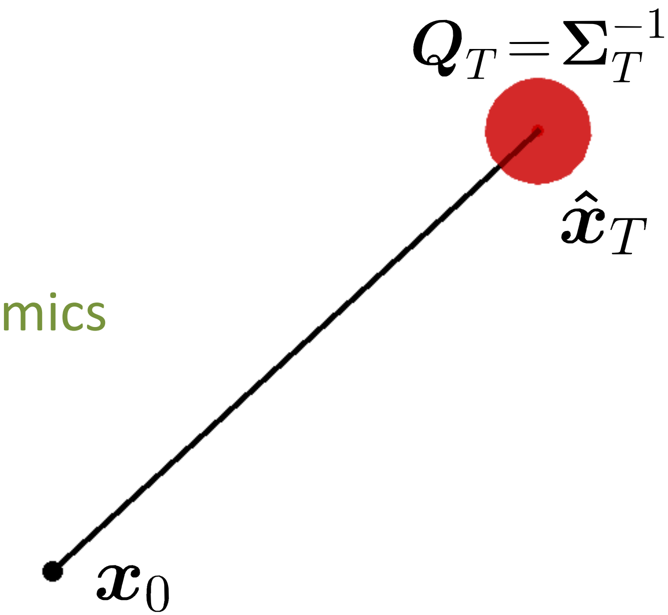
Linear quadratic tracking (LQT)

Track path! Use low control commands!

$$\min_{\boldsymbol{u}} \sum_{t=1}^{T} \left\|\boldsymbol{\mu}_t - \boldsymbol{x}_t\right\|^2_{\boldsymbol{Q}_t} + \left\|\boldsymbol{u}_t\right\|^2_{\boldsymbol{R}_t}$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t$$

System dynamics



Model predictive control (MPC):

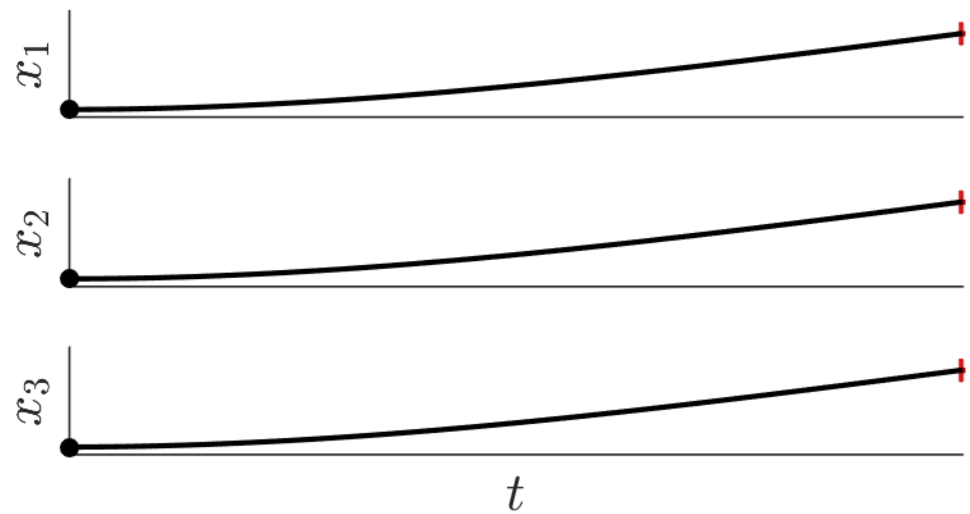
$\boldsymbol{x}_t$ state variable (position+velocity)

$\boldsymbol{\mu}_t$ desired state

$\boldsymbol{u}_t$ control command (acceleration)

$\boldsymbol{Q}_t$ precision matrix

$\boldsymbol{R}_t$ control weight matrix



How to solve this objective function?

Track path! Use low control commands!

$$\min_{\boldsymbol{u}} \sum_{t=1}^{T} \left\|\boldsymbol{\mu}_t - \boldsymbol{x}_t\right\|_{\boldsymbol{Q}_t}^2 + \left\|\boldsymbol{u}_t\right\|_{\boldsymbol{R}_t}^2$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t$$

System dynamics

Pontryagin's maximum principle → Riccati equation

(the Physicist perspective)

Dynamic programming

(the Computer Scientist perspective)

Linear algebra

(the Algebraist perspective)

Let's first re-organize the objective function...

$$c = \sum_{t=1}^{T} \Big((\boldsymbol{\mu}_t - \boldsymbol{x}_t)^\top \boldsymbol{Q}_t (\boldsymbol{\mu}_t - \boldsymbol{x}_t) \; + \; \boldsymbol{u}_t^\top \boldsymbol{R}_t \, \boldsymbol{u}_t \Big)$$

$$= (\boldsymbol{\mu} - \boldsymbol{x})^\top \boldsymbol{Q} (\boldsymbol{\mu} - \boldsymbol{x}) \; + \; \boldsymbol{u}^\top \boldsymbol{R} \boldsymbol{u}$$

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{Q}_T \end{bmatrix} \qquad \boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{R}_T \end{bmatrix}$$

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \vdots \\ \boldsymbol{\mu}_T \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_T \end{bmatrix}$$

Let's then re-organize the constraint...

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\,\boldsymbol{x}_t + \boldsymbol{B}\,\boldsymbol{u}_t$$

$$\begin{aligned}
\boldsymbol{x}_2 &= \boldsymbol{A}\boldsymbol{x}_1 + \boldsymbol{B}\boldsymbol{u}_1 \\
\boldsymbol{x}_3 &= \boldsymbol{A}\boldsymbol{x}_2 + \boldsymbol{B}\boldsymbol{u}_2 = \boldsymbol{A}(\boldsymbol{A}\boldsymbol{x}_1 + \boldsymbol{B}\boldsymbol{u}_1) + \boldsymbol{B}\boldsymbol{u}_2 \\
&\vdots \\
\boldsymbol{x}_T &= \boldsymbol{A}^{T-1}\boldsymbol{x}_1 + \boldsymbol{A}^{T-2}\boldsymbol{B}\boldsymbol{u}_1 + \boldsymbol{A}^{T-3}\boldsymbol{B}\boldsymbol{u}_2 + \cdots + \boldsymbol{B}_{T-1}\boldsymbol{u}_{T-1}
\end{aligned}$$

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^{T-1} \end{bmatrix}}_{\boldsymbol{S}^{x}} \boldsymbol{x}_1 + \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{A}^{T-2}\boldsymbol{B} & \boldsymbol{A}^{T-3}\boldsymbol{B} & \cdots & \boldsymbol{B} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{S}^{u}} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_T \end{bmatrix}$$

$$\boldsymbol{x} = \boldsymbol{S}^{x}\boldsymbol{x}_1 + \boldsymbol{S}^{u}\boldsymbol{u}$$

Linear quadratic tracking: Analytic solution

The constraint can then be put into the objective function:

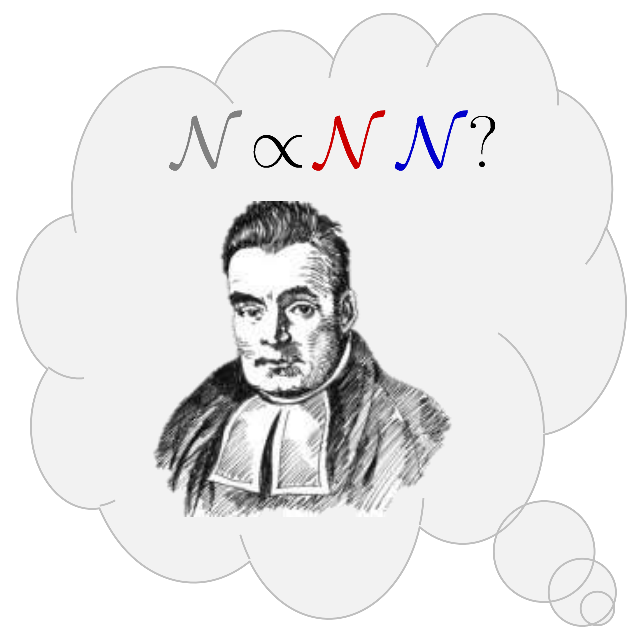
$$\boldsymbol{x} = \boldsymbol{S}^{\boldsymbol{x}}\boldsymbol{x}_1 + \boldsymbol{S}^{\boldsymbol{u}}\boldsymbol{u}$$

$$c = (\boldsymbol{\mu} - \boldsymbol{x})^{\top}\boldsymbol{Q}(\boldsymbol{\mu} - \boldsymbol{x}) \;+\; \boldsymbol{u}^{\top}\boldsymbol{R}\boldsymbol{u}$$

$$= \left(\boldsymbol{\mu} - \boldsymbol{S}^{\boldsymbol{x}}\boldsymbol{x}_1 - \boldsymbol{S}^{\boldsymbol{u}}\boldsymbol{u}\right)^{\top}\boldsymbol{Q}\left(\boldsymbol{\mu} - \boldsymbol{S}^{\boldsymbol{x}}\boldsymbol{x}_1 - \boldsymbol{S}^{\boldsymbol{u}}\boldsymbol{u}\right) + \boldsymbol{u}^{\top}\boldsymbol{R}\boldsymbol{u}$$

Solving for *u* results in the analytic solution:

$$\hat{\boldsymbol{u}} = \left(\boldsymbol{S}^{\boldsymbol{u}\top}\boldsymbol{Q}\boldsymbol{S}^{\boldsymbol{u}} + \boldsymbol{R}\right)^{-1}\boldsymbol{S}^{\boldsymbol{u}\top}\boldsymbol{Q}\left(\boldsymbol{\mu} - \boldsymbol{S}^{\boldsymbol{x}}\boldsymbol{x}_1\right)$$



MPC/LQT as a product of Gaussians

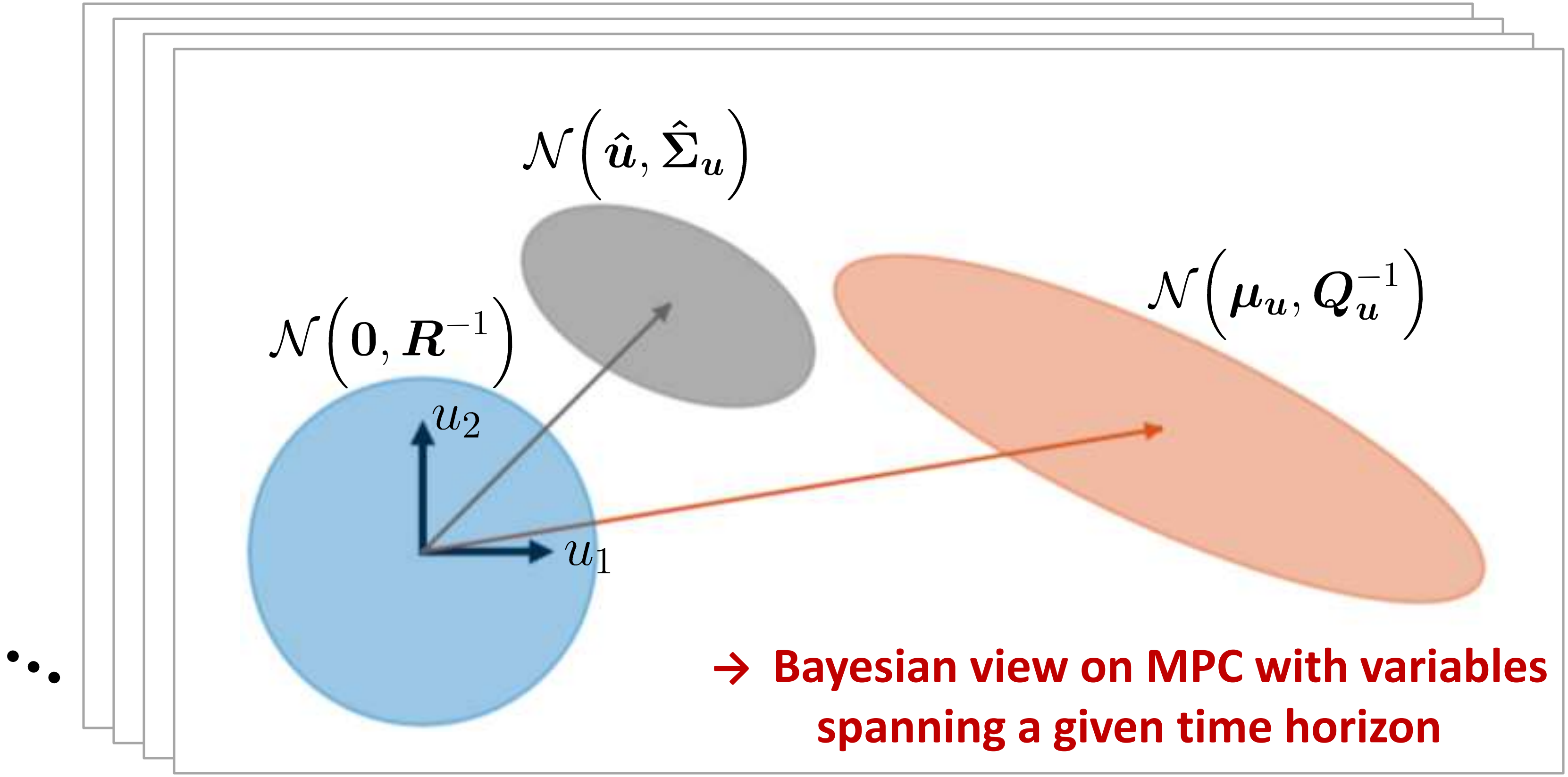
Track path! Use low control commands!

$$\min_{\boldsymbol{u}} \sum_{t=1}^{T} \left\|\boldsymbol{\mu}_t - \boldsymbol{x}_t\right\|_{\boldsymbol{Q}_t}^2 + \left\|\boldsymbol{u}_t\right\|_{\boldsymbol{R}_t}^2$$

$$\mathcal{N}\left(\hat{\boldsymbol{u}}, \hat{\Sigma}^{\boldsymbol{u}}\right) \propto \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{u}}, \boldsymbol{Q}_{\boldsymbol{u}}^{-1}\right) \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{R}^{-1}\right)$$

$$\boldsymbol{\mu}_{\boldsymbol{u}} \triangleq \boldsymbol{S}^{\boldsymbol{u}\dagger}(\boldsymbol{\mu} - \boldsymbol{S}^{\boldsymbol{x}}\boldsymbol{x}_1)$$

$$\boldsymbol{Q}_{\boldsymbol{u}} \triangleq \boldsymbol{S}^{\boldsymbol{u}\top}\boldsymbol{Q}\boldsymbol{S}^{\boldsymbol{u}}$$



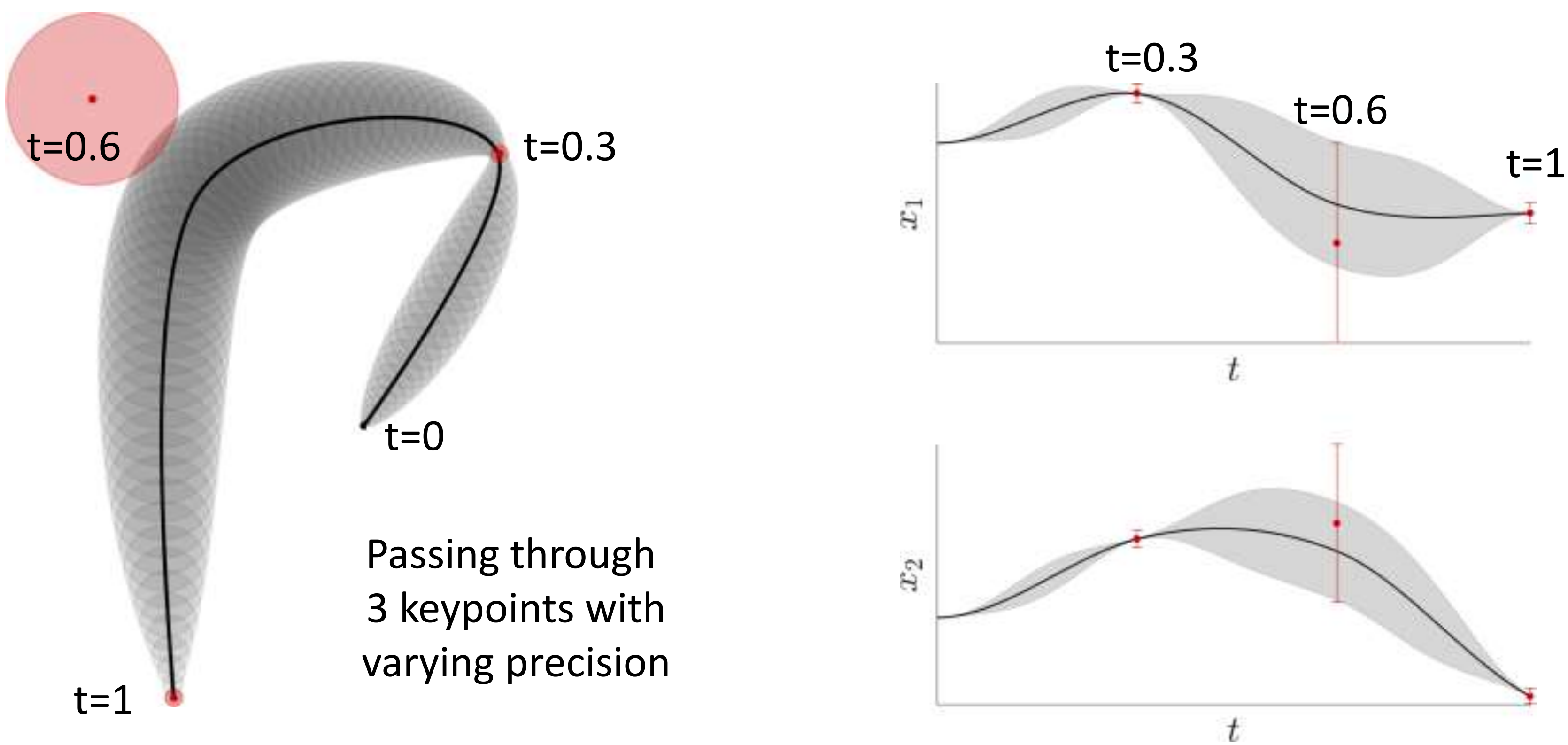
→ **Bayesian view on MPC with variables spanning a given time horizon**

Probabilistic representation of MPC/LQT

$$\hat{\boldsymbol{u}} = \left(\boldsymbol{S}^{u\top}\boldsymbol{Q}\boldsymbol{S}^{u} + \boldsymbol{R}\right)^{-1}\boldsymbol{S}^{u\top}\boldsymbol{Q}\left(\boldsymbol{\mu} - \boldsymbol{S}^{x}\boldsymbol{x}_1\right)$$
$$\hat{\boldsymbol{\Sigma}}^{u} = \left(\boldsymbol{S}^{u\top}\boldsymbol{Q}\boldsymbol{S}^{u} + \boldsymbol{R}\right)^{-1}$$

$$\hat{\boldsymbol{x}} = \boldsymbol{S}^{x}\boldsymbol{x}_1 + \boldsymbol{S}^{u}\hat{\boldsymbol{u}}$$
$$\hat{\boldsymbol{\Sigma}}^{x} = \boldsymbol{S}^{u}\left(\boldsymbol{S}^{u\top}\boldsymbol{Q}\boldsymbol{S}^{u} + \boldsymbol{R}\right)^{-1}\boldsymbol{S}^{u\top}$$

The distribution in control space can be projected back to the state space



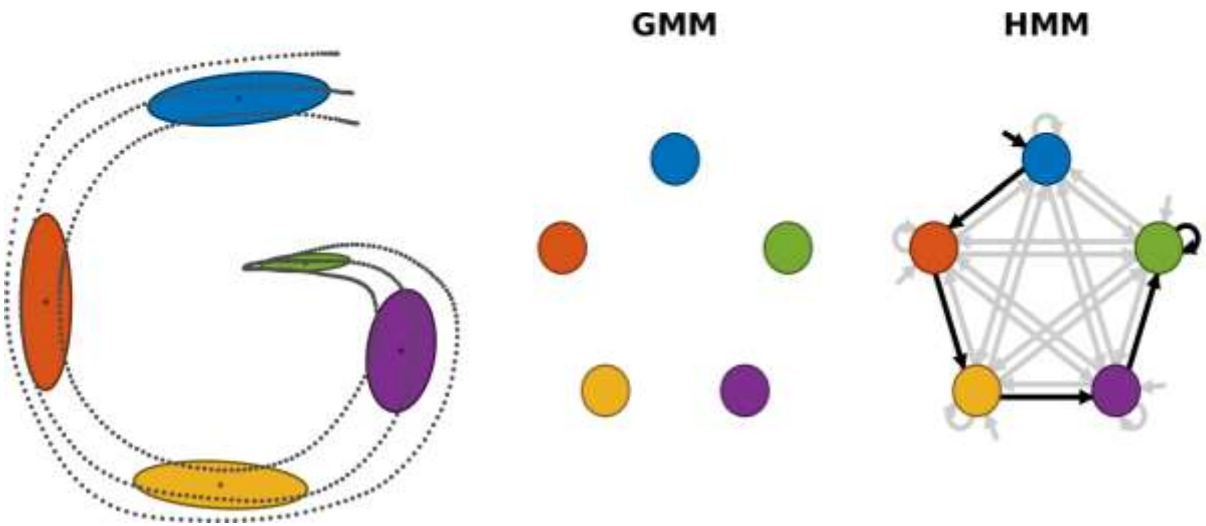
Passing through 3 keypoints with varying precision

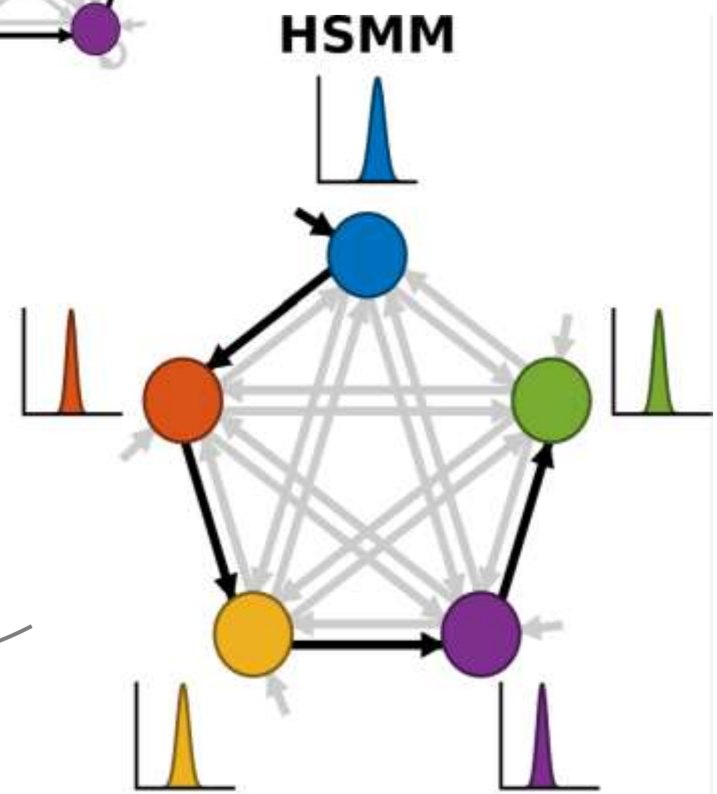
Model Predictive Control (MPC) combined with probabilistic representation of movement primitives

$$c = (\boldsymbol{\mu} - \boldsymbol{x})^{\top} \boldsymbol{Q} (\boldsymbol{\mu} - \boldsymbol{x}) \; + \; \boldsymbol{u}^{\top} \boldsymbol{R} \boldsymbol{u}$$

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \vdots \\ \boldsymbol{\mu}_T \end{bmatrix} \quad \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_2 & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{Q}_T \end{bmatrix}$$

Hidden semi-Markov model (HSMM)

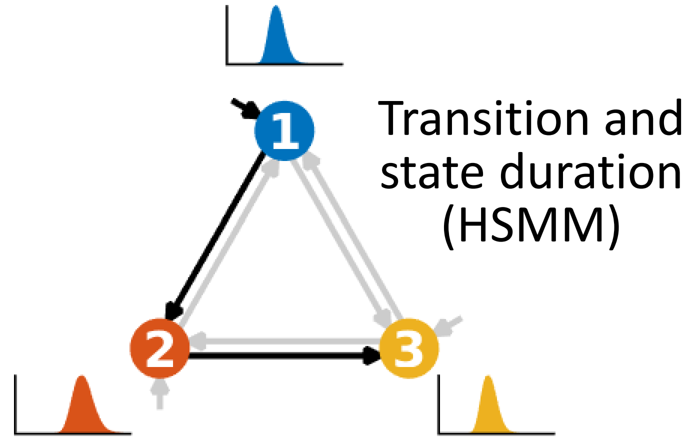




HSMM provides a model of the state duration instead of relying on self-transition probabilities as in standard HMM

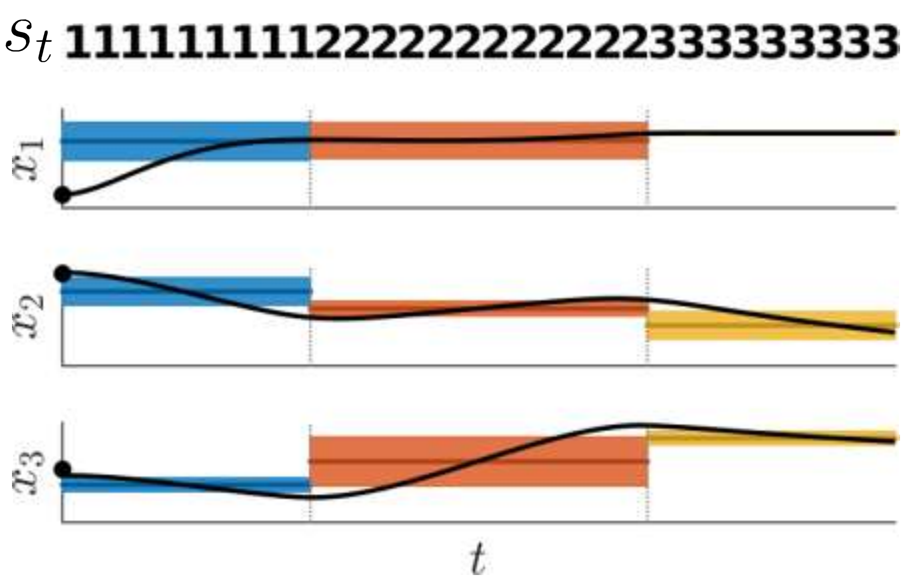
Learning minimal intervention controllers

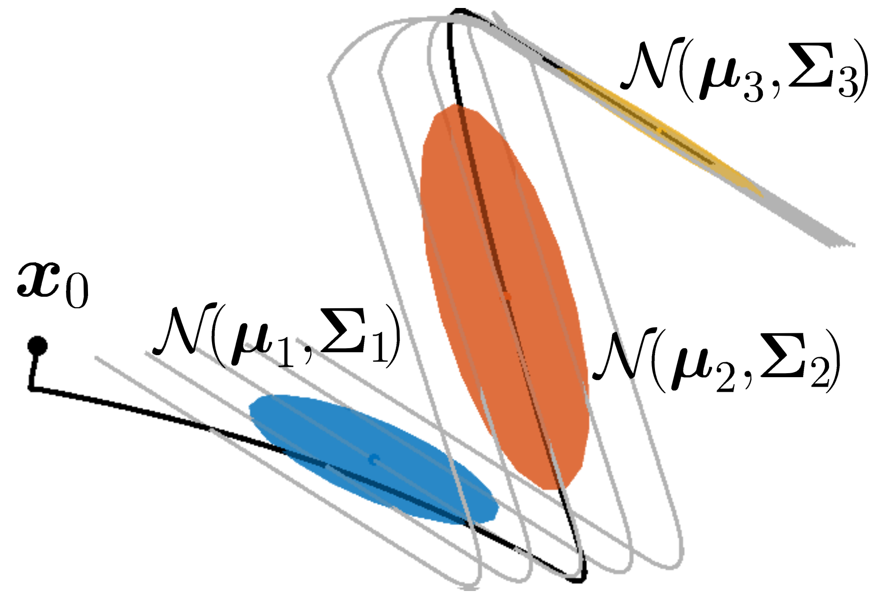
➡ **Analytical solution to generate movements by following minimal intervention control principle**



Stepwise reference path given by:

$$\hat{\boldsymbol{x}}_t = \boldsymbol{\mu}_{s_t} \quad \boldsymbol{Q}_t = \boldsymbol{\Sigma}_{s_t}^{-1}$$



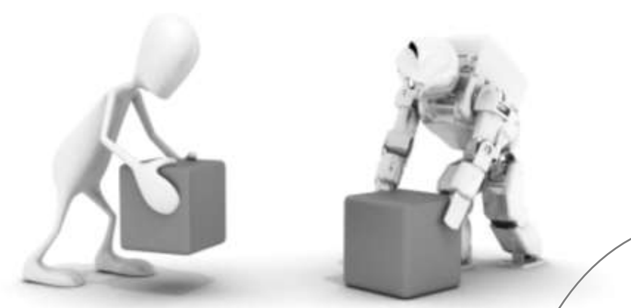


$\boldsymbol{\mu}_i$ center of the Gaussian

$\boldsymbol{\Sigma}_i$ covariance matrix

Learning minimal intervention controllers

$$\boldsymbol{s} = \{1, 1, 1, 2, \ldots, 4\}$$



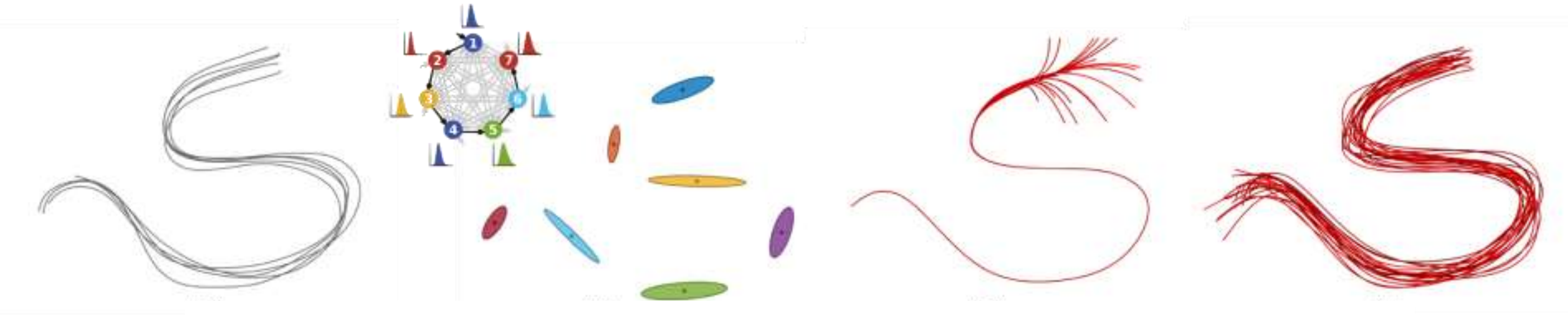
$$\begin{bmatrix} \boldsymbol{\Sigma}_1^{-1} & 0 & 0 & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Sigma}_1^{-1} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}_1^{-1} & 0 & \cdots & 0 \\ 0 & 0 & 0 & \boldsymbol{\Sigma}_2^{-1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \boldsymbol{\Sigma}_K^{-1} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \vdots \\ \boldsymbol{\mu}_K \end{bmatrix}$$

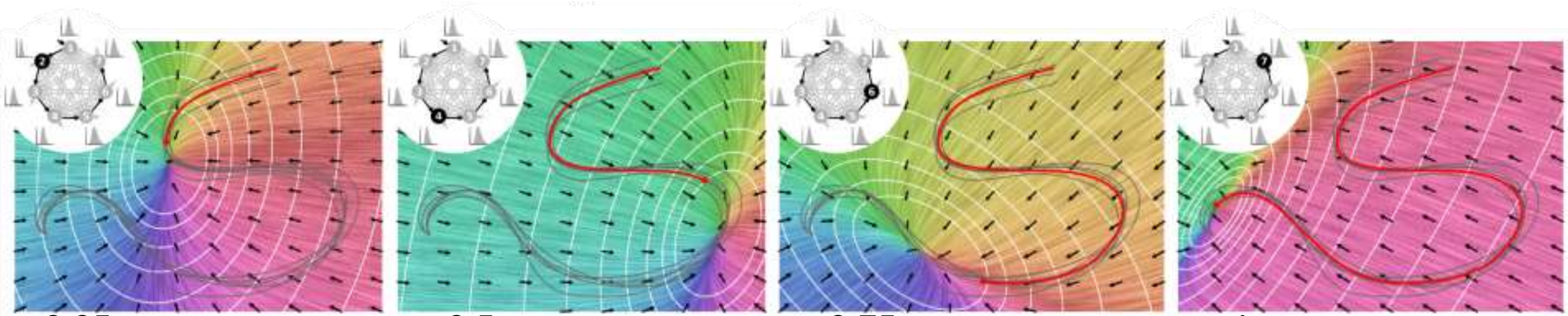
$$\hat{\boldsymbol{u}} = \left(\boldsymbol{S}^{\boldsymbol{u}\top}\boldsymbol{Q}\boldsymbol{S}^{\boldsymbol{u}} + \boldsymbol{R}\right)^{-1} \boldsymbol{S}^{\boldsymbol{u}\top}\boldsymbol{Q} \left(\boldsymbol{\mu} - \boldsymbol{S}^{\boldsymbol{x}}\boldsymbol{x}_1\right)$$

$$\begin{bmatrix} \hat{\boldsymbol{u}}_1 \\ \hat{\boldsymbol{u}}_2 \\ \hat{\boldsymbol{u}}_3 \\ \vdots \\ \hat{\boldsymbol{u}}_{T-1} \end{bmatrix}$$

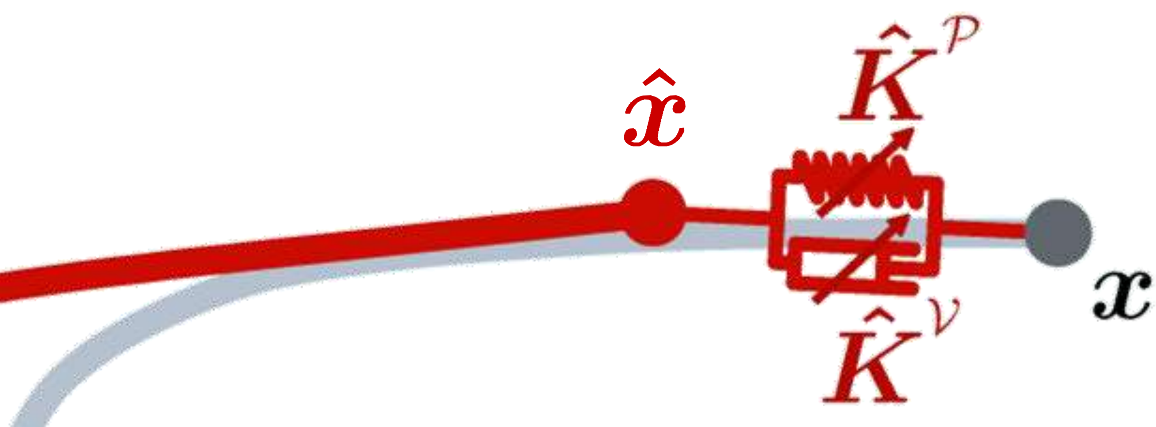
$$\begin{bmatrix} 0 & 0 & \cdots & 0 \\ \boldsymbol{B} & 0 & \cdots & 0 \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{T-2}\boldsymbol{B} & \boldsymbol{A}^{T-3}\boldsymbol{B} & \cdots & \boldsymbol{B} \end{bmatrix} \quad \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^{T-1} \end{bmatrix}$$

Learning controllers instead of trajectories

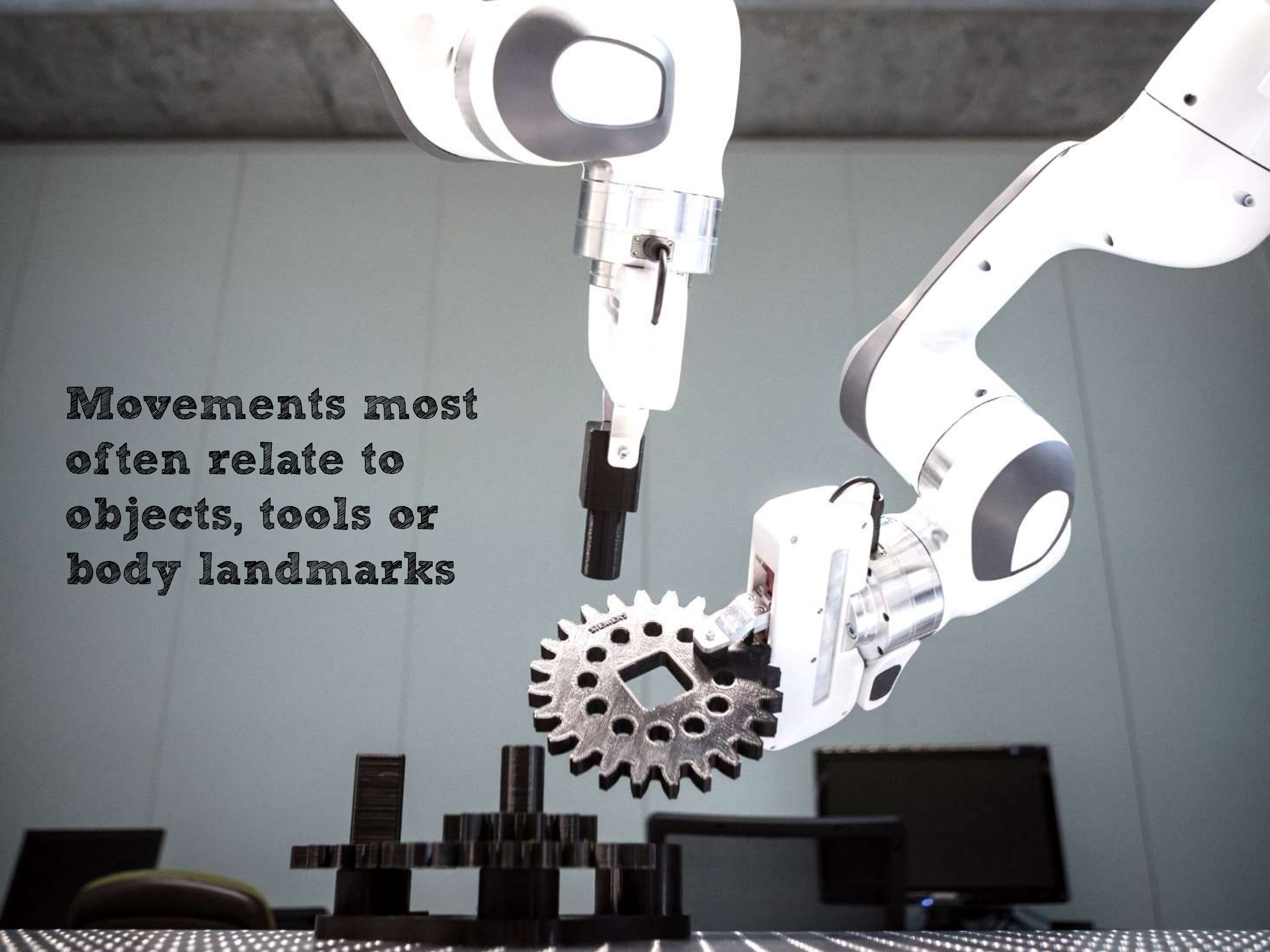




t=0.25 t=0.5 t=0.75 t=1



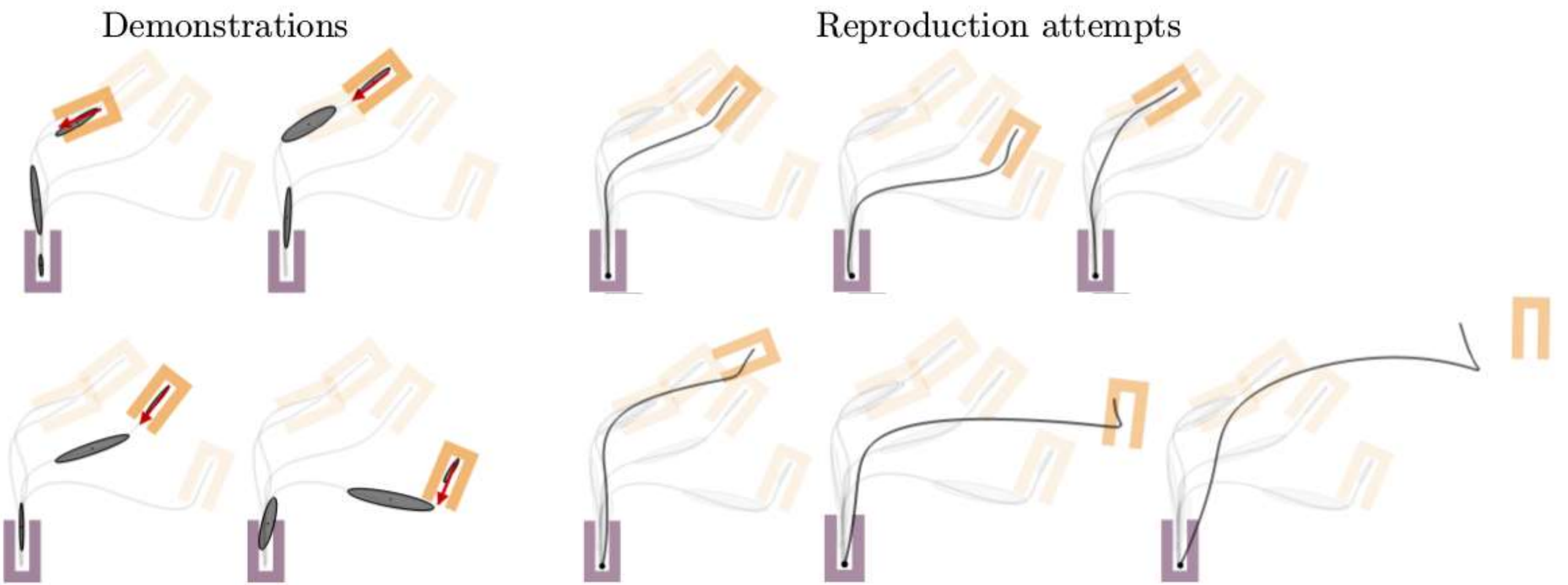
Task-parameterized movement models


Movements most often relate to objects, tools or body landmarks

Conditioning-based approach

Regression with a context variable *c*:

- Learning of $\mathcal{P}(\boldsymbol{c}, \boldsymbol{x})$
- Retrieval with $\mathcal{P}(\boldsymbol{x}|\boldsymbol{c})$



→ **Generic approach, but limited generalization capability**

MPC/LQT in multiple coordinate systems

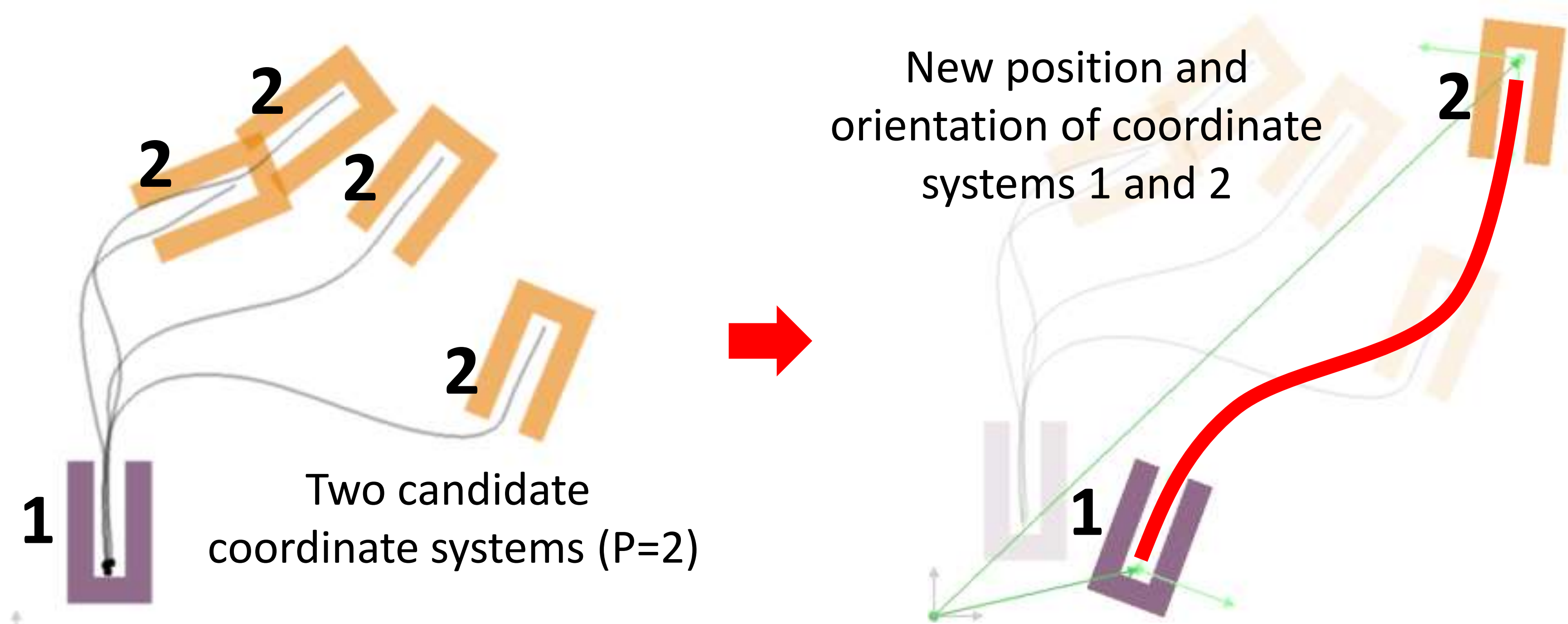
Track path in coordinate system j

$$\min_{\boldsymbol{u}} \sum_{t=1}^{T} \sum_{j=1}^{P} \left\| \boldsymbol{\mu}_t^{(j)} - \boldsymbol{x}_t \right\|^2_{\boldsymbol{Q}_t^{(j)}} + \left\| \boldsymbol{u}_t \right\|^2_{\boldsymbol{R}_t}$$

Use low control commands

$$\text{s.t.} \quad \boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t$$

System dynamics



Set of demonstrations **Reproduction in new situation**

MPC/LQT in multiple coordinate systems

Track path in coordinate system j

$$\min_{\boldsymbol{u}} \sum_{t=1}^{T} \sum_{j=1}^{P} \left\| \boldsymbol{\mu}_t^{(j)} - \boldsymbol{x}_t \right\|_{\boldsymbol{Q}_t^{(j)}}^2 + \left\| \boldsymbol{u}_t \right\|_{\boldsymbol{R}_t}^2$$

Use low control commands

$$\text{s.t.} \quad \boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t$$

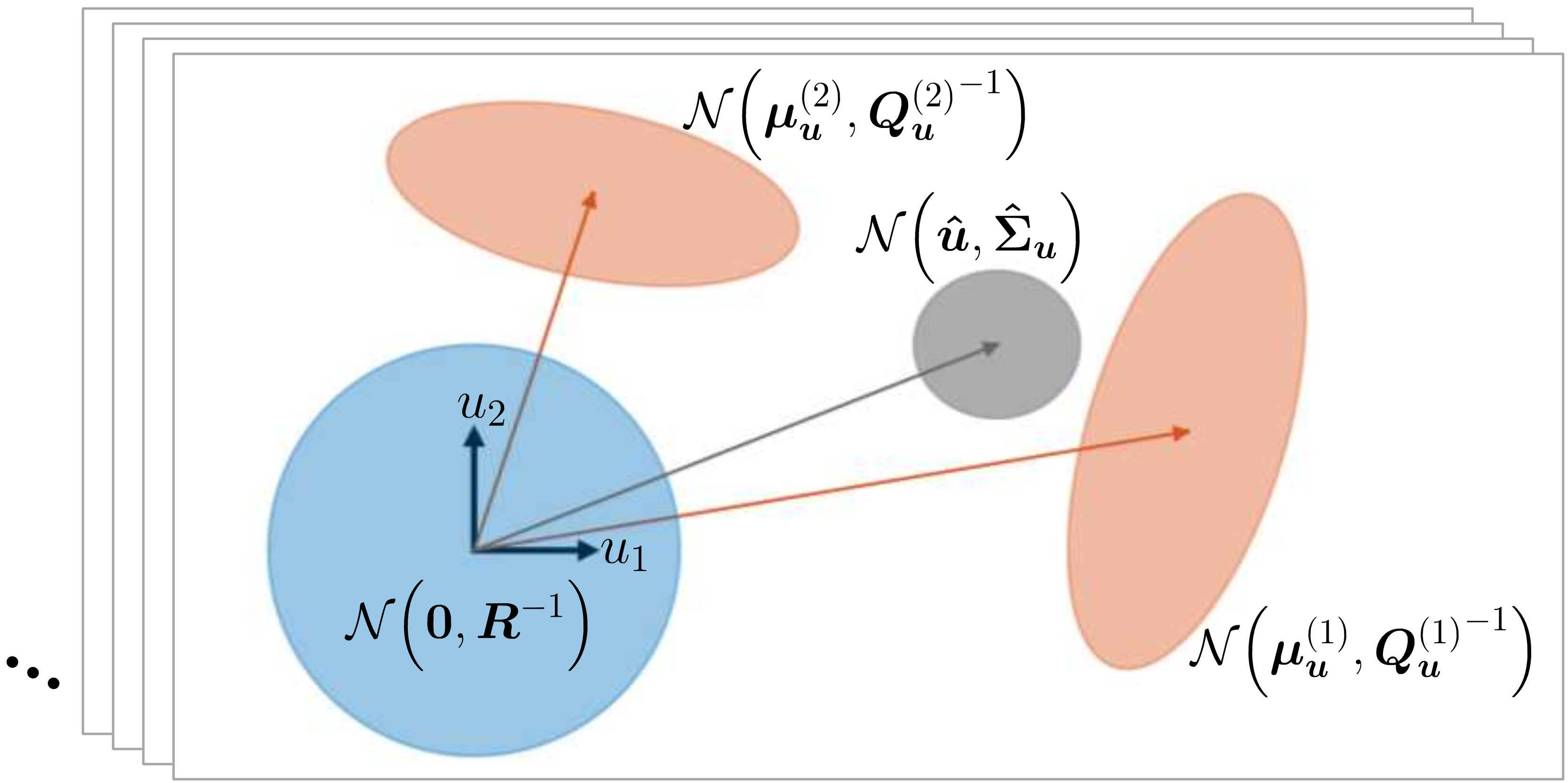
System dynamics

MPC/LQT in multiple coordinate systems

Track path in coordinate system j

$$\min_{\boldsymbol{u}} \sum_{t=1}^{T} \sum_{j=1}^{P} \left\| \boldsymbol{\mu}_t^{(j)} - \boldsymbol{x}_t \right\|^2_{\boldsymbol{Q}_t^{(j)}} + \left\| \boldsymbol{u}_t \right\|^2_{\boldsymbol{R}_t}$$

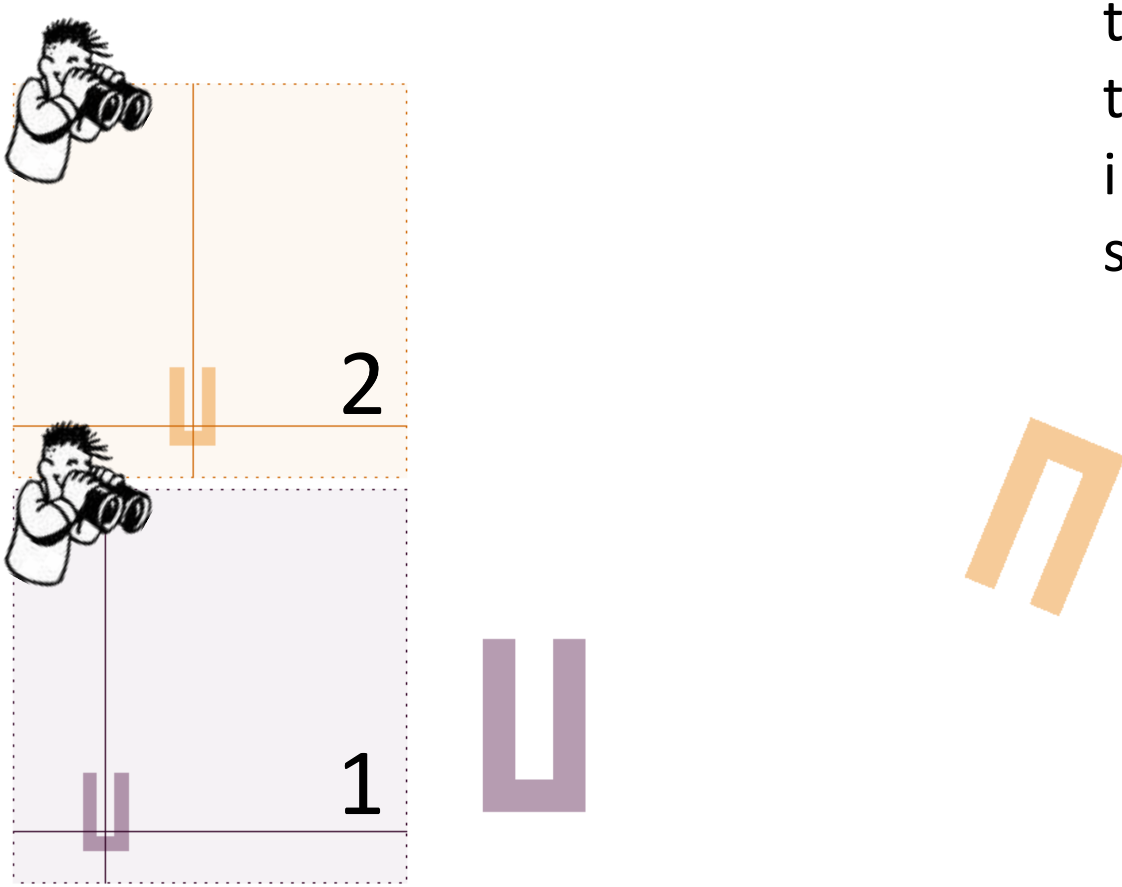
Use low control commands



MPC/LQT in multiple coordinate systems

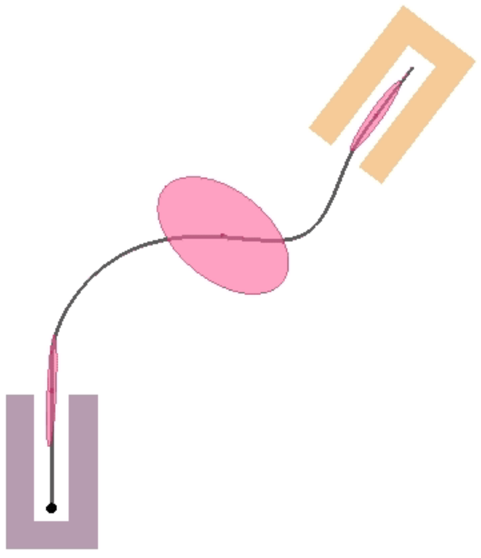
$$\min_{\boldsymbol{u}} \sum_{t=1}^{T} \sum_{j=1}^{P} \left\| \boldsymbol{\mu}_t^{(j)} - \boldsymbol{x}_t \right\|^2_{\boldsymbol{Q}_t^{(j)}} + \left\| \boldsymbol{u}_t \right\|^2_{\boldsymbol{R}_t}$$

In many robotics problems, the parameters describing the task or situation can be interpreted as coordinate systems

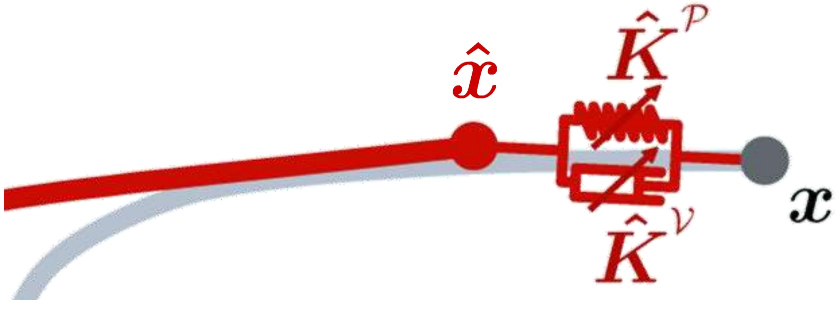


MPC/LQT in multiple coordinate systems

$$\min_{\boldsymbol{u}} \sum_{t=1}^{T} \sum_{j=1}^{P} \left\| \boldsymbol{\mu}_t^{(j)} - \boldsymbol{x}_t \right\|^2_{\boldsymbol{Q}_t^{(j)}} + \left\| \boldsymbol{u}_t \right\|^2_{\boldsymbol{R}_t}$$



➨ **Learning of a controller**
(instead of learning a trajectory)
that adapts to new situations
while regulating the gains
according to the precision and
coordination required by the task



MPC/LQT in multiple coordinate systems

$$\min_{\boldsymbol{u}} \sum_{t=1}^{T} \sum_{j=1}^{P} \left\| \boldsymbol{\mu}_t^{(j)} - \boldsymbol{x}_t \right\|^2_{\boldsymbol{Q}_t^{(j)}} + \left\| \boldsymbol{u}_t \right\|^2_{\boldsymbol{R}_t}$$

➡ **Retrieval of control commands in the form of trajectory distributions, facilitating exploration and adaptation (in either control or state space)**

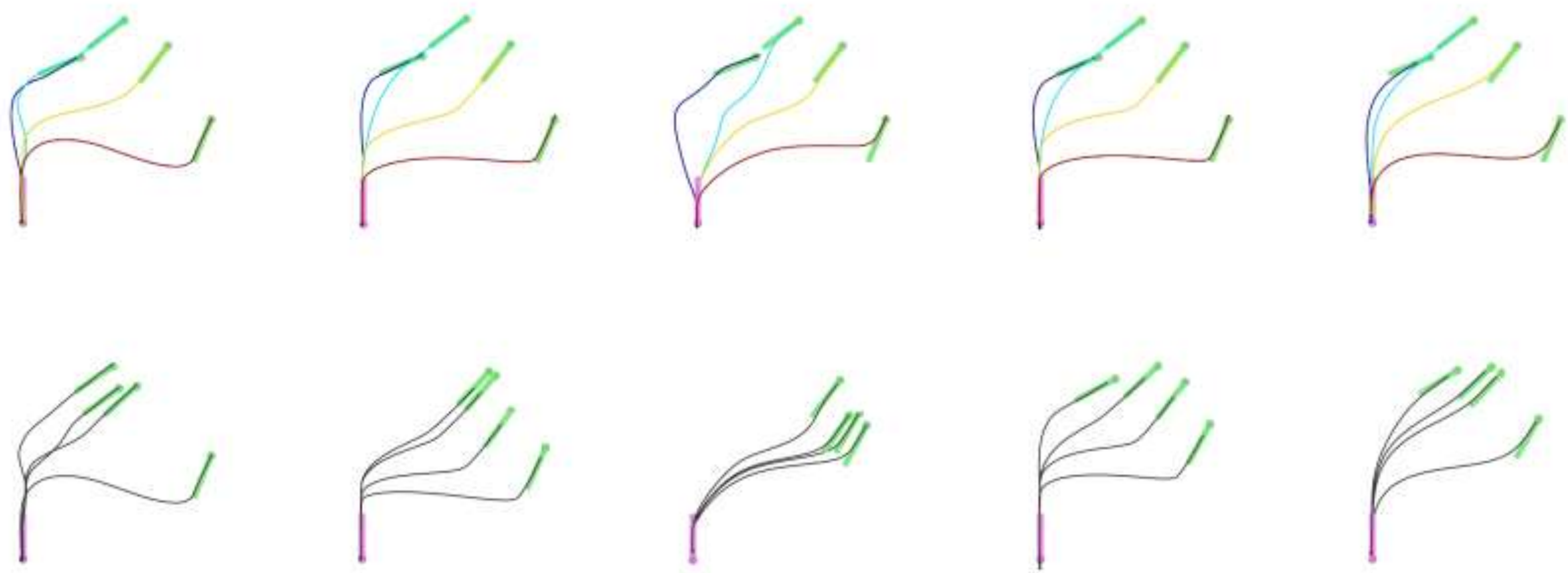
Exploitation in other probabilistic models

Demonstrations

TP model with raw trajectory distribution | TP model with MPC | TP model with GMR | TP model with Trajectory-HMM | TP model with ProMP

Reproductions in same situations

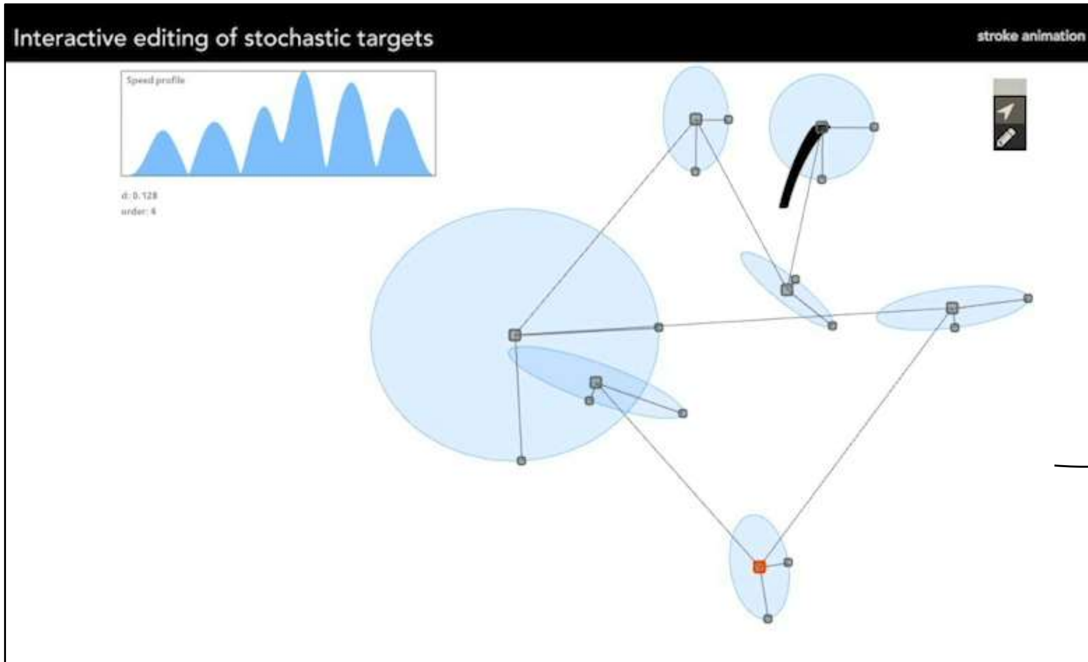
Reproductions in new situations



http://www.idiap.ch/software/pbdlib/

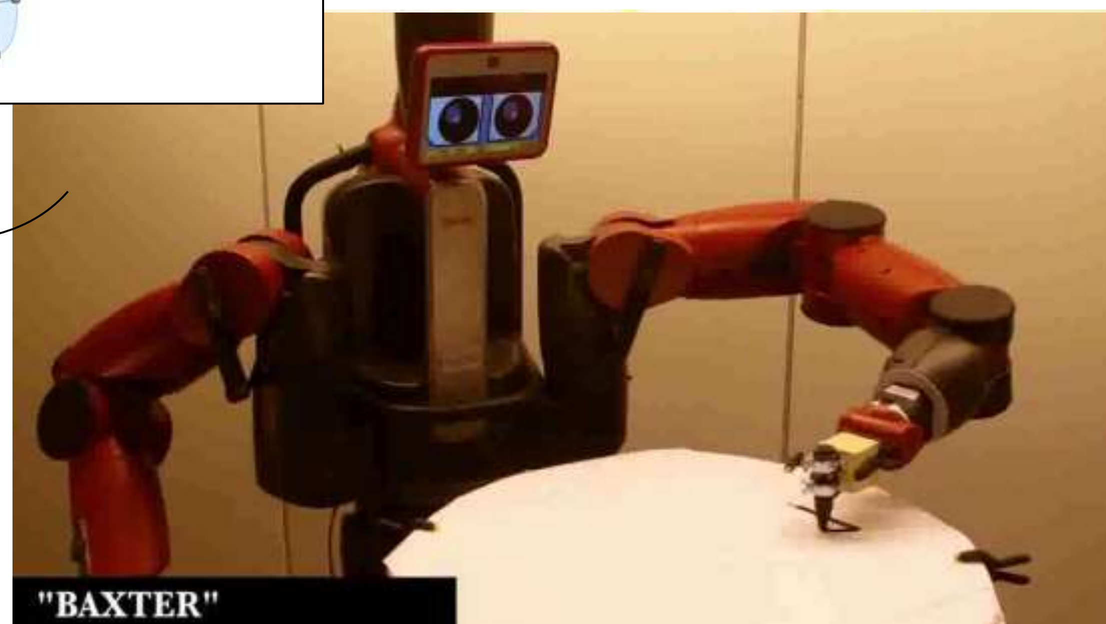
Robot application examples

Application: Editing movements with variations



User interface to edit and generate natural and dynamic motions by considering variation and coordination

Compliant controller to retrieve safe and human-like motions



Daniel Berio

Frederic Fol Leymarie

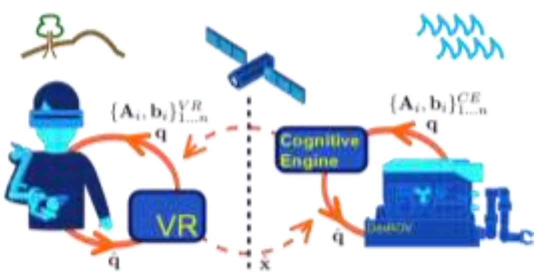
idiap

Goldsmiths UNIVERSITY OF LONDON

[Berio, Calinon and Leymarie, IROS'2016] [Berio, Calinon and Leymarie , MOCO'2017]

Application: Shared control

DexROV

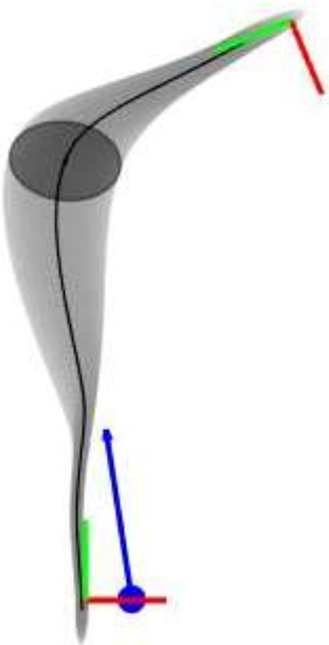


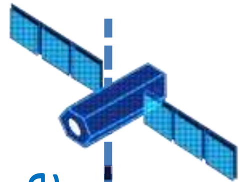
http://dexrov.eu

EC, H2020 (2015-2018)

Application: Shared control

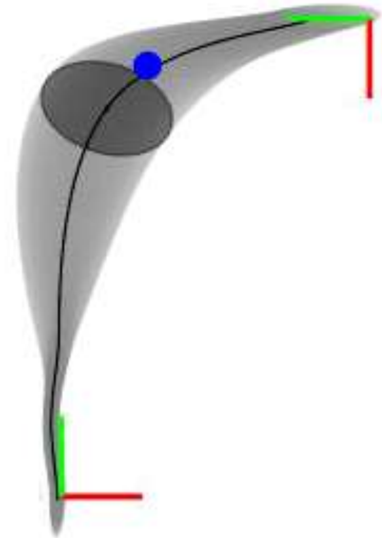
DexROV http://dexrov.eu
EC, H2020 (2015-2018)





Teleoperator side

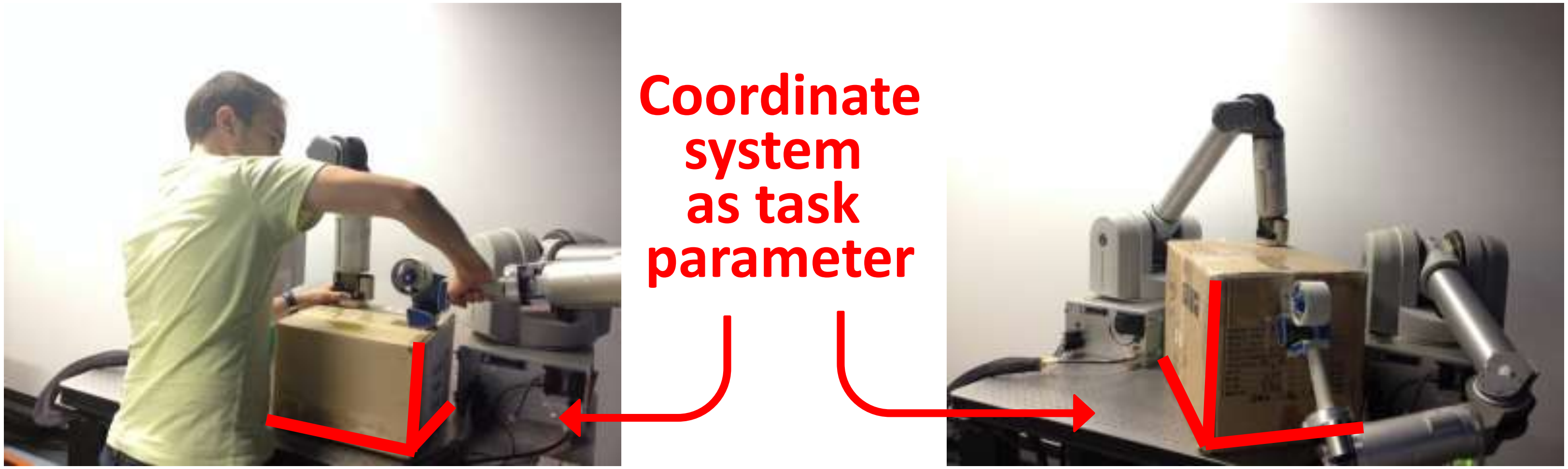
Robot side

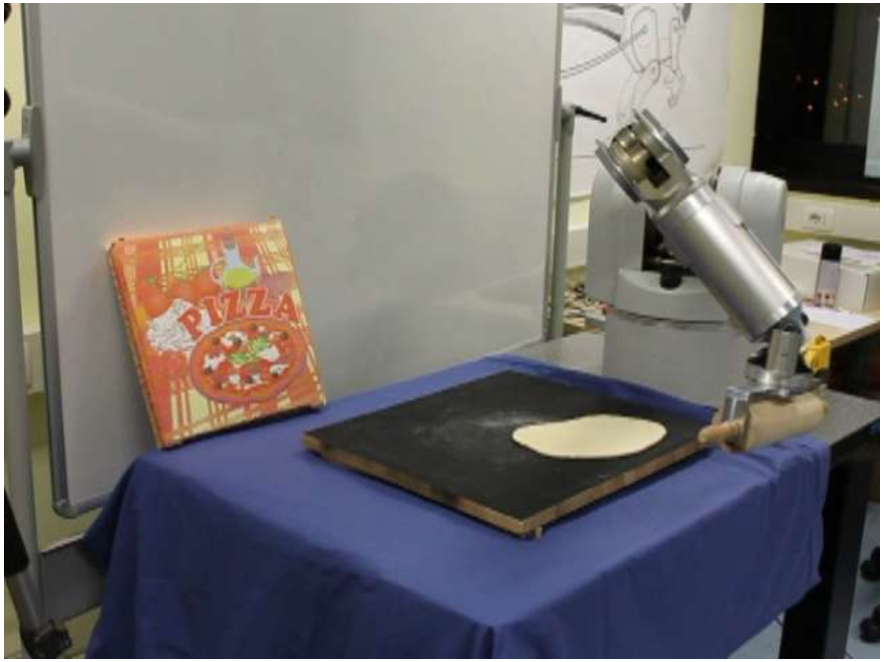


only Gaussian ID
is transmitted

[Birk *et al.*, IEEE Robotics and Autom. Magazine, 2018 (in press)] [Havoutis & Calinon, Autonomous Robots, 2018]

Adaptation to different object shapes

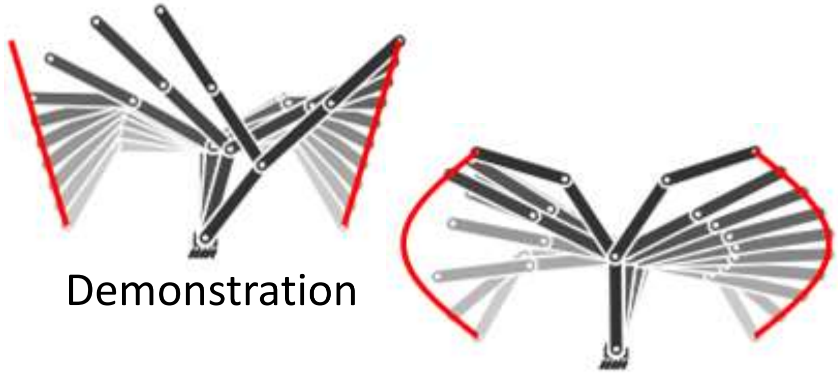




[Calinon, Alizadeh and Caldwell, IROS’2013]

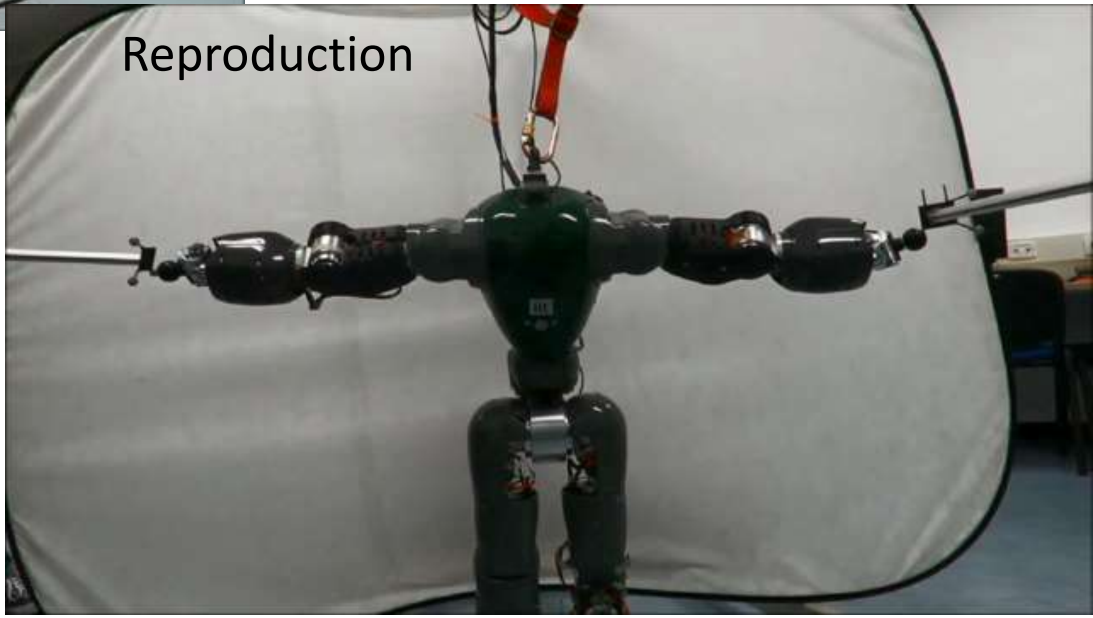
Learning & generalizing tasks prioritization

Priority on left hand



$$\hat{\dot{q}} = \underbrace{\begin{bmatrix} J_1^{\dagger} & N_1 J_2^{\dagger} \end{bmatrix}}_{\text{Candidate hierarchy } A_1} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$$

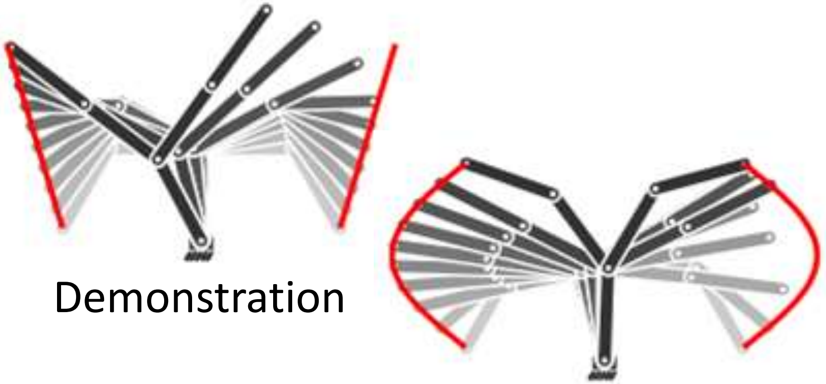
$$\hat{\dot{q}} = \underbrace{\begin{bmatrix} N_2 J_1^{\dagger} & J_2^{\dagger} \end{bmatrix}}_{\text{Candidate hierarchy } A_2} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$$



[Silvério, Calinon, Rozo and Caldwell (2018), Arxiv 1707.06791] [Calinon, ISRR'15]

Learning & generalizing tasks prioritization

Priority on right hand

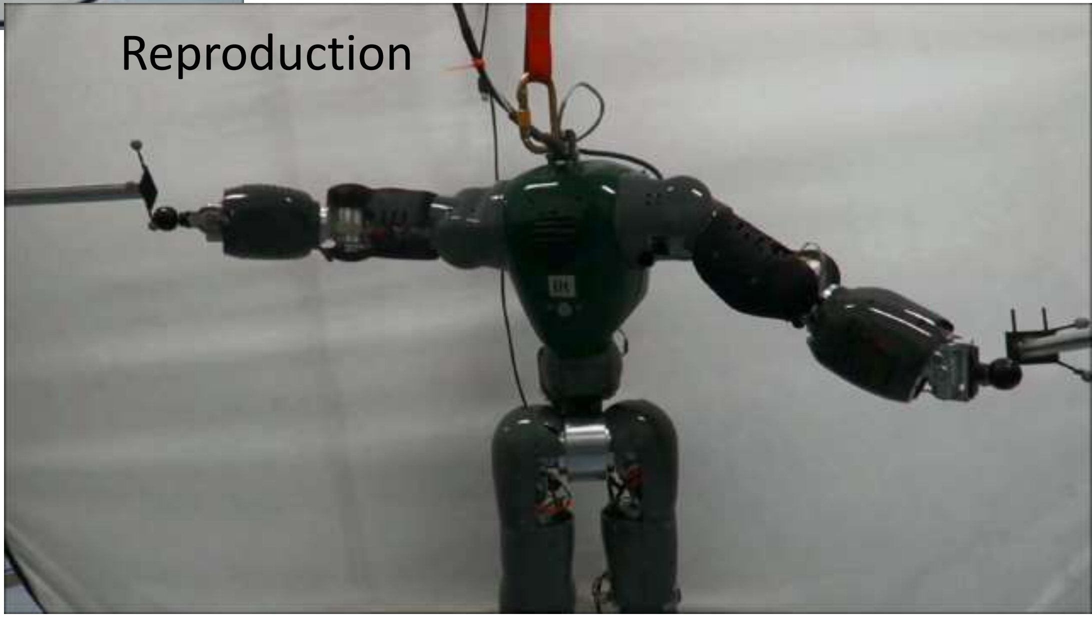


Demonstration

Reproduction

$$\hat{\dot{q}} = \underbrace{\begin{bmatrix} J_1^\dagger & N_1 J_2^\dagger \end{bmatrix}}_{\text{Candidate hierarchy } A_1} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$$

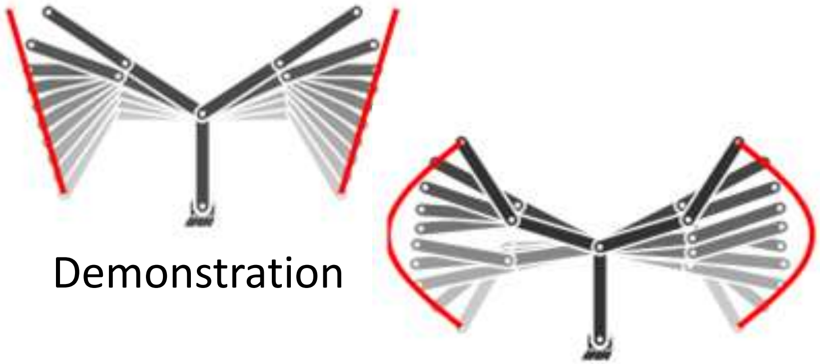
$$\hat{\dot{q}} = \underbrace{\begin{bmatrix} N_2 J_1^\dagger & J_2^\dagger \end{bmatrix}}_{\text{Candidate hierarchy } A_2} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$$



[Silvério, Calinon, Rozo and Caldwell (2018), Arxiv 1707.06791] [Calinon, ISRR'15]

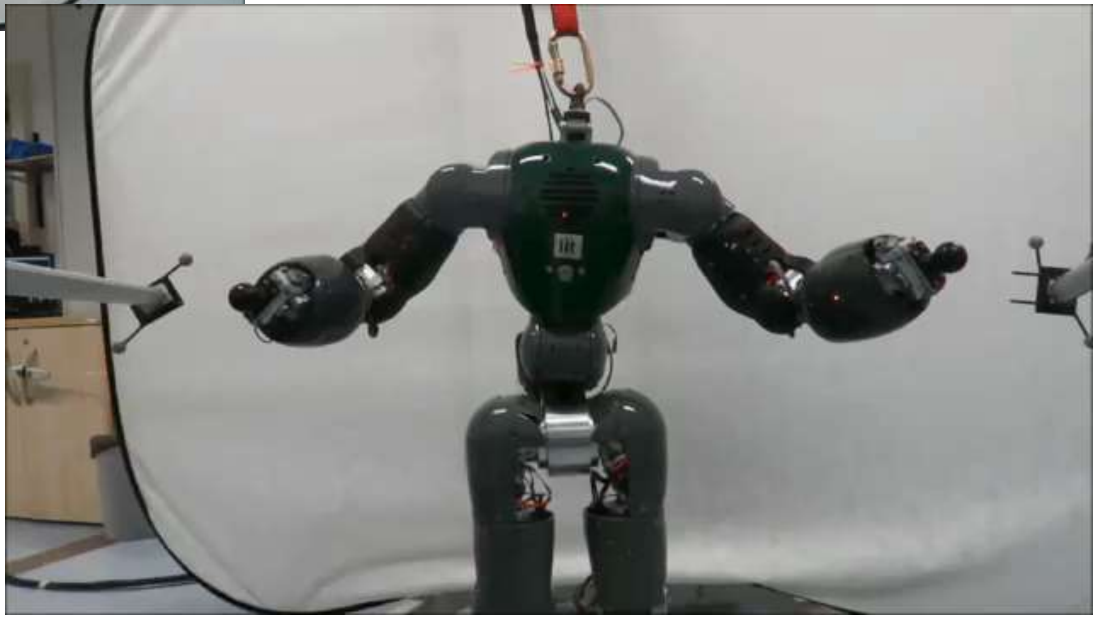
Learning & generalizing tasks prioritization

Equal priority



$$\hat{\dot{q}} = \underbrace{\begin{bmatrix} J_1^{\dagger} & N_1 J_2^{\dagger} \end{bmatrix}}_{\text{Candidate hierarchy } A_1} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$$

$$\hat{\dot{q}} = \underbrace{\begin{bmatrix} N_2 J_1^{\dagger} & J_2^{\dagger} \end{bmatrix}}_{\text{Candidate hierarchy } A_2} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$$



[Silvério, Calinon, Rozo and Caldwell (2018), Arxiv 1707.06791] [Calinon, ISRR'15]

Summary

Combination as fusion problem

- Application I: Ridge regression
- Application II: Model predictive control
- Application III: Task-parameterized models

Further extensions and open issues

- Non-Gaussian distributions
  (e.g., L1-norm instead of L2-norm)
- Multimodal distributions
  (e.g., controllers with options)
- Approximation with linearization and quadratization

References

Calinon, S. and Lee, D. (2018). **Learning Control**. Vadakkepat, P. and Goswami, A. (eds.). Humanoid Robotics: a Reference. Springer.

Calinon, S. (2016). **A Tutorial on Task-Parameterized Movement Learning and Retrieval**. Intelligent Service Robotics (Springer), 9:1, 1-29.

Calinon, S. (2016). **Stochastic learning and control in multiple coordinate systems**. Intl Workshop on Human-Friendly Robotics.

Source codes

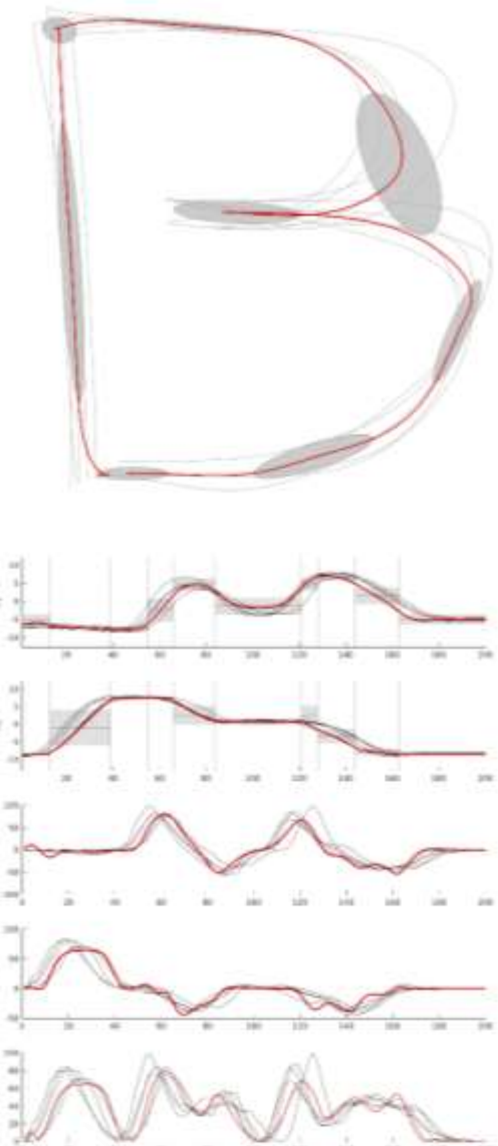
http://www.idiap.ch/software/pbdlib/

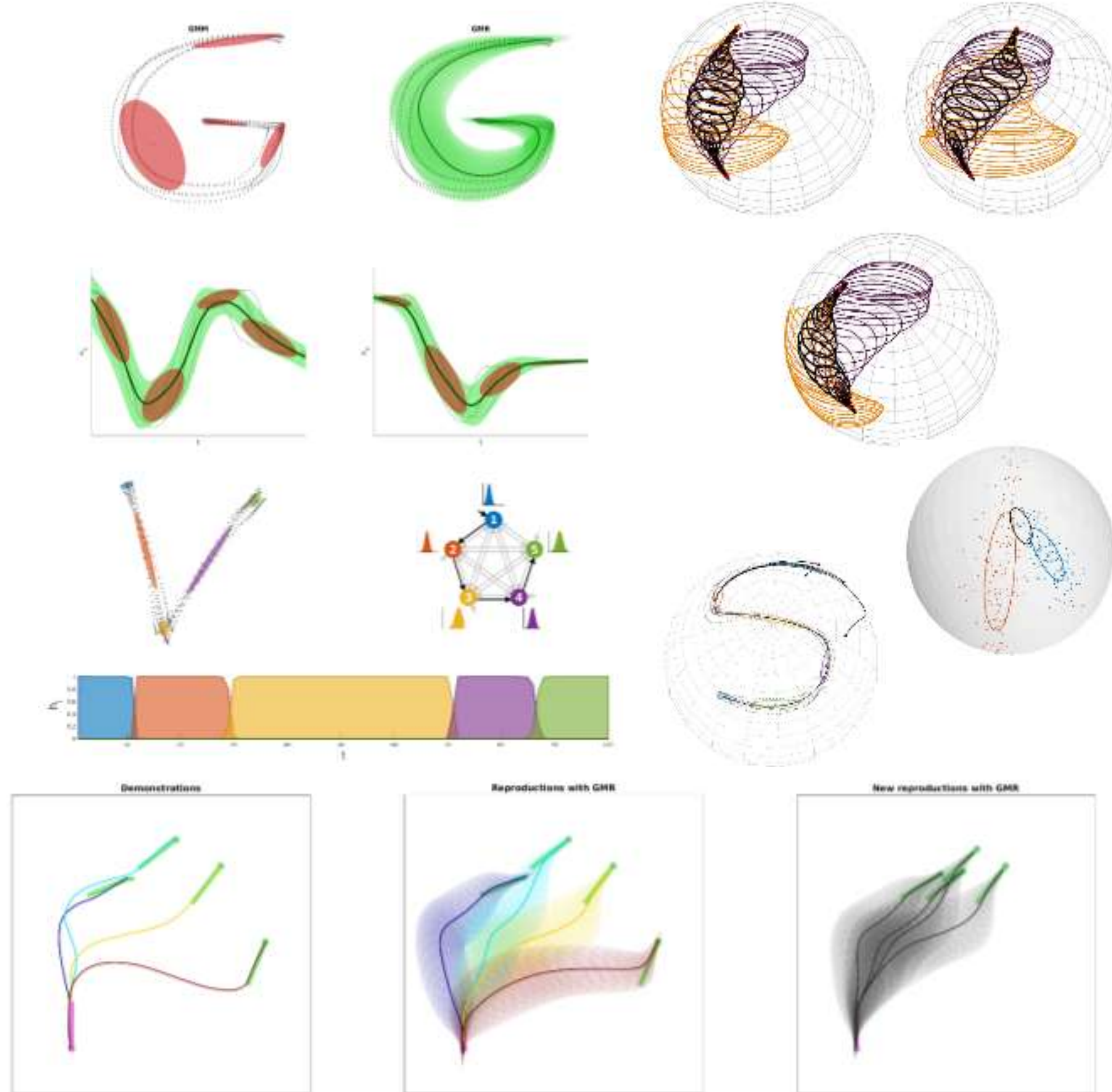
Matlab / GNU Octave

C++

Python

Pbdlib





http://www.idiap.ch/software/pbdlib/

PbDlib - Mozilla Firefox

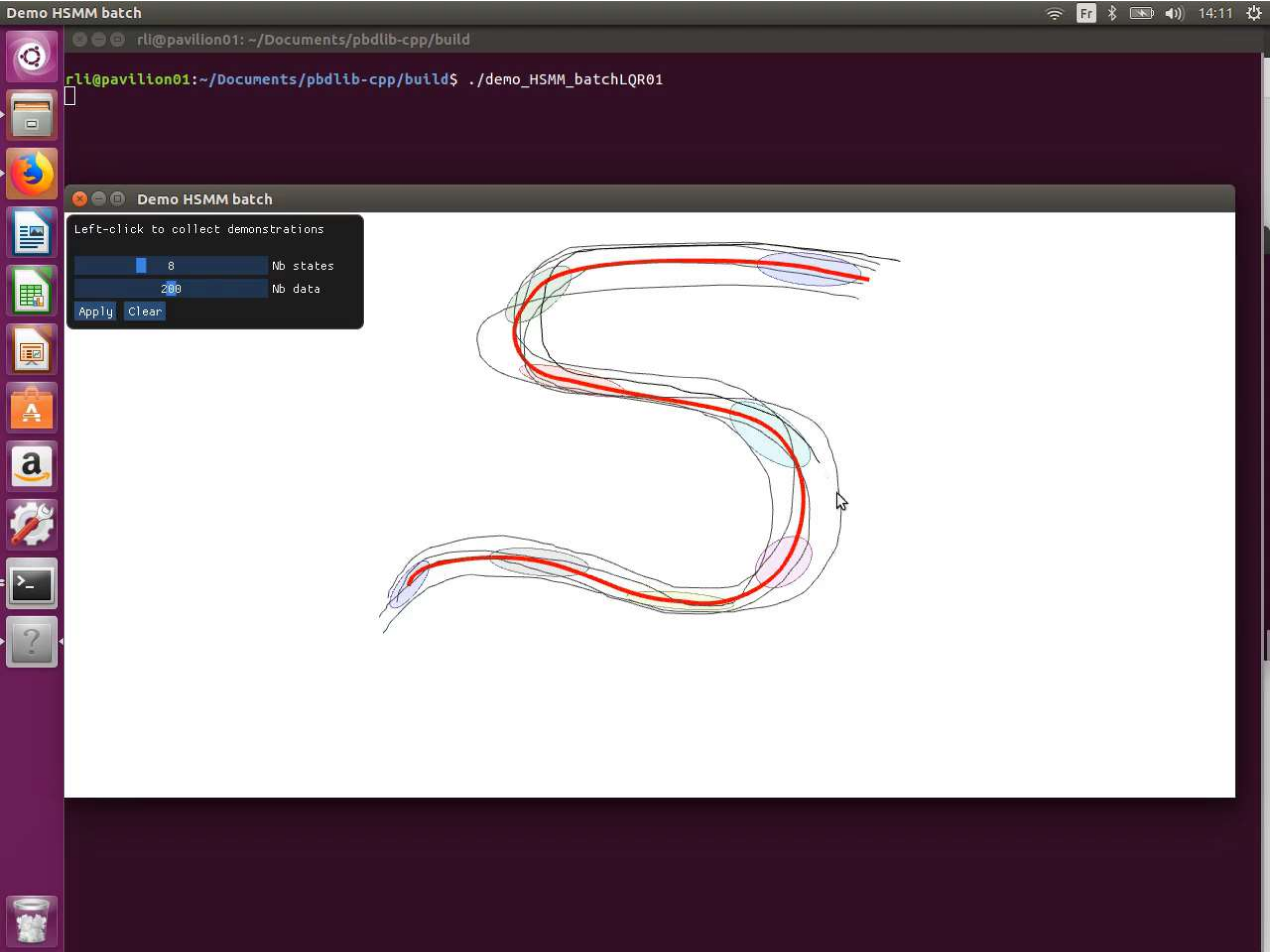
idiap.ch/software/pbdlib/

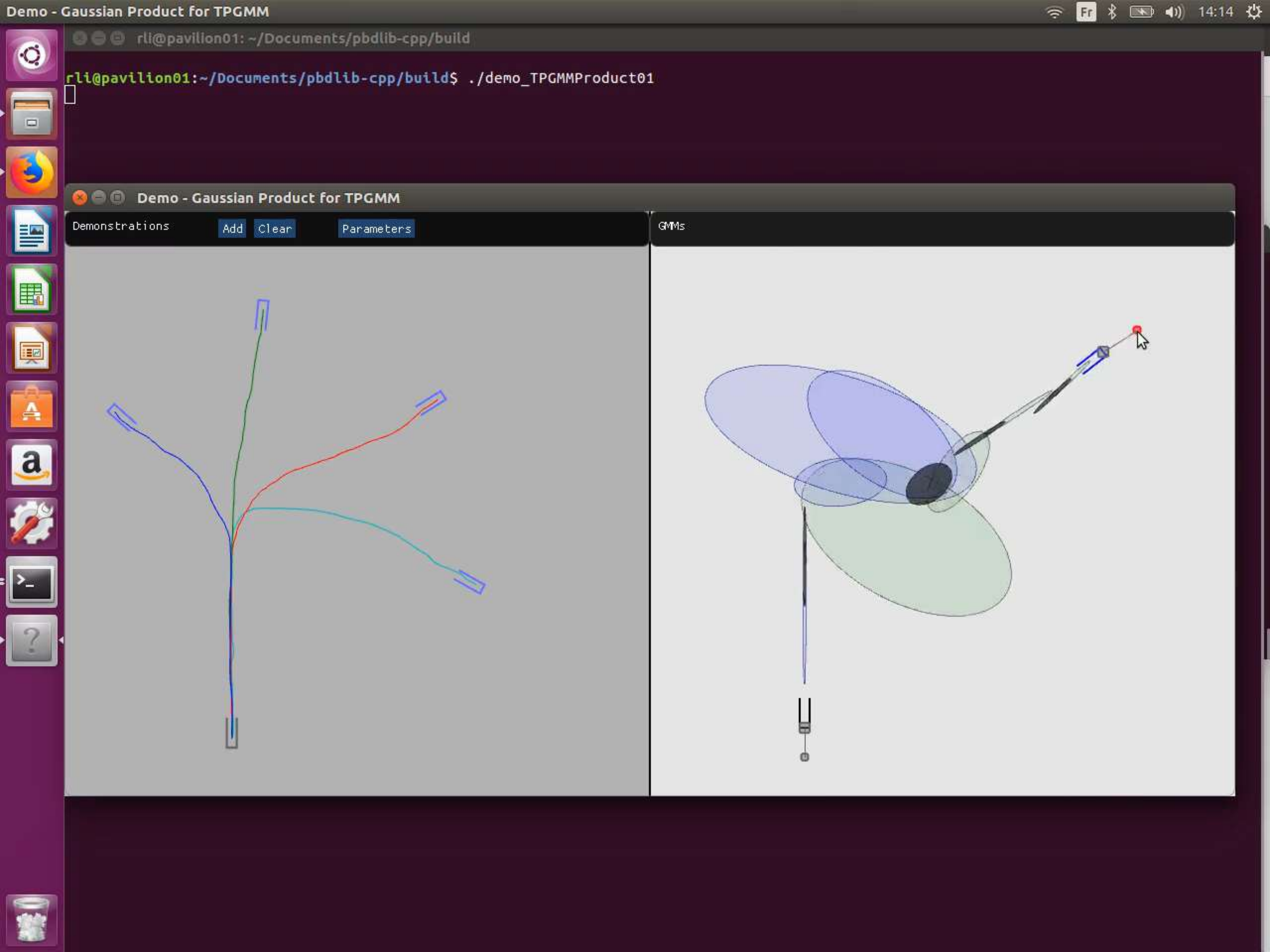
PbDlib

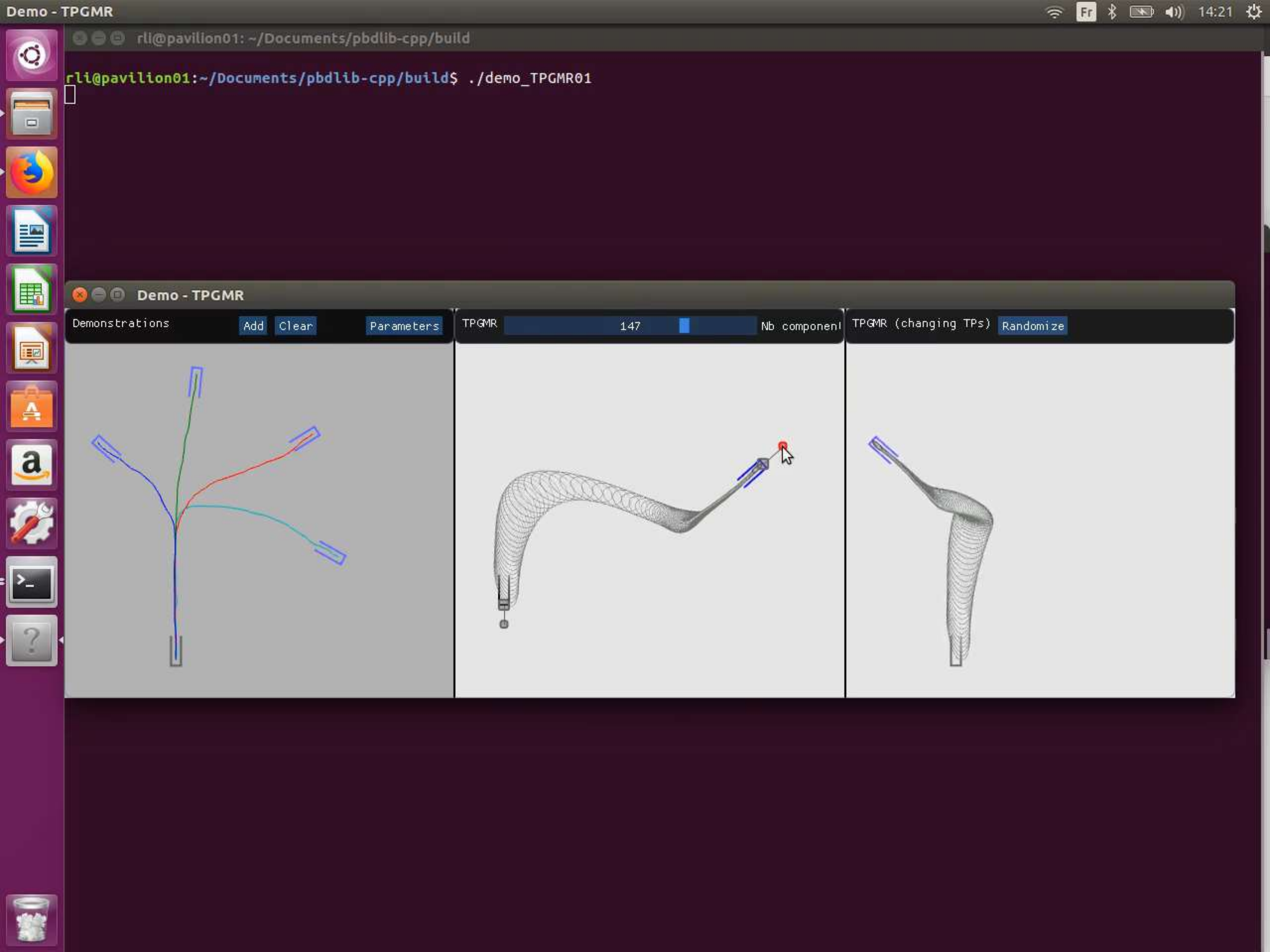
PbDlib is a collection of source codes for robot programming by demonstration (learning from demonstration). It includes a varied set of functionalities at the crossroad of statistical learning, dynamical systems, optimal control and differential geometry. It is available in the following languages:

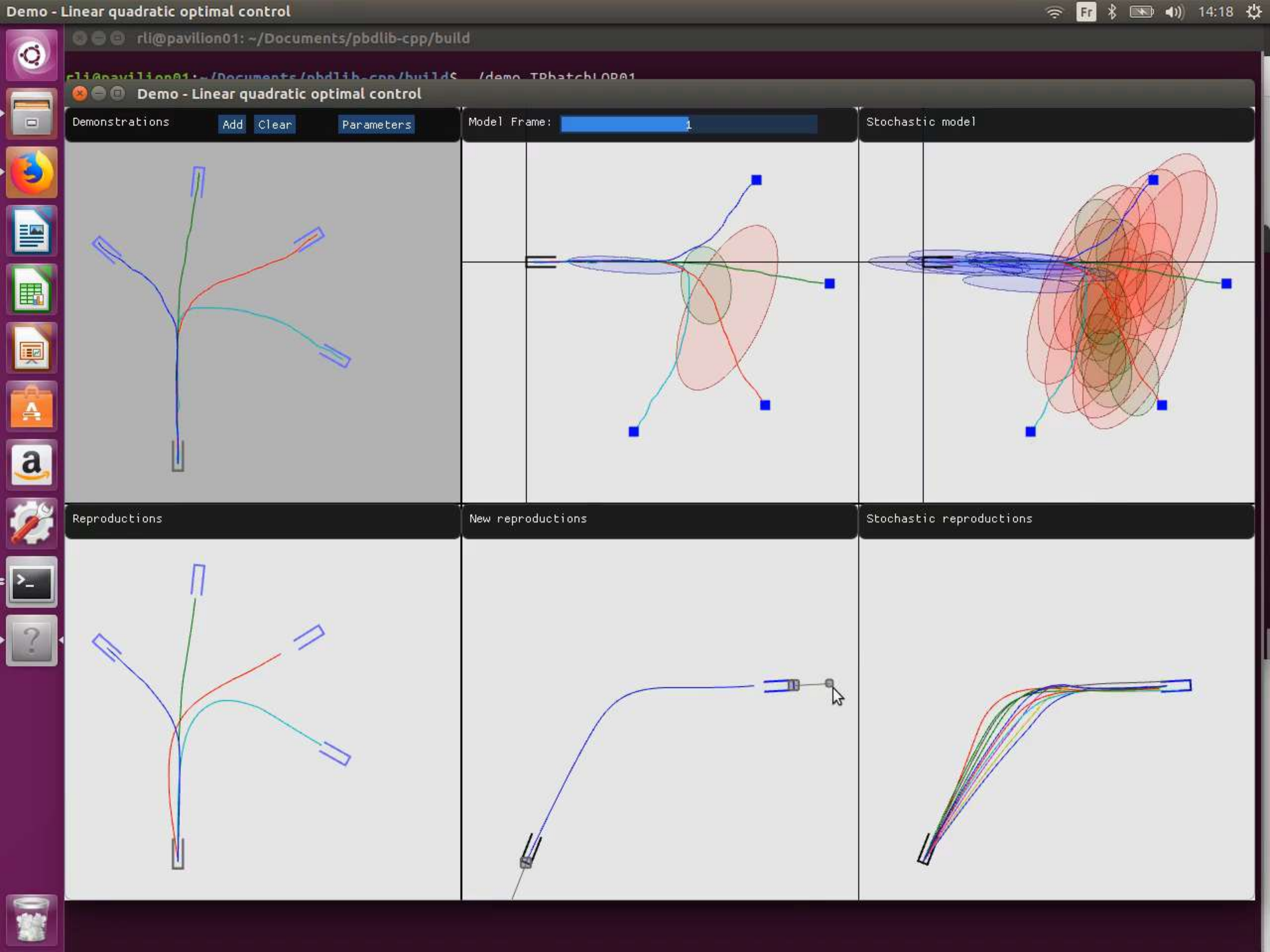
- Matlab / GNU Octave
- C++
- Python

PbDlib can be used in applications requiring task adaptation, human-robot skill transfer, safe controllers based on minimal intervention principle, as well as for probabilistic motion analysis and synthesis in multiple coordinate systems.


Demo HSMM batch
14:11
rli@pavilion01: ~/Documents/pbdlib-cpp/build
rli@pavilion01:~/Documents/pbdlib-cpp/build$ ./demo_HSMM_batchLQR01
Demo HSMM batch
Left-click to collect demonstrations
8
Nb states
200
Nb data
Apply
Clear


Demo - Gaussian Product for TPGMM
14:14
rli@pavilion01: ~/Documents/pbdlib-cpp/build
rli@pavilion01:~/Documents/pbdlib-cpp/build$ ./demo_TPGMMProduct01
Demo - Gaussian Product for TPGMM
Demonstrations
Add
Clear
Parameters
GMMs


Demo - TPGMR
14:21
rli@pavilion01: ~/Documents/pbdlib-cpp/build
rli@pavilion01:~/Documents/pbdlib-cpp/build$ ./demo_TPGMR01
Demo - TPGMR
Demonstrations
Add
Clear
Parameters
TPGMR
147
Nb componen
TPGMR (changing TPs)
Randomize


Demo - Linear quadratic optimal control
14:18
rli@pavilion01: ~/Documents/pbdlib-cpp/build
Demo - Linear quadratic optimal control
Demonstrations
Add
Clear
Parameters
Model Frame:
1
Stochastic model
Reproductions
New reproductions
Stochastic reproductions

Source codes (Matlab/Octave, C++ and Python):

http://www.idiap.ch/software/pbdlib/

Contact:

sylvain.calinon@idiap.ch

http://calinon.ch

Photo: Basilio Noris