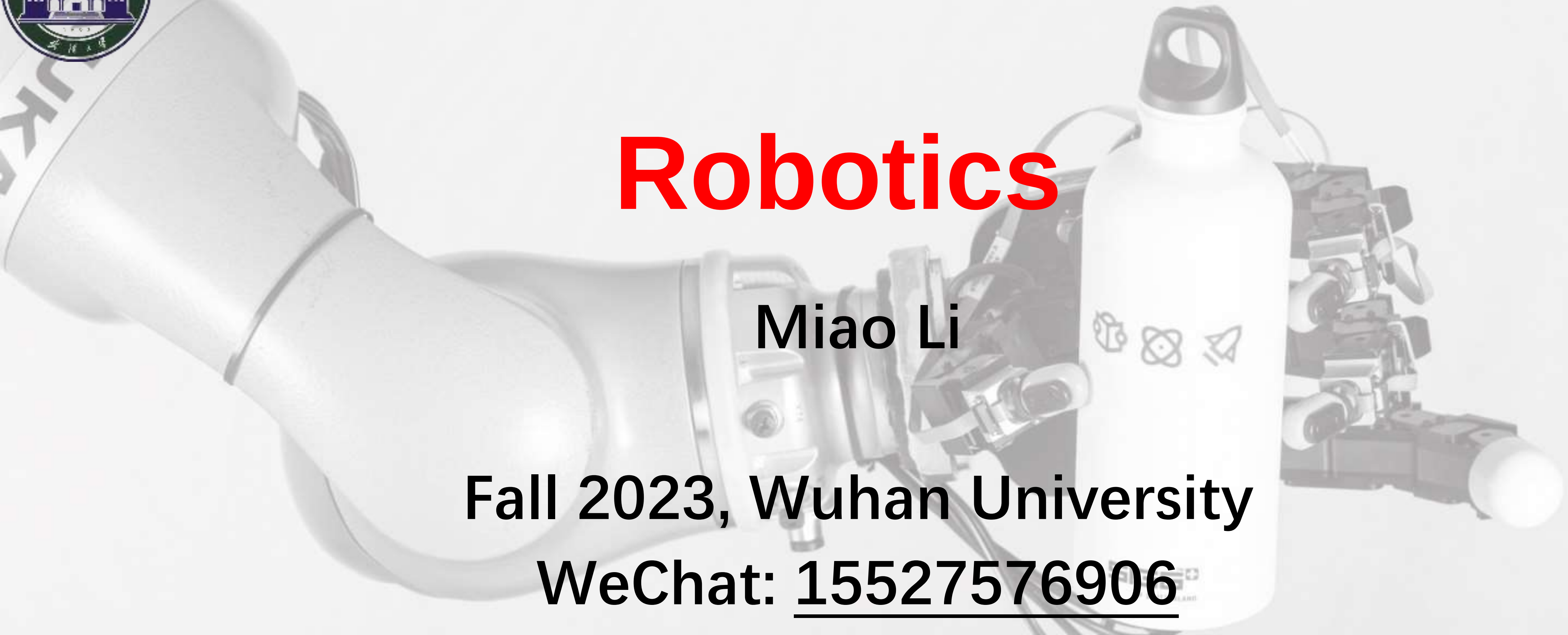


Robotics

Miao Li

Fall 2023, Wuhan University

WeChat: 15527576906

Email: limiao712@gmail.com

2023-11-6

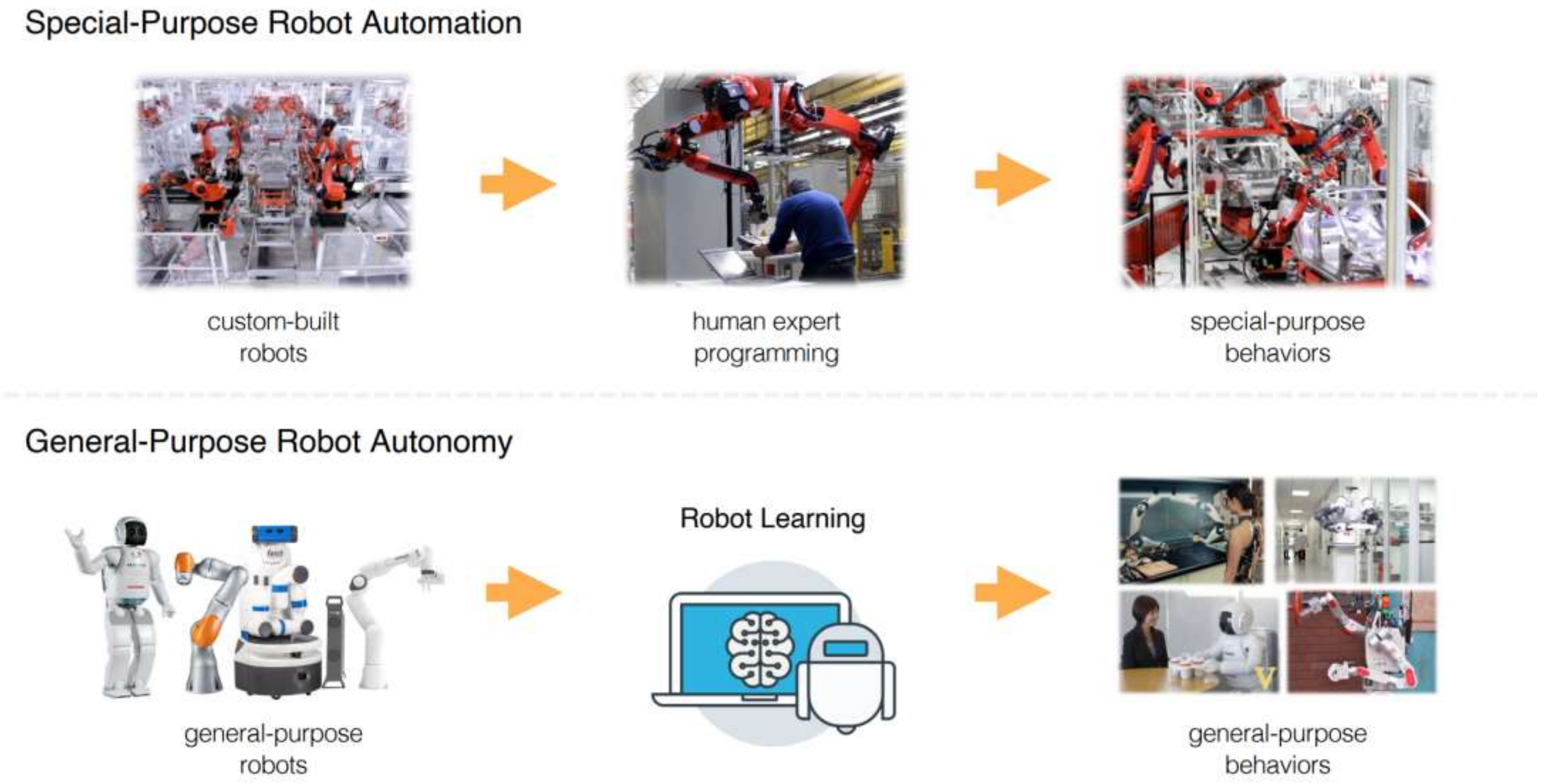
Goal for this course

- **Design: soft hand design x1**
- **Perception: vision, point cloud, tactile, force/torque x1**
- **Planning: sampling-based, optimization-based, learning-based x3**
- **Control: feedback, multi-modal x2**
- **Learning: imitation learning, RL x2**
- **Simulation tool (pybullet, matlab, OpenRAVE, Issac Nvidia, Gazebo)**
- **How to get a robot moving!**

Today agenda

- **Paper reading (~30 mins)**
- **Why imitation learning (IL) (~5)**
- **Key ingredients of IL (~5)**
- **Data collection (~5)**
- **Learning algorithms (~20)**
- **Limits of IL (~5)**
- **Examples and applications (~20)**
  - **Motion**
  - **Hand IK**
  - **Force-relevant task**
  - **Multi-modal task**

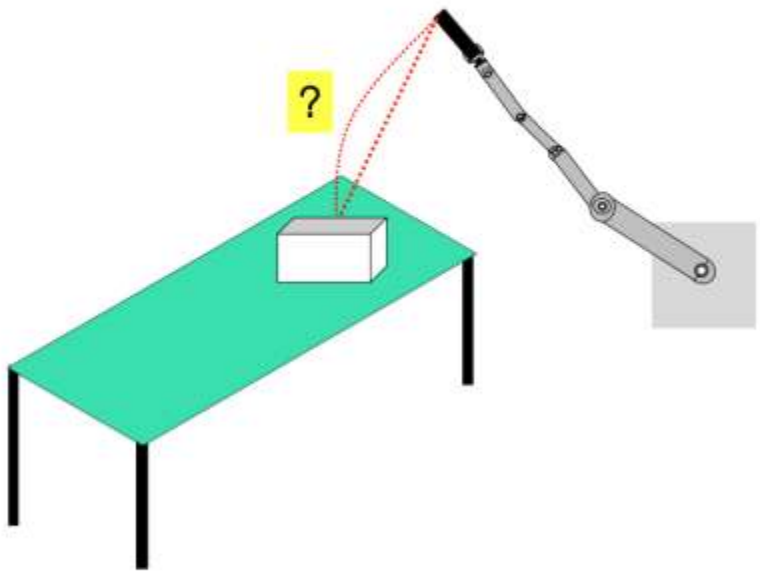
Why imitation learning?



Why imitation learning?

Motivation

How can we learn optimal controllers to perform a task from data?



EPFL

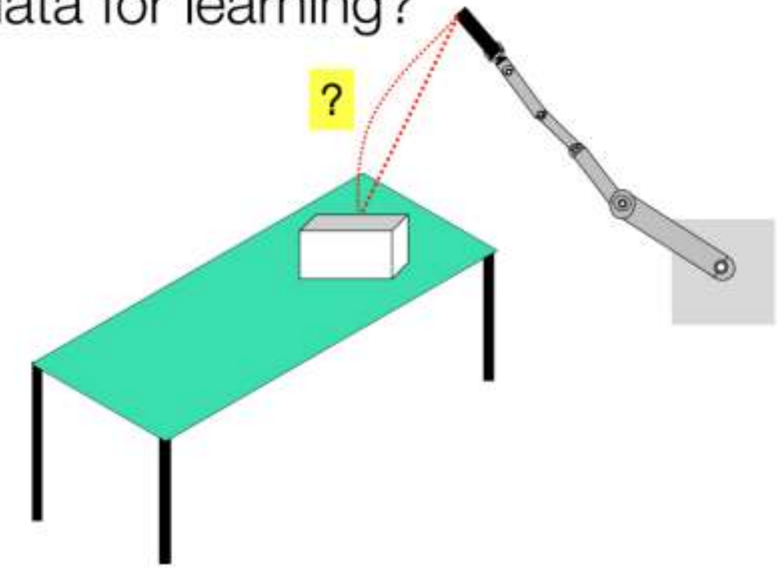
Billard A., Calinon S., Dillmann R., Schaal S. (2008) Robot Programming by Demonstration. In: Siciliano B., Khatib O. (eds) Springer Handbook of Robotics. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-30301-5_60

Why imitation learning?

Motivation

How can we learn optimal controllers to perform a task from data?

- Use data-driven approaches to learn optimal controllers
- How do we gather data for learning?



EPFL

Billard A., Calinon S., Dillmann R., Schaal S. (2008) Robot Programming by Demonstration. In: Siciliano B., Khatib O. (eds) Springer Handbook of Robotics. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-30301-5_60

Why imitation learning?

Learning is critical for getting robots to work in the real world.

object variation

environment uncertainty

adaptation

https://www.cs.utexas.edu/~yukez/cs391r_fall2020/slides/lecture_intro.pdf

Why imitation learning?

Robots should have the ability to learn skills and adapt these skills to new scenarios.

https://sites.google.com/view/icml2018-imitation-learning/

Why imitation learning?

Imitation is a crucial aspect of skill development, because it allows us to learn new things quickly and efficiently by watching those around us. Most children learn everything from gross motor movements, to speech, to interactive play skills by watching parents, caregivers, siblings, and peers perform these behaviors.

https://www.mayinstitute.org/news/acl/asd-and-dd-child-focused/what-is-imitation-and-why-is-it-important/#:~:text=Imitation%20is%20a%20crucial%20aspect,and%20peers%20perform%20these%20behaviors.

Why imitation learning?

Imitation Learning in a Nutshell

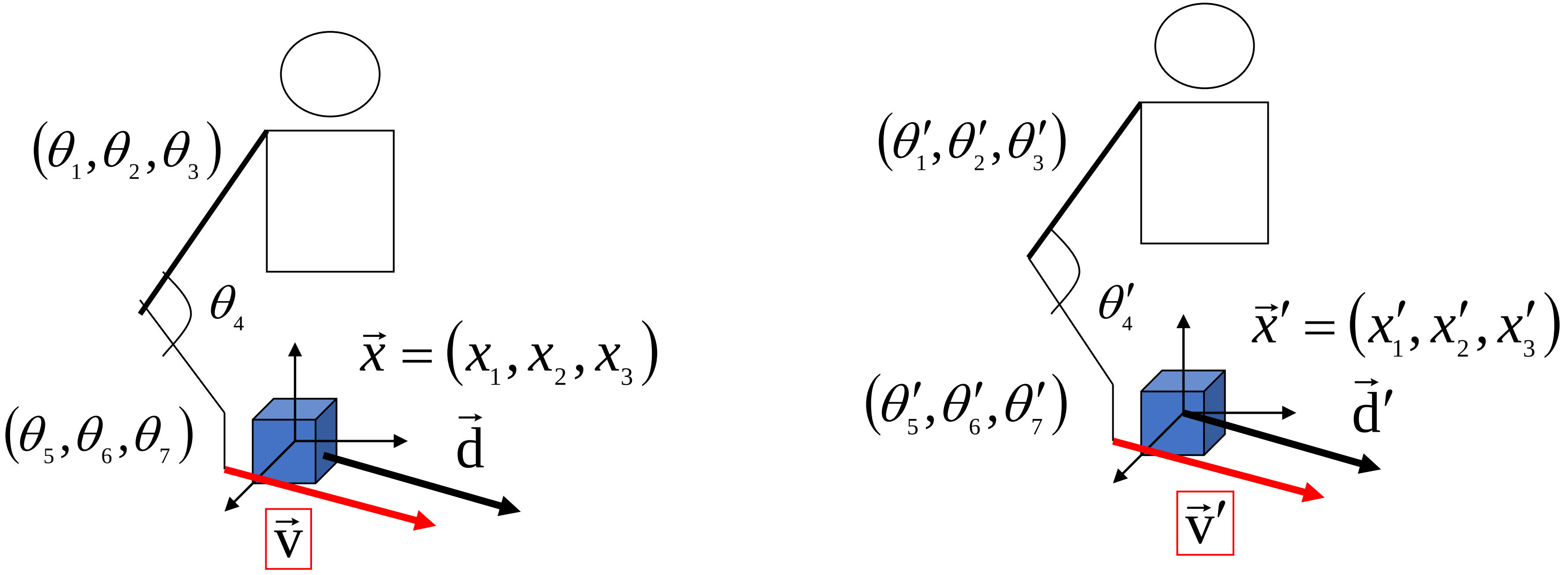
Given: demonstrations or demonstrator

Goal: train a policy to mimic demonstrations

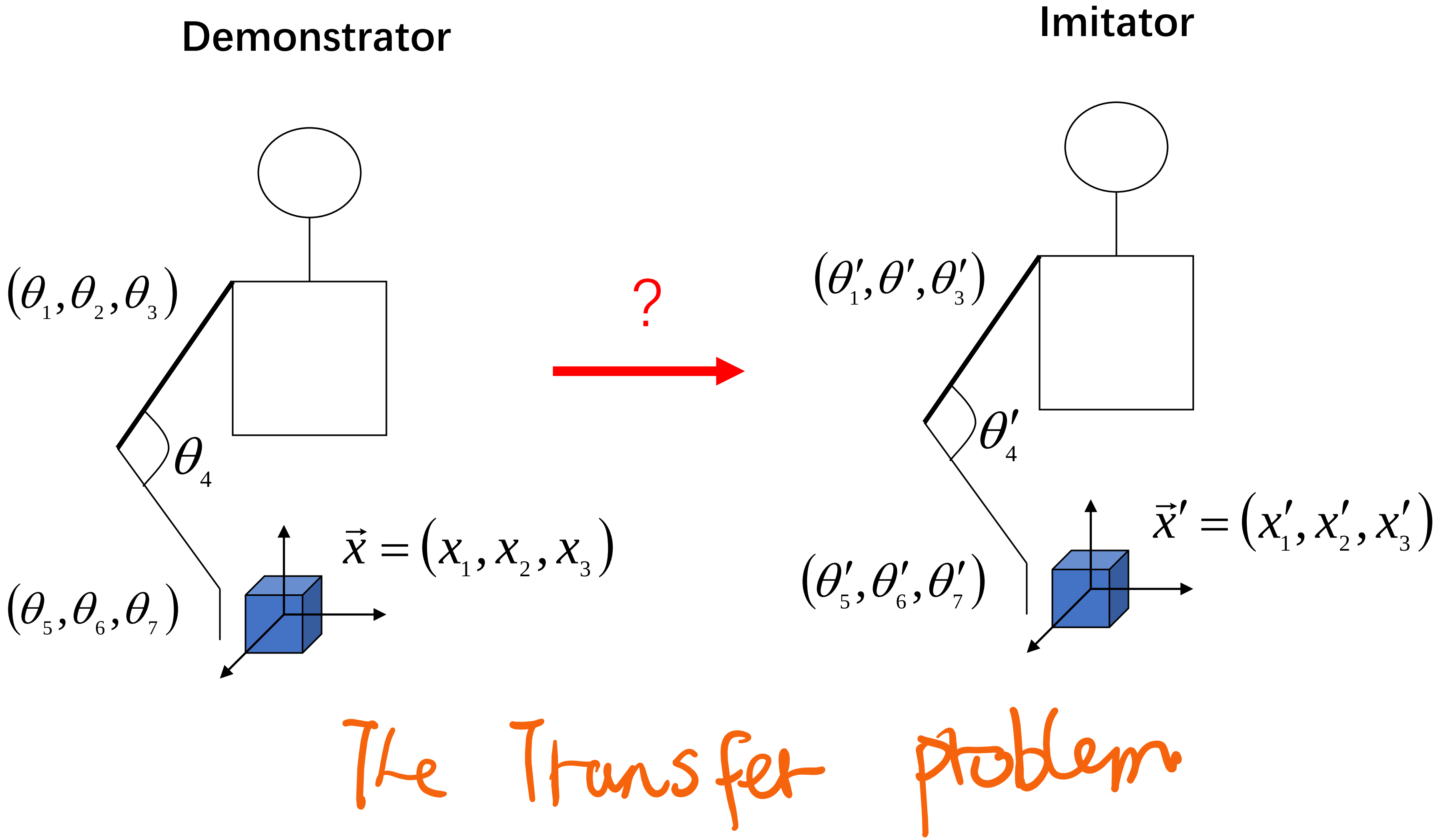
Images from Stephane Ross

Imitation learning

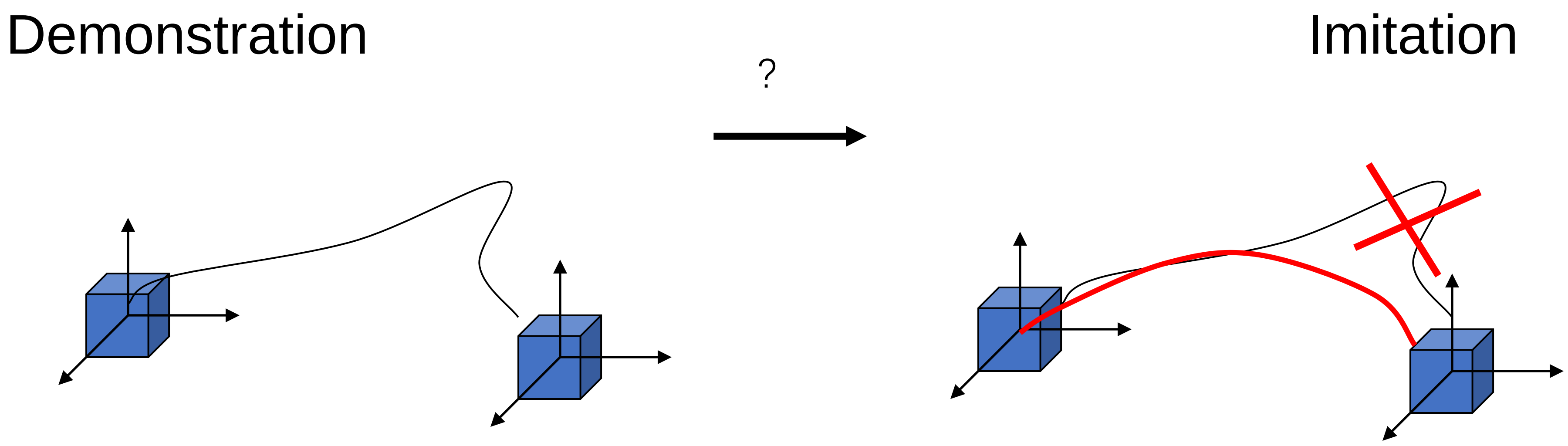
$\vec{x} = \vec{x}'$	**Same Object, same target location**
$\vec{d} = \vec{d}'$	**Same direction of motion**
$\vec{v} = \vec{v}'$	**Same speed, same force**
$\vec{\theta} = \vec{\theta}'$	**Same posture**



Imitation learning



Imitation learning

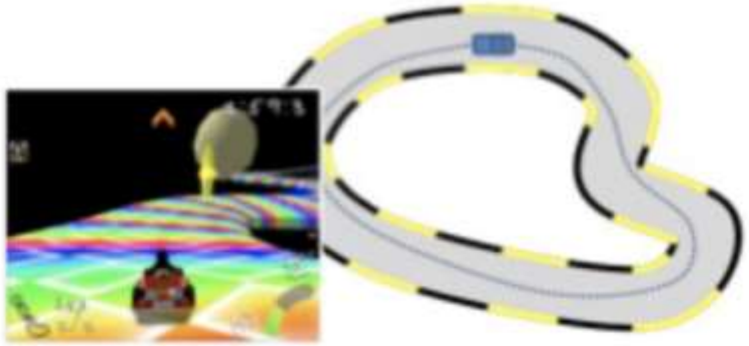


No solutions (smaller range of motion)

→ Find the closest solution according to a metric

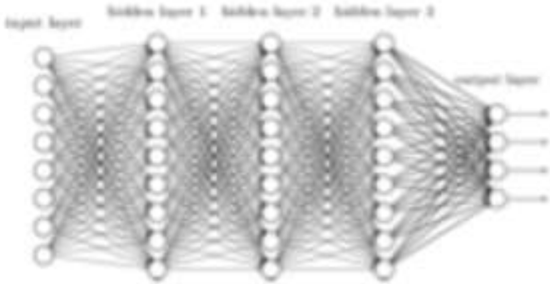
How to Imitate?
The correspondence problem

Key ingredients of IL

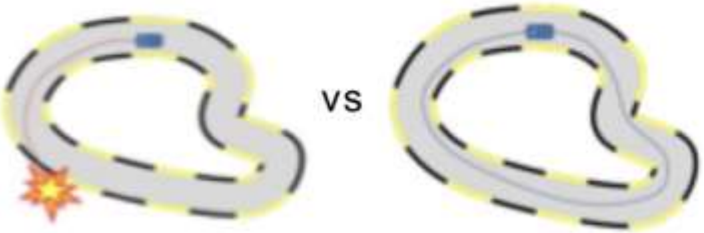


(1) Demonstrations or Demonstrator

Environment / Simulator (2)

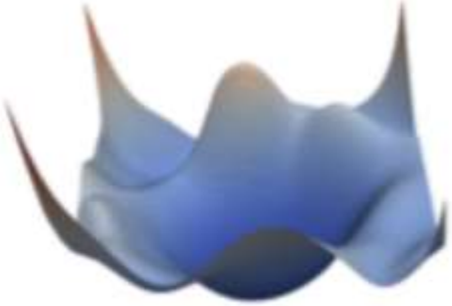


Policy Class (3)



Loss Function (4)

+



Learning Algorithm (5)

Key ingredients of IL

Considerations

Learning human skills through LFD requires the following questions:

- What/Who to imitate?
- How to imitate?
- When to imitate?

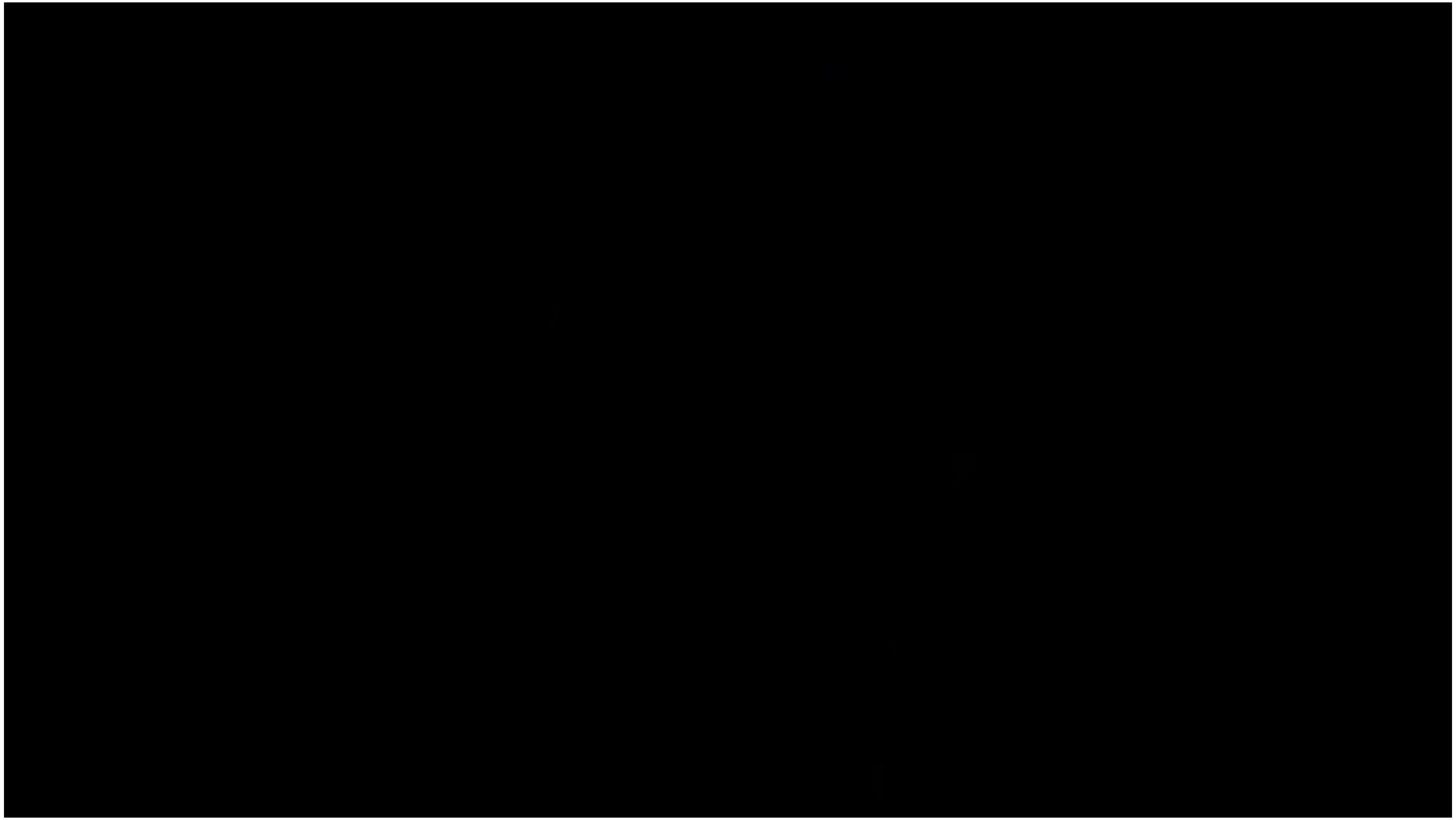
Key ingredients of IL

Demonstrator

Teleoperation
+
Data glove

Key ingredients of IL

Demonstrator



Key ingredients of IL

Demonstrator

Teleoperation
+
EMG

Key ingredients of IL

Demonstrator

Teleoperation

Teleoperation Interfaces

- Graphical user interface/Tablet
- Joysticks
- More complex devices (e.g., exoskeleton)

Davinci Surgical Robot

Key ingredients of IL

Demonstrator

Kinesthetic

https://www.youtube.com/channel/UCqnvGUfdlr94mddDQamEBGA

Key ingredients of IL

Demonstrator

Kinesthetic

\+

Tactile

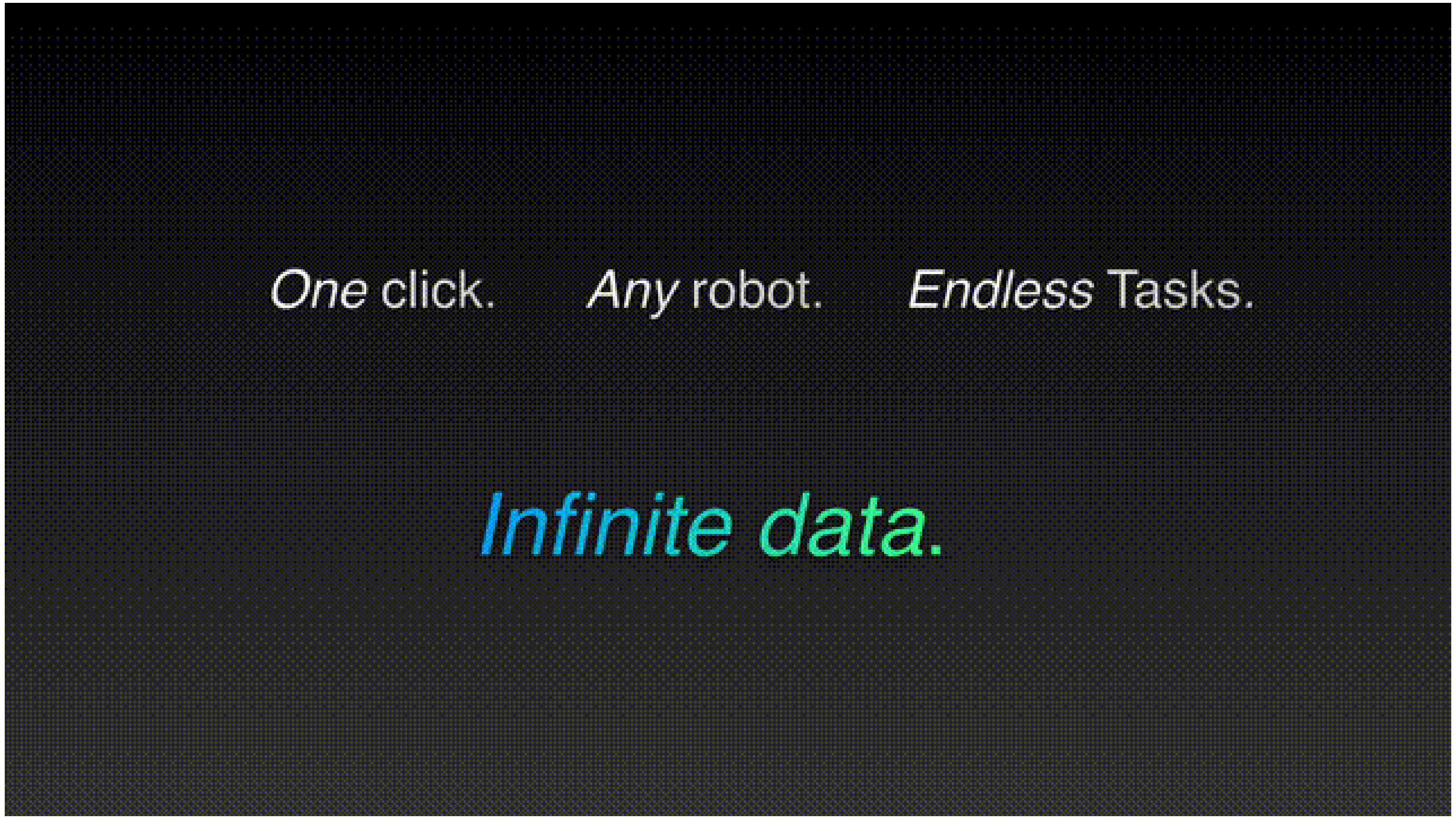
Key ingredients of IL

Demonstrator

Simulation

CMU清华MIT引爆全球首个Agent无限流，机器人「007」加班自学停不下来！具身智能被革命

Original 新智元 新智元 2023-11-04 14:27 Posted on 北京



Computer Science > Robotics

[Submitted on 2 Nov 2023]

RoboGen: Towards Unleashing Infinite Data for Automated Robot Learning via Generative Simulation

Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, Chuang Gan

Key ingredients of IL

Demonstrator

Simulation

Key ingredients of IL

Demonstrator

Motion capture

Key ingredients of IL

Demonstrator

Motion capture

Key ingredients of IL

Demonstrator

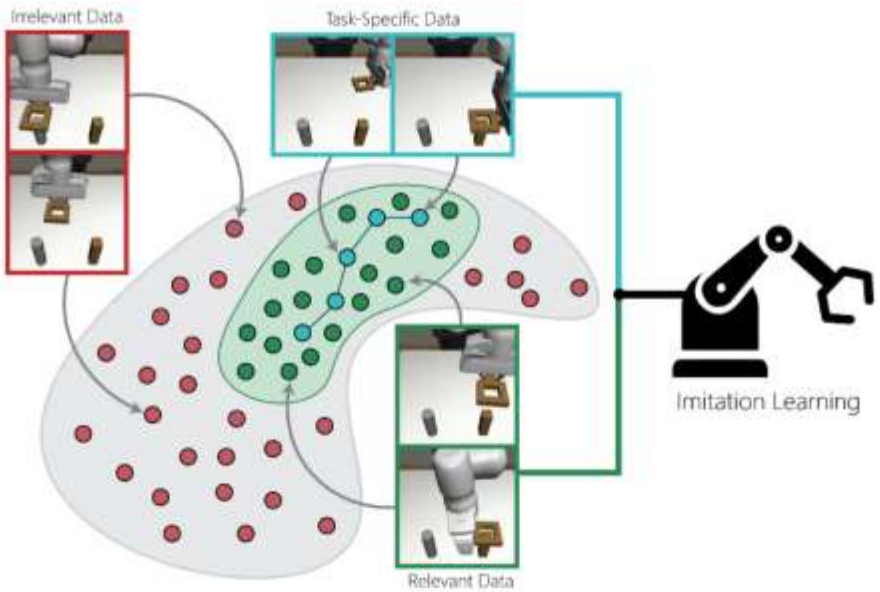
Web video

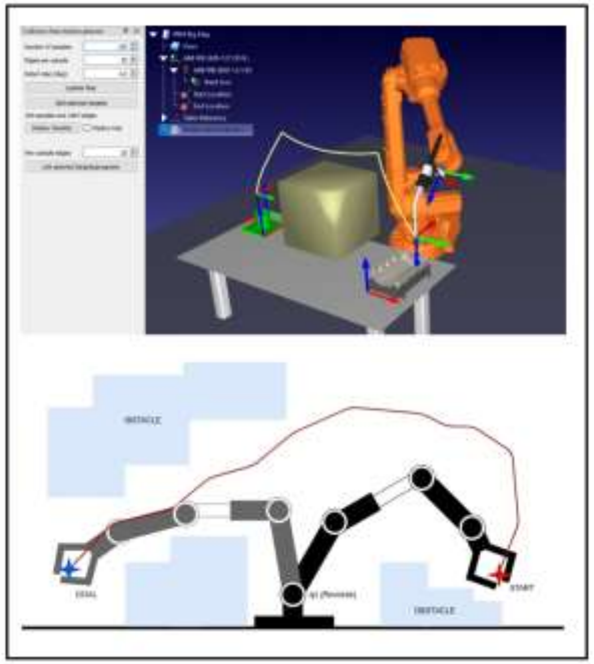
Salt Bae
7.2M views

POV Chefs Cooking 500+ Meals #food #chef #cookin...
19K views

Difference between teppanyaki and hibachi

Data collection





Task distribution

Data collection

Imitation learning is very good at in-distribution tasks, but not so good at out-distribution tasks.

Data collection

or exp design

- **Task variations**
- **Environments**
- **Demonstrator variance**
- **Invariant relation**

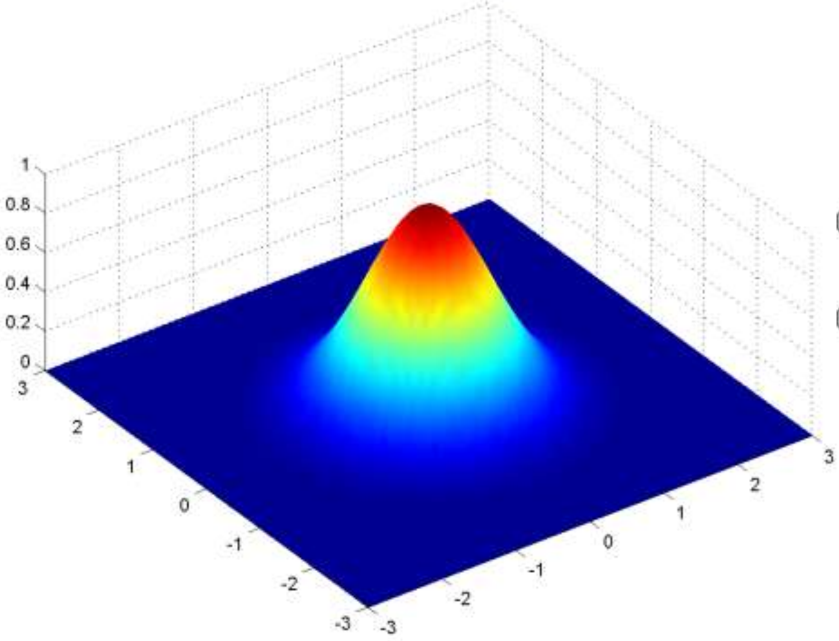
We need to design the exps according to these rules.

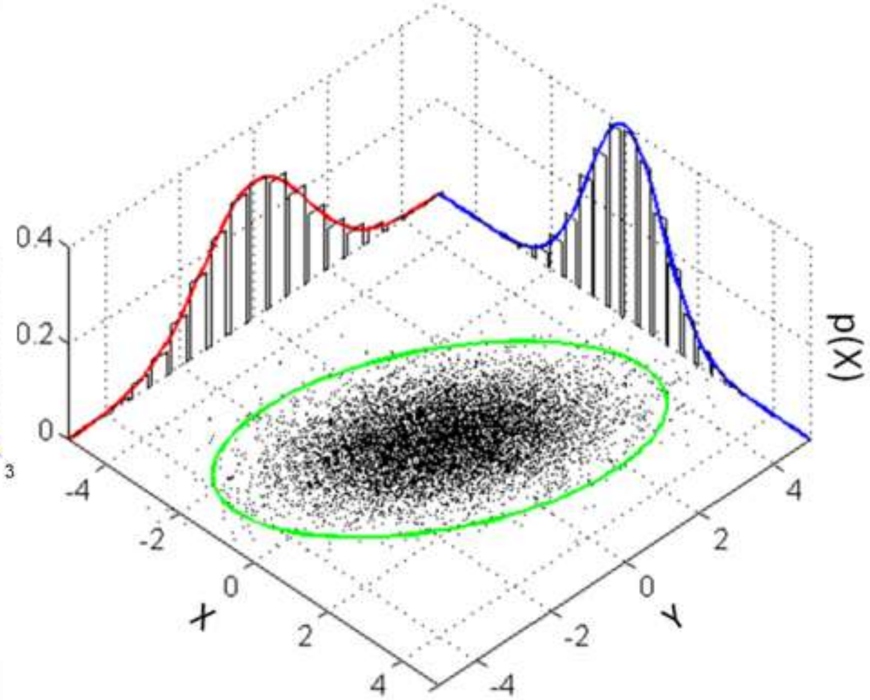
Learning algorithms

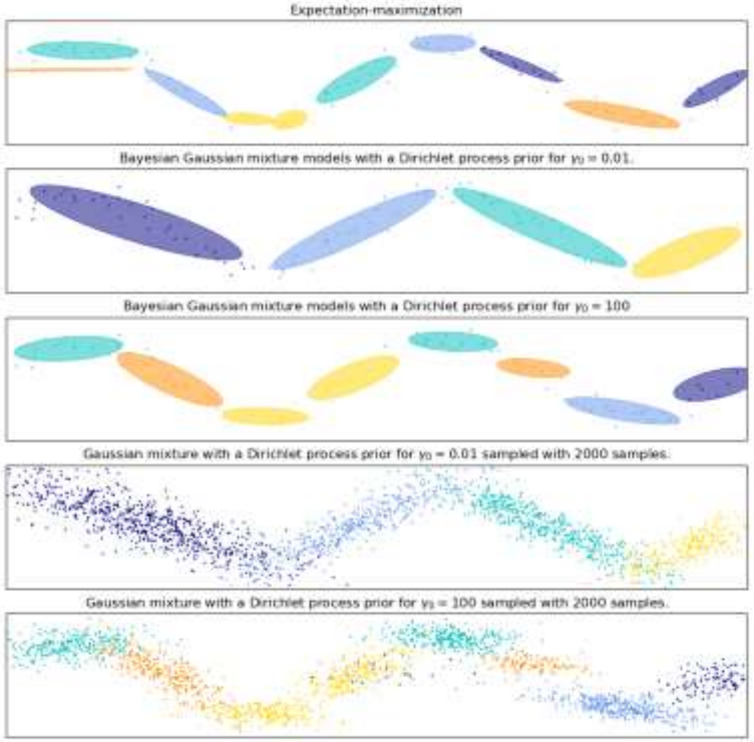
GMM

- Recall the Gaussian distribution:

$$P(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})\right)$$







Learning algorithms

Multivariate Gaussian distribution

Univariate Gaussian distribution:

$$\mathcal{N}(\mu, \sigma^2) = (2\pi\sigma^2)^{-\frac{1}{2}} \boxed{\exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)}$$ Radial basis function (RBF)

$x \in \mathbb{R}$	Datapoint	
$\mu \in \mathbb{R}$	Center (or mean)	Parameters $\{\mu, \sigma^2\}$
$\sigma^2 \in \mathbb{R}$	Variance	

Multivariate Gaussian distribution:

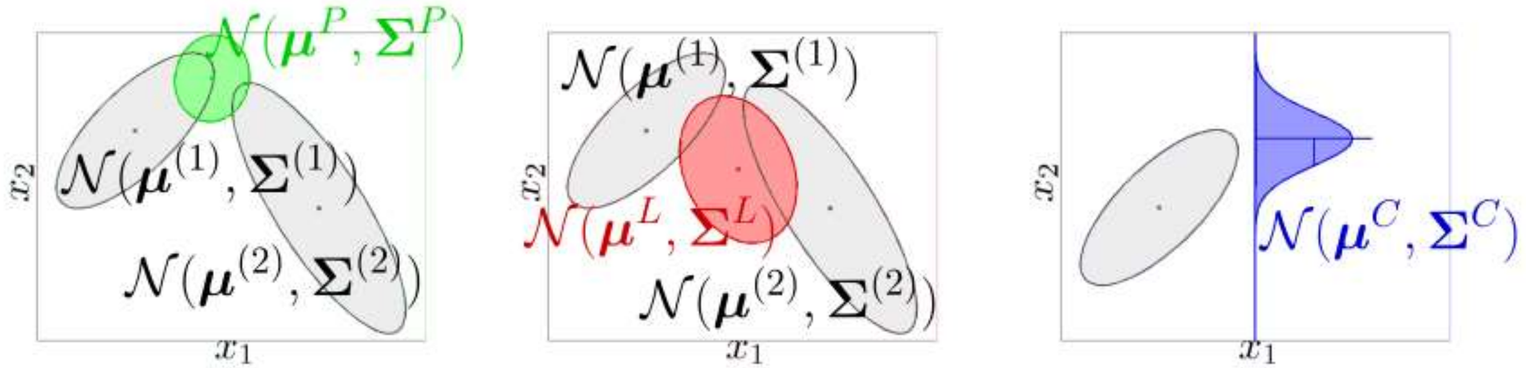
$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

$\boldsymbol{x} \in \mathbb{R}^D$	Datapoint	
$\boldsymbol{\mu} \in \mathbb{R}^D$	Center (or mean)	Parameters $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$
$\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$	Covariance matrix	

https://calinon.ch/misc/EE613/EE613-nonlinearRegression.pdf

Learning algorithms

Properties of Gaussian distributions



Linear combination:

$$\mathcal{N}(\boldsymbol{\mu}^L, \boldsymbol{\Sigma}^L) \sim \frac{1}{2}\,\mathcal{N}(\boldsymbol{\mu}^{(1)}, \boldsymbol{\Sigma}^{(1)}) + \frac{1}{2}\,\mathcal{N}(\boldsymbol{\mu}^{(2)}, \boldsymbol{\Sigma}^{(2)})$$

Product of Gaussians:

$$c\,\mathcal{N}(\boldsymbol{\mu}^P, \boldsymbol{\Sigma}^P) \sim \mathcal{N}(\boldsymbol{\mu}^{(1)}, \boldsymbol{\Sigma}^{(1)}) \cdot \mathcal{N}(\boldsymbol{\mu}^{(2)}, \boldsymbol{\Sigma}^{(2)})$$

Conditional probability:

$$\mathcal{N}(\boldsymbol{\mu}^C, \boldsymbol{\Sigma}^C) \sim \mathcal{P}(\boldsymbol{x}_2 | \boldsymbol{x}_1)$$

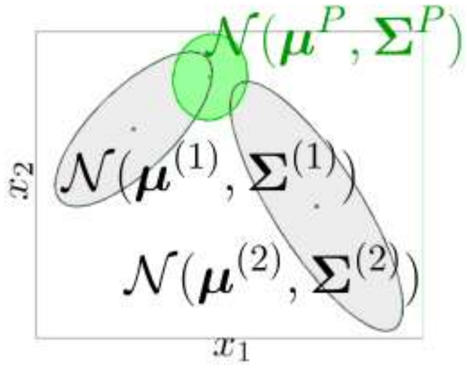
Learning algorithms

Product of Gaussians

The product of two Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}^{(1)}, \boldsymbol{\Sigma}^{(1)})$ and $\mathcal{N}(\boldsymbol{\mu}^{(2)}, \boldsymbol{\Sigma}^{(2)})$ is defined by

$$c\,\mathcal{N}(\boldsymbol{\mu}^{P}, \boldsymbol{\Sigma}^{P}) = \mathcal{N}(\boldsymbol{\mu}^{(1)}, \boldsymbol{\Sigma}^{(1)}) \cdot \mathcal{N}(\boldsymbol{\mu}^{(2)}, \boldsymbol{\Sigma}^{(2)}),$$

$$\begin{aligned} \text{with} \qquad c &= \mathcal{N}(\boldsymbol{\mu}^{(1)} | \boldsymbol{\mu}^{(2)}, \boldsymbol{\Sigma}^{(1)} + \boldsymbol{\Sigma}^{(2)}), \\ \boldsymbol{\Sigma}^{P} &= \left(\boldsymbol{\Sigma}^{(1)^{-1}} + \boldsymbol{\Sigma}^{(2)^{-1}}\right)^{-1}, \\ \boldsymbol{\mu}^{P} &= \boldsymbol{\Sigma}^{P}\left(\boldsymbol{\Sigma}^{(1)^{-1}}\boldsymbol{\mu}^{(1)} + \boldsymbol{\Sigma}^{(2)^{-1}}\boldsymbol{\mu}^{(2)}\right). \end{aligned}$$



Learning algorithms

Conditional probability

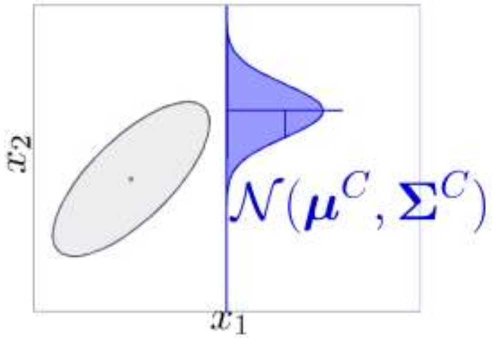
Let $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be defined by

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix}, \; \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \; \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}.$$

The conditional probability $\mathcal{P}(\boldsymbol{x}_2|\boldsymbol{x}_1)$ is defined by

$$\mathcal{P}(\boldsymbol{x}_2|\boldsymbol{x}_1) \;\sim\; \mathcal{N}(\boldsymbol{\mu}^C, \boldsymbol{\Sigma}^C),$$

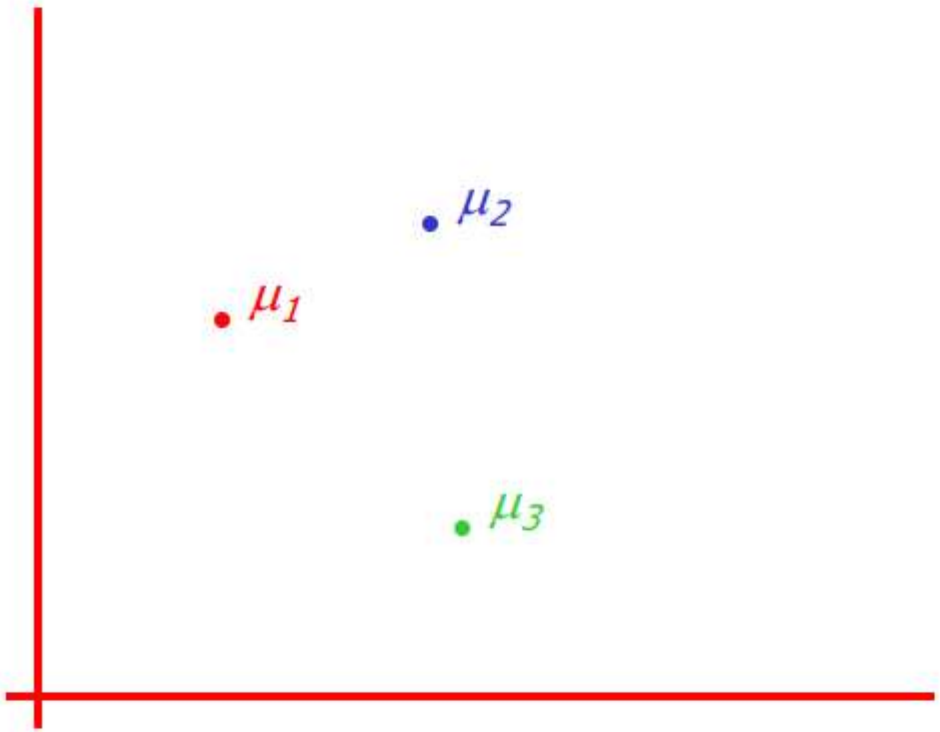
$$\text{with} \quad \begin{aligned} \boldsymbol{\mu}^C &= \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}(\boldsymbol{\Sigma}_{11})^{-1}(\boldsymbol{x}_1 - \boldsymbol{\mu}_1), \\ \boldsymbol{\Sigma}^C &= \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}(\boldsymbol{\Sigma}_{11})^{-1}\boldsymbol{\Sigma}_{12}. \end{aligned}$$



Learning algorithms

The GMM assumption

- There are k components. The i' th component is called ω_i
- Component ω_i has an associated mean vector μ_i

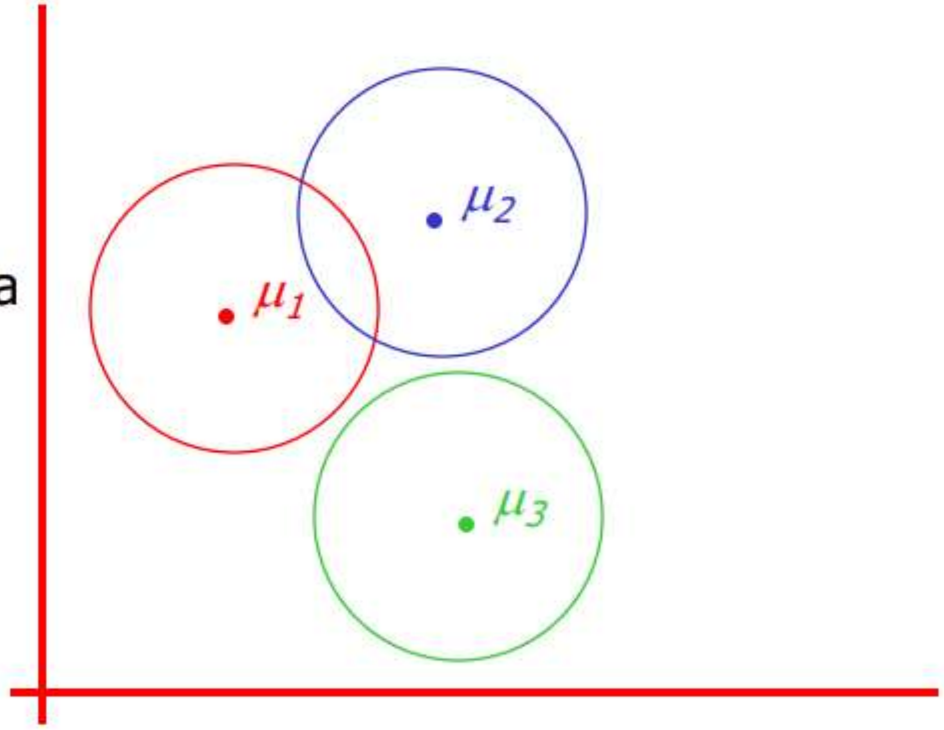


Learning algorithms

The GMM assumption

- There are k components. The i'th component is called ω_i
- Component ω_i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 \boldsymbol{I}$

Assume that each datapoint is generated according to the following recipe:



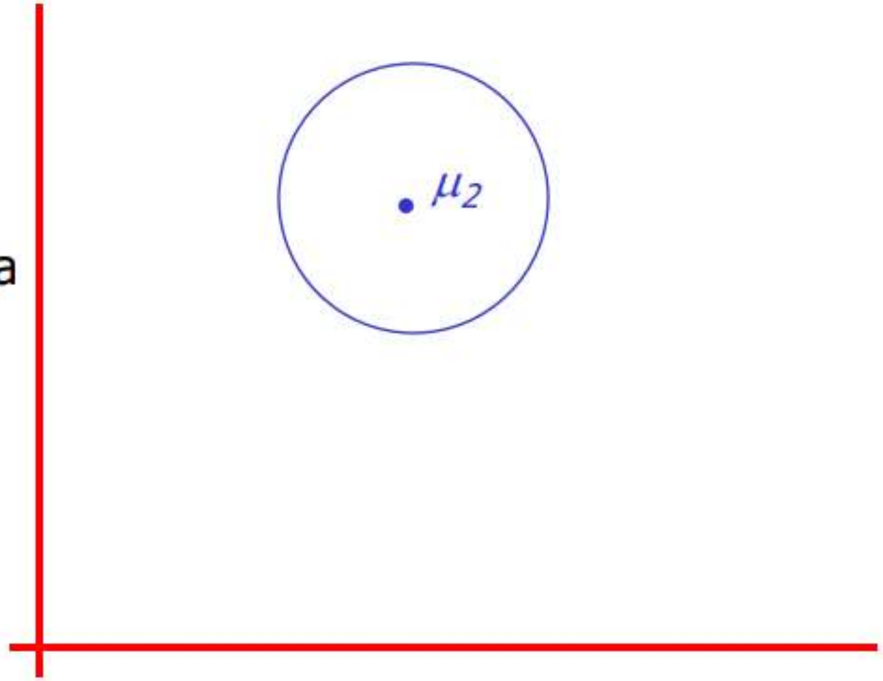
Learning algorithms

The GMM assumption

- There are k components. The i' th component is called ω_i
- Component ω_i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 \boldsymbol{I}$

Assume that each datapoint is generated according to the following recipe:

1. Pick a component at random: choose component *i* with probability $P(\omega_i)$.



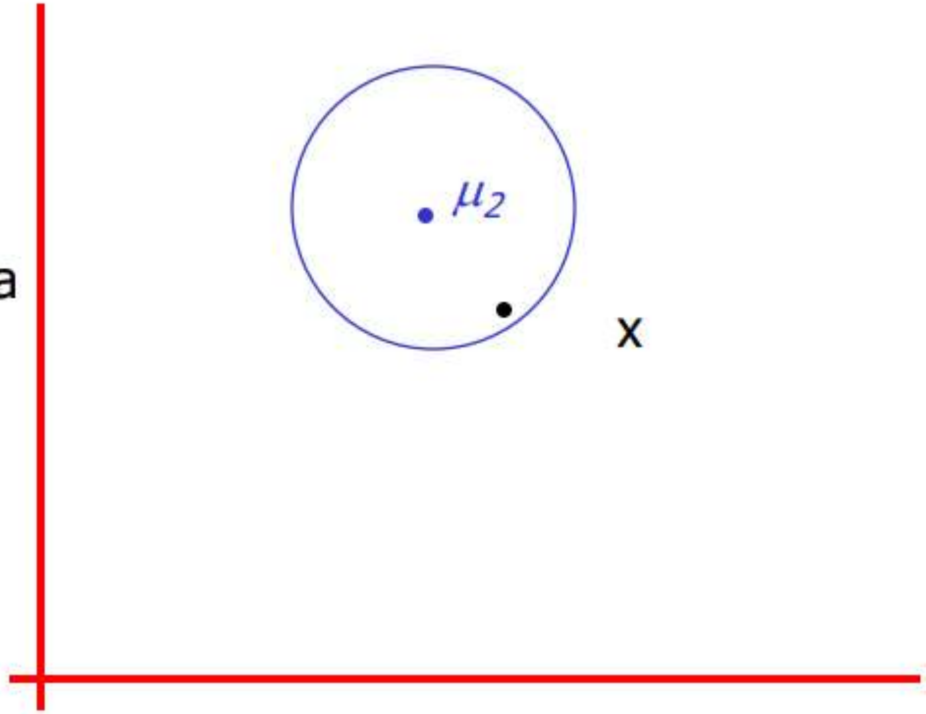
Learning algorithms

The GMM assumption

- There are k components. The i' th component is called ω_i
- Component ω_i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix $\sigma^2 \boldsymbol{I}$

Assume that each datapoint is generated according to the following recipe:

1. Pick a component at random: choose component *i* with probability $P(\omega_i)$.
2. Datapoint ~ $N(\mu_i, \sigma^2 \boldsymbol{I})$



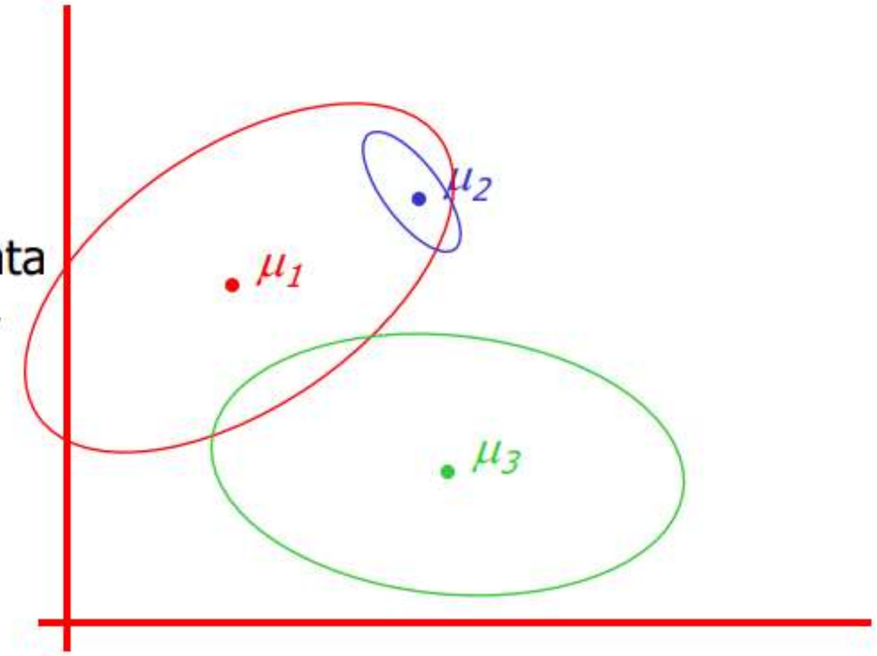
Learning algorithms

The General GMM assumption

- There are k components. The i' th component is called ω_i
- Component ω_i has an associated mean vector μ_i
- Each component generates data from a Gaussian with mean μ_i and covariance matrix Σ_i

Assume that each datapoint is generated according to the following recipe:

1. Pick a component at random: choose component i with probability $P(\omega_i)$.
2. Datapoint ~ N(μ_i, Σ_i)



Learning algorithms

Mixture Models

- Formally a Mixture Model is the weighted sum of a number of pdfs where the weights are determined by a distribution π

$$p(x) = \pi_0 f_0(x) + \pi_1 f_1(x) + \pi_2 f_2(x) + \ldots + \pi_k f_k(x)$$

where $\sum_{i=0}^{k} \pi_i = 1$

$$\boxed{p(x) = \sum_{i=0}^{k} \pi_i f_i(x)}$$

Learning algorithms

Gaussian Mixture Models

- GMM: the weighted sum of a number of Gaussians where the weights are determined by a distribution π

$$p(x) = \pi_0 N(x|\mu_0, \Sigma_0) + \pi_1 N(x|\mu_1, \Sigma_1) + \ldots + \pi_k N(x|\mu_k, \Sigma_k)$$

$$\text{where } \sum_{i=0}^{k} \pi_i = 1$$

$$\boxed{p(x) = \sum_{i=0}^{k} \pi_i N(x|\mu_k, \Sigma_k)}$$

Learning algorithms

E.M. for General GMMs

$p_i(t)$ is shorthand for estimate of $P(\omega_i)$ on t' th iteration

Iterate. On the t' th iteration let our estimates be

$$\lambda_t = \{ \mu_1(t), \mu_2(t) \dots \mu_c(t), \Sigma_1(t), \Sigma_2(t) \dots \Sigma_c(t), p_1(t), p_2(t) \dots p_c(t) \}$$

① E-step: Compute "expected" clusters of all datapoints

Just evaluate a Gaussian at x_k

$$P(w_i | x_k, \lambda_t) = \frac{p(x_k | w_i, \lambda_t) P(w_i | \lambda_t)}{p(x_k | \lambda_t)} = \frac{p(x_k | w_i, \mu_i(t), \Sigma_i(t)) p_i(t)}{\sum_{j=1}^{c} p(x_k | w_j, \mu_j(t), \Sigma_j(t)) p_j(t)}$$

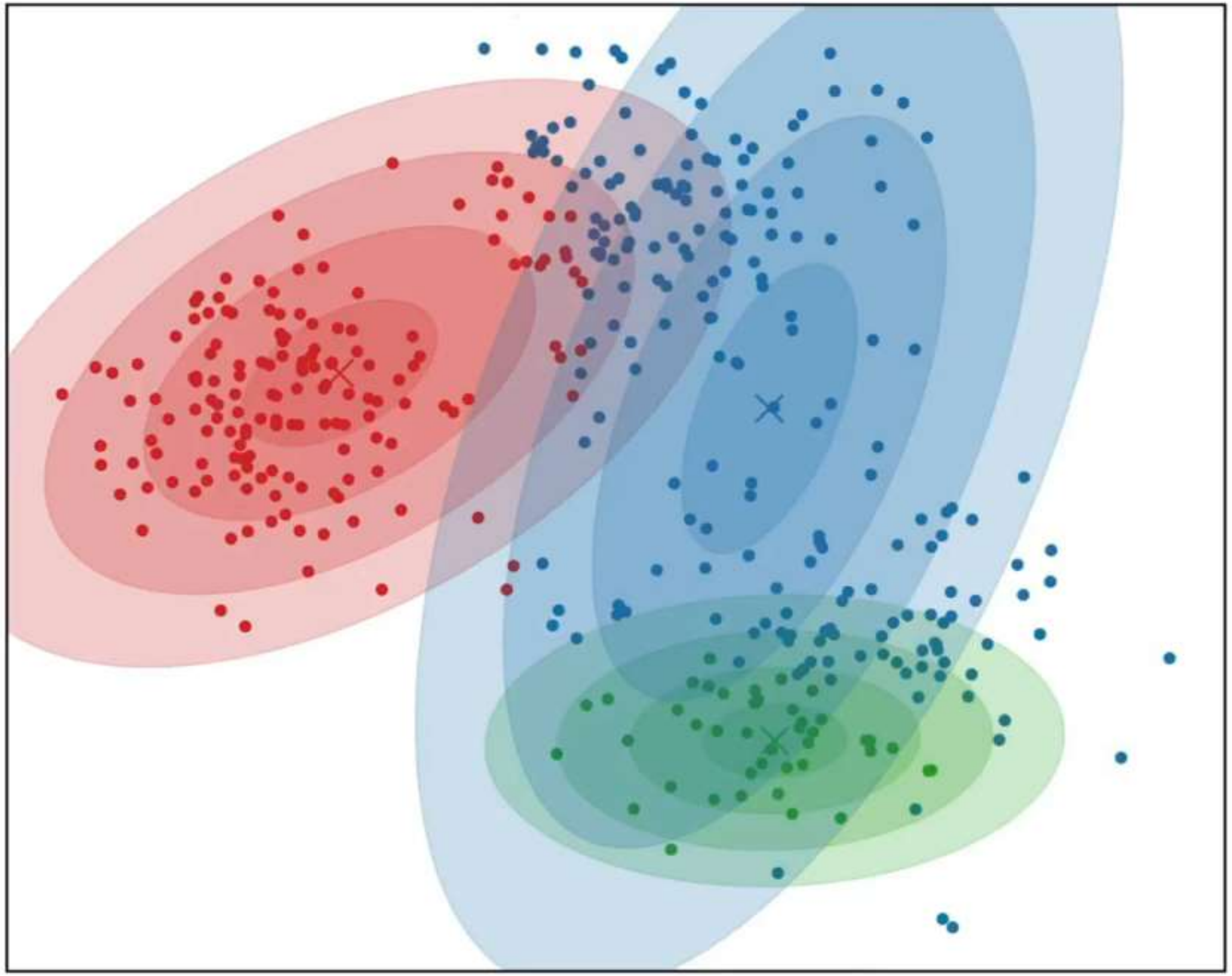
② M-step: Estimate **μ, Σ** given our data's class membership distributions

$$\mu_i(t+1) = \frac{\sum_k P(w_i | x_k, \lambda_t) x_k}{\sum_k P(w_i | x_k, \lambda_t)} \qquad \Sigma_i(t+1) = \frac{\sum_k P(w_i | x_k, \lambda_t) [x_k - \mu_i(t+1)][x_k - \mu_i(t+1)]^T}{\sum_k P(w_i | x_k, \lambda_t)}$$

$$p_i(t+1) = \frac{\sum_k P(w_i | x_k, \lambda_t)}{R}$$

R = #records

Learning algorithms (video)



Gaussian Mixture Models

Learning algorithms

Example:

$\dot{x} = f(x)$

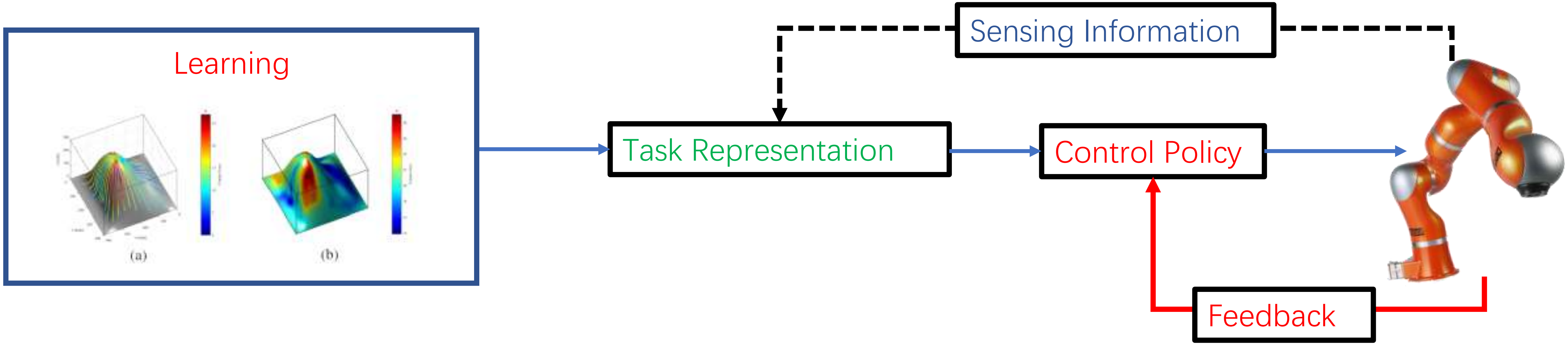
→ exp design (Human intelligence)

→ collect data $\{\dot{x}_k, x_k\}_{k=1,\dots,N}$ (pre-processing)

→ GMM $(\dot{x}_k, x_k)$ Model training

→ Online prediction $\dot{x}_k \leftarrow GMR(x_k)$

How to Implement?



Leverage the power of learning techniques and nonlinear control

Learning algorithms

LWR

C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning for control. Artificial Intelligence Review, 11(1-5):75–113, 1997

W.S. Cleveland. Robust locally weighted regression and smoothing scatterplots. American Statistical Association 74(368):829–836, 1979

GMR

Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via an EM approach. In Advances in Neural Information Processing Systems (NIPS), volume 6, pages 120–127, 1994

S. Calinon. Mixture models for the analysis, edition, and synthesis of continuous time series. Mixture Models and Applications, Springer, 2019

GPR

C.K.I. Williams and C.E. Rasmussen. Gaussian processes for regression. In Advances in Neural Information Processing Systems (NIPS), pages 514–520, 1996

C.E. Rasmussen and C.K.I. Williams. Gaussian processes for machine learning. MIT Press, Cambridge, MA, USA, 2006

S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. Philosophical Trans. of the Royal Society A, 371(1984):1–25, 2012

GPIS

O. Williams and A. Fitzgibbon. Gaussian Process Implicit Surfaces. In Gaussian Processes in Practice, 2007

Limitation of traditional learning algorithms

- **Limited training data**
- **Can only handle vector state**
- **Typically assume a Gaussian distribution**
- **Assume continuous system**
  - **Difficult to model hybrid system**
- **Can not deal with multi-modal control**
- **Good at modeling motion primitive or low-level physical skill**

Learning algorithms

exp design:

- hardware
- sensor
- protocol.
- intention
- interface

data collection:

- joint angles
- pos/ori
- force
- tactile
- vision
- ⋮

Learning Alg

- GMM
- GP
- SVM
- ⋮
- Deep learning
- LLM
- RT-2.

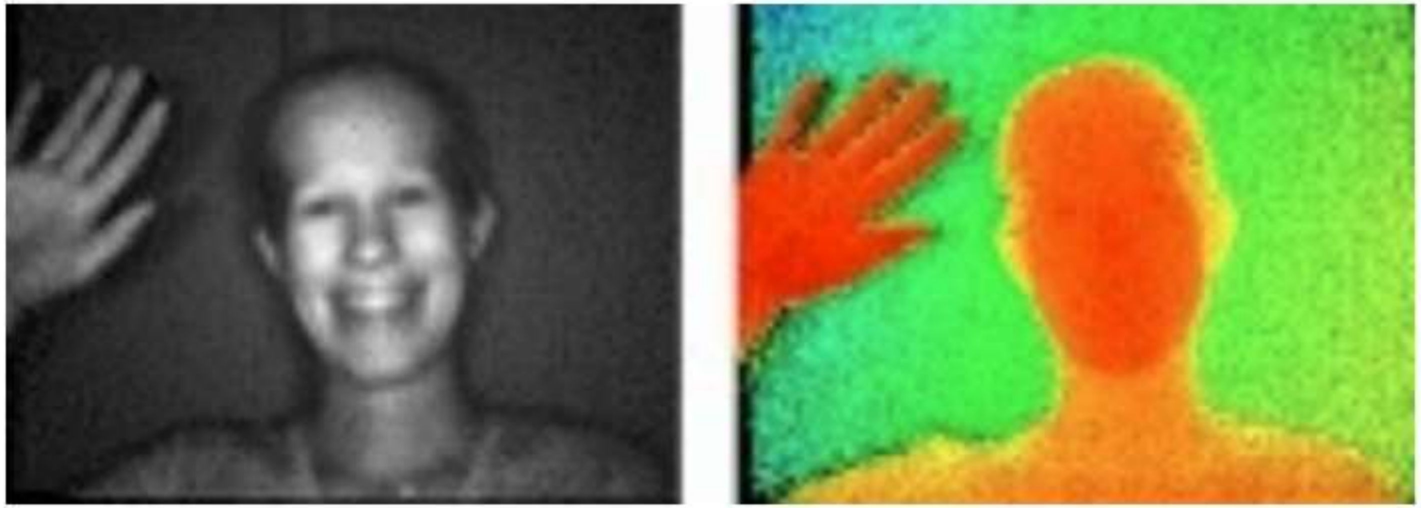
Limits of IL

Problem 1: Correspondence Problem

Even when the robot looks more like the human, its body does not have the same range and dynamics of motion.

Limits of IL

Problem 1: Correspondence Problem



Robots do not perceive things like we do.

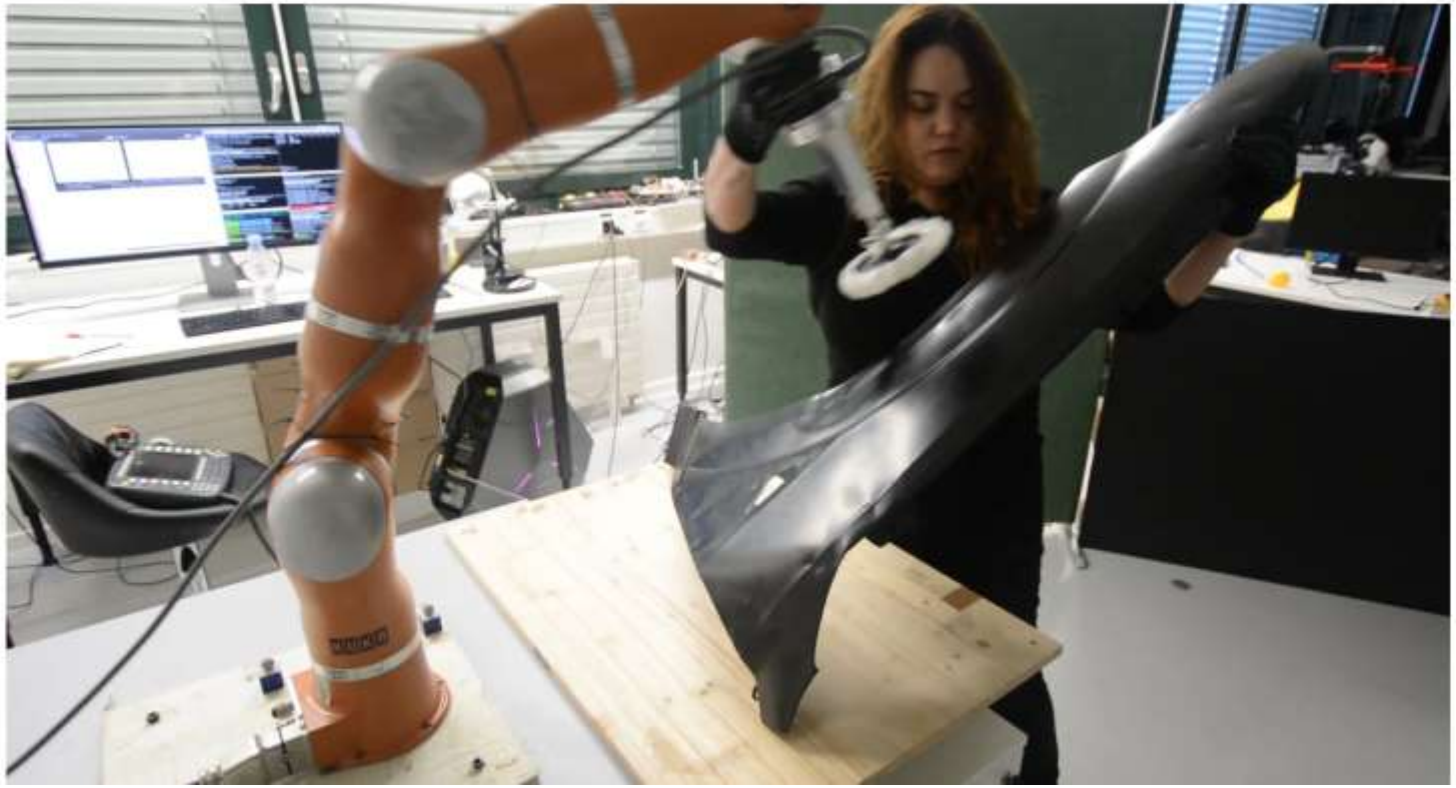
Sonars, infrared sensors, lasers are common on robots and easier to process than information from cameras.

Limits of IL

Problem 1: Correspondence Problem

- Teachers need to train themselves before training the robots.

Sometimes super hard!!!



Limits of IL

Problem 2: Learning is Data-Sensitive

- Data is robot-dependent

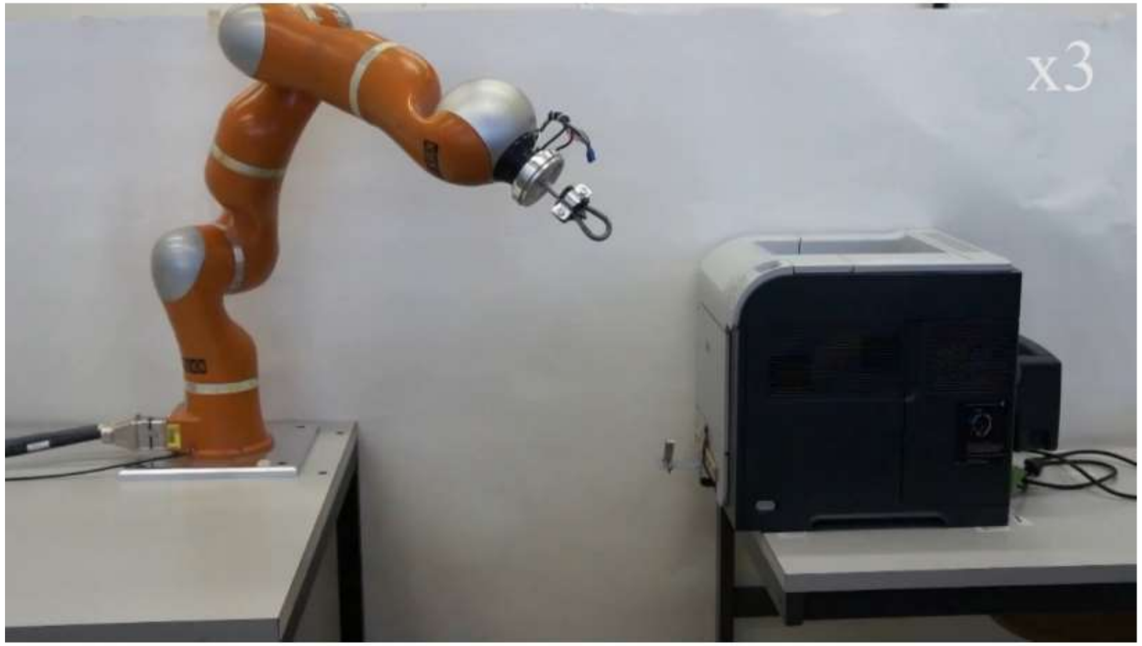
UR5: 6DOF

Franka Panda: 7DOF

Limits of IL

Problem 2: Learning is Data-Sensitive

- Data is environment-dependent



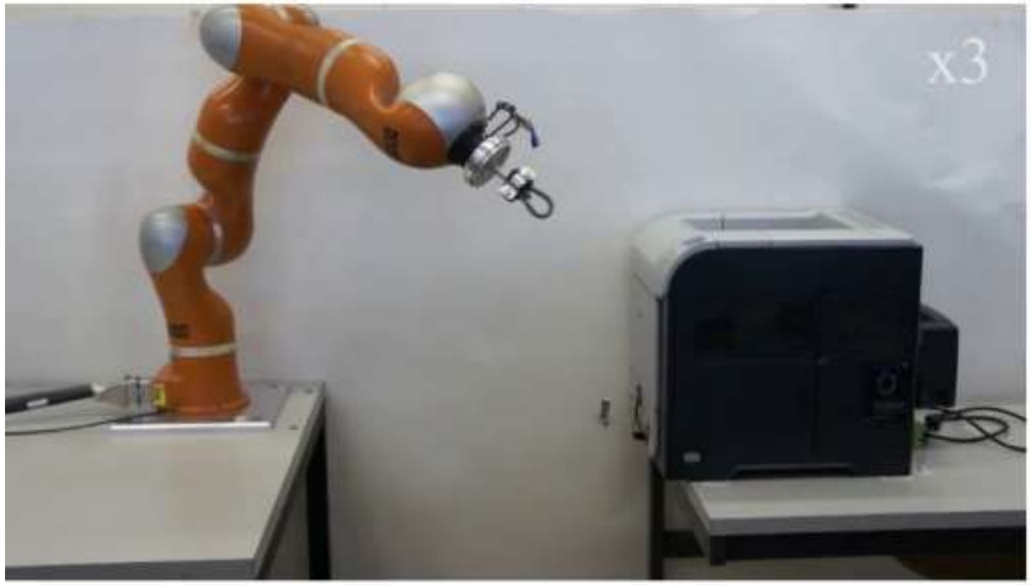
Model Learned at EPFL

Model transferred at AIST/JRL

Limits of IL

Problem 2: Learning is Data-Sensitive

Need Transfer Learning methods



Model Learned at EPFL

Model transferred at AIST/JRL

Limits of IL

Problem 3: Variability in Task Definition

- Question: **What does it mean to perform a task?**
- Multiple ways to accomplish a task:
  - multiple motions

Limits of IL

Problem 3: Variability in Task Definition

- Question: **What does it mean to perform a task?**
- Multiple ways to accomplish a task:
  - multiple motions
  - multiple tools

Limits of IL

Current/Future Research Directions: Learn from Small Datasets

- Learn from small datasets: Reduce the number of demonstrations needed
- Combine heterogeneous data types
- Improve teaching interactions

one-shot learning

Today agenda

- **Paper reading (~30 mins)**
- **Why imitation learning (IL) (~5)**
- **Key ingredients of IL (~5)**
- **Data collection (~5)**
- **Learning algorithms (~20)**
- **Limits of IL (~5)**
- **Examples and applications (~20)**
  - **Motion**
  - **Hand IK**
  - **Force-relevant task**
  - **Multi-modal task**

Applications

Modelling Hitting Task using Dynamical Systems-Based Control

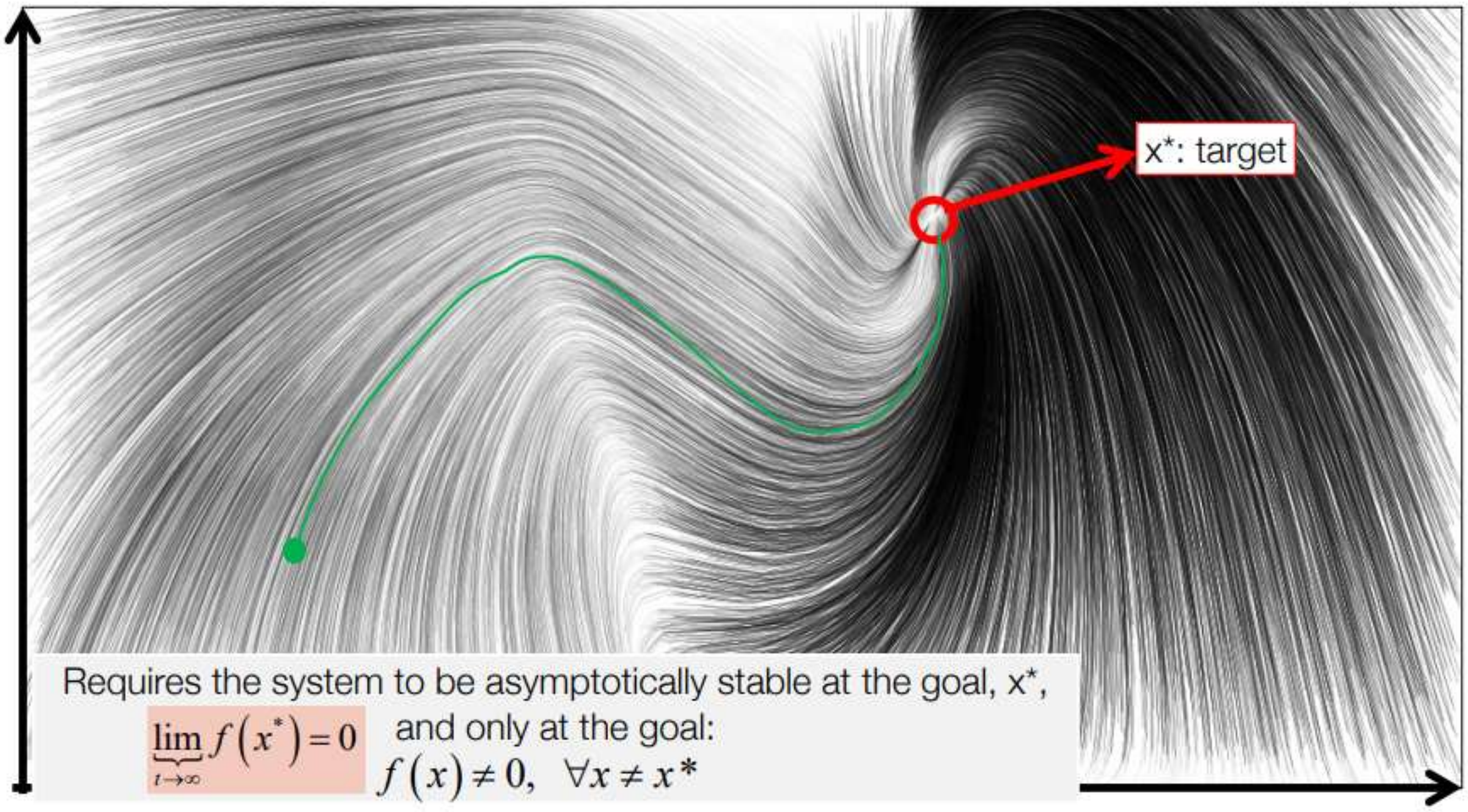
- Collect Demonstrations of hitting a golf ball using kinesthetic teaching

- Collect the recorded robot states and velocity at each time step

- We could generate a dynamical system representing this motion: $\dot{x} = f(x)$

Applications

Modelling Hitting Task using Dynamical Systems-Based Control

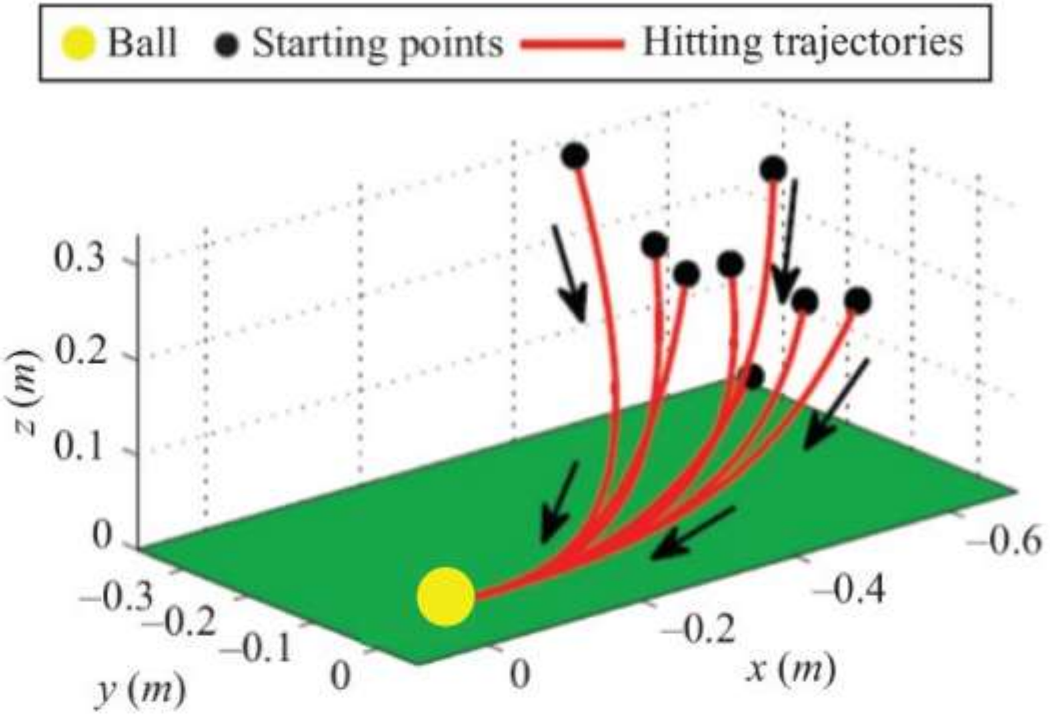


EPFL

Applications

Modelling Hitting Task using Dynamical Systems-Based Control

- We could generate a dynamical system representing this motion: $\dot{x} = f(x)$
- Guarantees asymptotically reaching and stabilizing at attractor: $\lim_{\{t \to \infty\}} x = x^*$, where x^*: Ball Location



EPFL

Applications

Teaching Compliant Control: What happens when stiffness not considered?

Too stiff: Liquid spills from jerking

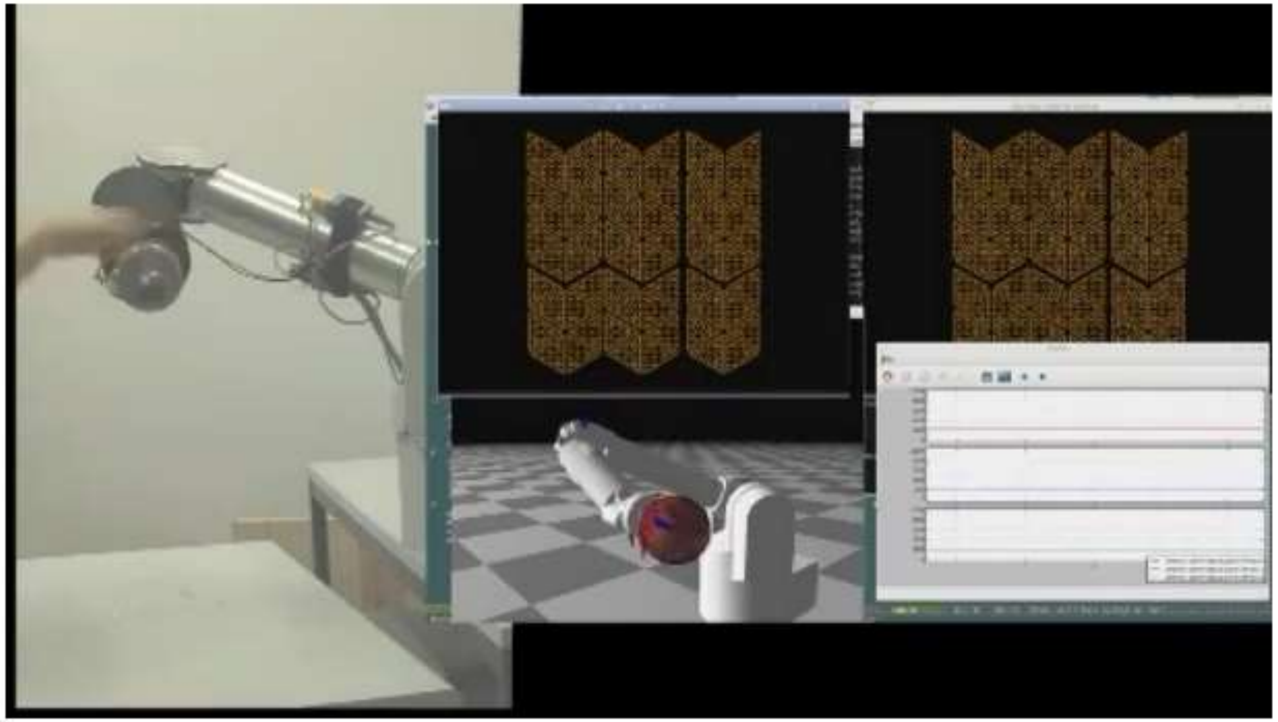
Too compliant: Liquid spills from glass

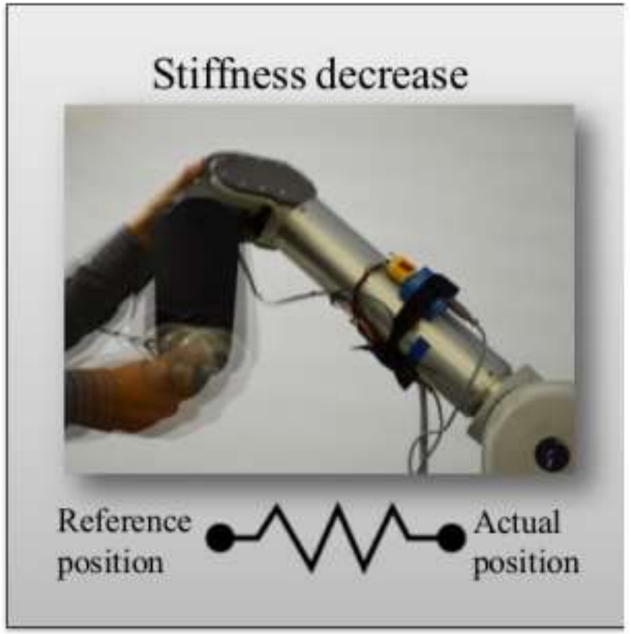
How can we teach robot when to increase and decrease compliance?

Applications

Teaching Compliant Control: Adding Compliance

Teaching ***decrease*** in stiffness by wiggling the robot

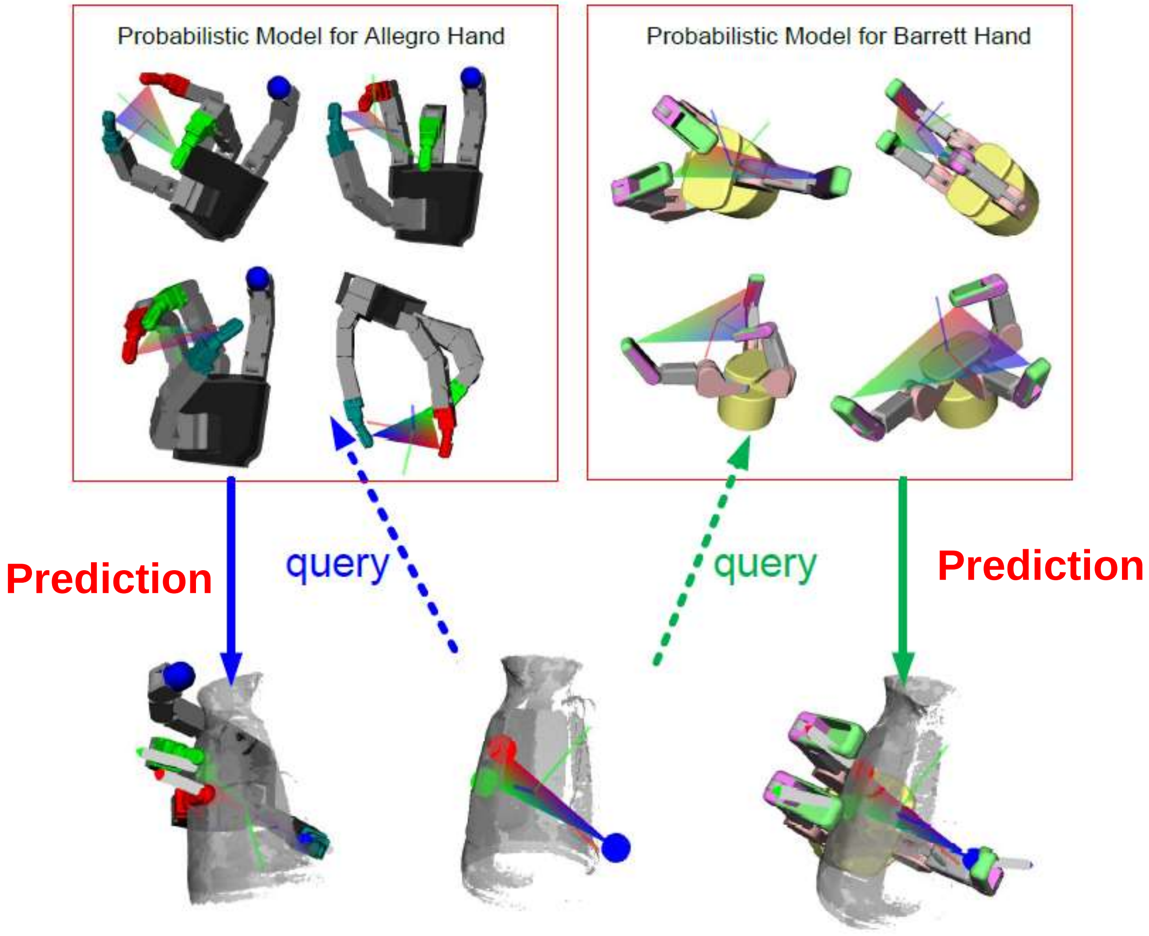




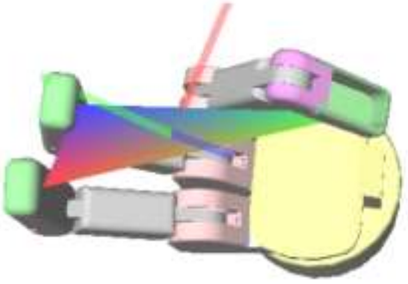
Kronander and Billard, Robot Learning from Demonstration for Compliant Manipulation, ICRA 2012

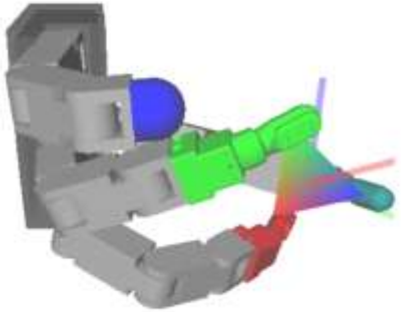
Applications

Probabilistic Hand Inverse Kinematics



Applications





Virtual Frame

$$T_{VF}^{Hand} = \begin{bmatrix} R^o & \mathbf{p}^o \\ [0,0,0] & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

$$\{\mathcal{G}^i, i = 1 \cdots N_g\}$$

Grasp Configuration $\mathcal{G} = \{\Theta, L, N\}$

$$L = [L_1, L_2, L_3] \in \mathbb{R}^3$$
$$L_i = \|\mathbf{p}^i - \mathbf{p}^o\|$$

$$N = [N_1, N_2, N_3] \in \mathbb{R}^3$$
$$N_2 = \mathbf{n}^1 \cdot \mathbf{n}^3$$

Applications

$$\{\mathcal{G}^i, i = 1 \cdots N_g\}$$

Grasp Configuration $\mathcal{G} = \{\Theta, L, N\}$

GMM

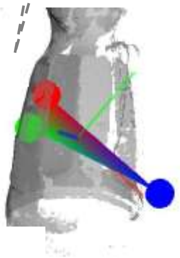
$$p(\mathcal{G}_*|\Omega) = \sum_{i=1} \pi_i \mathcal{N}(\mathcal{G}_*|\mu_i, \Sigma_i)$$

Learn Density Function

Feasibility

$$f_i(q, \Omega) = \frac{1}{2}(q - \mu_{q,i})^T \Sigma_{q,i}^{-1} (q - \mu_{q,i})$$

Query Feasibility

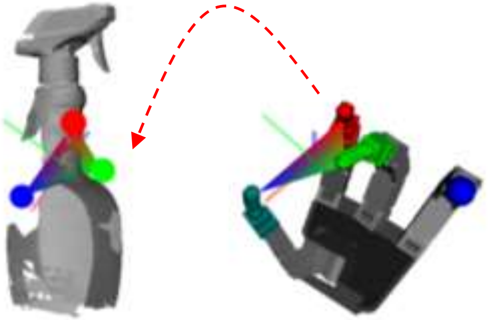


GMR

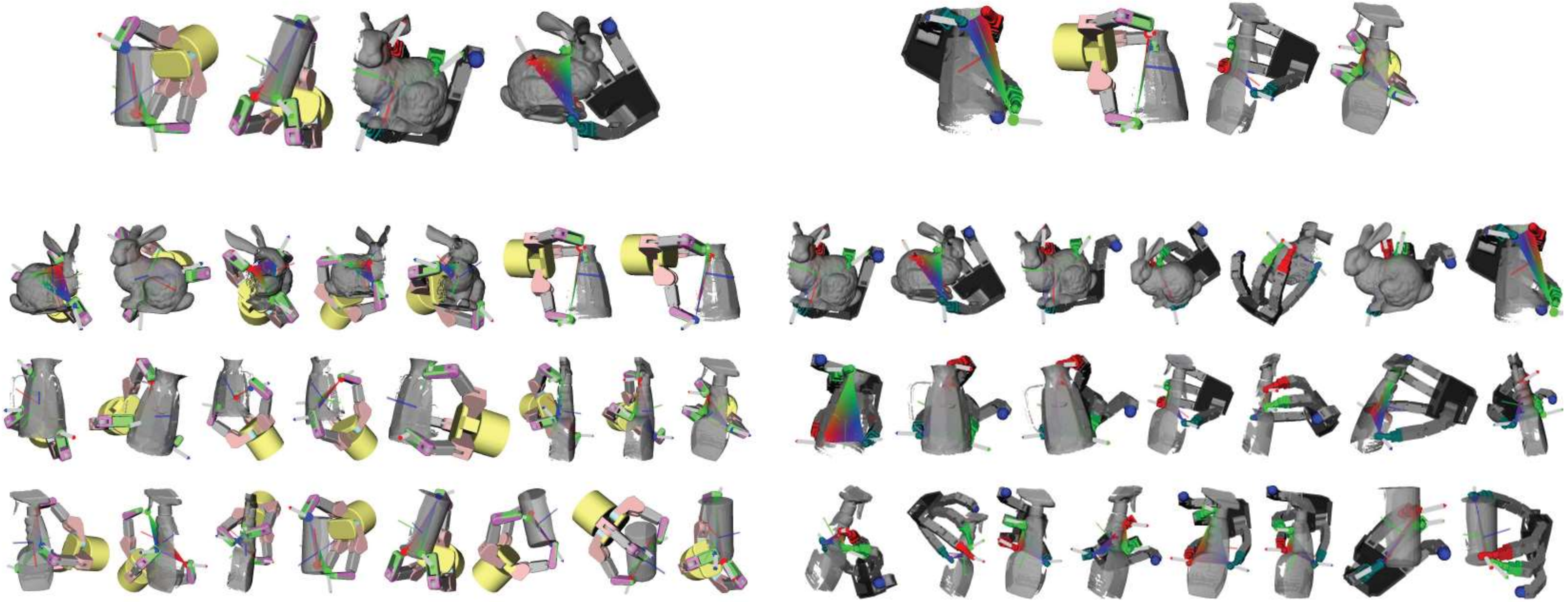
$$E\{p(\Theta|L, N, \Omega)\} = \sum_{i=1}^{m} h_i(\mu_{\Theta,i} + \Sigma_{\Theta q,i} \Sigma_{q,i}^{-1} (q - \mu_{q,i}))$$

Hand Pose

$$T_{Hand}^{Obj} = T_{VF}^{Obj} T_{VF}^{Hand^{-1}}$$



Applications

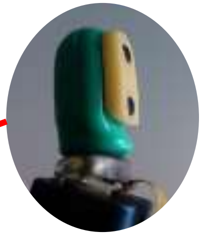


Applications

Dynamic Uncertainty

$$\mathbf{M}_r(\mathbf{x}_r)\ddot{\mathbf{x}}_r + \mathbf{C}_r(\mathbf{x}_r, \dot{\mathbf{x}}_r)\dot{\mathbf{x}}_r + \mathbf{g}_r(\mathbf{x}_r) = \mathbf{G}\mathbf{f} + \mathbf{f}_{\text{ext}}$$

$$I_{\mathcal{G}} = g(\mathbf{G}, \mathbf{f}, \mathbf{S})$$

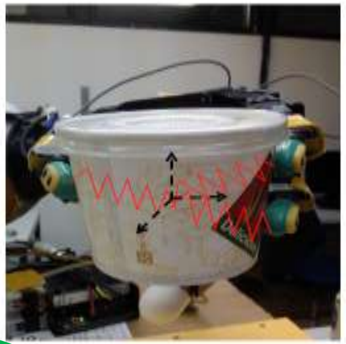


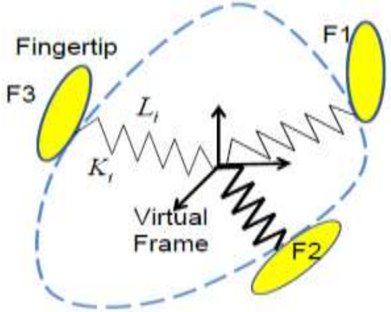
Grasp Matrix

Grasping Force

Tactile Sensing

Object-level Impedance Controller





Grasp Experience

$$D = \{(K^i, L^i, S^i)\}_{i=1\dots N}$$

Applications

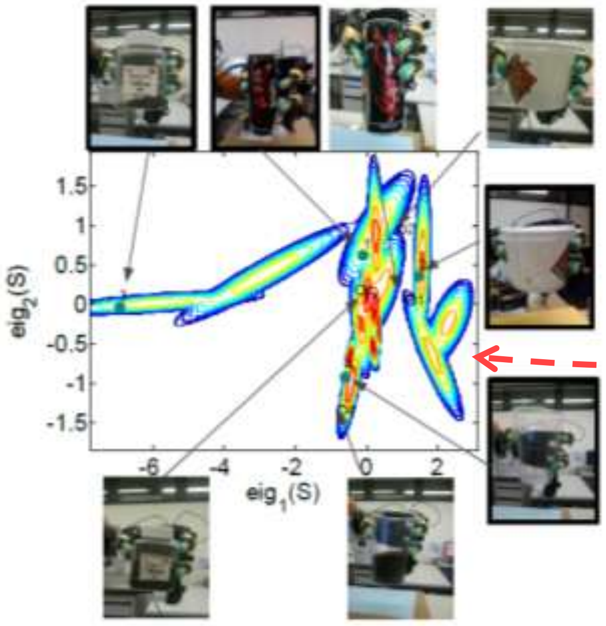
$$D = \{(K^i, L^i, S^i)\}_{i=1\dots N}$$

Grasp Experience

$$X_* = (K_*, L_*, S_*)$$

$$p(X_*|\Omega) = \sum_{i=1}^{m} \pi_i \mathcal{N}(X_*|\mu_i, \Sigma_i)$$

Learn Density Function

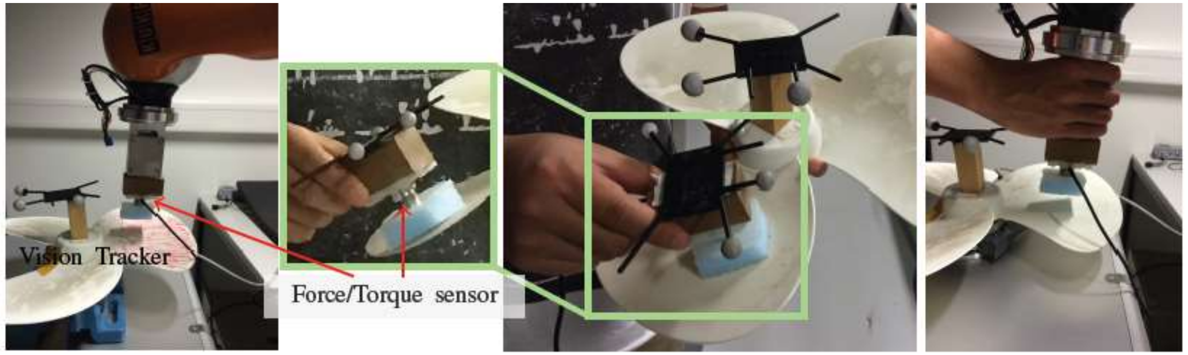


Stability Estimation

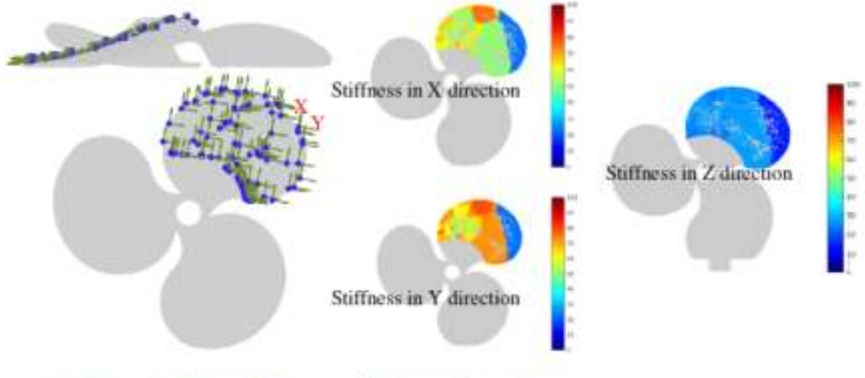
Applications

Experimental Results

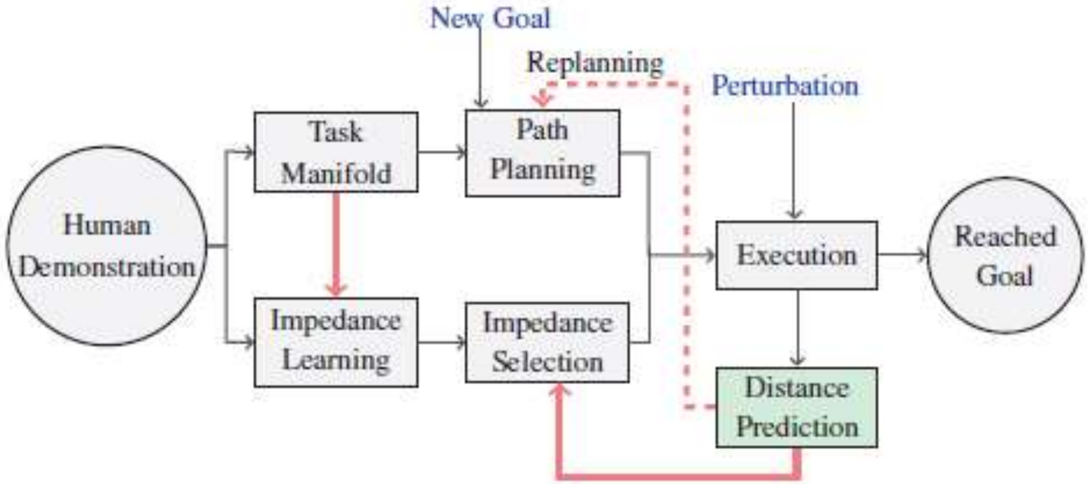
Fan Blade Cleaning

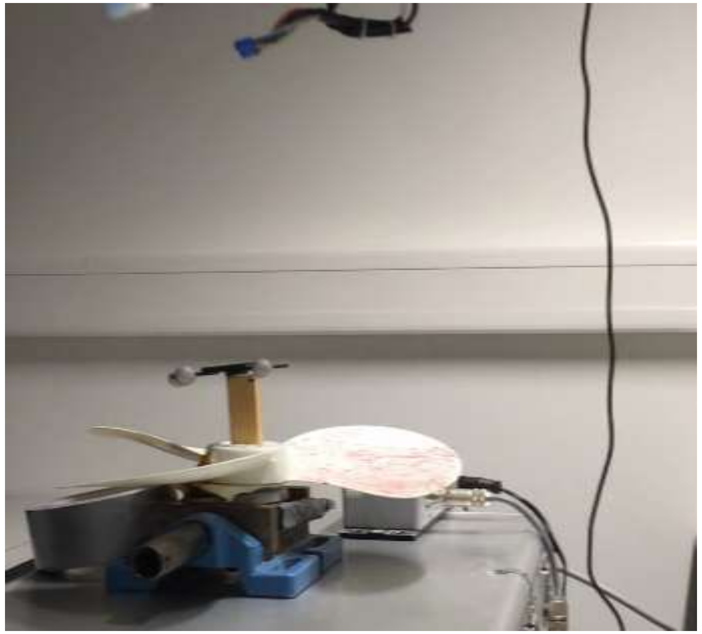


(a) Robot setup (b) Human demonstration (c) Kinesthetic teaching



(a) Learned task manifold (b) Learned impedance





M. Li *et al*. “Learning task manifolds for constrained object manipulation”, Autonomous Robots 2016

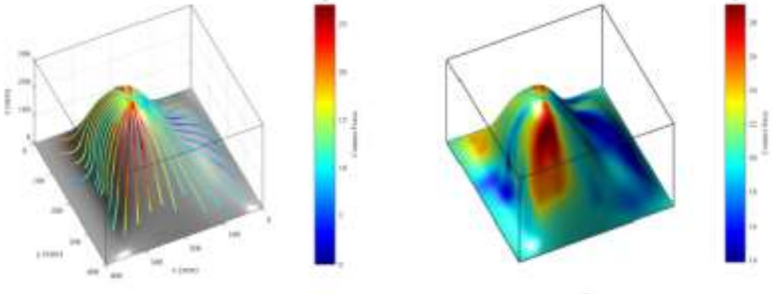
Polishing

Learning force-dominant skills from human demonstration

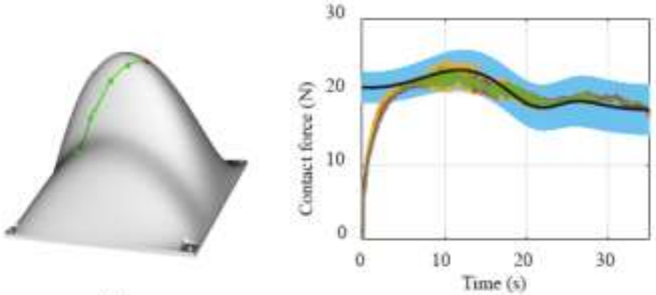
Xiao Gao, Jie Ling, Xiaohui Xiao and Miao Li

Xiao Gao

This video is submitted to IROS 2018



(a) (b)



(a) (b)

X. Gao *et al*. "Learning Force-dominant Skills from Human Demonstration", Submitted to IROS 2018

Assembly

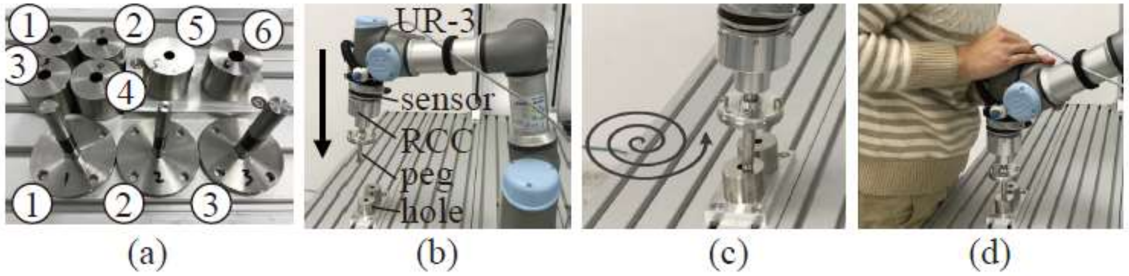
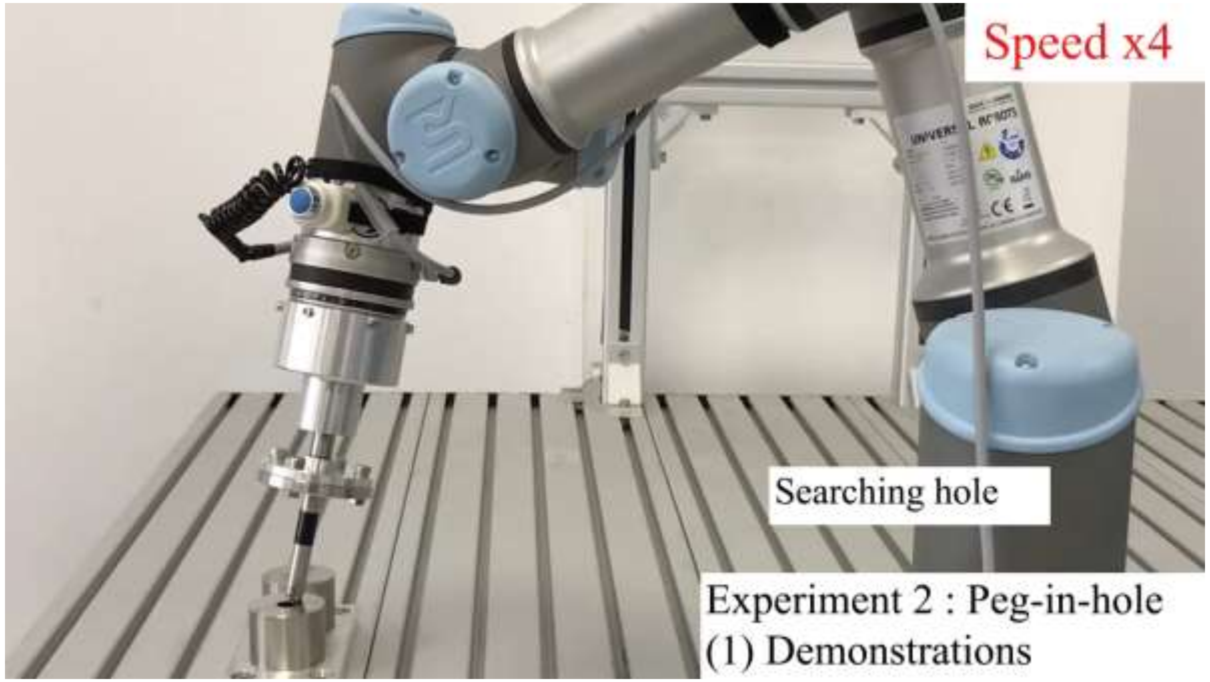
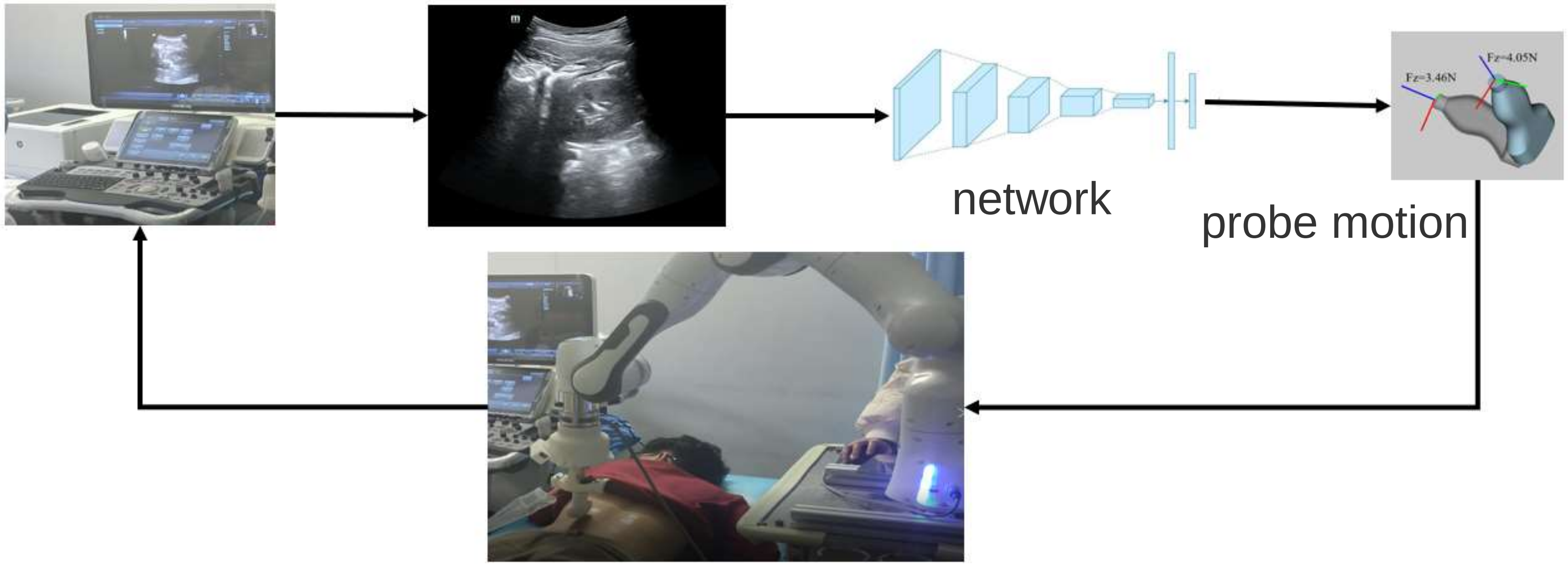


Fig. 12. Experiment setup and demonstration phase by collaborative insertions. **a**: The three pegs and six holes. **b**: The peg was moving towards the hole. **c**: Searching the hole by an Archimedean spiral movement. **d**: Collaborative insertions.



Learning the moving strategy of probe

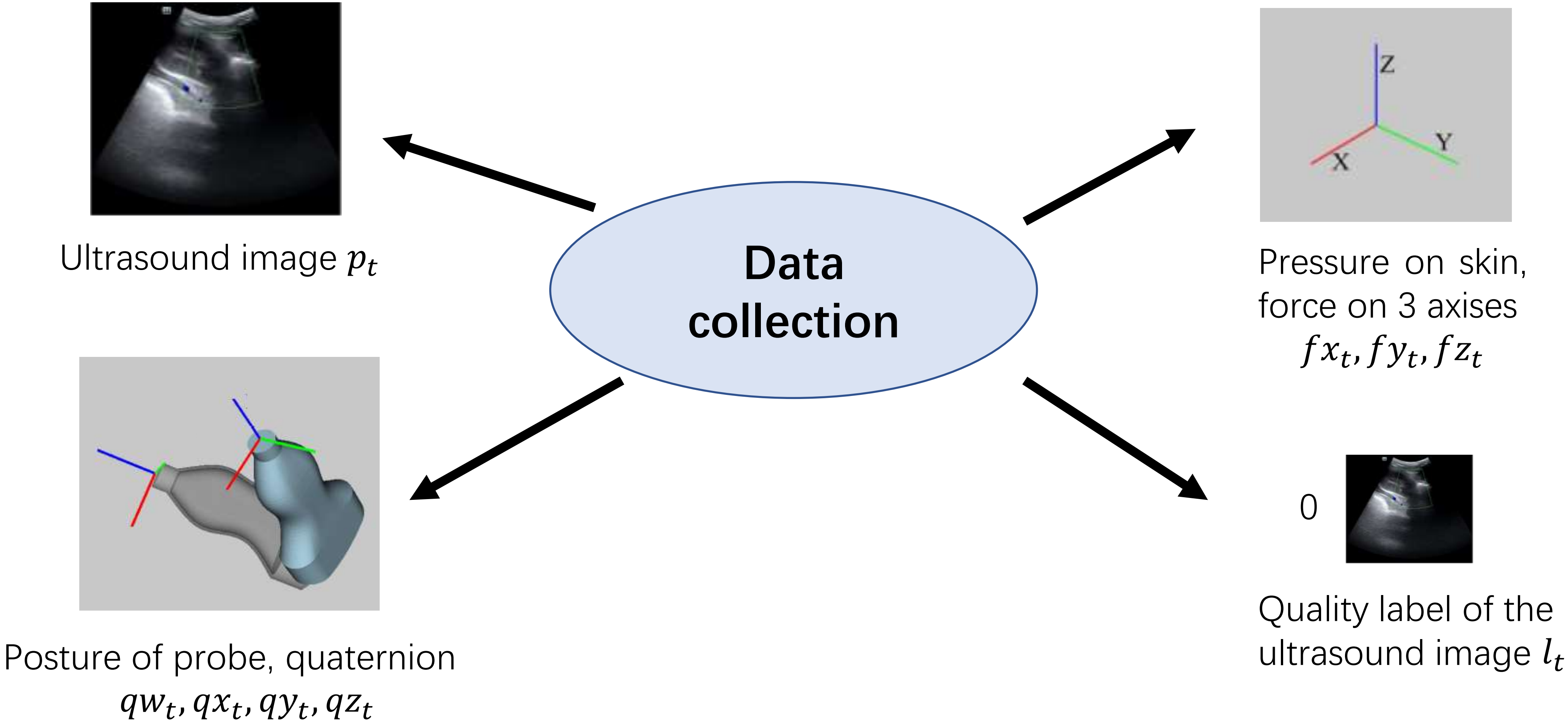
The kidney area in image changed from fuzzy to clear.



Predict the next motion of probe, and enable the robot to move the probe.

Collect the probe motion data

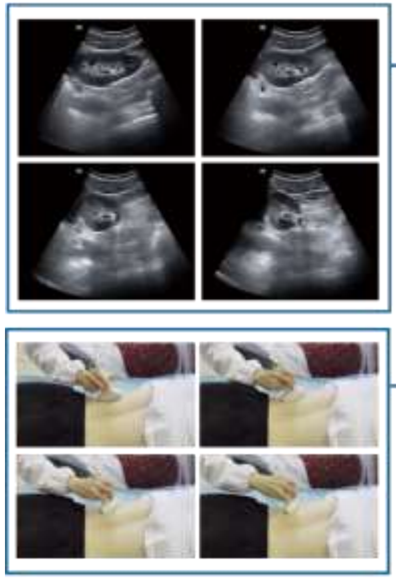
Keep the contact point between the probe and the human body unchanged when collecting data.



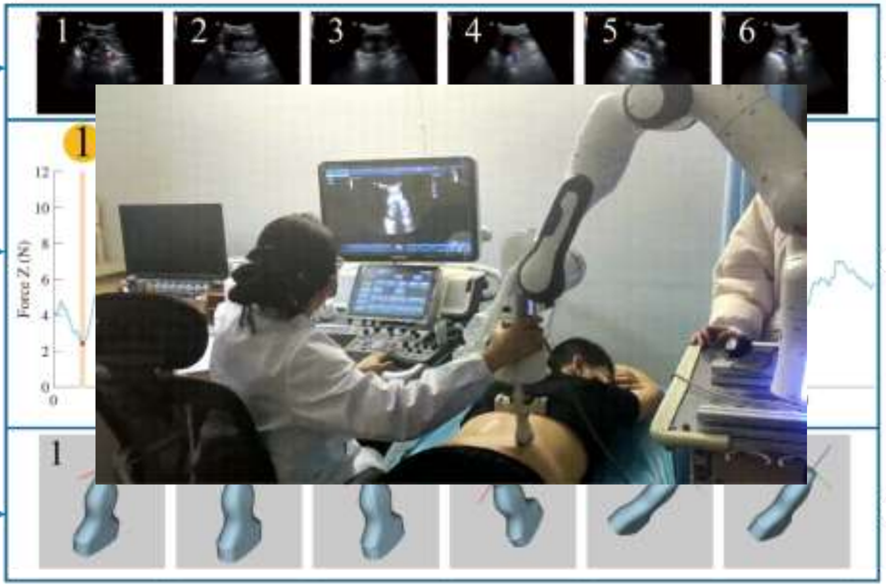
Collect data from 5 persons

Person	1	2	3	4	5
Quantity of data	776	1348	596	919	1552

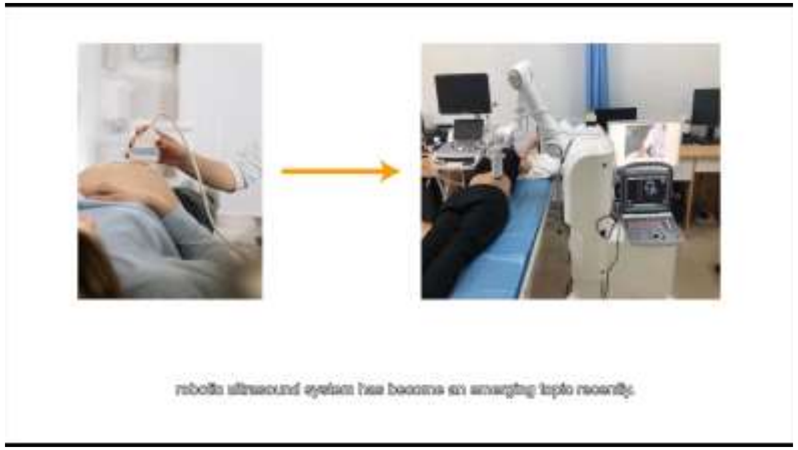
临床数据采集

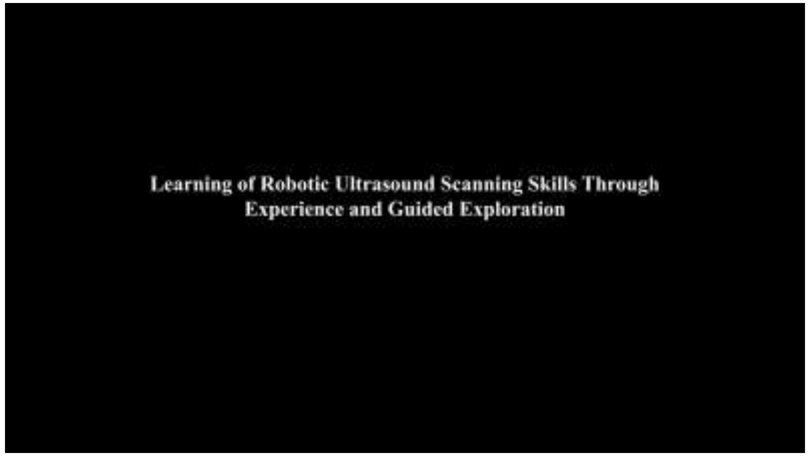


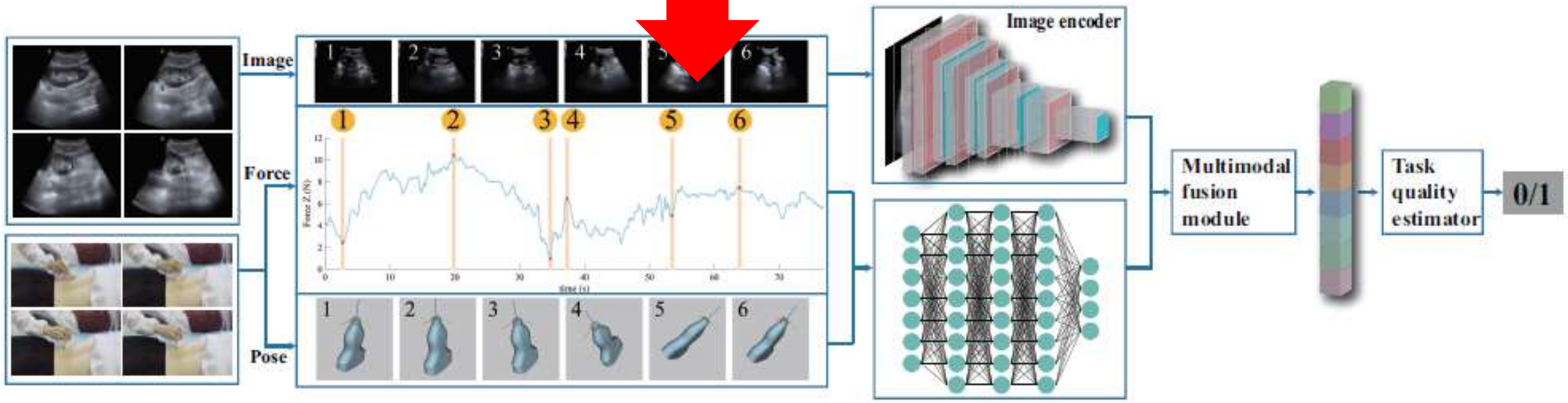
多模态技能学习



样机测试场景







TRO 2023 (Under Review)

Goal for this course

- **Design：soft hand design x1**
- **Perception: vision, point cloud, tactile, force/torque x1**
- **Planning: sampling-based, optimization-based, learning-based x3**
- **Control: feedback, multi-modal x2**
- **Learning: imitation learning, RL x2**
- **Simulation tool (pybullet, matlab, OpenRAVE, Issac Nvidia, Gazebo)**
- **How to get a robot moving!**

Robotics today!

Learning Object-level Impedance Control for Robust Grasping and Dexterous Manipulation

Miao Li*, Hang Yin*, Kenji Tahara+, and Aude Billard*

*Learning Algorithms and Systems Laboratory (LASA)

Ecole Polytechnique Federale de Lausanne (EPFL)

+Faculty of Engineering, Kyushu University, Japan

ICRA-2014, HongKong

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Overview

“Learning Object-level Impedance Control for Robust Grasping and Dexterous Manipulation”

Motivation

Model — Object-level Impedance Controller

Approach — Learning from Human Demonstration

Experiments and implementation

Conclusion

Motivation

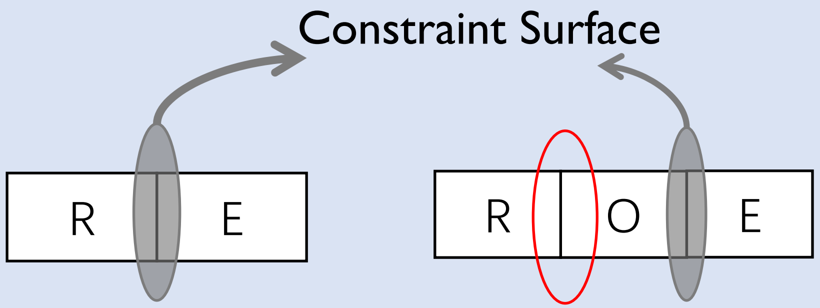
How to specify the proper impedance for a given task?

Our Answer:

The desired object-level impedance can be learnt from human demonstration

Motivation

Contact Task:



- R: robot
- O: object
- E: environment

Object-level Impedance Control

Robust Grasping

Dexterous Manipulatio
n

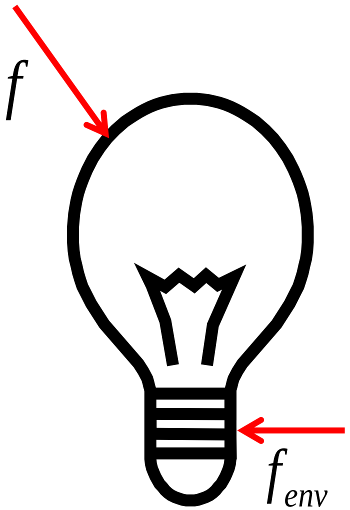
Keep object stable

Object-Centric

Move object to desired configuration

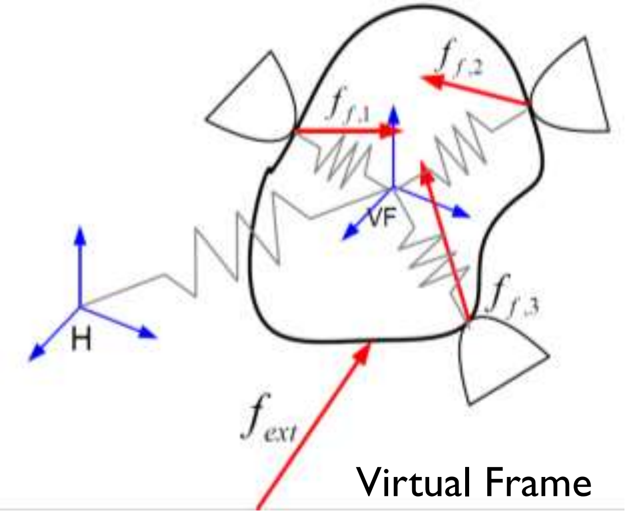
The desired interactions are represented in the object frame

Object-level Impedance Control



Object Dynamics:

$$\mathbf{f} + \mathbf{f}_{env} = m\ddot{\mathbf{x}}$$



Desired Behavior:

$$\mathbf{f}_{env} = M\ddot{\mathbf{x}} + D(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) + K(\mathbf{x} - \mathbf{x}_d)$$

Object-level Impedance Control

$$\mathbf{f} = mM^{-1}D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + mM^{-1}K(\mathbf{x}_d - \mathbf{x}) + (mM^{-1} - I)\mathbf{f}_{env}$$

Object-level Impedance Control

$$\mathbf{f} = mM^{-1}D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + mM^{-1}K(\mathbf{x}_d - \mathbf{x}) + (mM^{-1} - I)\mathbf{f}_{env}$$

Actual Trajectory

Object-level Impedance Control

$$\mathbf{f} = mM^{-1}D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + mM^{-1}K(\mathbf{x}_d - \mathbf{x}) + (mM^{-1} - I)\mathbf{f}_{env}$$

Desired Trajectory

Actual Trajectory

Object-level Impedance Control

$$\mathbf{f} = mM^{-1}D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + mM^{-1}K(\mathbf{x}_d - \mathbf{x}) + (mM^{-1} - I)\mathbf{f}_{env}$$

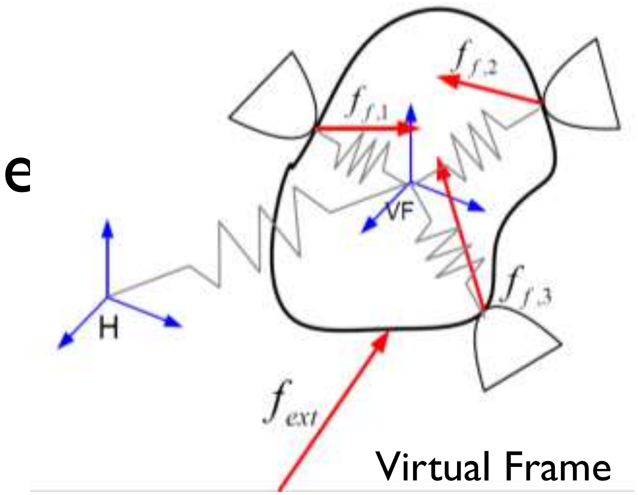
Desired Trajectory

Actual Trajectory

Object-level Impedance Control

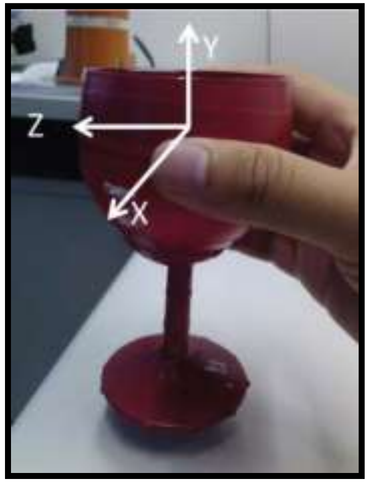
$$\mathbf{f} = mM^{-1}D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + mM^{-1}K(\mathbf{x}_d - \mathbf{x}) + (mM^{-1} - I)\mathbf{f}_{env}$$

Desired Trajectory   Actual Traje



Robust Grasping

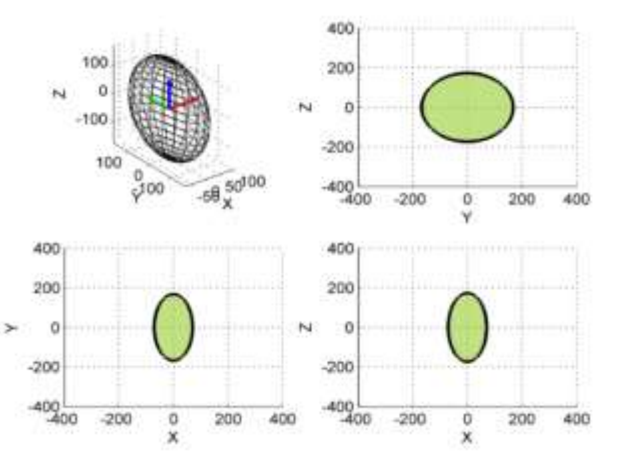
$$\mathbf{f} = D(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + K(\mathbf{x}_d - \mathbf{x})$$



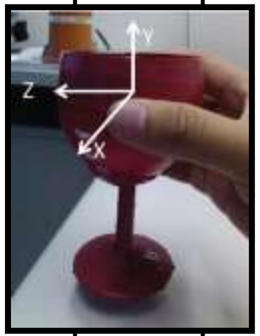
Relative Stiffness: the object stiffness in one direction is inversely proportional to the variance of displacement under perturbation in the corresponding direction

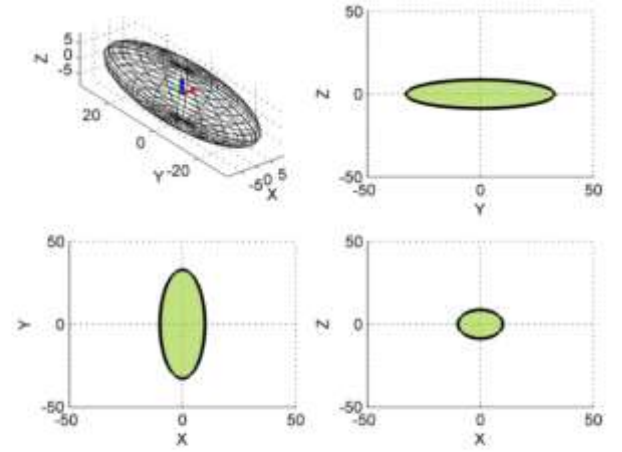
Robust Grasping

$$K = \alpha\{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^i - \mathbf{x}_r)(\mathbf{x}^i - \mathbf{x}_r)^T\}^{-1}$$



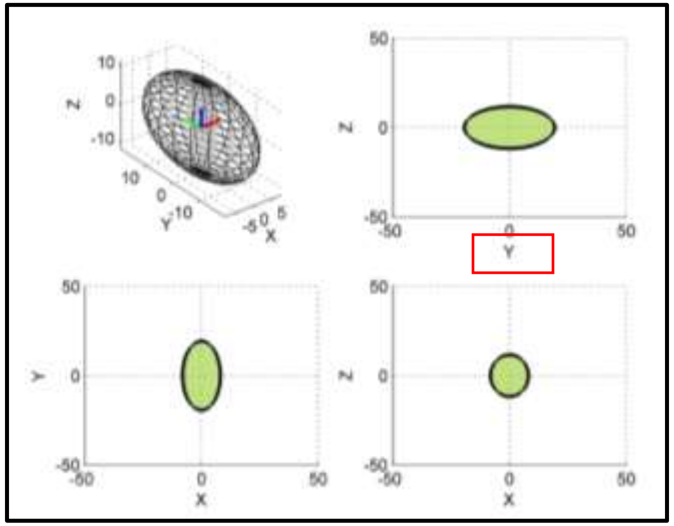
Relative Translational Stiffness



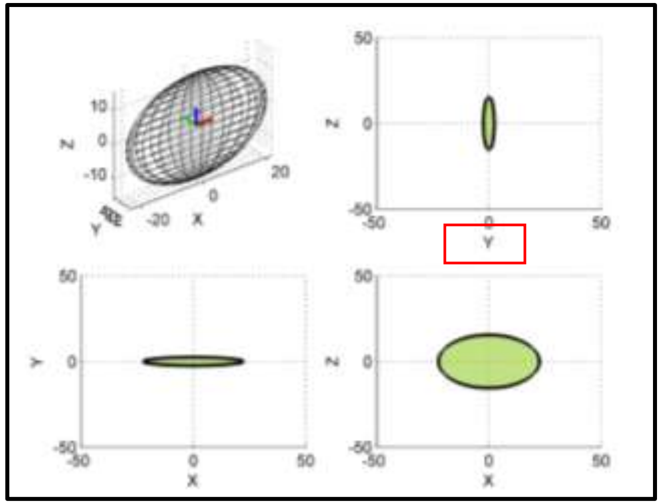


Relative Rotational Stiffness

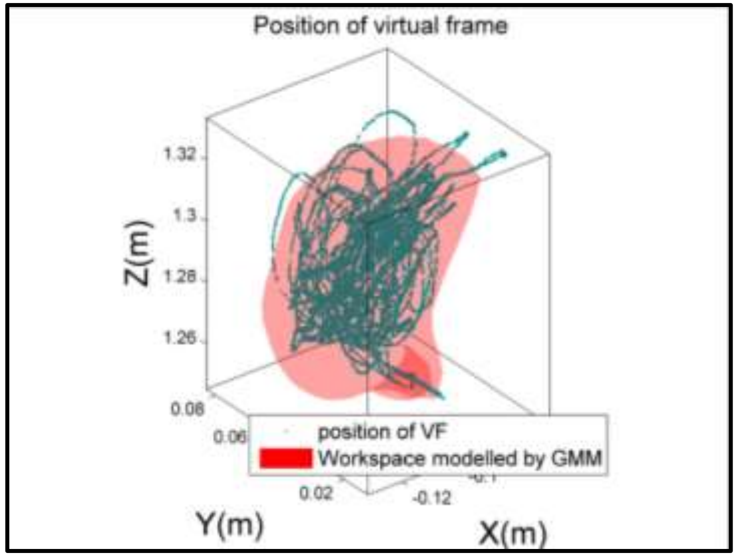
Robust Grasping



Relative Rotational Stiffness



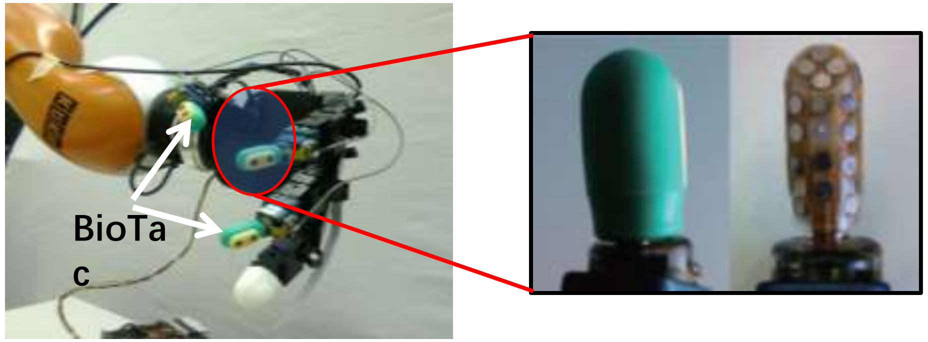
Robust Grasping: Workspace

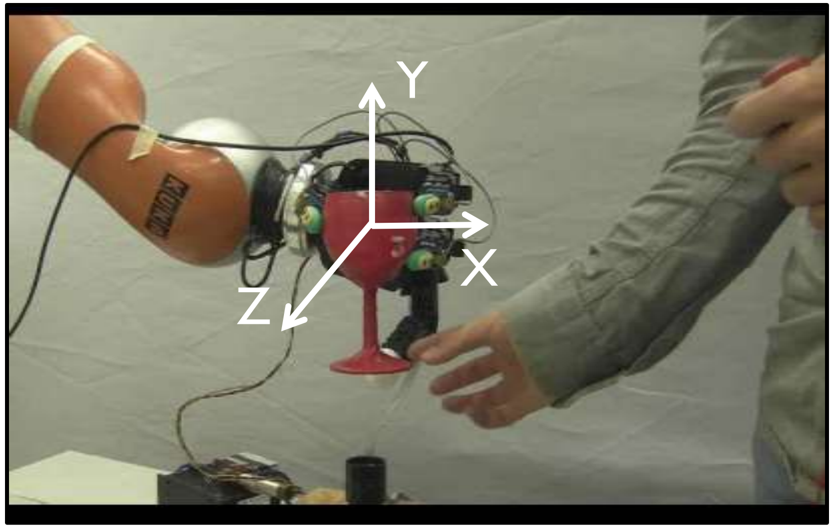


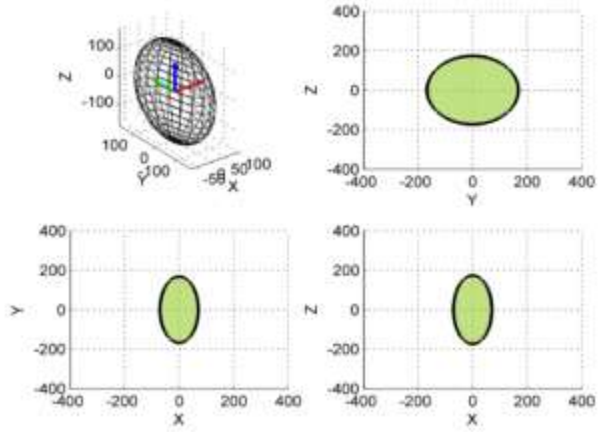
$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$K = \alpha\{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^i - \mathbf{x}_r)(\mathbf{x}^i - \mathbf{x}_r)^T\}^{-1}$$

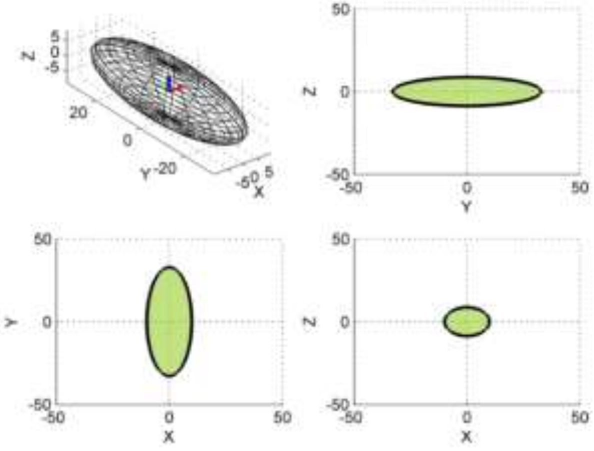
Robust Grasping





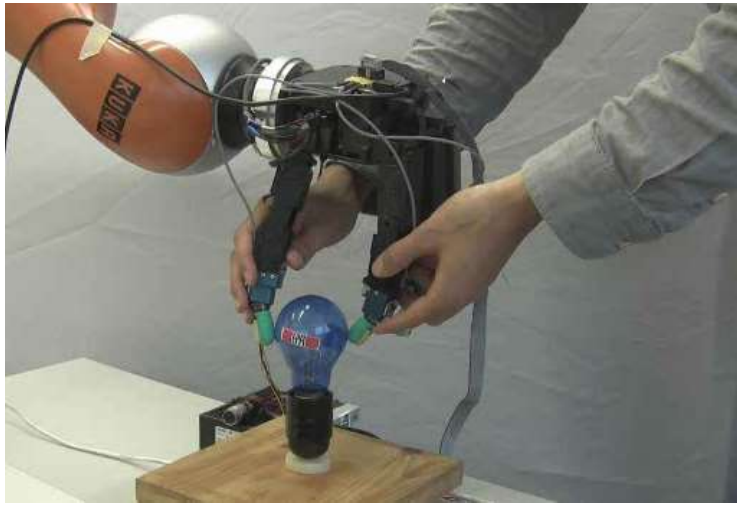


Relative Translational Stiffness



Relative Rotational Stiffness

Dexterous Manipulation



Human demonstration

Optimization:

$$\min_{K,\mathbf{x}_r} \sum_{i=1}^{N_t} \|\mathbf{f}_{f,o}(i) - \{K(\mathbf{x}_r - \mathbf{x}(i))\}\|^2$$

s.t.

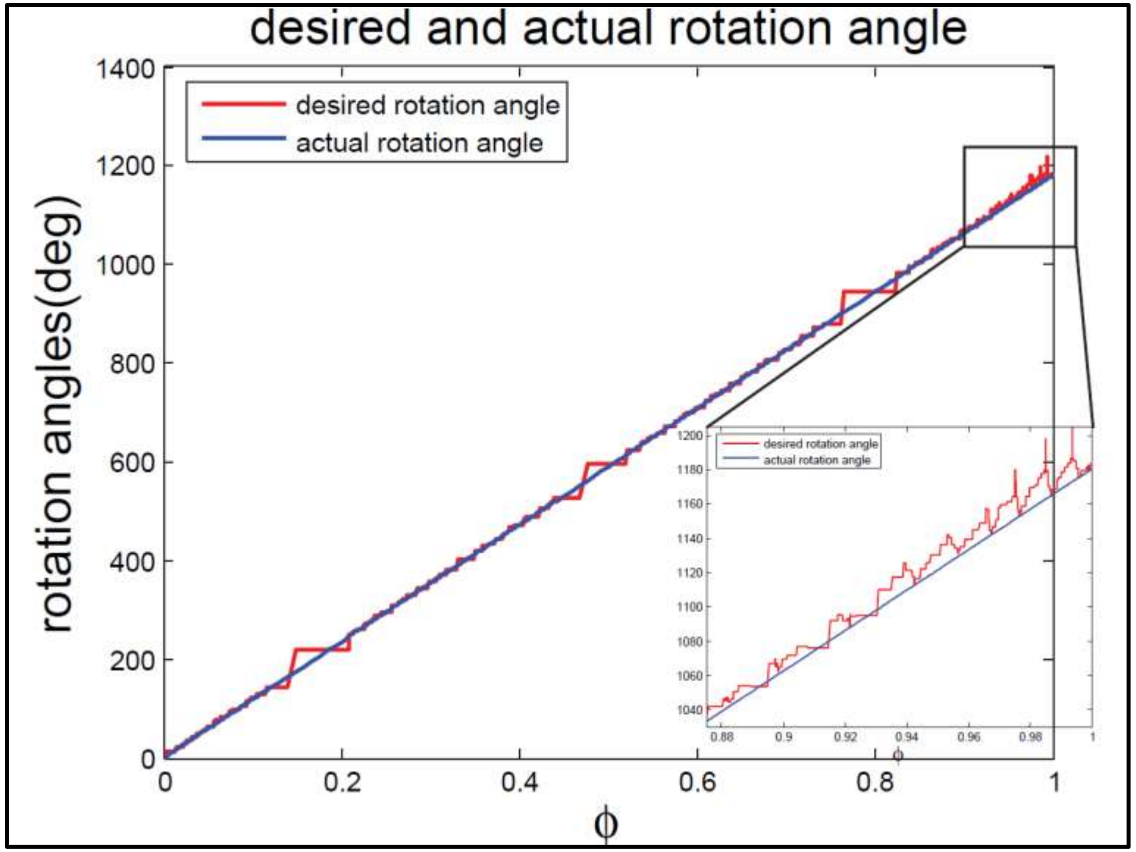
$$K_{i,j} \leq k_{lim}, \quad i = 1...6, j = 1...6;$$

$$\|\mathbf{x}_r - \mathbf{x}(i))\| \leq \Delta x_{lim}, \quad i = 1...N_t;$$

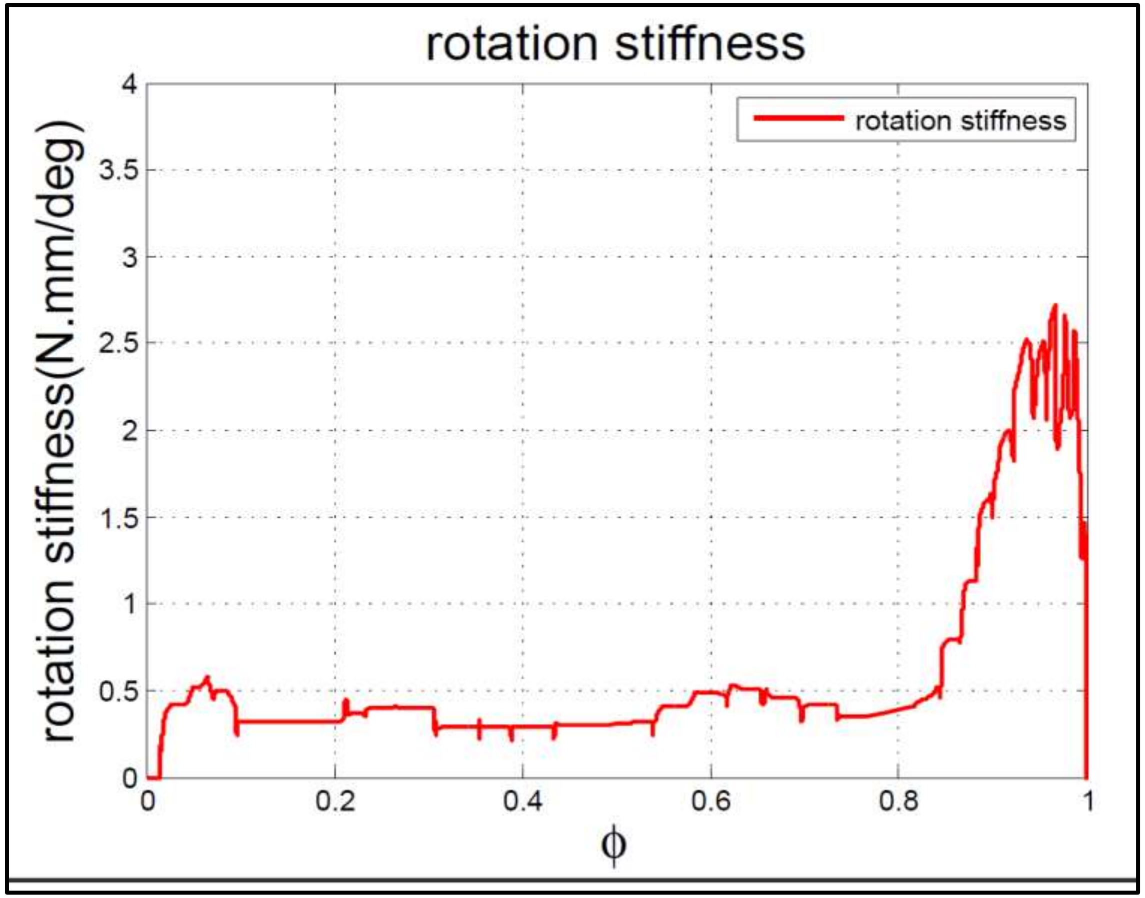
$$\|\dot{\mathbf{x}}_r - \dot{\mathbf{x}}(i)\| \leq \Delta \dot{x}_{lim}, \quad i = 1...N_t;$$

Stiffness Learning: the object force and motion are recorded from human demonstration, and used to learn an impedance model.

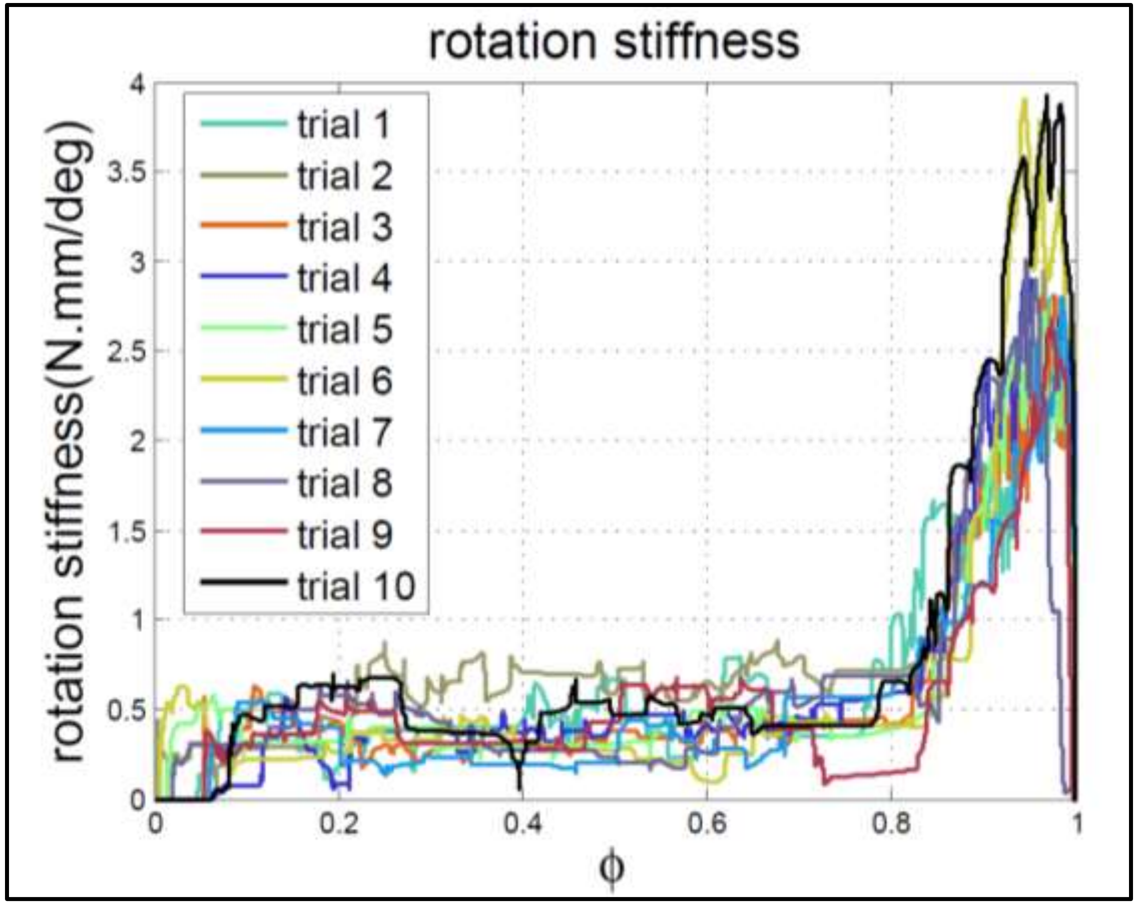
Dexterous Manipulation



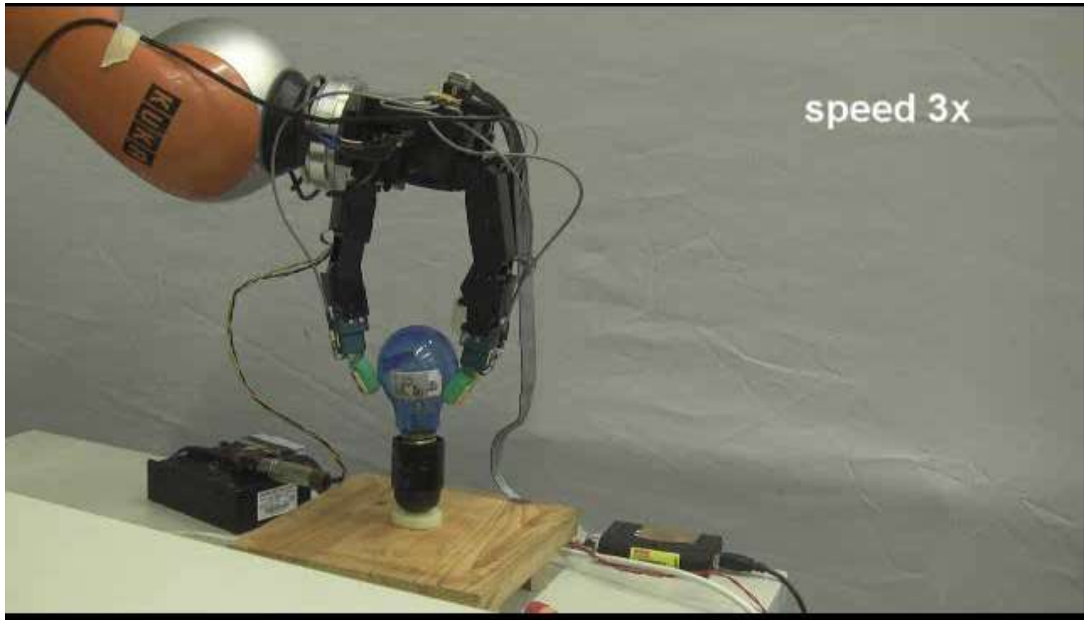
Dexterous Manipulation



Dexterous Manipulation



Dexterous Manipulation



Conclusion

- We introduced an object-level impedance learning approach for robust grasping and dexterous manipulation.
- We modeled the boundary of the workspace using a Gaussian Mixture Model.
- This learning approach could be applied in multiple ways, such as grasp adaptation (IROS 2014 paper), grasp synthesis and tool use tasks.

Miao Li, Yasemin Bekiroglu, Danica Kragic and Aude Billard, "Learning of Grasp Adaptation through Experience and Tactile Sensing", IROS 2014

Thanks for your attention!

Swiss National Center of Robotics Research

JSPS Grant-in-Aid for Young Scientists
(A) (25700028)