Transmitting Human Skills to Robots

Learning and Adaptive Control for Robotics
Ahalya Prabhakar

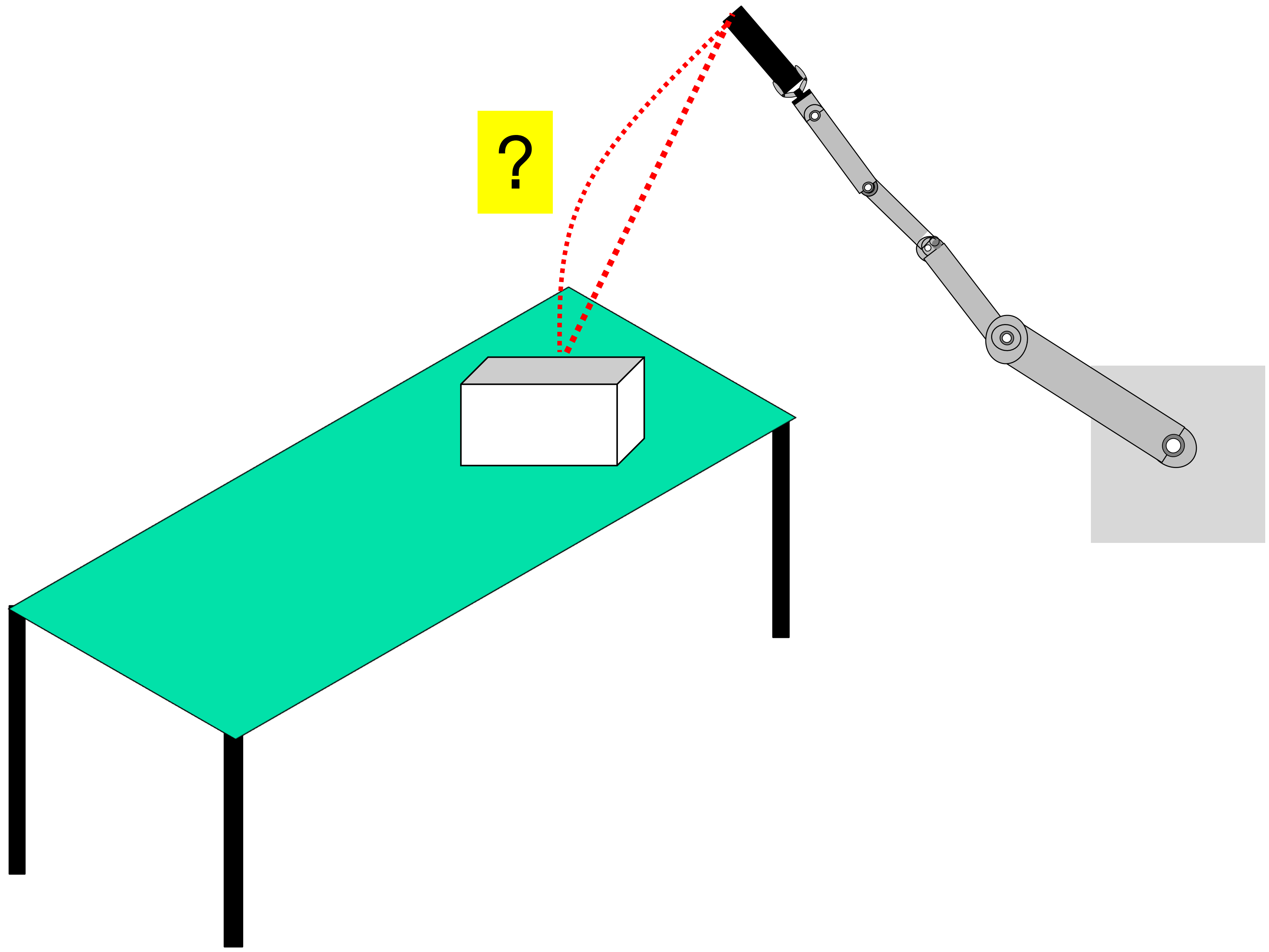
EPFL

Overview

1. Introduction of Data-driven Learning
2. Modern Approaches for Data-Gathering and Data-driven Learning
3. Interfaces for Gathering Demonstration Data
4. Challenges with Learning from Demonstration (LfD)
5. Walk through some examples
6. General Considerations and Research Directions

Motivation

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



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

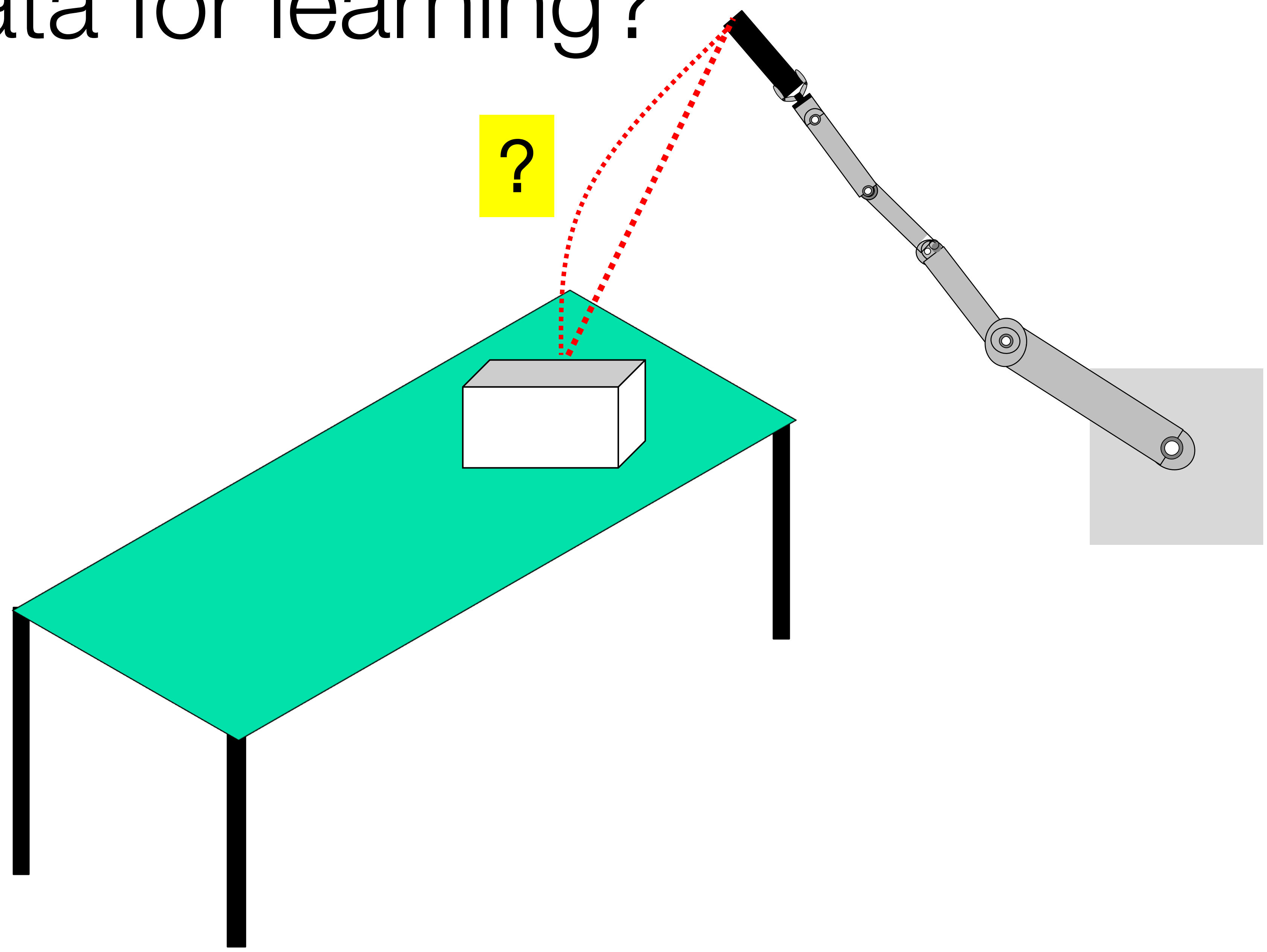
Nomenclature

- Reward function: function that provides a numerical score representing task success
- Optimal Controller: the controller that generates the action that maximizes the reward function
- Cost function represents the negative of reward (want to minimize)

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?



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

Initial Approaches

- Programming by Demonstration
  - Started in the 1980s
  - Primarily used teleoperation to provide demonstrations to the robot
  - Demonstrations consisted of position and orientation that robot would track

Programming By Demonstration

- Teleoperation
  - Manually move robot through motions for task
  - Generate Motion Primitives to enable task segmentation
- Symbolic Reasoning
  - Generate state-action-state sequences to represent task
  - Use ”if-then” rules to construct symbolic task representation
  - Originally these symbolic representations were defined as prior knowledge to the system, not learned

Data-Driven Learning from Demonstration

- Moved from copying demonstrated movements to generalizing across demonstration sets
- How to generate demonstrations that the robots can understand?
- Generate a sufficient number of examples to generalize?
- How can we generate successful robot controller from the demonstration data?

Modern Approaches

- Teach the robot how to do it through demonstrations or let the robot learn on its own

Modern Approaches

Reinforcement Learning

Let a robot explore on its own and learn an optimal controller through trial and error

Google Robotics Arm Farm

EPFL

Modern Approaches

Reinforcement Learning

Let a robot explore on its own and learn an optimal controller through trial and error

Challenges:

- Large amounts of data
- Time and energy to collect experimental data
- Safety during learning
- Typically use simulation: sim2real reality gap
- Defining reward function

Google Robotics Arm Farm

Modern Approaches

- Learning from Demonstration (LfD) & Inverse Reinforcement Learning (IRL)
  - Inverse Reinforcement Learning: Infer goal/reward function of the environment from the demonstrations

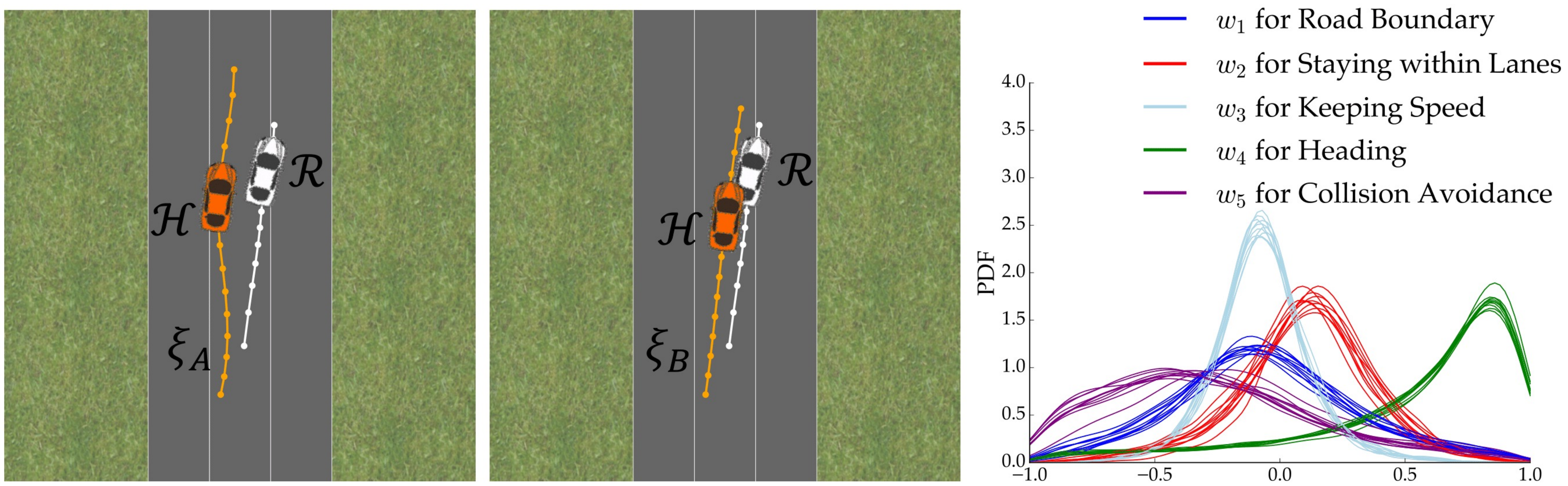


Image Source: https://iliad.stanford.edu/research/humans

Modern Approaches

- Learning from Demonstration (LfD) & Inverse Reinforcement Learning (IRL)
  - Inverse Reinforcement Learning: Infer goal/reward function of the environment from the demonstrations
    - Needs to be paired with (reinforcement) learning to learn the optimal controller

Modern Approaches

- Learning from Demonstration (LfD) & Inverse Reinforcement Learning (IRL)
  - Inverse Reinforcement Learning: Infer goal/reward function of the environment from the demonstrations
    - Feature Expectation Matching
    - Maximum Entropy IRL
    - Generative Adversarial Imitation learning
    - and many more…

Modern Approaches

Learning from Demonstration (LfD) & Inverse Reinforcement Learning (IRL)

LfD: Learn controller to imitate behavior of demonstrations

Learn motion representations from task demonstrations:

- Statistical Representation (HMMs)
- Dynamical movement primitives
- Time-Invariant Dynamical Systems

Modern Approaches

- Learn Motion Representations from Task Demonstrations

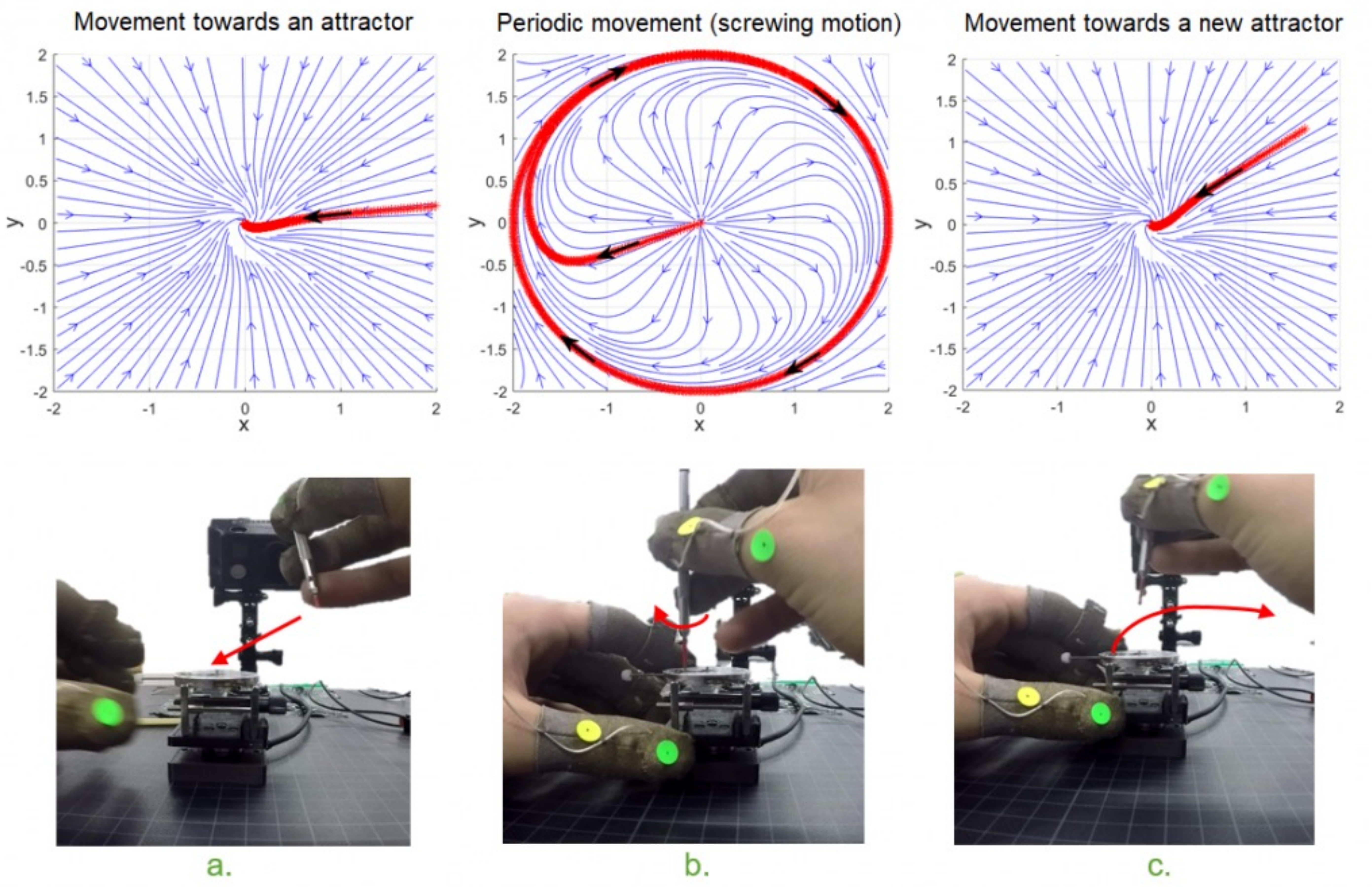
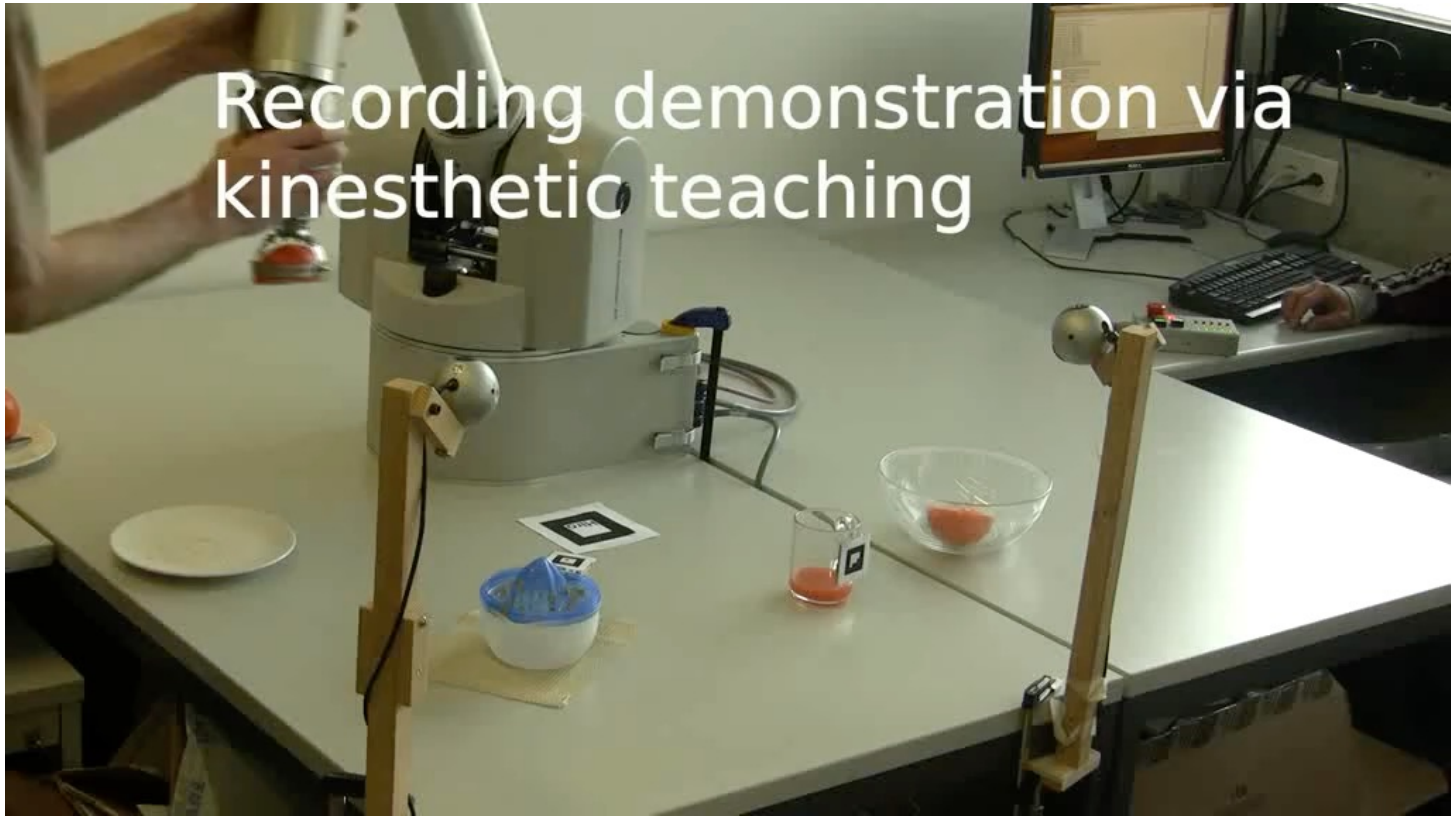


Image source: https://www.epfl.ch/labs/lasa/sahr/research/

DS-based Control to learn subtasks

- Record demonstrations of task via kinesthetic teaching to learn subtasks and dynamical systems-based controllers for task



Optimal Control

- Can be used to generate a large set of feasible trajectories that can perform the task

- Requires knowledge of a model of the task and of the dynamics of the robot

- Can require complex solvers for nonconvex optimization or integer and constraint programming

How do we gather data for learning?

Method to generate the data	**Online mode**	**Need model of robot or world**	**Trainer**	**Number of training examples**
Learning from human demonstrations	YES	NO	Anyone	<20
Optimal control	NO	YES	Skilled programmer	>100
RL (live)	NO	YES (model-based RL) NO (model-free RL)	Anyone (reward)	>100
RL (simulation)	YES	YES	Skilled programmer	>1,000

How do we gather data for learning?

Method to generate the data	Online mode	Need model of robot or world	Trainer	Number of training examples
Learning from human demonstrations	YES	NO	Anyone	<20
Optimal control	NO	YES	Skilled programmer	>100
RL (live)	NO	YES (model-based RL) NO (model-free RL)	Anyone (reward)	>100
RL (simulation)	YES	YES	Skilled programmer	>1,000

Modern Approaches

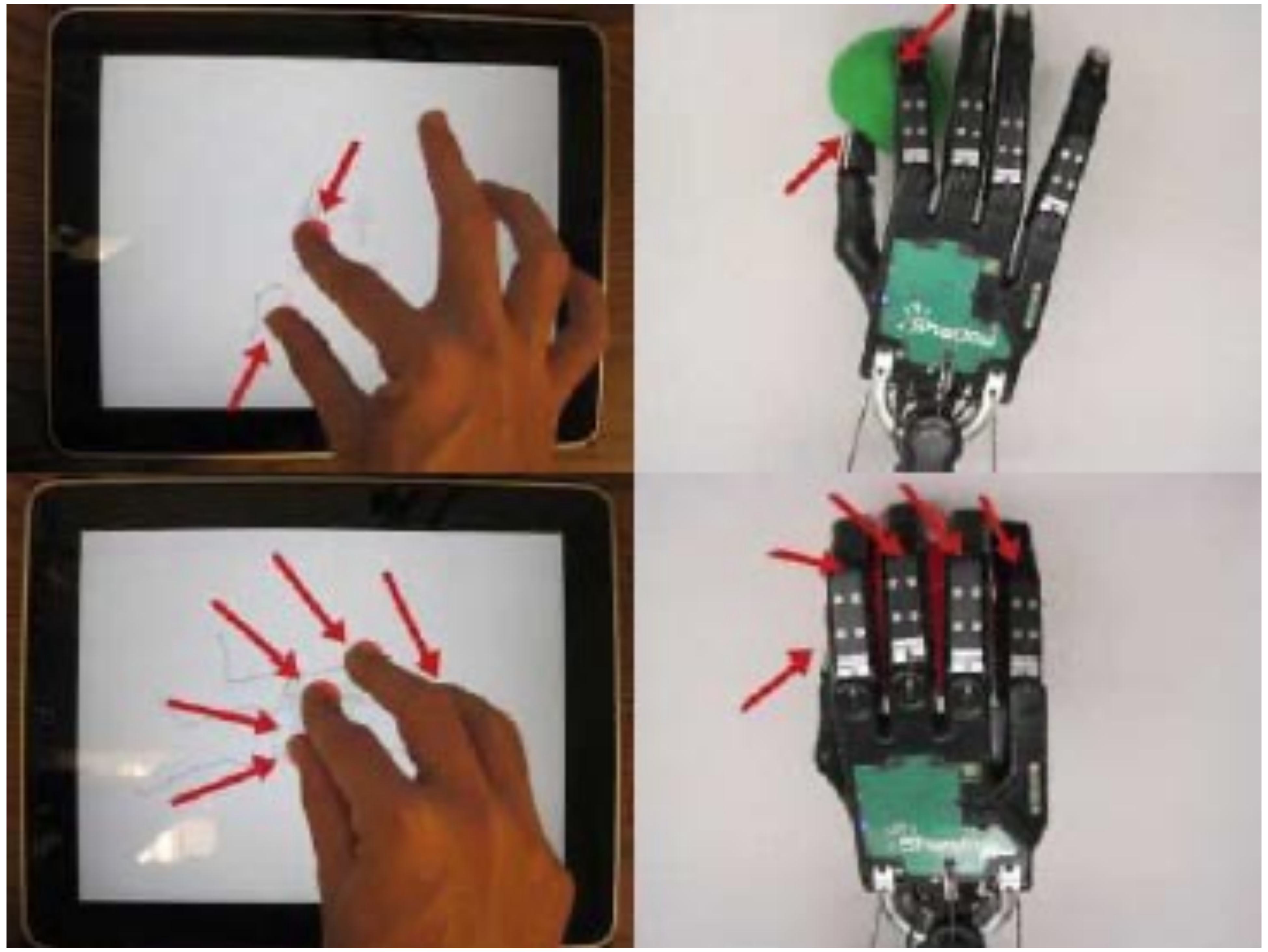
- Learning from Demonstration (LfD) & Inverse Reinforcement Learning (IRL)
- How we gather data has not changed too much…
  - Teleoperation: user controls the robot through interface
  - Kinesthetic Teaching: user physically moves the robot
  - Observational learning: robot learns from visual observation of demonstration

Teleoperation

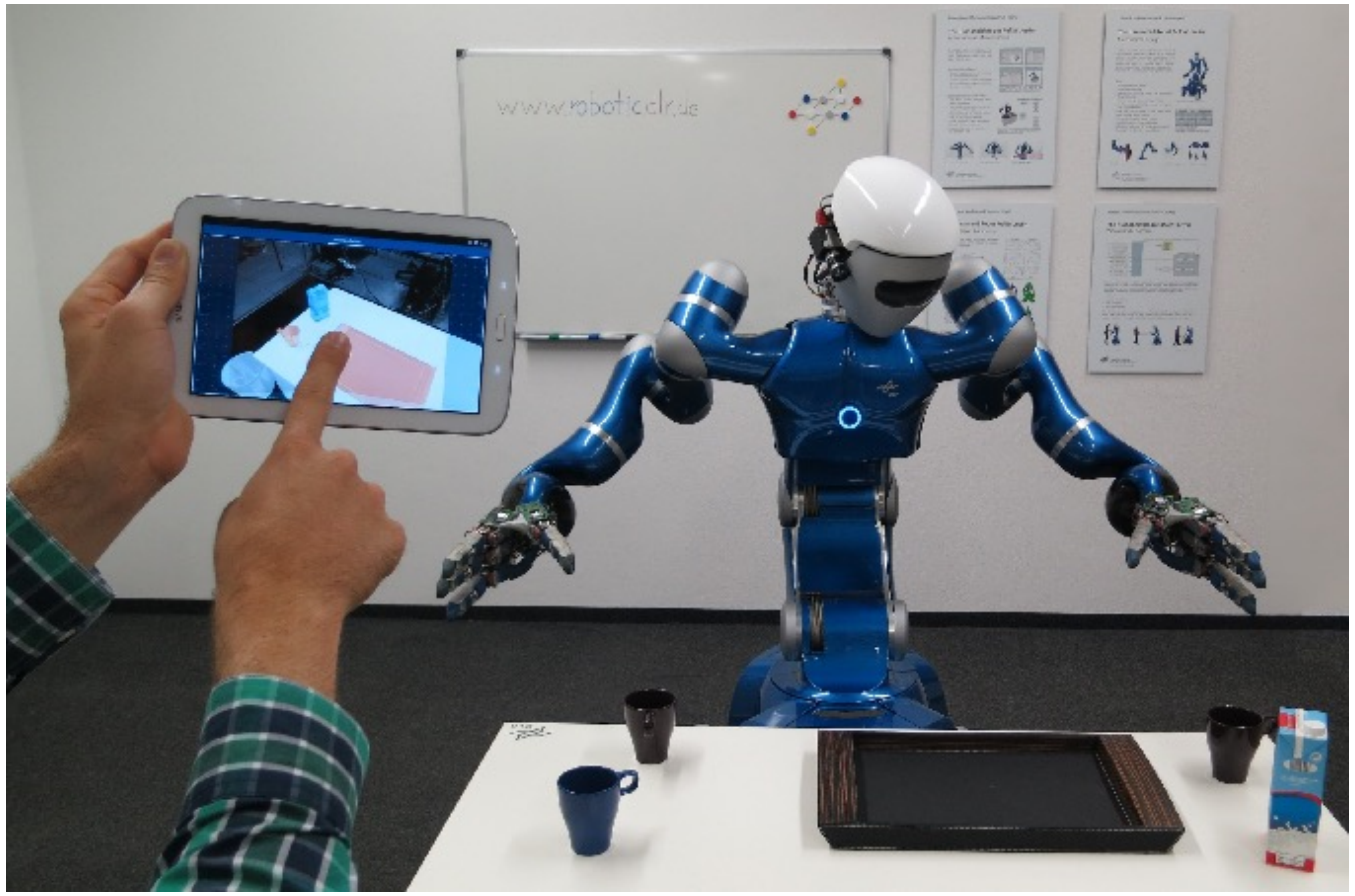
- Users control robots using some interface to perform task
- Demonstrations are used in LfD algorithm
- The quality of learning and performance is sensitive to:
    - Interface design
    - Teacher experience

Teleoperation Interfaces

- Graphical user interface/Tablet



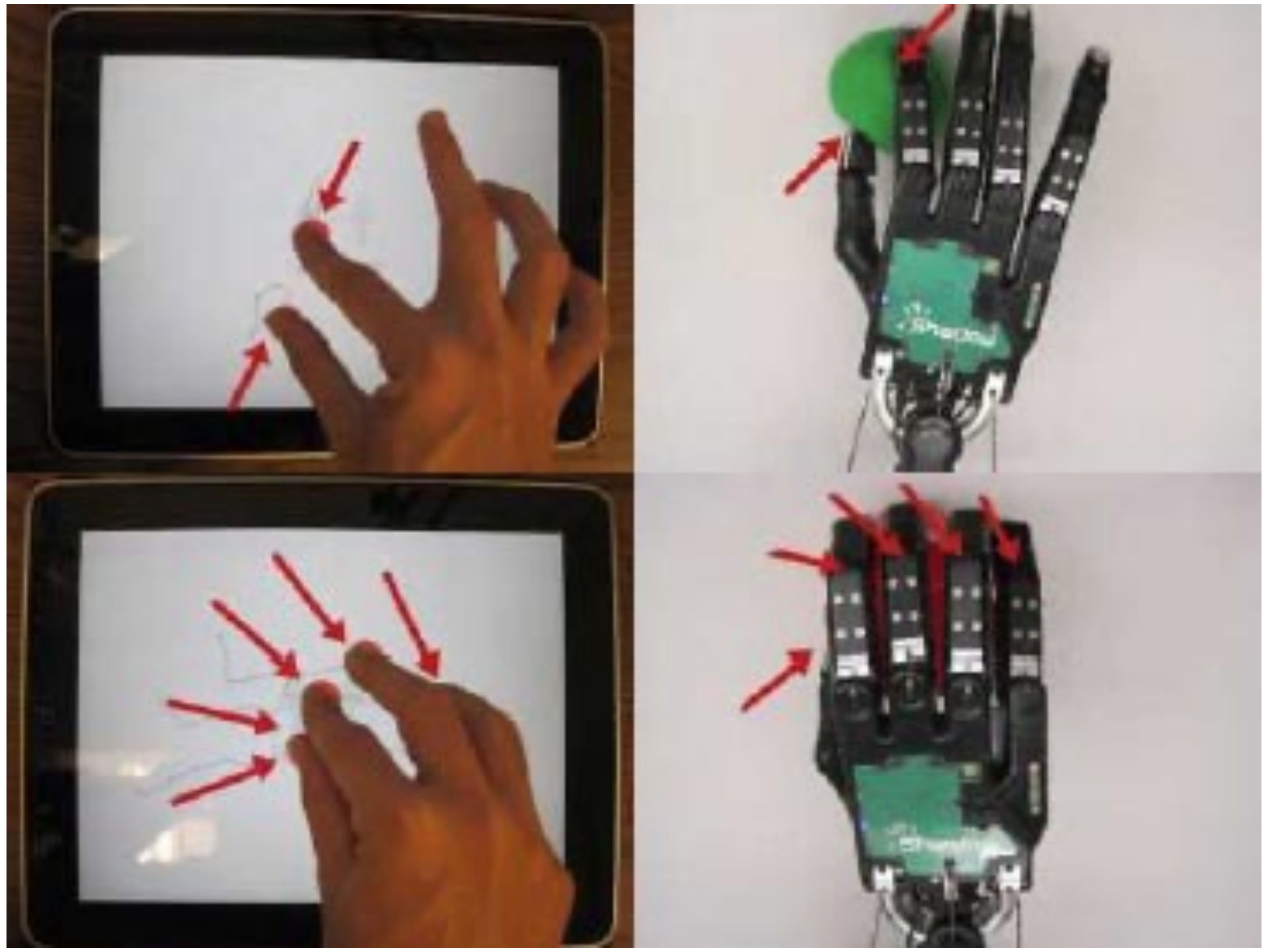
Dexterous Telemanipulation With a Multi-Touch Interface. Toh et al.
http://graphics.cs.cmu.edu/?p=223



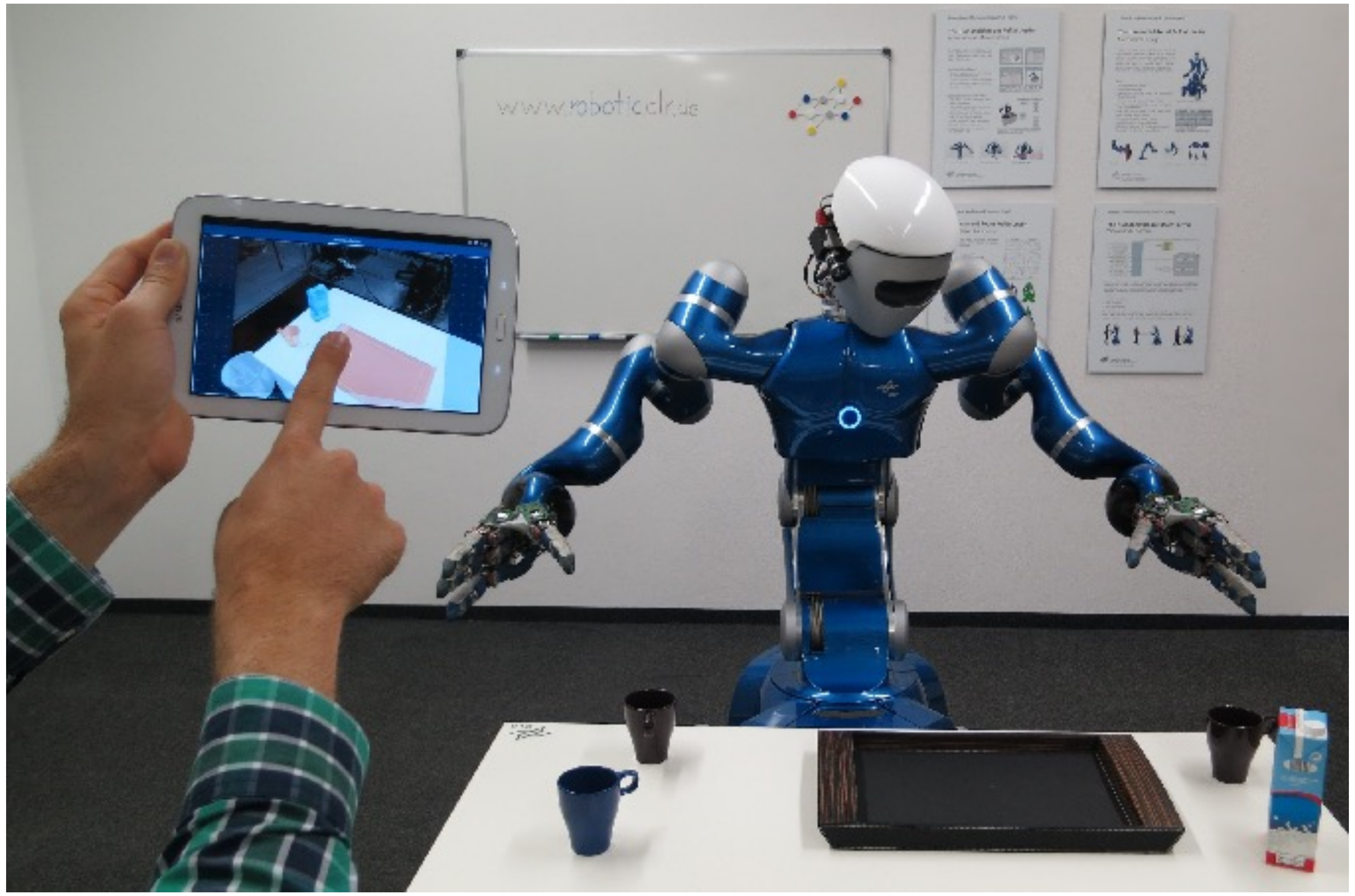
A Knowledge-Driven Shared Autonomy Human-Robot Interface for Tablet Computers. Birkenkampf et al.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7041352

Teleoperation Interfaces

- Graphical user interface/Tablet
  - Can communicate the desired motion by mimicking the motion on the tablet



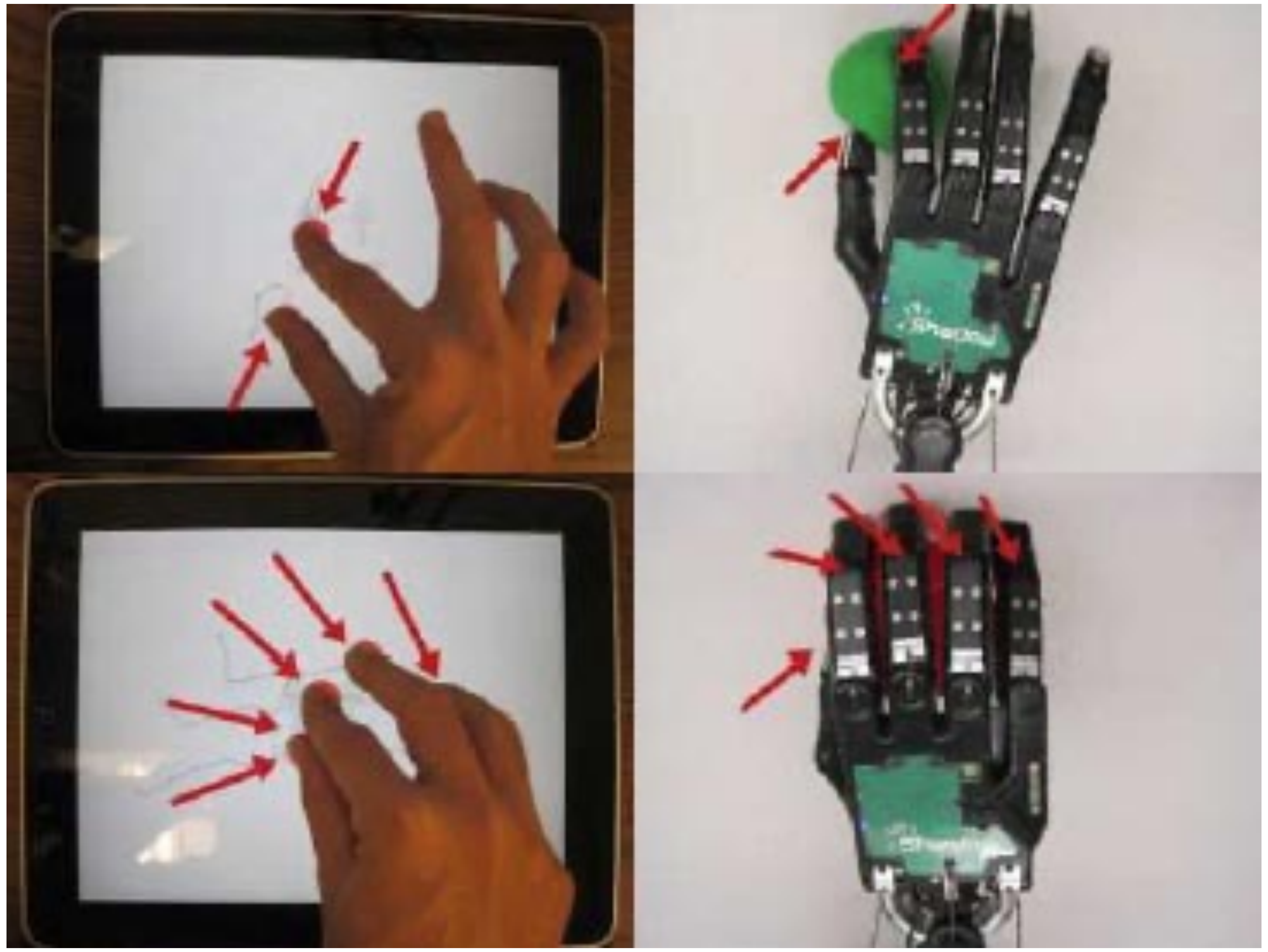
Dexterous Telemanipulation With a Multi-Touch Interface. Toh et al.
http://graphics.cs.cmu.edu/?p=223



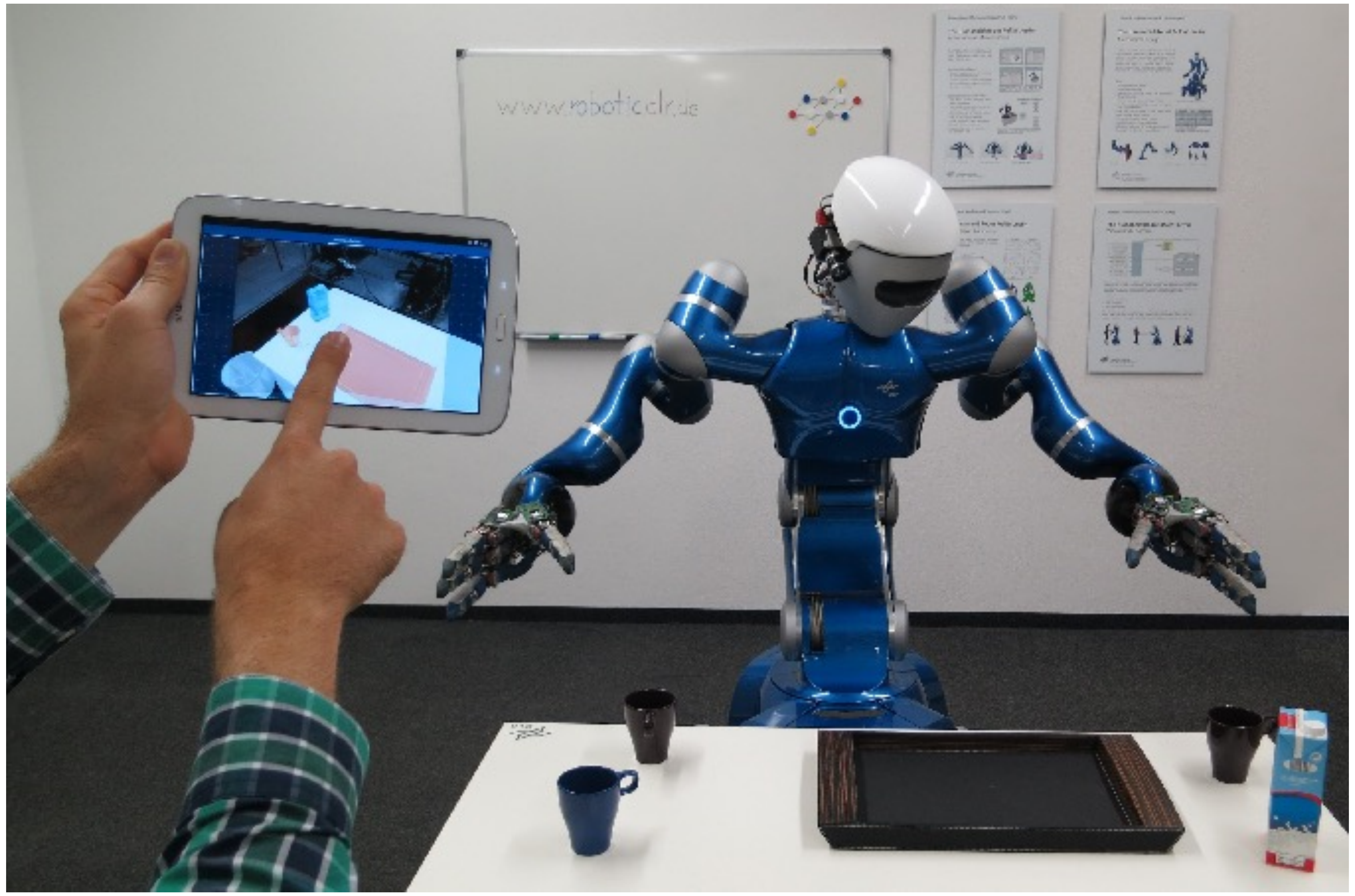
A Knowledge-Driven Shared Autonomy Human-Robot Interface for Tablet Computers. Birkenkampf et al.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7041352

Teleoperation Interfaces

- Graphical user interface/Tablet
  - Communicate higher level information (symbolic actions or interactions in environment)



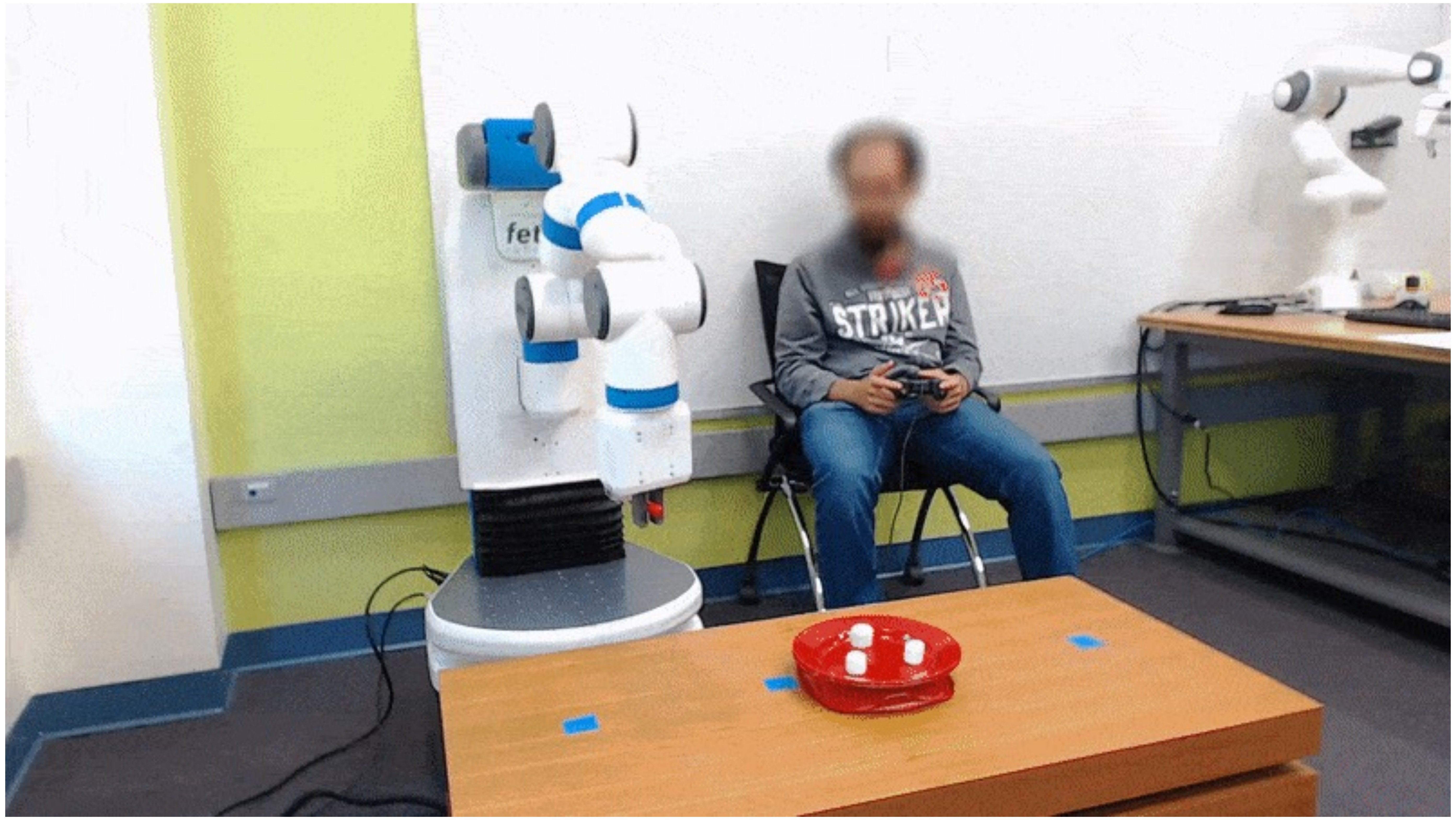
Dexterous Telemanipulation With a Multi-Touch Interface. Toh et al.
http://graphics.cs.cmu.edu/?p=223



A Knowledge-Driven Shared Autonomy Human-Robot Interface for Tablet Computers. Birkenkampf et al.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7041352

Teleoperation Interfaces

- Graphical user interface/Tablet
- Joysticks
  - Directly control a robot’s motion through joystick
  - Joints directly or end effector position



Video Source: https://iliad.stanford.edu/research/interactions

Losey, Dylan P., et al. "Controlling assistive robots with learned latent actions." 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020.

EPFL

Teleoperation Interfaces

- Graphical user interface/Tablet
- Joysticks
  - Sensitive to experience of teacher

A. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, E. Liang, Inverted autonomous helicopter flight via reinforcement learning, in: International Symposium on Experimental Robotics, 2004

Teleoperation Interfaces

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

Capio Upper Body Exoskeleton for Teleoperation by the DFKI GmbH Robotics Innovation Center.
https://robotik.dfki-bremen.de/en/research/projects/capio.html

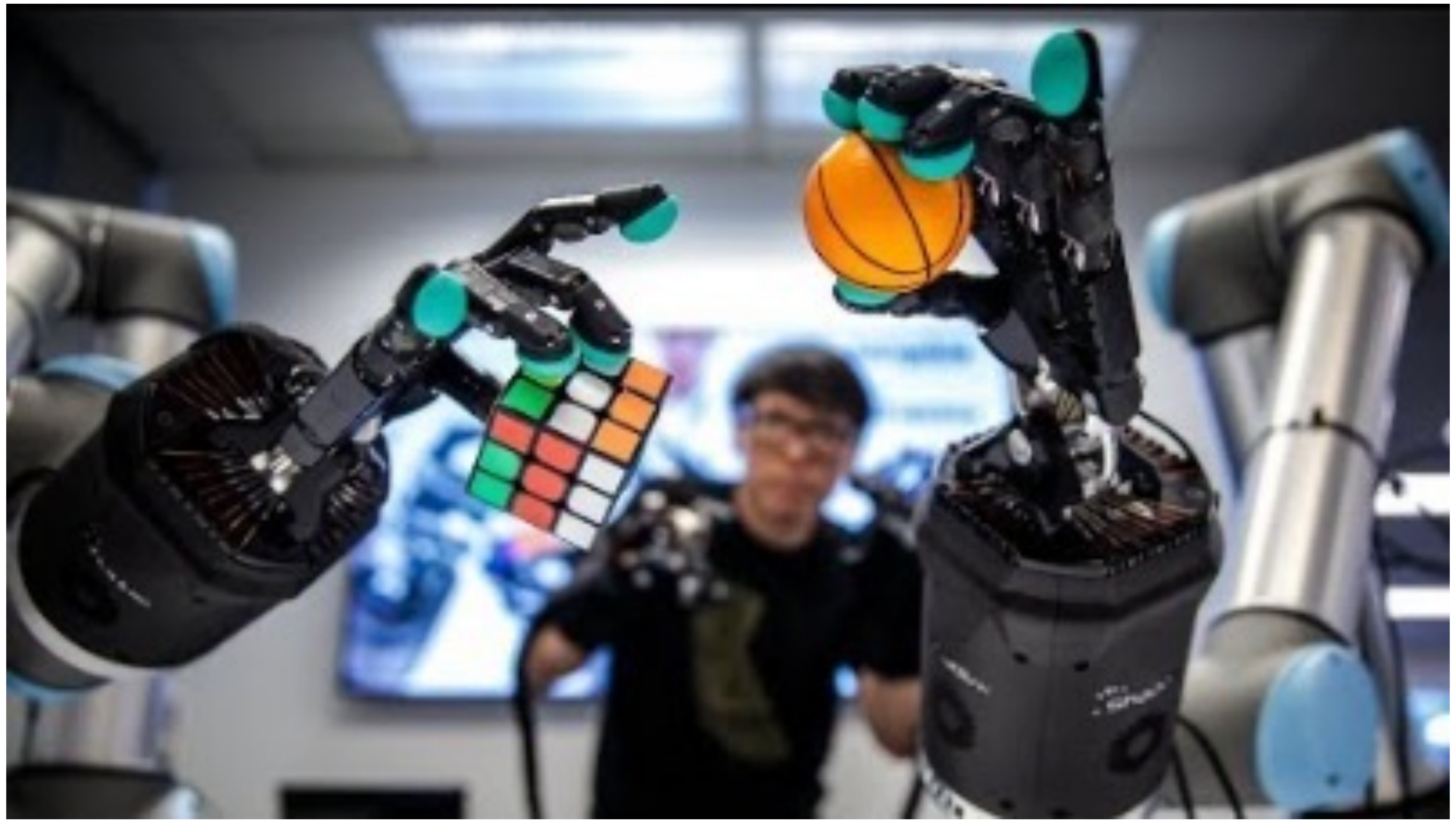
Teleoperation Interfaces

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

Davinci Surgical Robot

Teleoperation Interfaces

- Graphical user interface/Tablet
- Joysticks
- More complex devices (e.g., exoskeleton)
- Can also have feedback mechanisms/telepresence to allow the user to feel as if they are executing the task



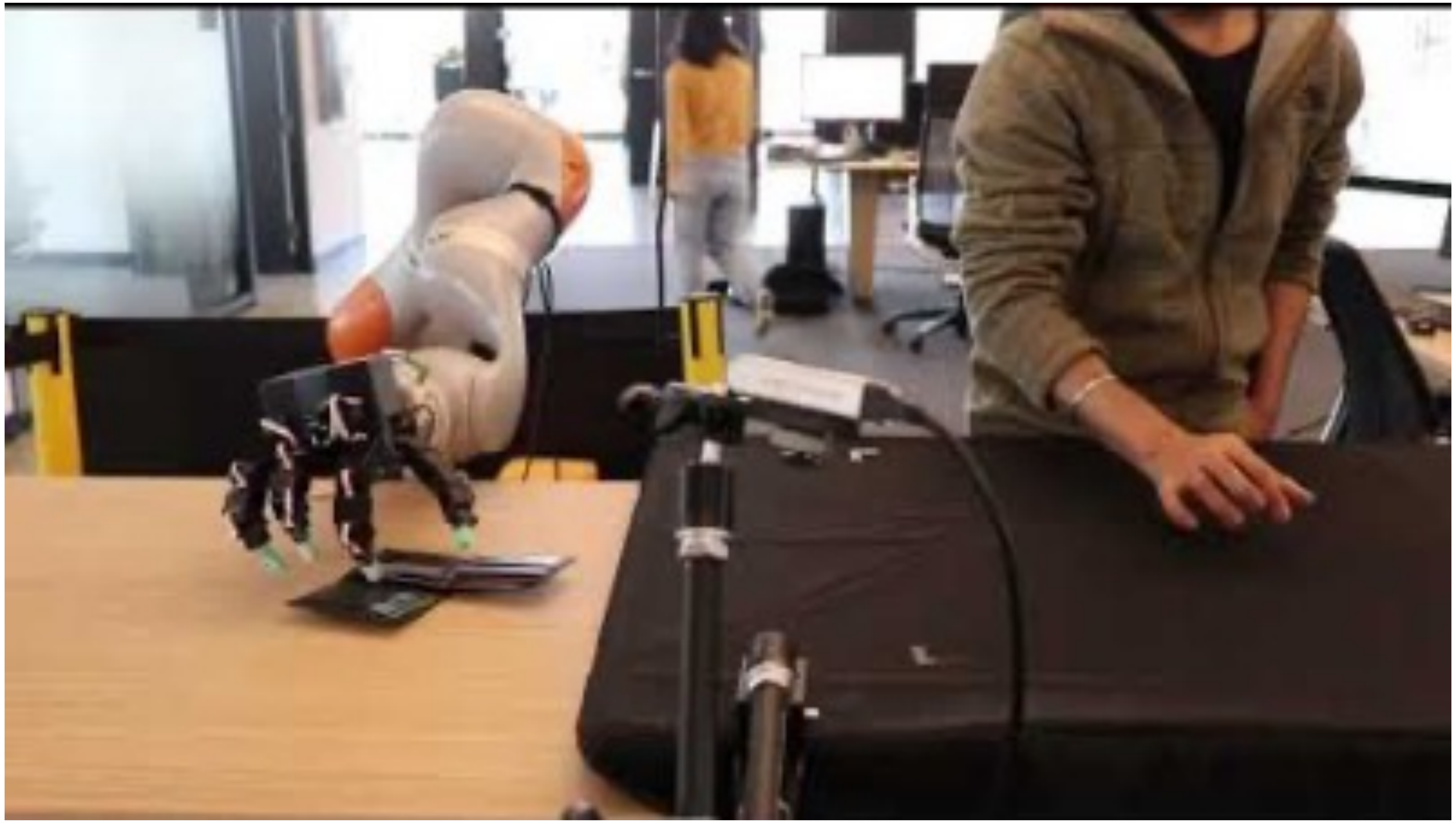
HaptX haptic glove integrated with Shadow Robot hand
Footage from Adam Savage's Tested + ShadowRobot Company and Syntouch
https://www.youtube.com/watch?v=rEoq7DMgaEc&t=24s

Kinesthetic Teaching

- Teacher physically moves robot
- Directly controlling motion of robot
- Challenging for the teacher to move the robot

Video Source: https://www.youtube.com/watch?v=xlK6U52TjRM
Learning by Demonstration, Bimanual Coordinated Task, LASA

Observational Learning: Vision Systems



DexPilot: Vision Based Teleoperation of Dexterous Robotic Hand-Arm System
Handa et al. ICRA 2020

Observational Learning: Motion Capture System

Video Source: https://youtu.be/LM4rDfW8-TU
HAL Robotics

Observational Learning: Motion Capture System

Video Source: https://youtu.be/ggLge1Rw2z4?t=77

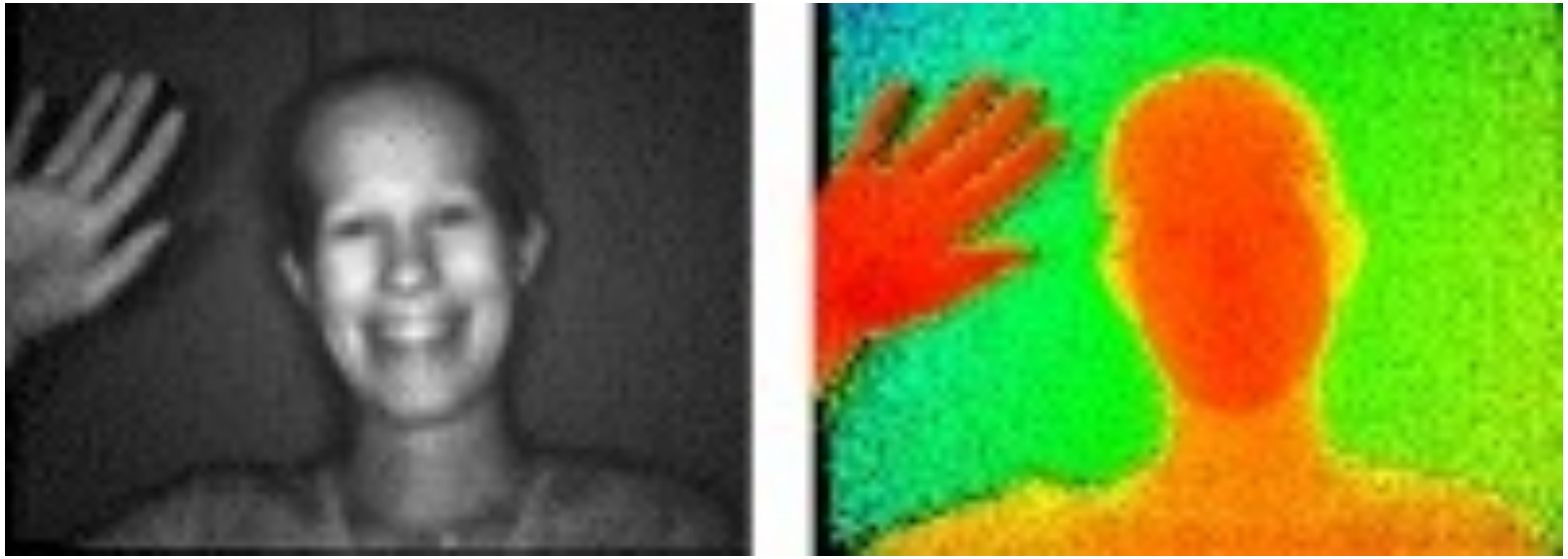
C. Stanton, A. Bogdanovych, E. Ratanasena: Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning. In proceedings of Australasian Conference on Robotics and Automation (ACRA 2012), Wellington, New Zealand, 3-5 December 2012.

Challenges of Learning from Demonstration

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.

Problem 1: Correspondence Problem

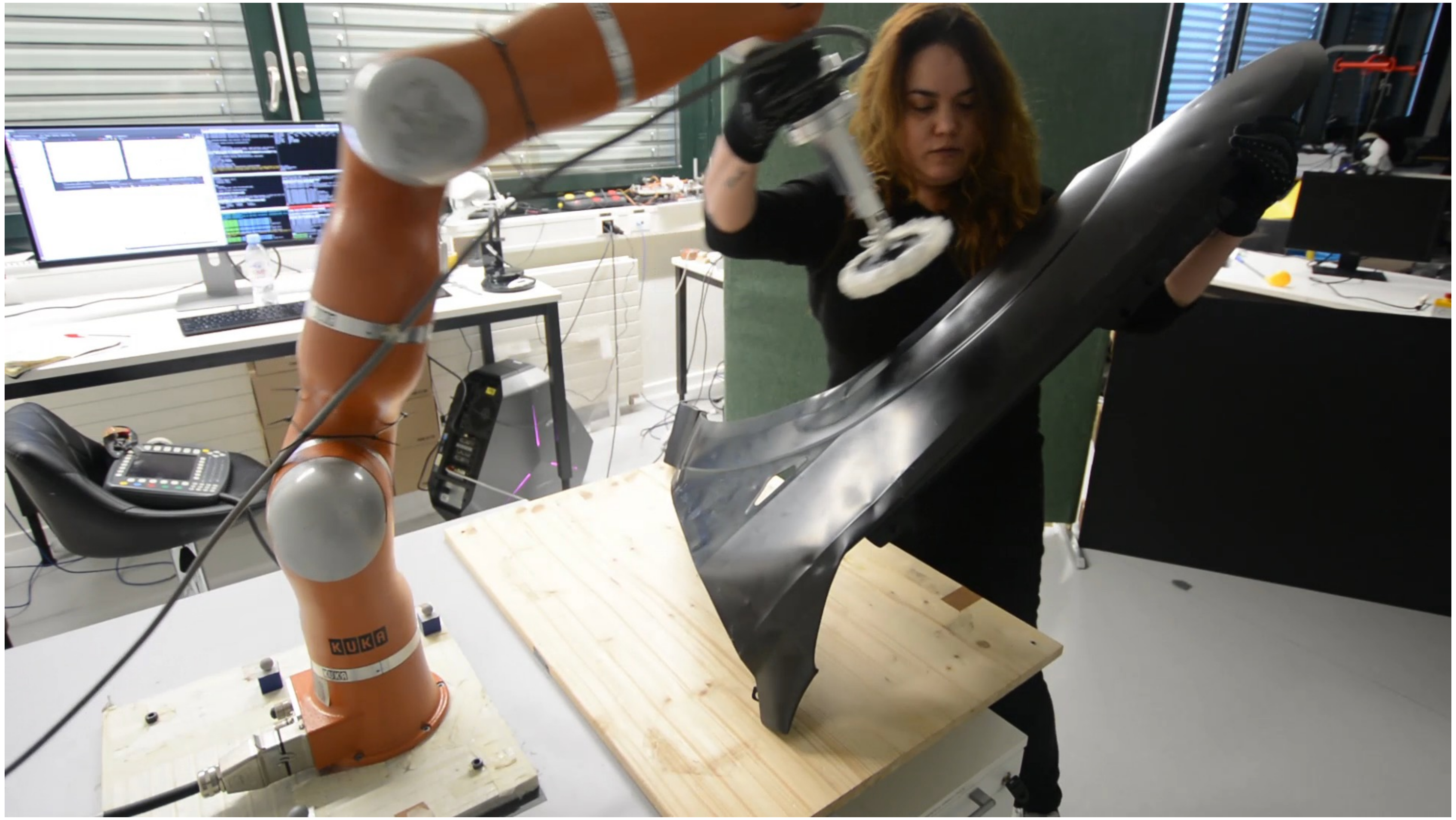


Robots do not perceive things like we do.

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

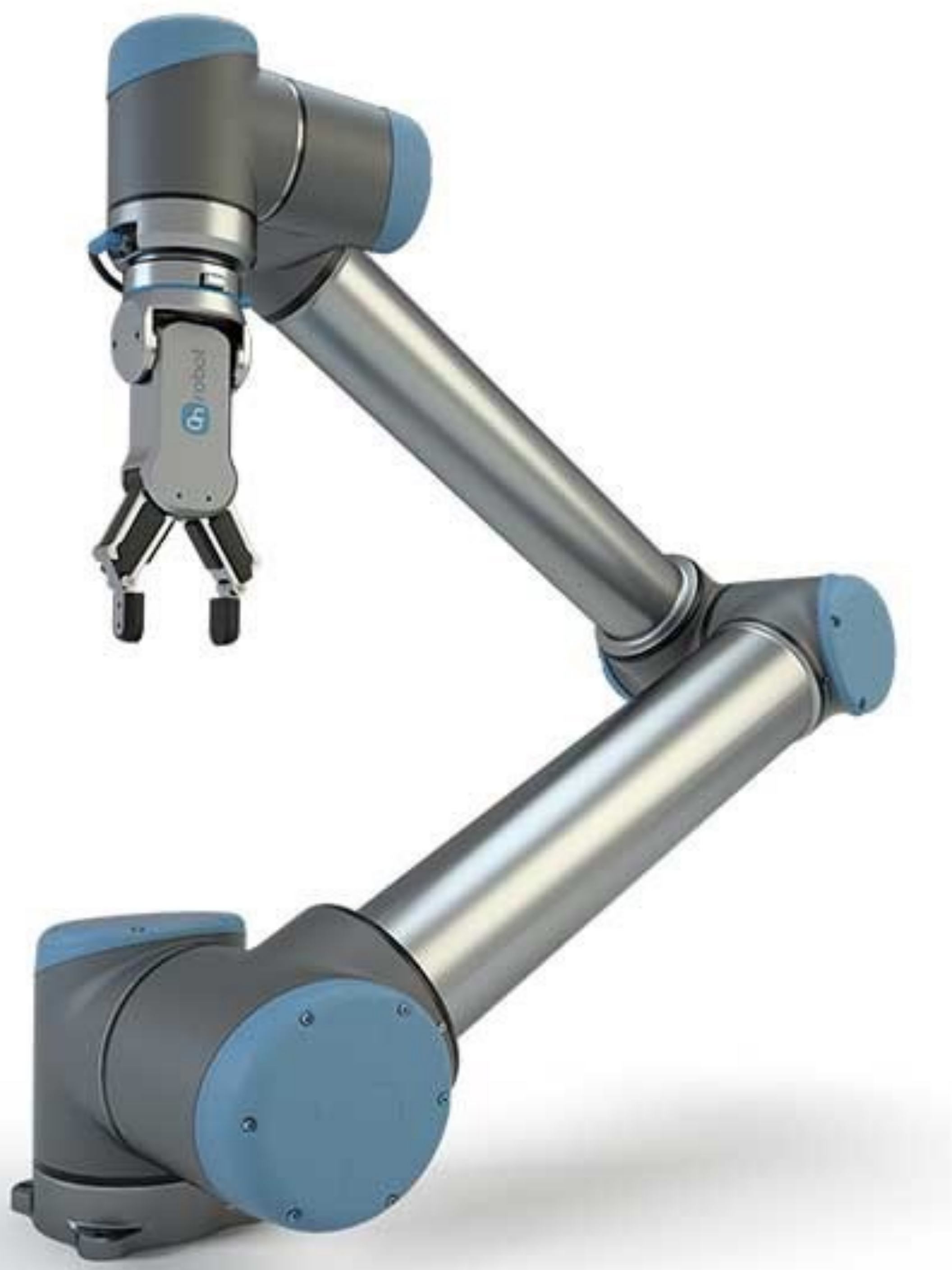
Problem 1: Correspondence Problem

- Teachers need to train themselves before training the robots.



Problem 2: Learning is Data-Sensitive

- Data is robot-dependent

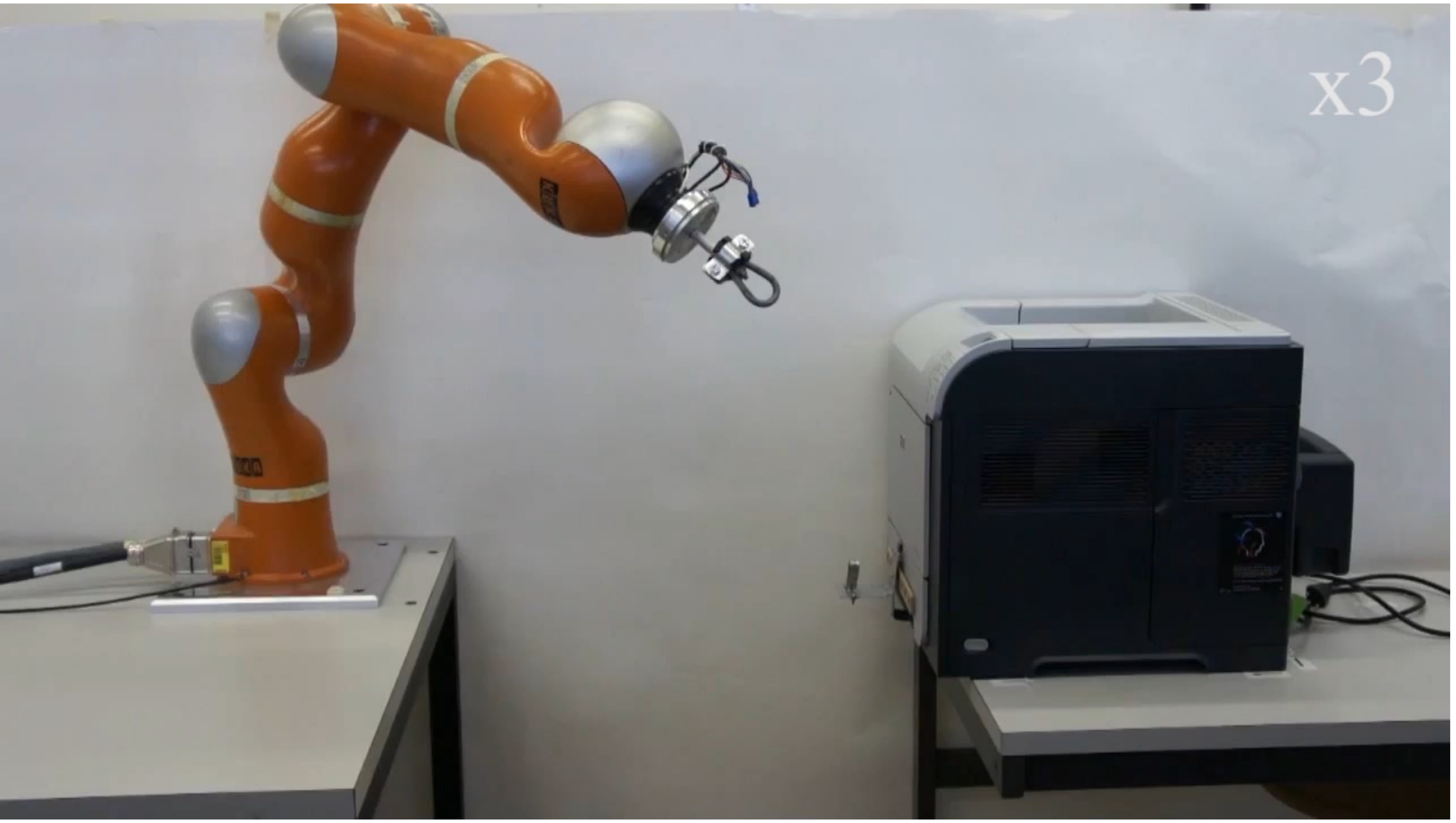


UR5: 6DOF

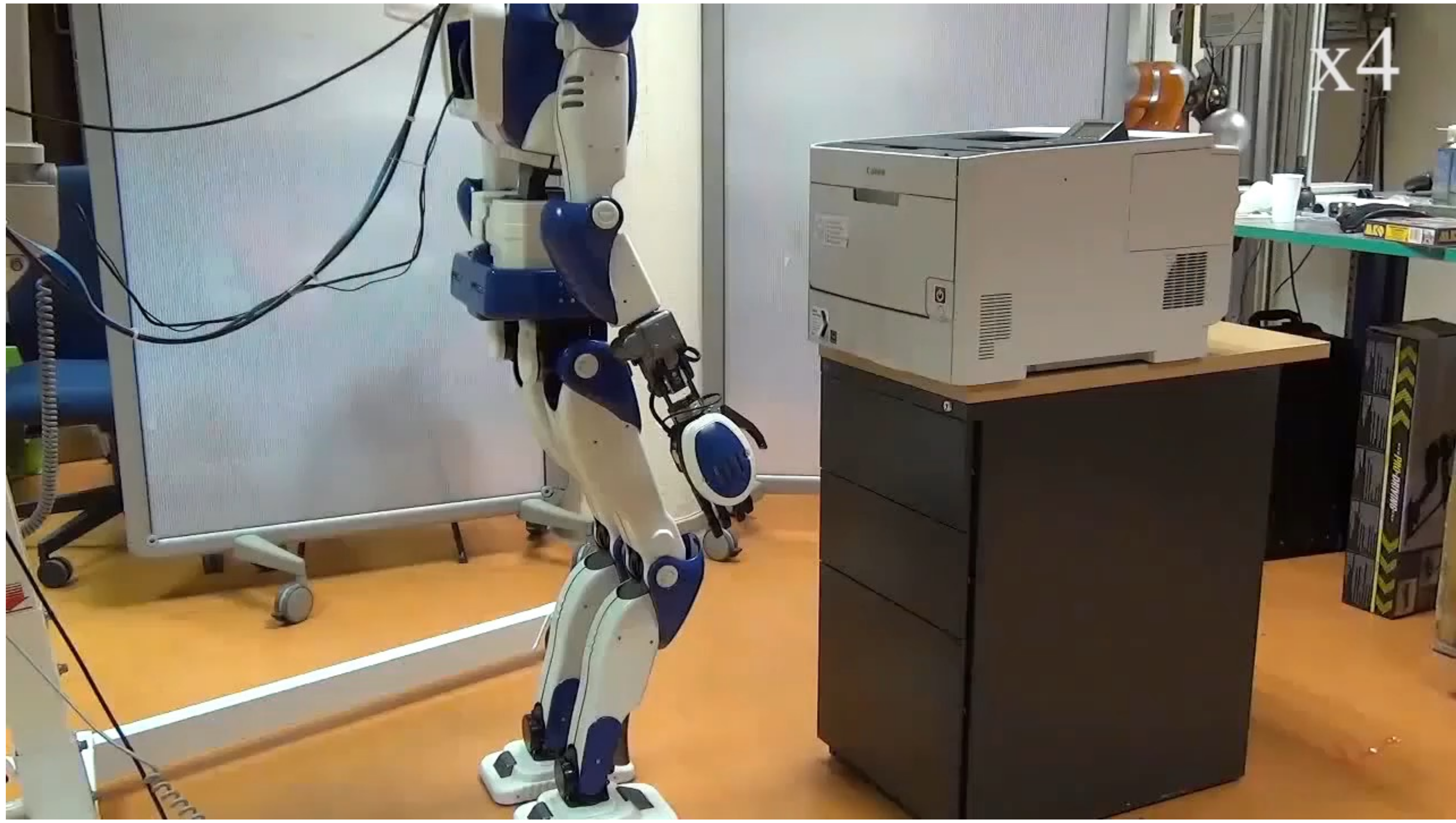
Franka Panda: 7DOF

Problem 2: Learning is Data-Sensitive

- Data is environment-dependent



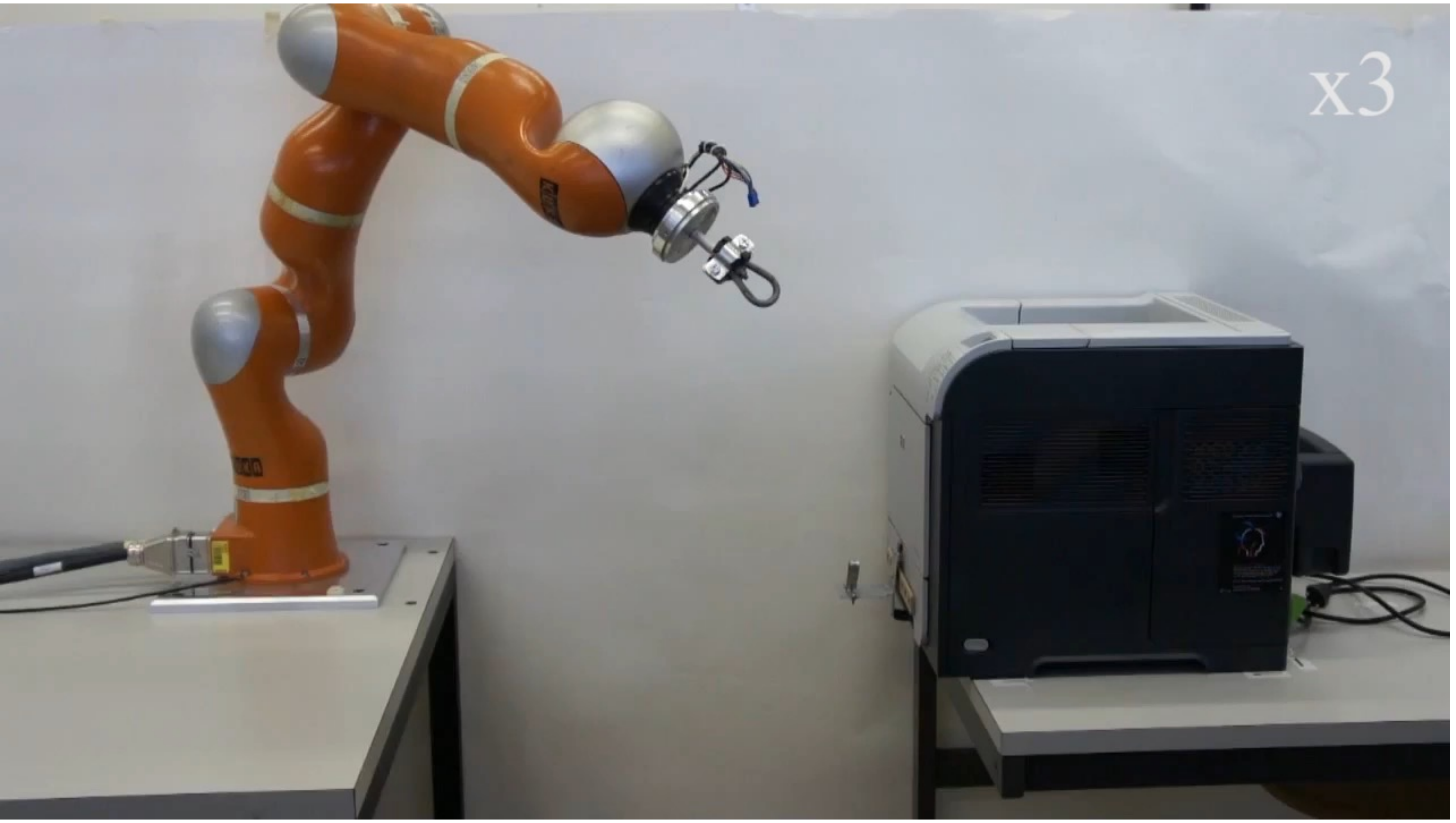
Model Learned at EPFL



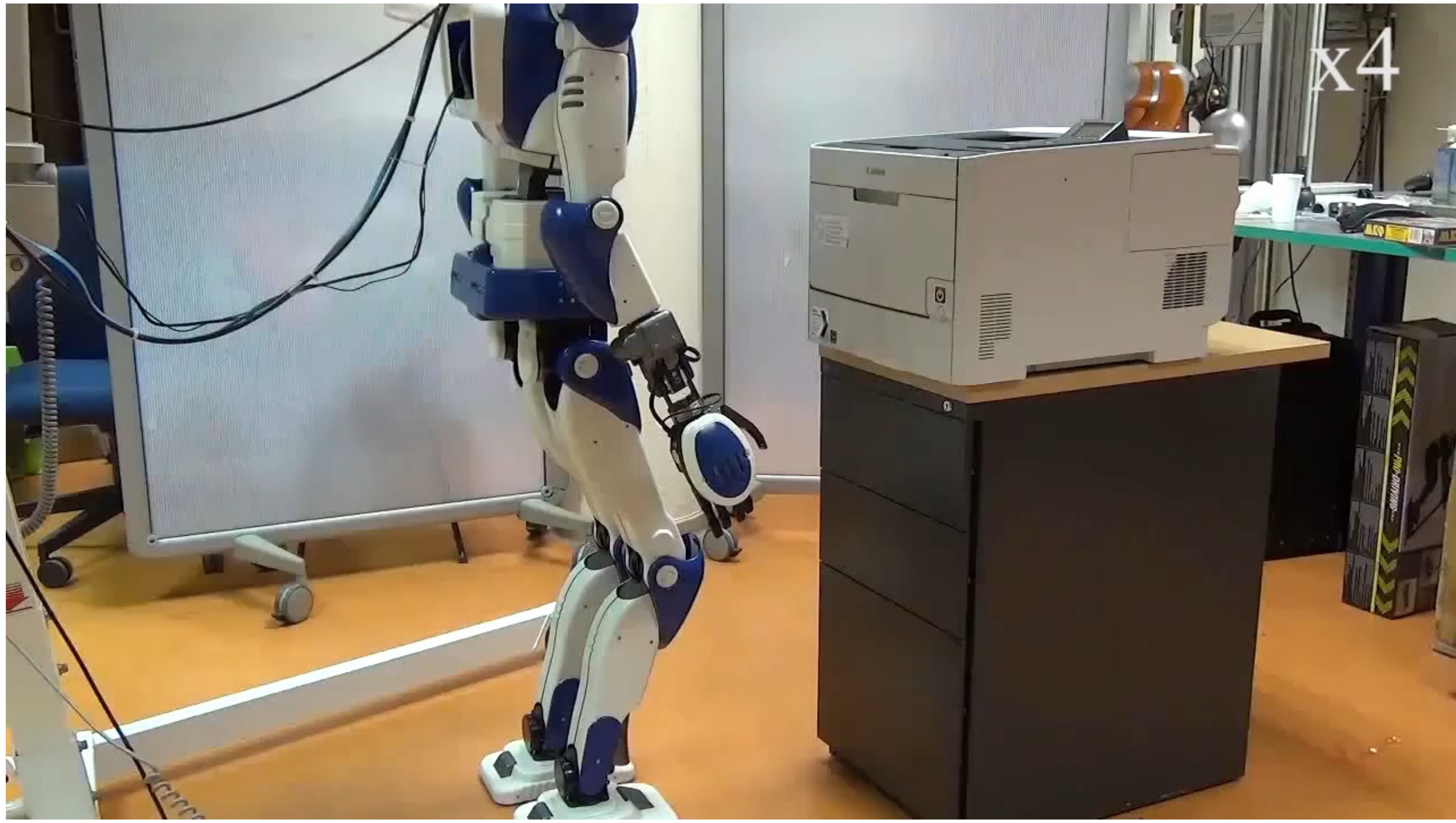
Model transferred at AIST/JRL

Problem 2: Learning is Data-Sensitive

Need Transfer Learning methods



Model Learned at EPFL



Model transferred at AIST/JRL

EPFL

Problem 3: Variability in Task Definition

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

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

Break Time

Learning from Demonstration: Examples

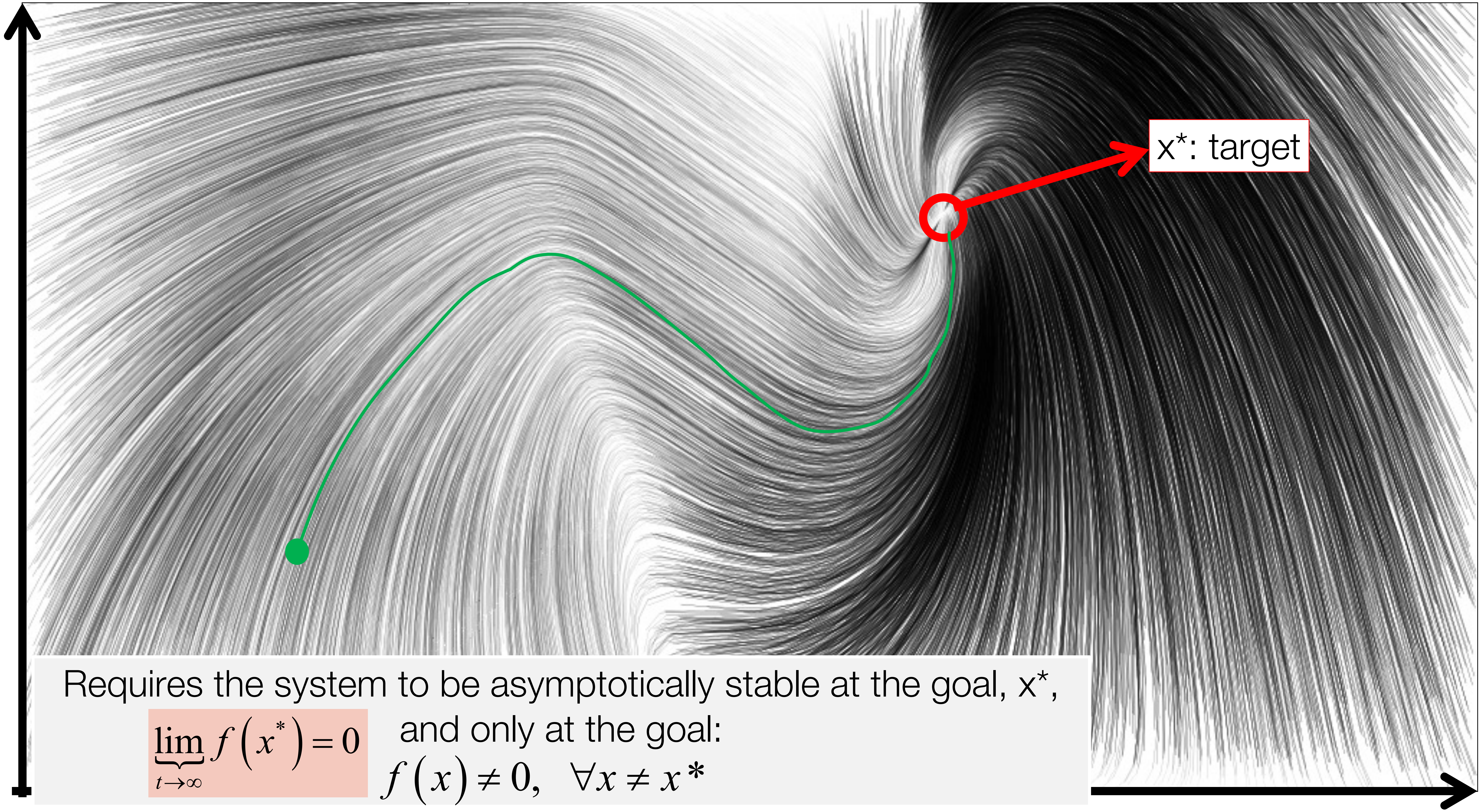
- How to use DS-based control to learn a task
- How to extend with compliant control to improve task performance

Learning from Demonstration: Using dynamical systems

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

Modelling Hitting Task using Dynamical Systems-Based Control

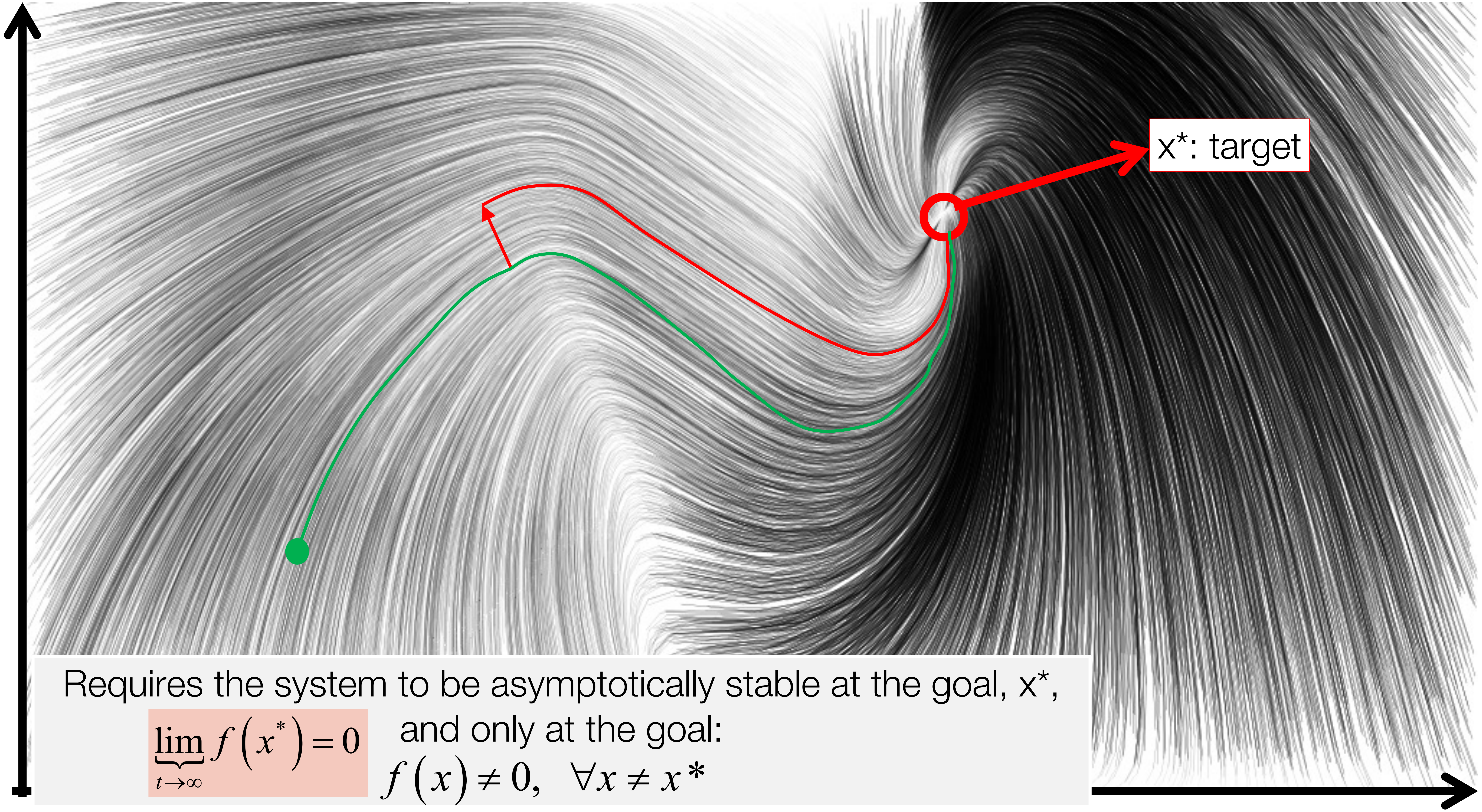


Requires the system to be asymptotically stable at the goal, x*,

$\underbrace{\lim}_{t\to\infty} f\left(x^*\right)=0$ and only at the goal:

$f\left(x\right)\neq 0, \quad \forall x \neq x^*$

Modelling Hitting Task using Dynamical Systems-Based Control



Requires the system to be asymptotically stable at the goal, x*,

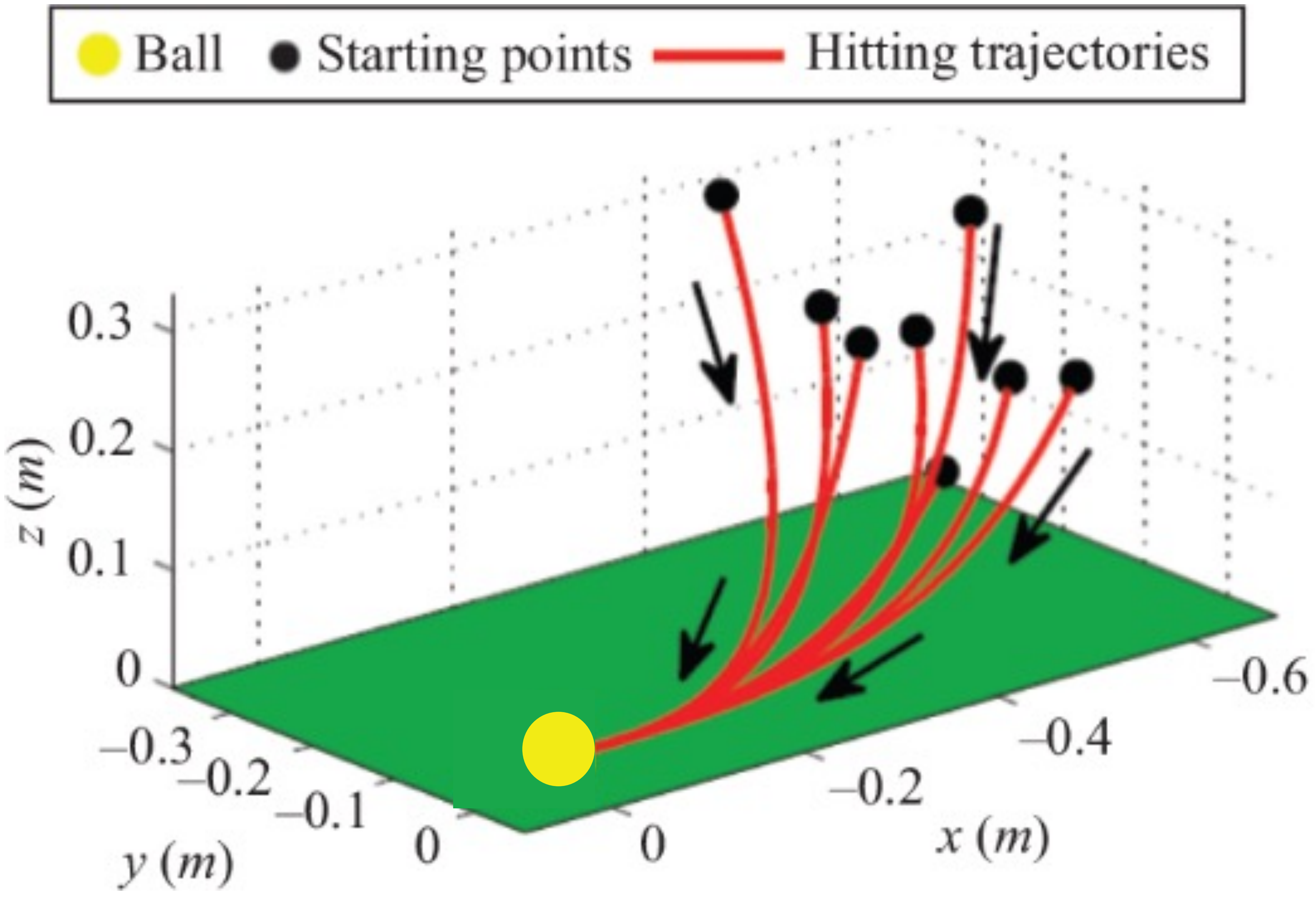
$\underbrace{\lim}_{t\to\infty} f\left(x^*\right) = 0$ and only at the goal:

$f\left(x\right) \neq 0, \quad \forall x \neq x^*$

Modelling Hitting Task using Dynamical Systems-Based Control

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

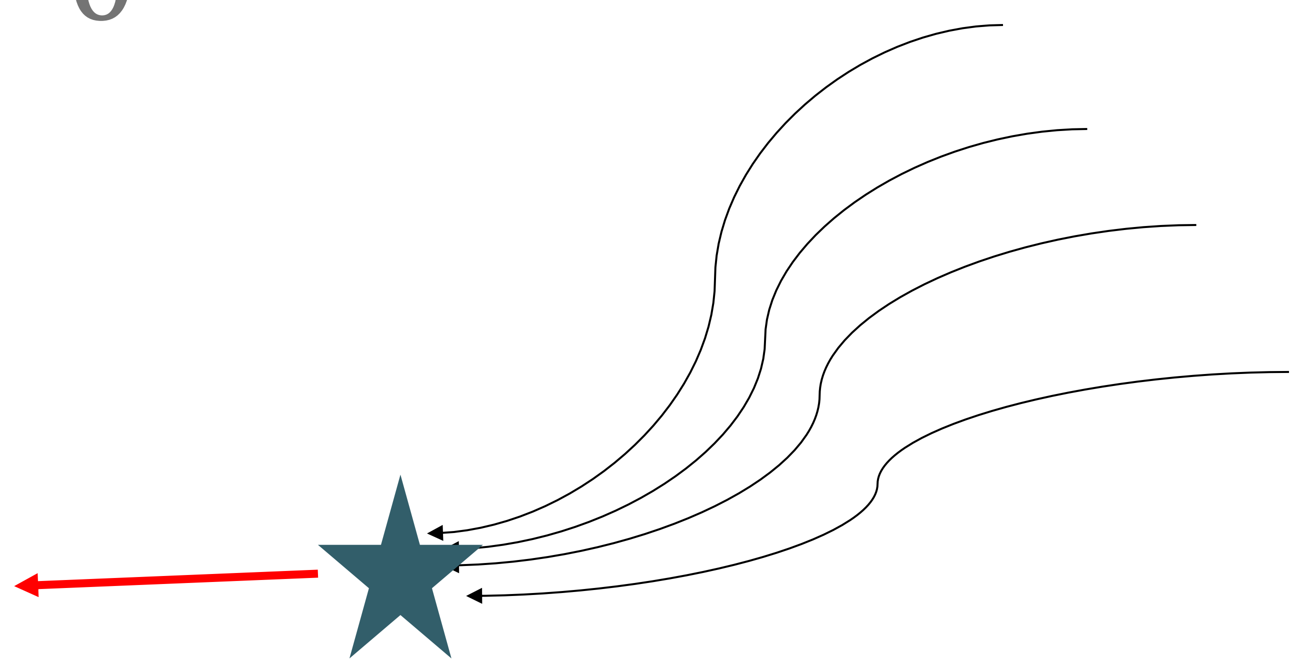
- Guarantees asymptotically reaching and stabilizing at attractor: $\lim_{\{t \to \infty\}} x = x^*$, where x^*: Ball Location

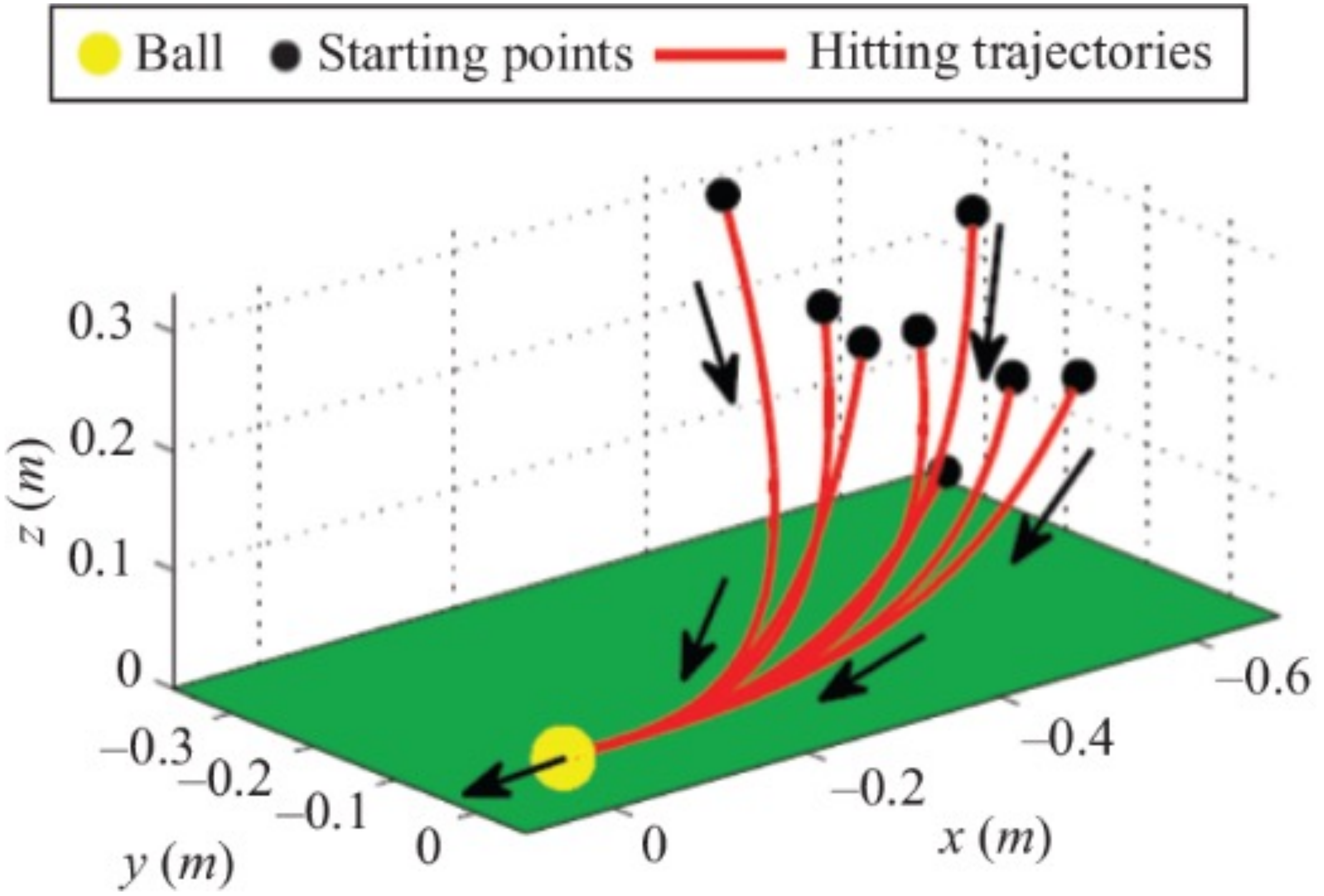


Modelling Hitting Task using Dynamical Systems-Based Control

- $\dot{x} = f(x)$ with a fixed point attractor $f(x^*) \neq 0$

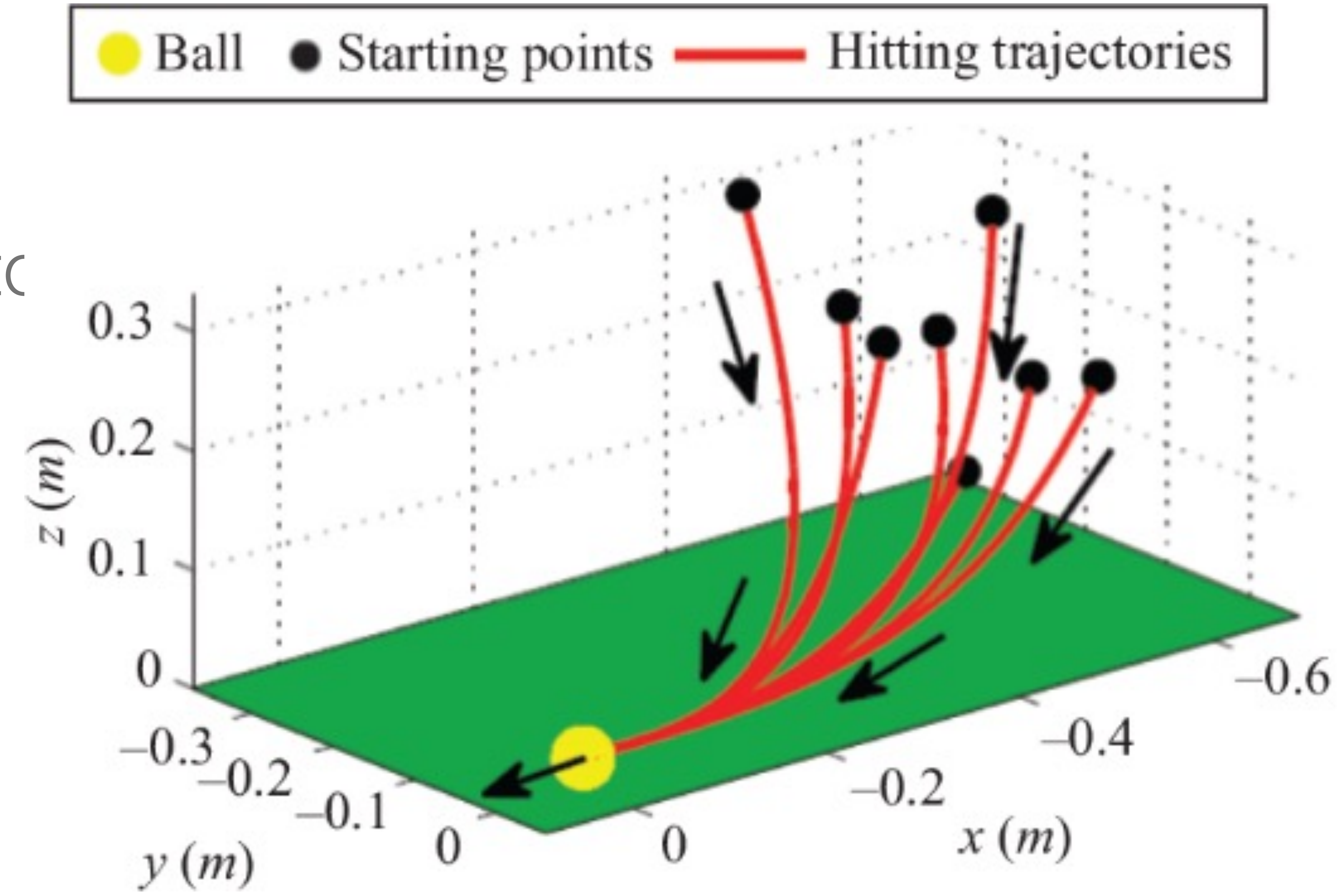
- Desired velocity at attractor
  $\dot{x}^* \neq 0$





Modelling Hitting Task using Dynamical Systems-Based Control

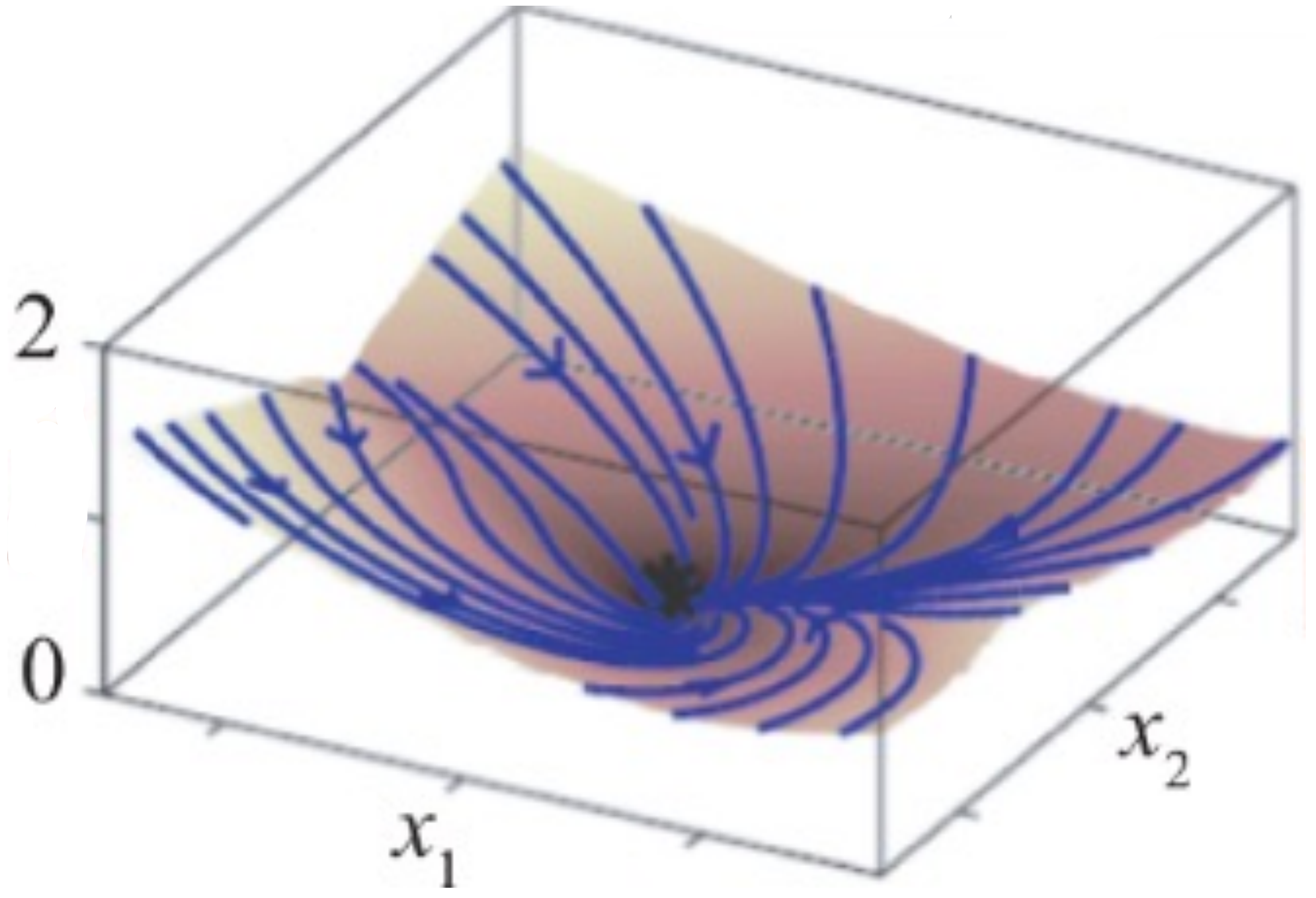
- Modulate the initial stable DS
- Dynamical system representing this motion using stable estimatc of dynamical systems (SEDS) method: $\dot{x} = g(x)$, $g(x^*) \neq 0$
- $f(x) = \dot{x} = M(x) * E(x)$
- Target field: $E(x)$
- Strength Factor: $M(x)$

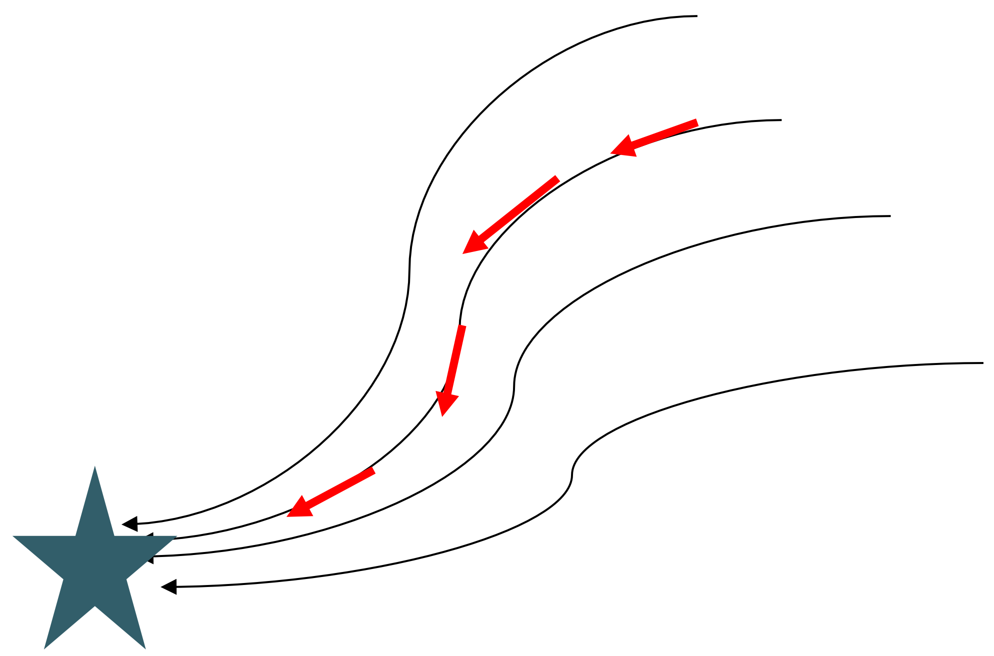


EPFL

Modelling Hitting Task using Dynamical Systems-Based Control

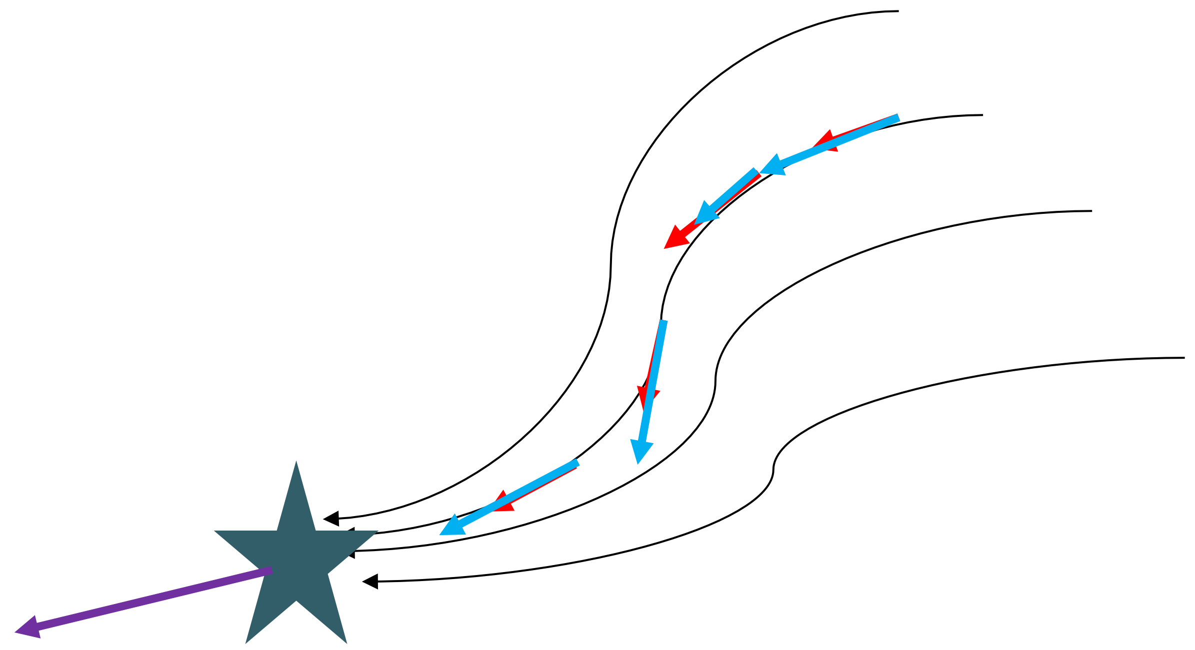
- Dynamical system representing this: $\dot{x} = g(x)$, $g(x^*) \neq 0$
- We only store the unitary vector field $\mathrm{E(x)} \coloneqq \|g(x)\| = 1 \; \forall x$
- $E(x) = \frac{g(x)}{\|g(x)\|}$
- Embeds the orientation at target and asymptotic stability





Modelling Hitting Task using Dynamical Systems-Based Control

- We also learn a function $M(x)$ from the demonstration set
- Embeds the amplitude of the velocity when approaching the ball during demonstration: $M(x) = ||\dot{x}||$
- Can be learned using any ML regression technique

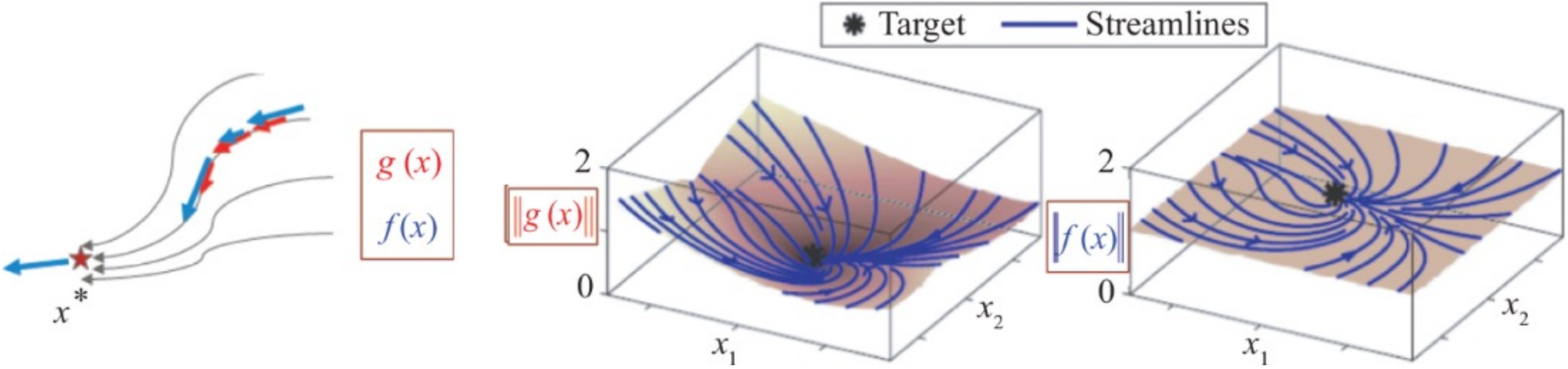


Modelling Hitting Task using Dynamical Systems-Based Control

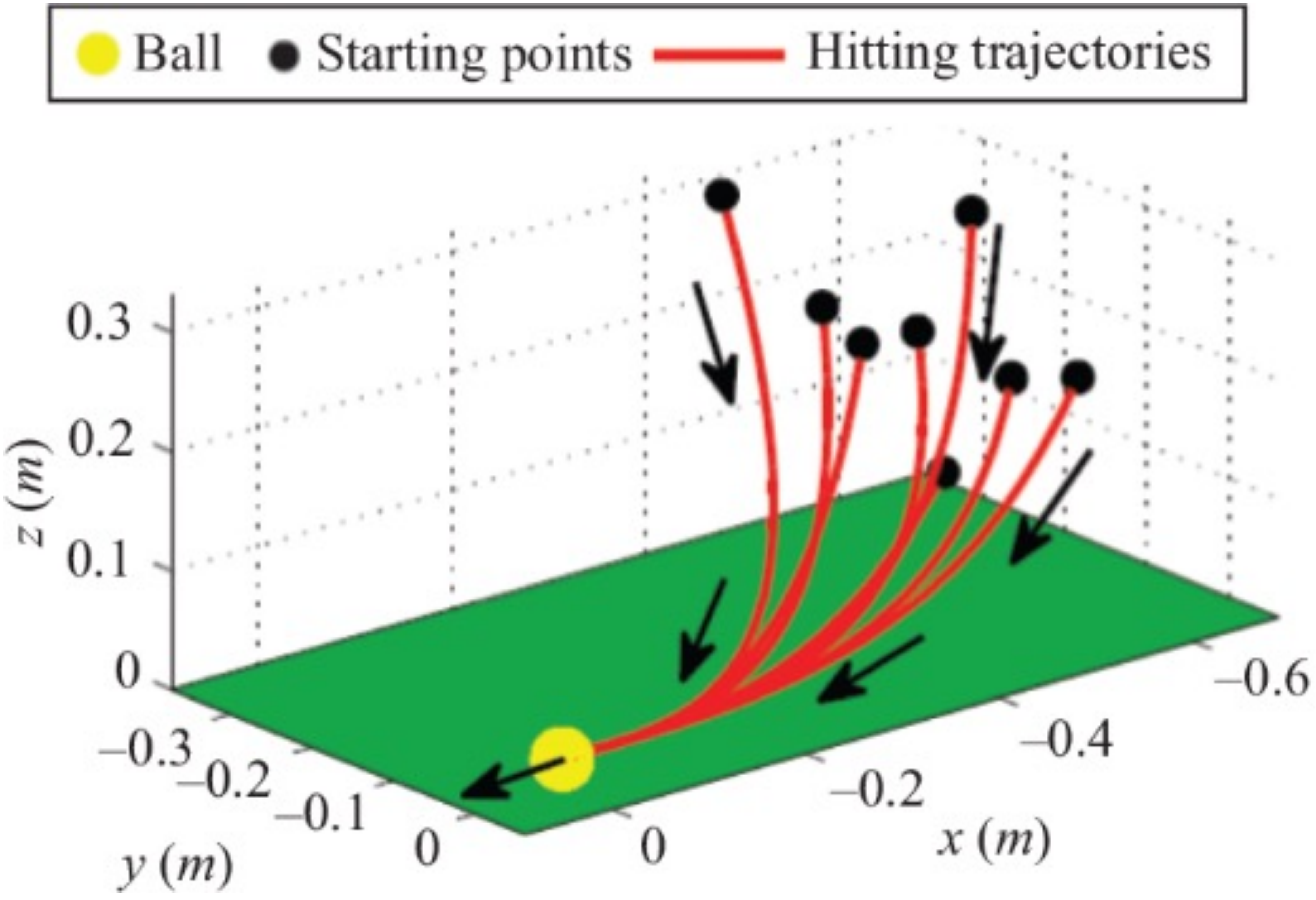
$$f(x) = M(x) * E(x)$$

Velocity at target

Orientation at target and asymptotic stability



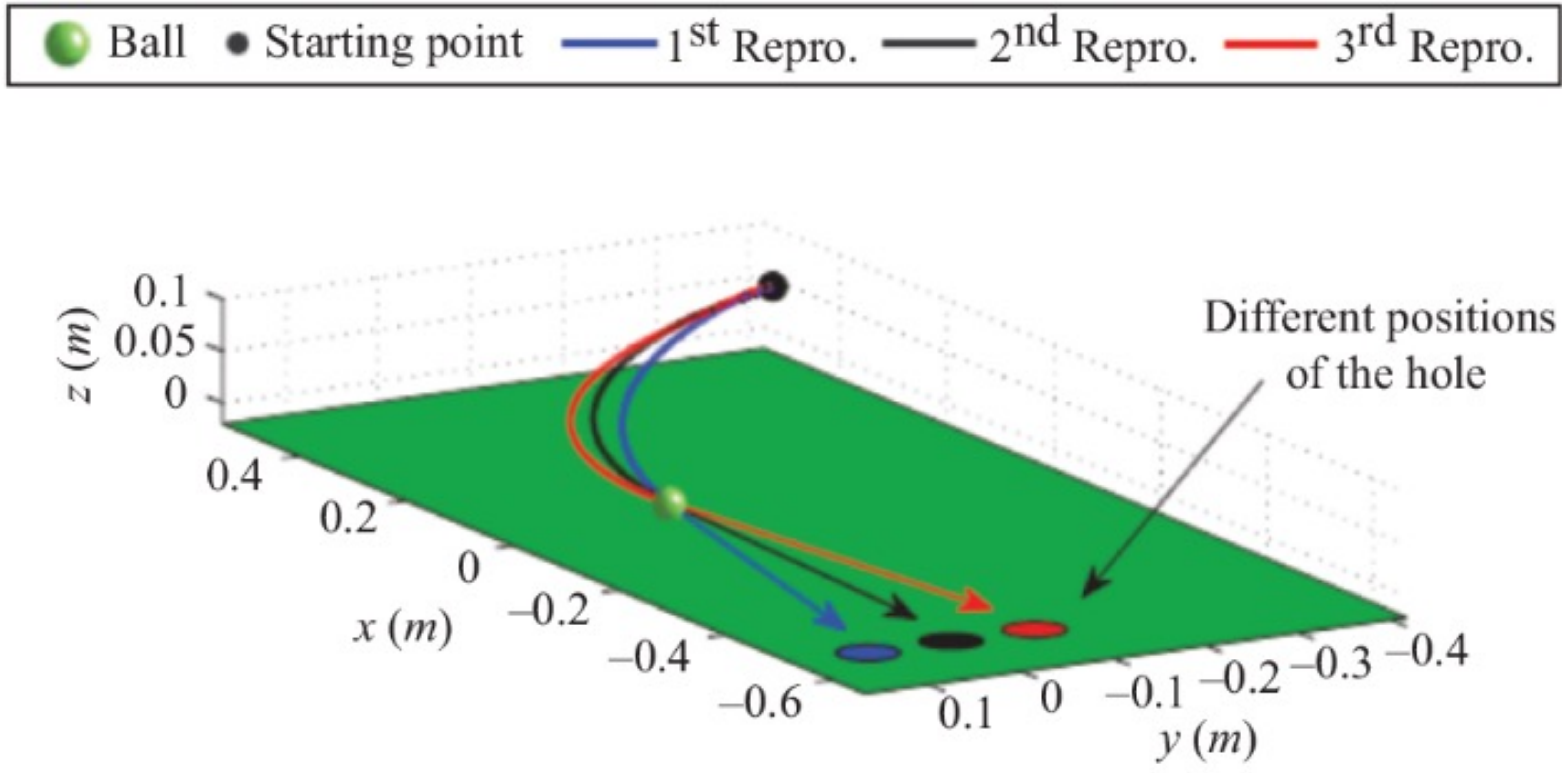
Modelling Hitting Task using Dynamical Systems-Based Control



Representation through DSs of trajectories to sink a golf ball.

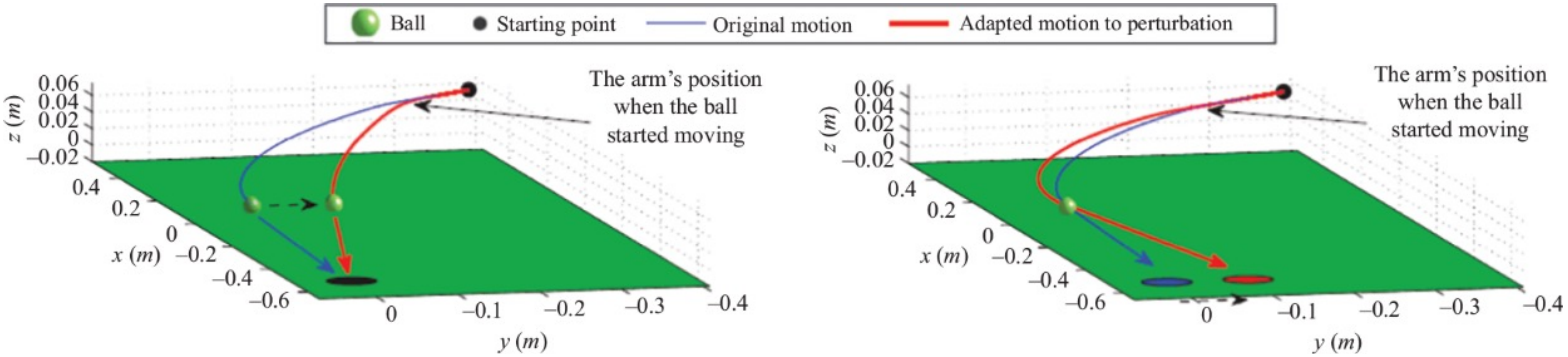
Modelling Hitting Task using Dynamical Systems-Based Control

- Expressing the dynamics from the relative position of the ball to the sink allows one to nicely generalize the orientation toward the ball without further demonstrations

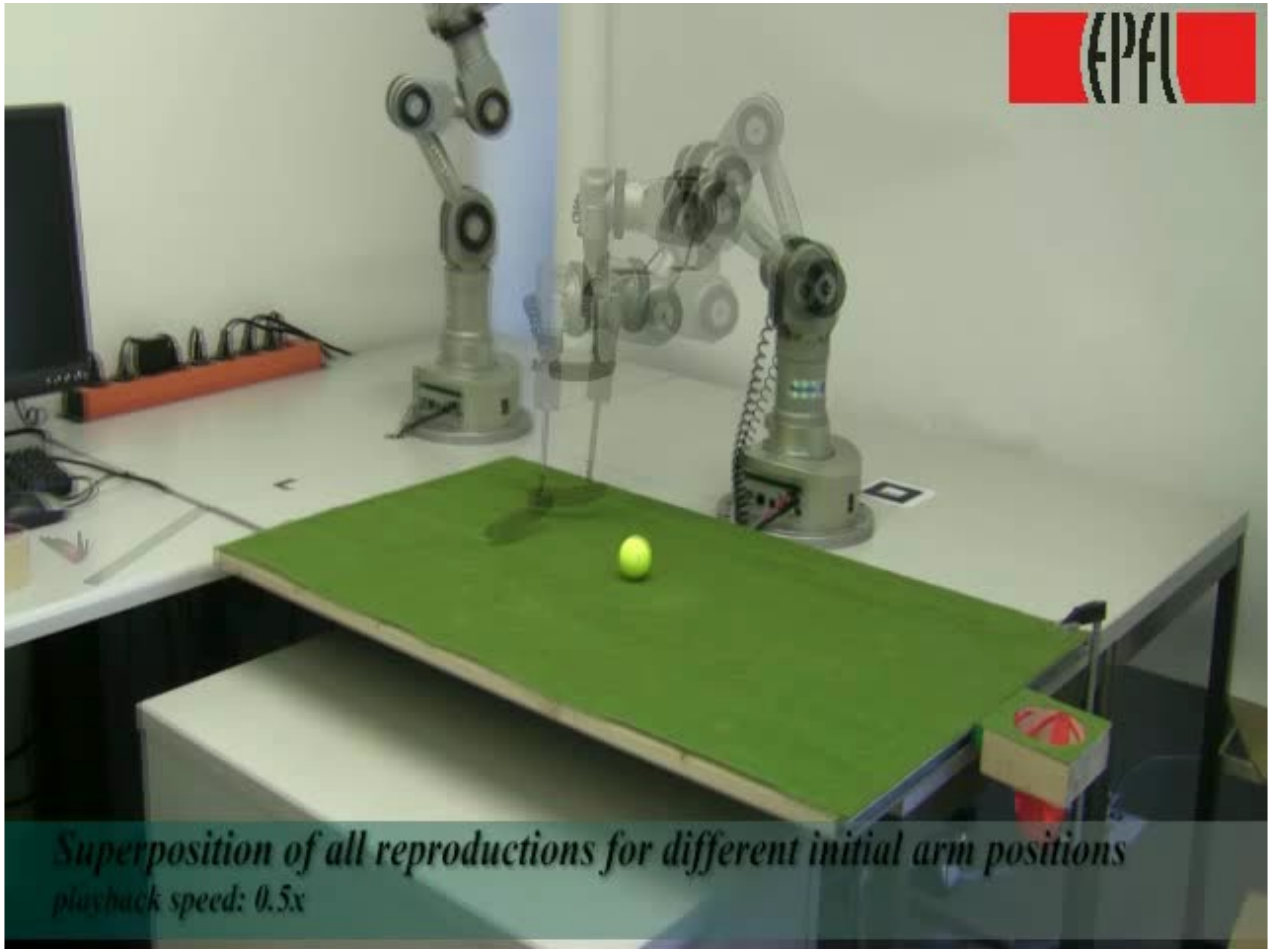


Modelling Hitting Task using Dynamical Systems-Based Control

- The learned DS can adapt at run time to perturbations, such as pushing the robot away from its trajectory (left) and moving the hole (right) by generating a new trajectory that reaches the target correctly

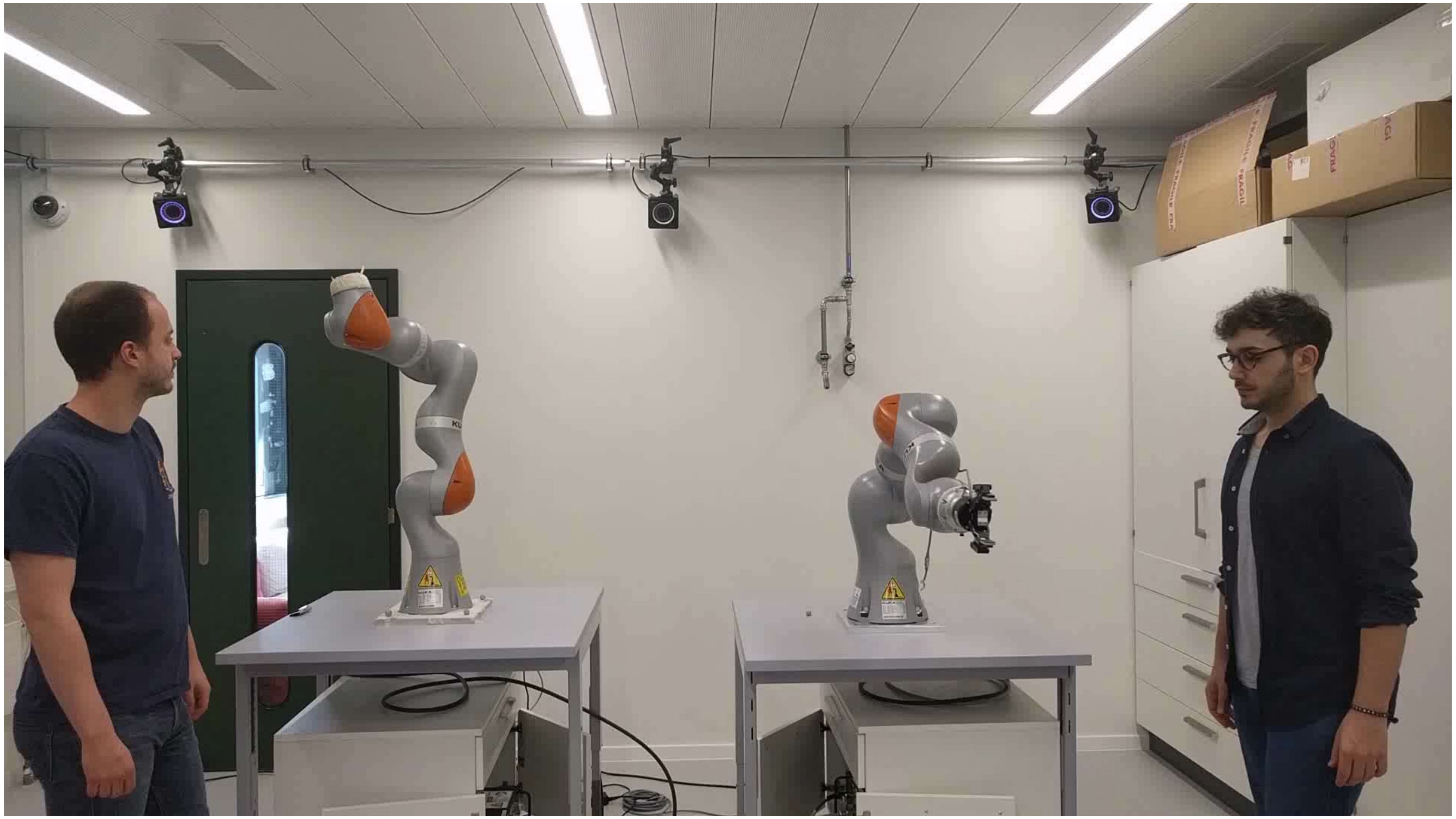


Modelling Hitting Task using Dynamical Systems-Based Control

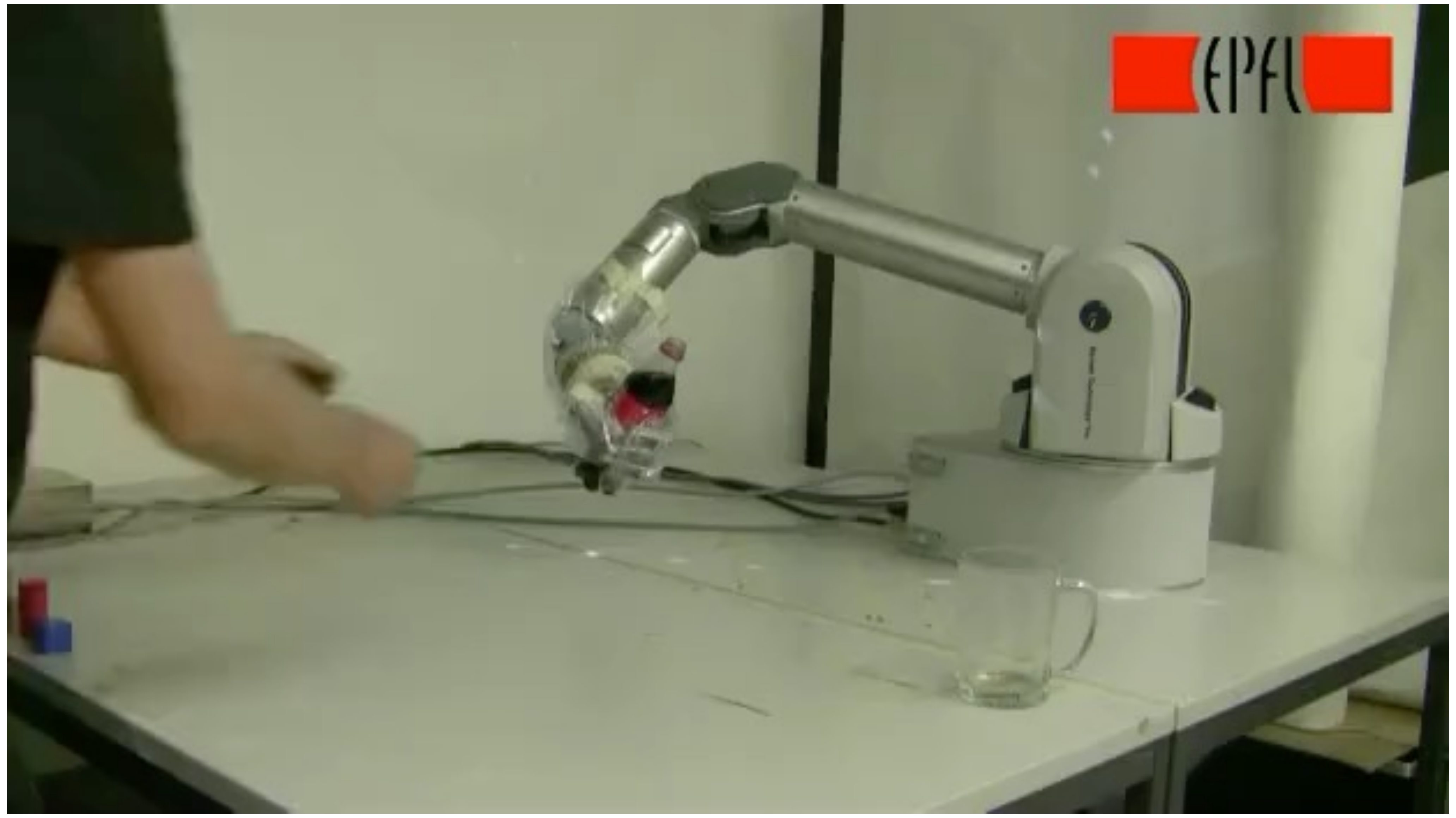


Teaching Compliant Control

Compliant Control



Teaching Compliant Control: What happens when stiffness not considered?



Too stiff: Liquid spills from jerking

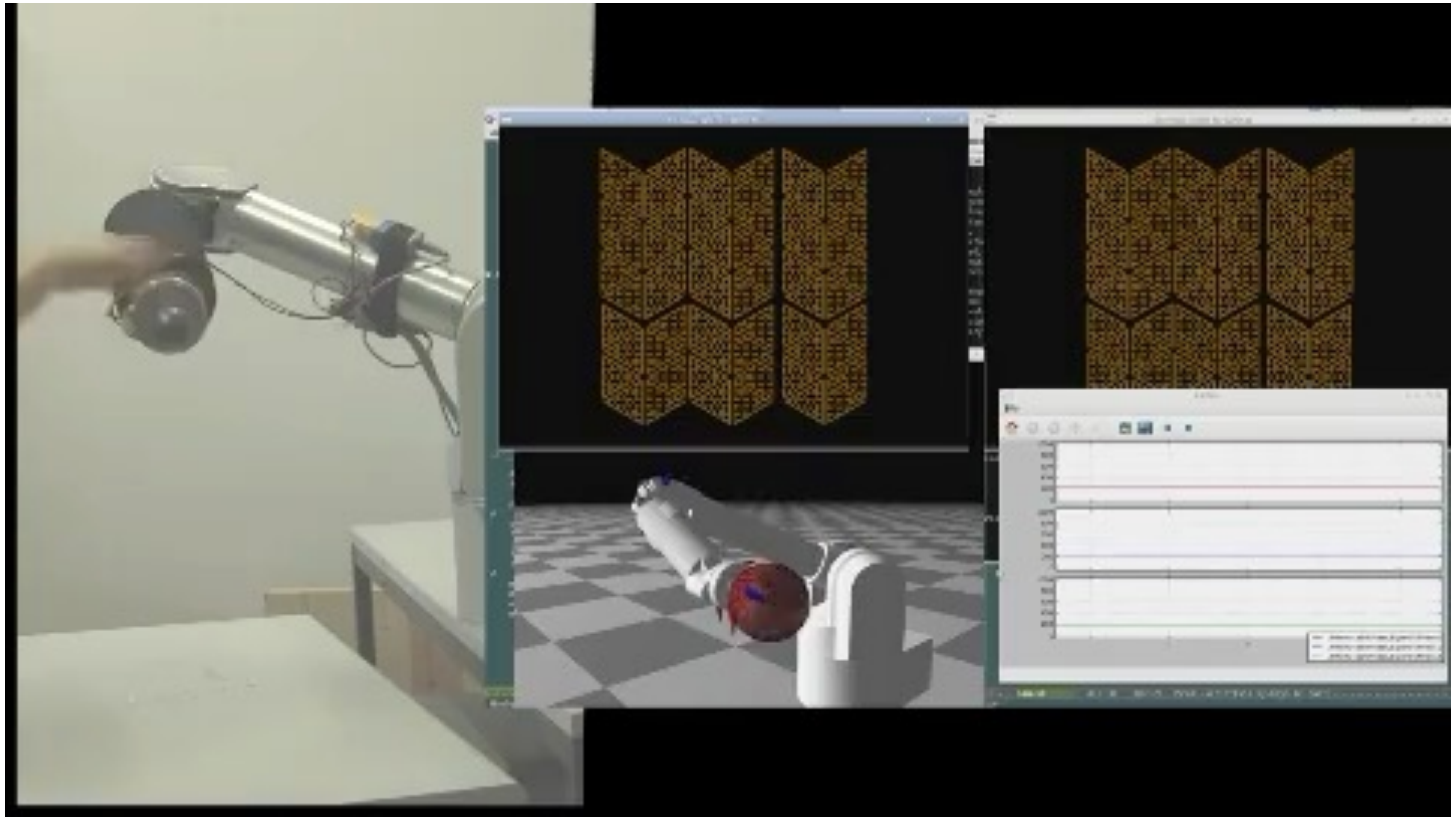
Too compliant: Liquid spills from glass

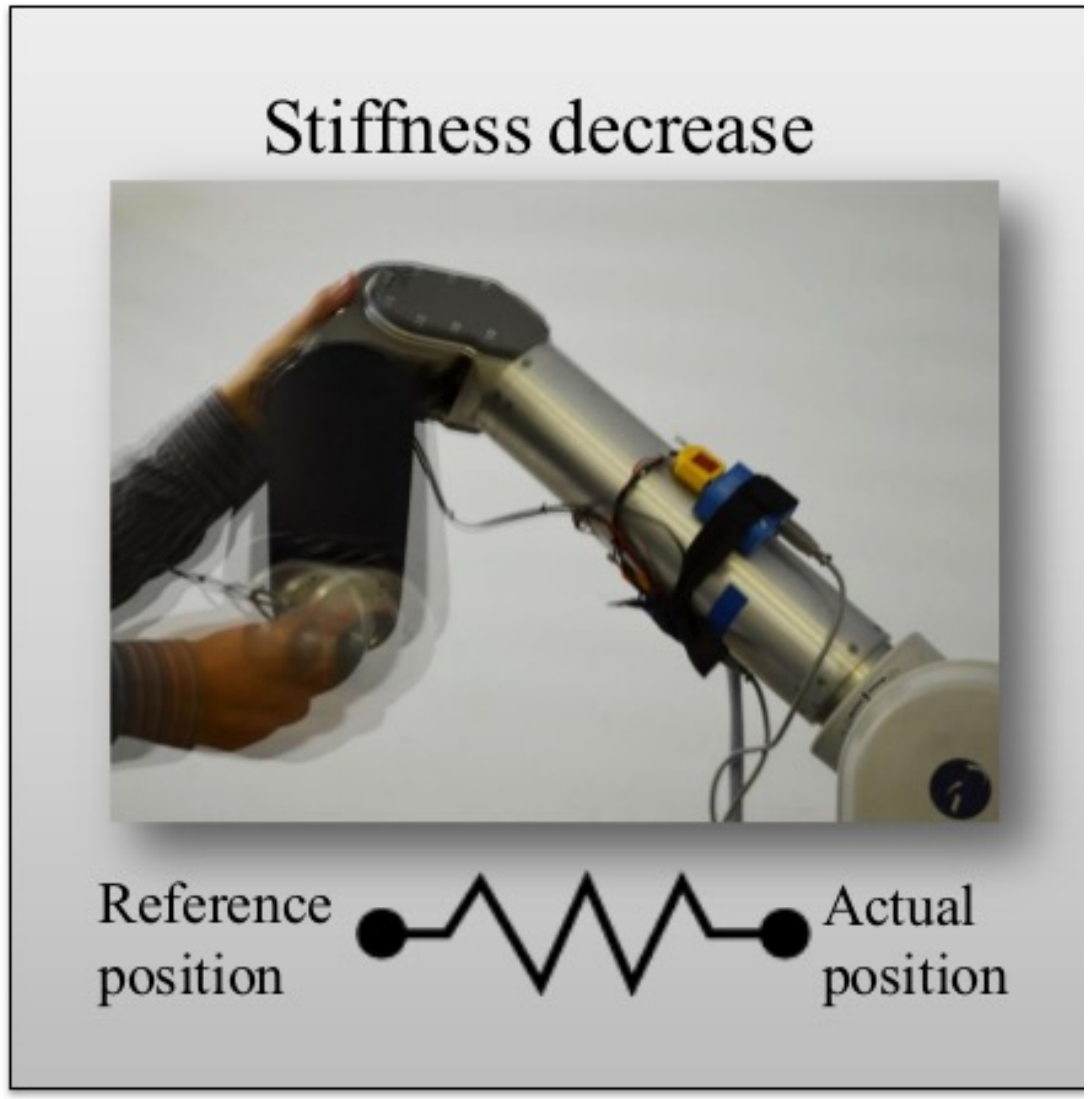
How can we teach robot when to increase and decrease compliance?

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

Teaching Compliant Control: Adding Compliance

Teaching *decrease* in stiffness by wiggling the robot

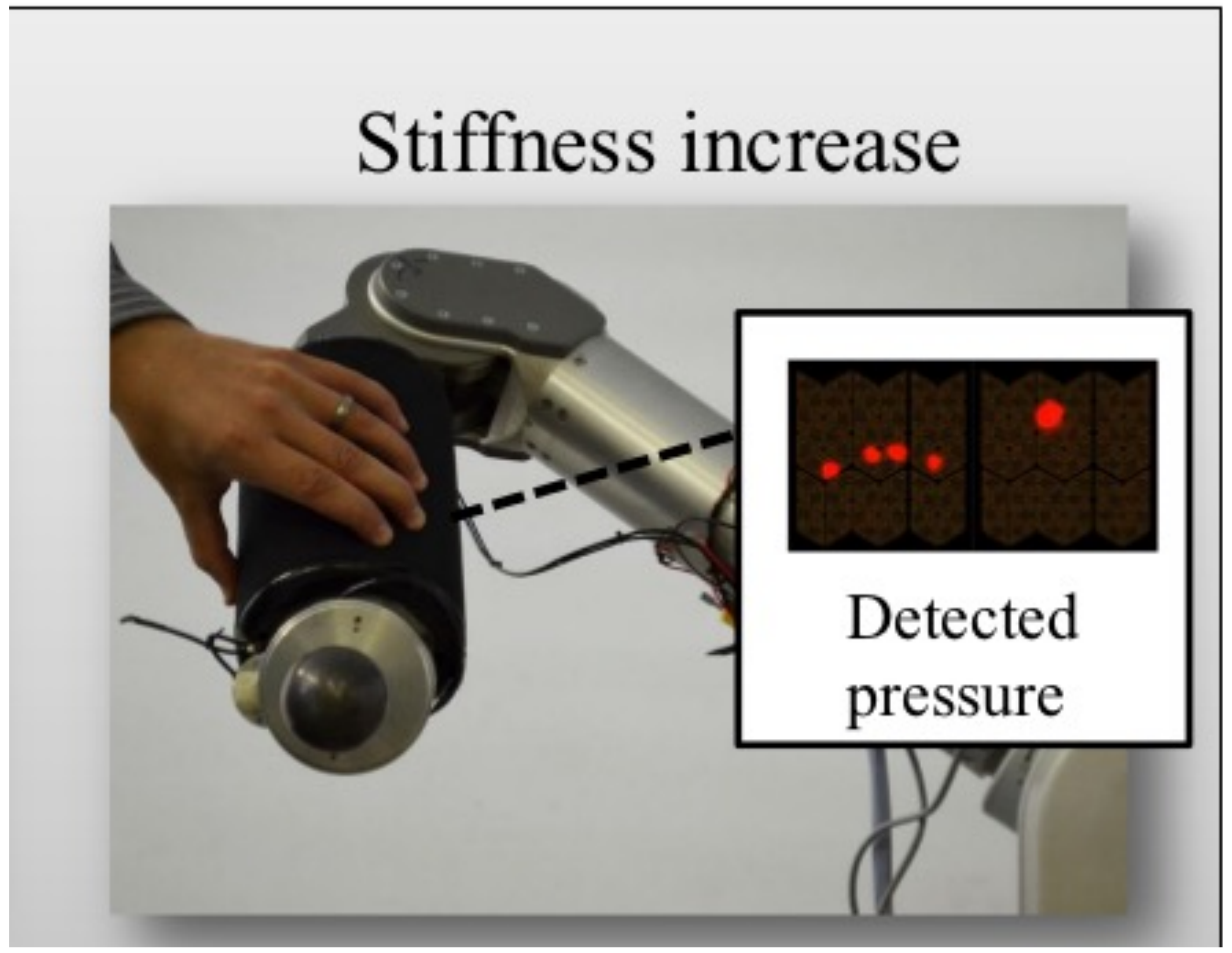




Kronander and Billard, Robot Learning from Demonstration for Compliant Manipulation, ICRA 2012

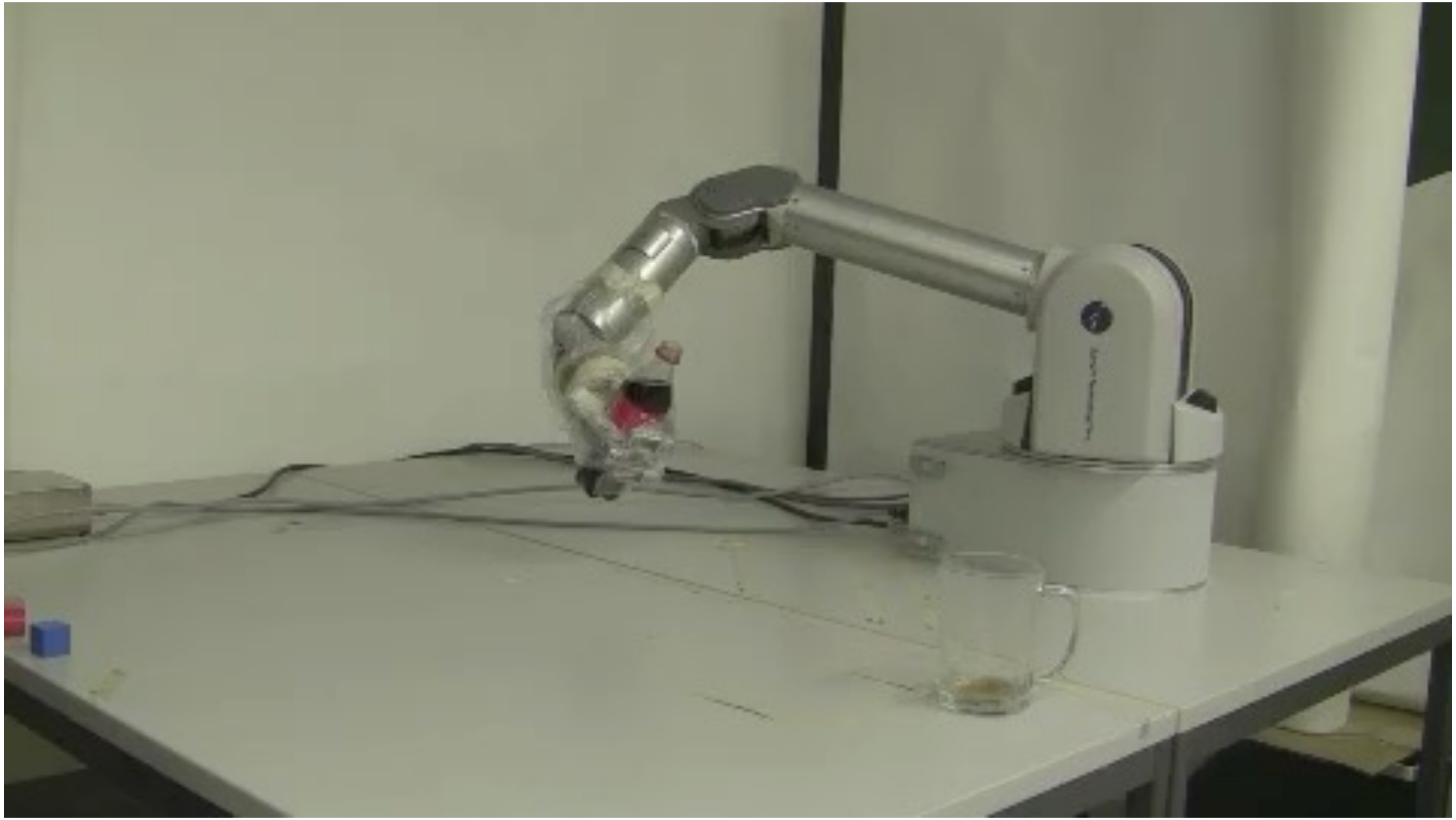
Teaching Compliant Control: Adding Stiffness

Teaching *increase* in stiffness
by exploiting tactile sensing on robot arm



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

Teaching Compliant Control: Final Result



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

Learning from Demonstration: General Considerations

Considerations

Learning human skills through LFD requires the following questions:

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

Answering these questions requires us to first address *the correspondence problem*.

EPFL

Recap

How can we learn controllers from data and how do we get the data?

- How can we can use demonstrations to learn task controllers?
- Different approaches to Learning from Demonstration
- How interface design affects data gathering and demonstration quality
- Examples of LfD
- General Considerations

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

EPFL