

Extensions to control with DS

Coupling DS & Examples of Applications

Multi-attractor DS, Switching across DS, On-line update of DS

Coupled DS

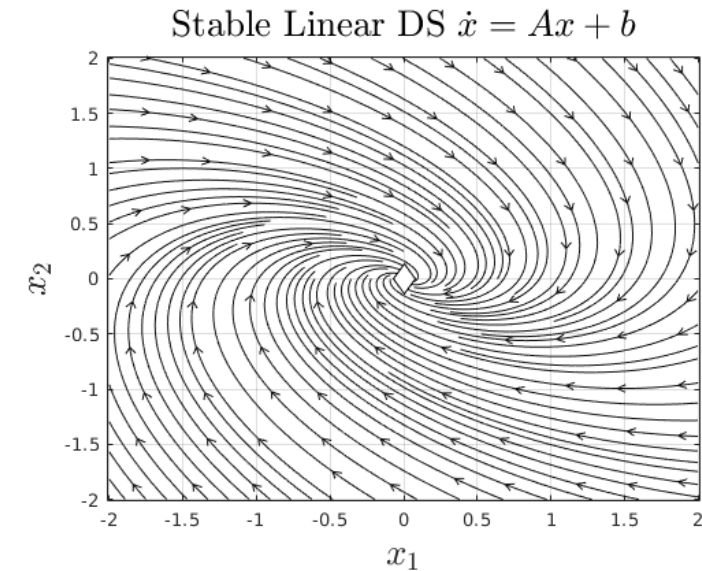
Isolated Dynamics

Until now, we have considered a DS control law in isolation.

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

The state of the system x at time t depends solely on the state of x at the previous time step.

$$x(t) = x(t-1) + f(x)dt$$



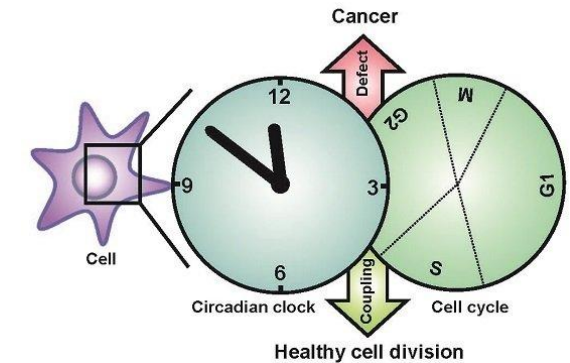
In general, systems do not function in isolation.

Their dynamics is influenced by the dynamics of their environment.

Coupled DS in Nature

In Biology, the circadian clock and the cell cycle are modelled as two periodic processes that are coupled with one another. “In mammalian cells, circadian clocks consist of autonomous feedback loop oscillators ticking with an average period of about 24 h and controlling many downstream cellular processes.”

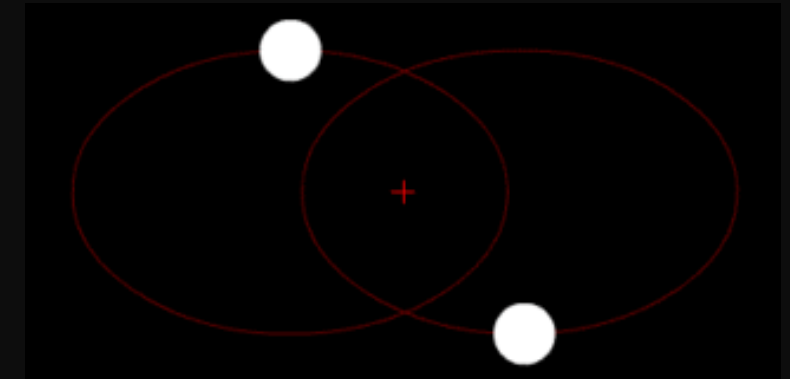
Droin, C., Paquet, E.R. & Naef, F. Low-dimensional dynamics of two coupled biological oscillators. *Nat. Phys.* 15, 1086–1094 (2019).



Shostak, A. Circadian Clock, Cell Division, and Cancer: From Molecules to Organism. *Int. J. Mol. Sci.* 2017, 18, 873.

In Astrophysics, a pair of close-by stars act as a coupled system.

The two stars' dynamics are influenced by their relative masses. In a perfectly balanced system, one would obtain a perfect oscillator, where each star rotates around an ellipse.



Heintz, Wulff Dieter. *Double stars*. Vol. 15. Springer Science & Business Media, 2012.

Coupled DS are often oscillatory in nature, but coupling can also be done to discrete movements, e.g. reach and grasp movements as well as bimanual reaching movements are coupled in amplitude and speed.

Jeannerod M (1984) The timing of natural prehension movements. *J Mot Behav* 16:235–254; Swinnen et al. " Behavioural Brain Research, 2001.

Coupled DS

In DS theory, the concept of “coupling” is used to express dependencies across dynamics.

Consider two variate x and y with dynamics:

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

$$\dot{y} = f_y(y)$$

x and y are coupled if any of the following happens:

$$\begin{cases} \dot{x} = f_x(x, y) \\ \dot{y} = f_y(y) \end{cases} \quad \begin{cases} \dot{x} = f_x(x) \\ \dot{y} = f_y(x, y) \end{cases} \quad \begin{cases} \dot{x} = f_x(x, y) \\ \dot{y} = f_y(x, y) \end{cases}$$

In the case of a linear DS on a multi-dimensional variable:

$$x \in \mathbb{R}^N, \quad \dot{x} = Ax$$

The variables follow an uncoupled dynamics only if A is diagonal.

Coupled DS: Stability

Stability can be inherited through coupling

Consider the system:

$$\begin{aligned}\dot{x} &= f_x(x), & \text{If } \lim_{t \rightarrow \infty} f_x(x) = 0 \text{ then, } \lim_{t \rightarrow \infty} f_y(y, x) = 0 \\ \dot{y} &= f_y(y, x) = A_y yx\end{aligned}$$

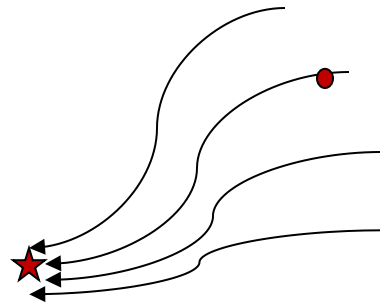
If the two DS are coupled, the stability of the coupled system must be studied.

$$\begin{cases} \dot{x} = f_x(x, y) \\ \dot{y} = f_y(x, y) \end{cases} \quad \begin{cases} \dot{x} = ax + by \\ \dot{y} = cx + dy \end{cases}$$

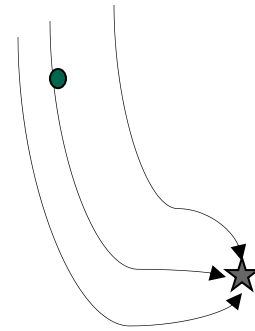
$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \underbrace{\begin{bmatrix} a & b \\ c & d \end{bmatrix}}_A \begin{pmatrix} x \\ y \end{pmatrix}$$

Study the eigenvalues of A.

Uncoupled DS



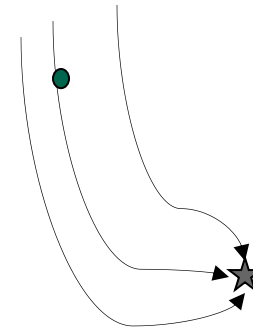
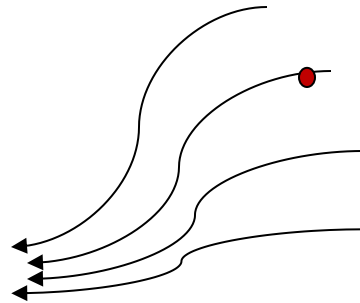
$$\dot{x} = f_x(x, y?)$$



$$\dot{y} = f_y(y, x?)$$

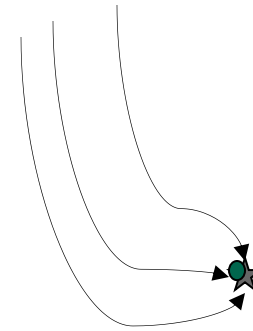
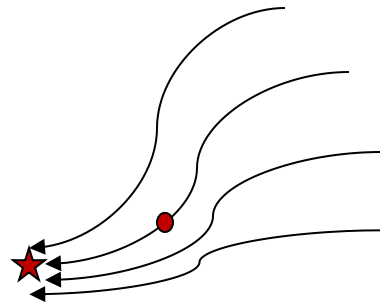
Uncoupled DS

To determine if the systems are coupled, generate a perturbation on one and observe if this affects dynamics of the other system, and conversely



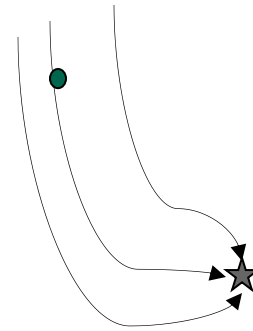
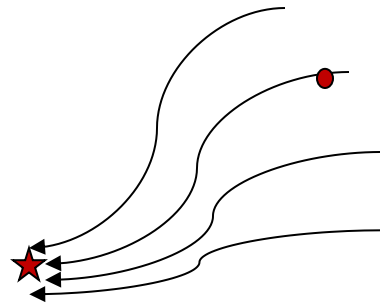
Uncoupled DS

To determine if the systems are coupled, generate a perturbation on one and observe if this affects dynamics of the other system, and conversely

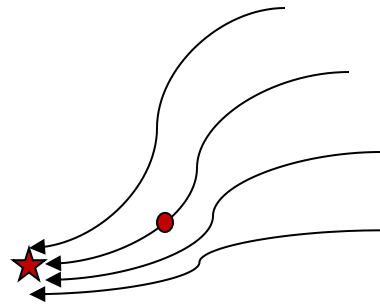


$$\dot{y} = f_y(y)$$

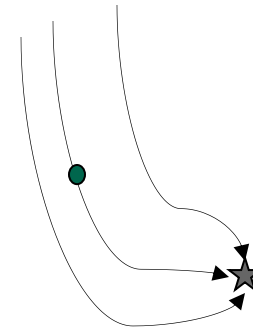
Coupled DS



Coupled DS



$$\dot{x} = f_x(x, y?)$$



$$\dot{y} = f_y(y, x)$$

Repeat the experiment but, this time,
by applying a perturbation on y .

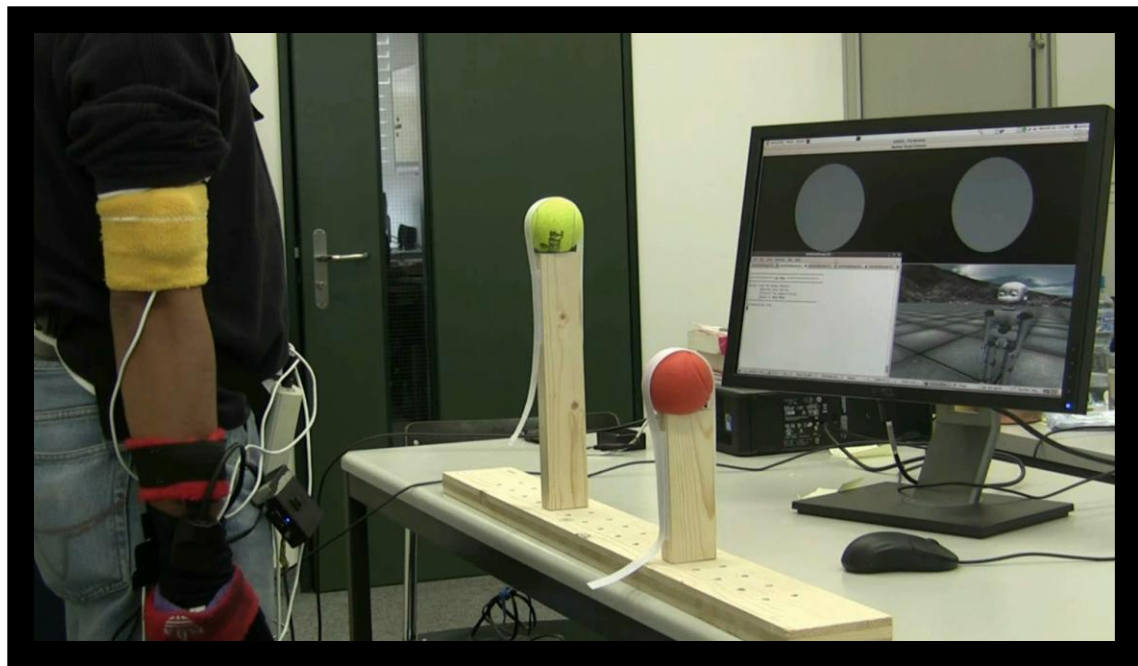
Example

Coupled DS for Hand-Arm Coordination

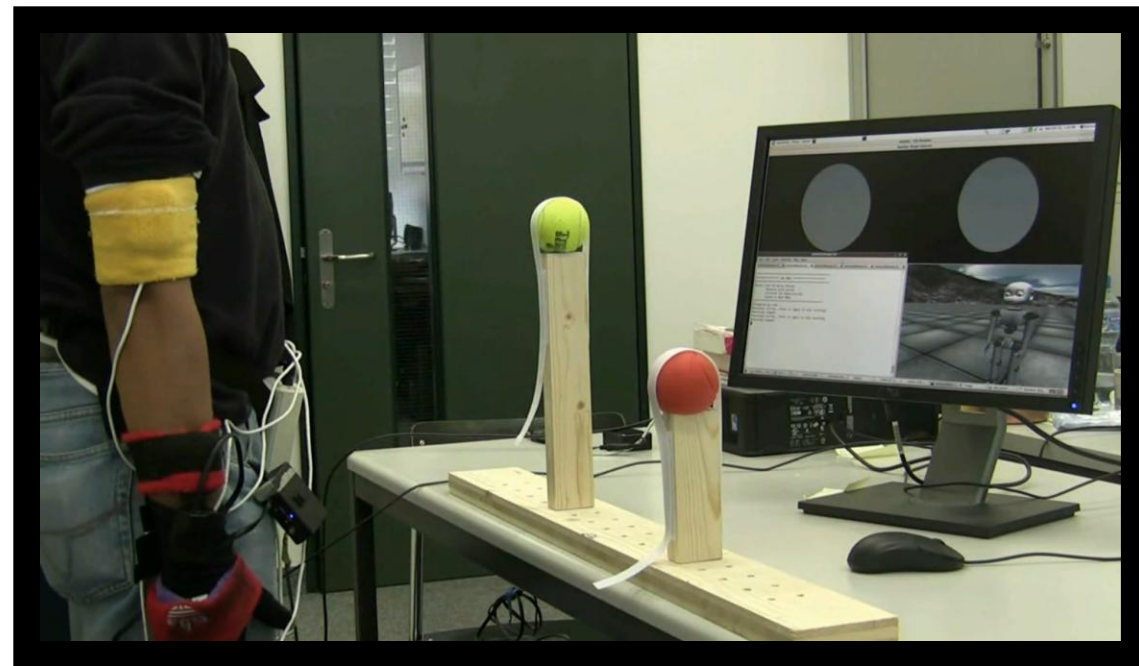
Arm-hand coordination



Hand-Arm Coupling under Disturbances



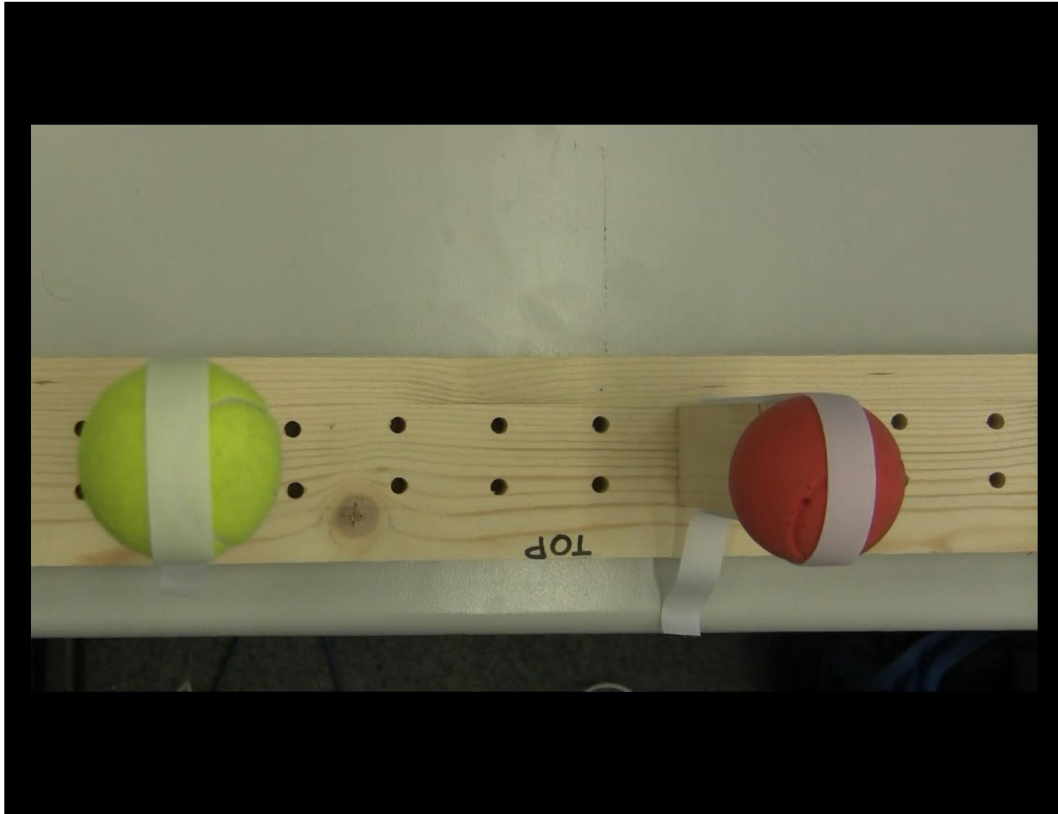
Unperturbed trial



Perturbed trial

Hand-finger coordination: Fingers start opening (preshape) for the final posture at about half of the reaching cycle motion. **Is this coupling preserved during perturbation?**

Hand-Arm Coupling under Disturbances

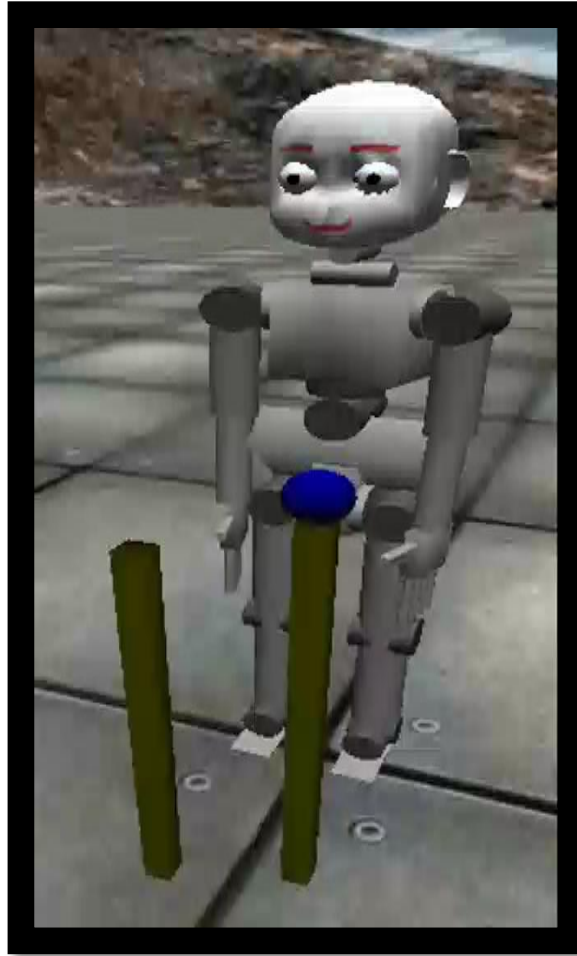


When the target is changed, subjects **re-open completely the fingers** while redirecting the hand to the new target's location.

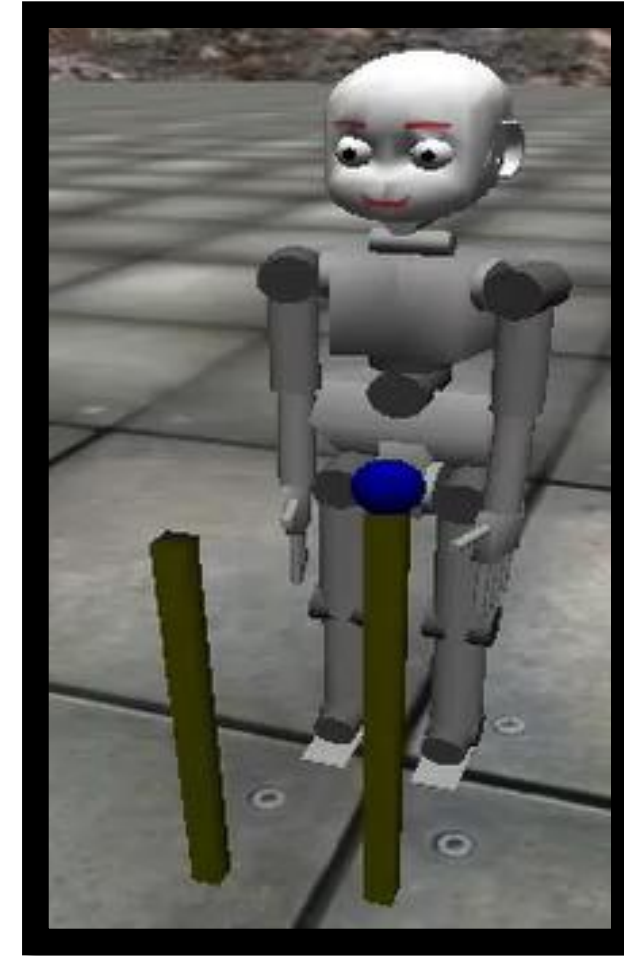
→ This may be advantageous **to adapt to a new configuration of the object that requires a larger hand aperture** with a different fingers' positioning.

→ **Coupling** across fingers and hand offers on-the-fly re-computation of hand motion.

Usefulness for Grasping in Robots



Arm/hand and fingers are not coupled

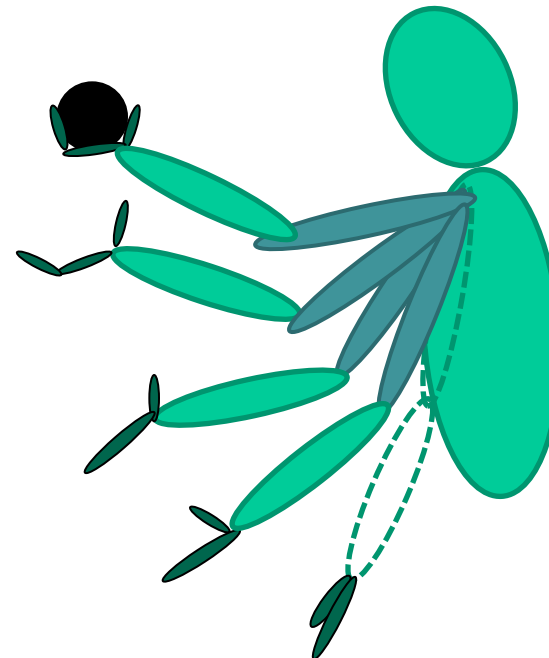


Arm/hand and fingers are explicitly coupled

Modeling Arm-Hand Coupling

$\dot{x} = f_x(x)$ **Controller for hand transport – attractor on object**

$\dot{y} = f_y(y)$ **Controller for finger motion – attractor in joint space**



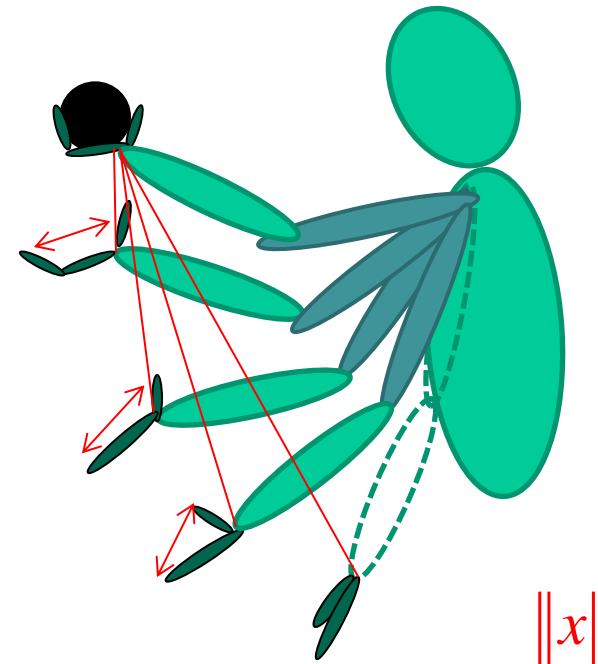
Modeling Arm-Hand Coupling

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

Couple finger-hand dynamics

$$\dot{y} = f_y(y, \|x\|)$$

Dependency on distance to target



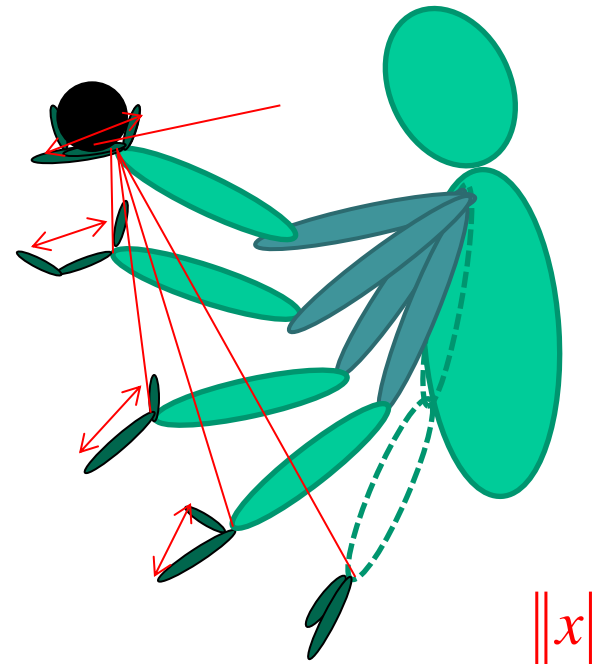
Modeling Arm-Hand Coupling

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

Couple finger-hand dynamics

$$\dot{y} = f_y(y, \|x\|)$$

Dependency on distance to target



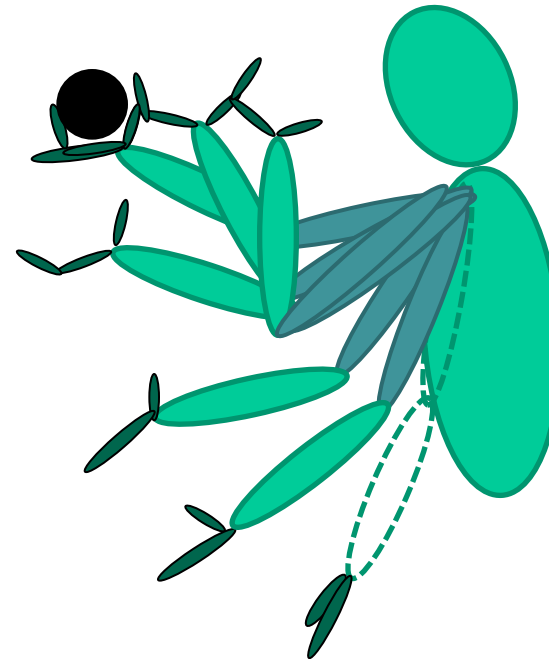
Modeling Arm-Hand Coupling

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

Couple finger-hand dynamics

$$\dot{y} = f_y(y, \|x\|)$$

Dependency on distance to target



Learning the coupling

Couple finger-hand dynamics

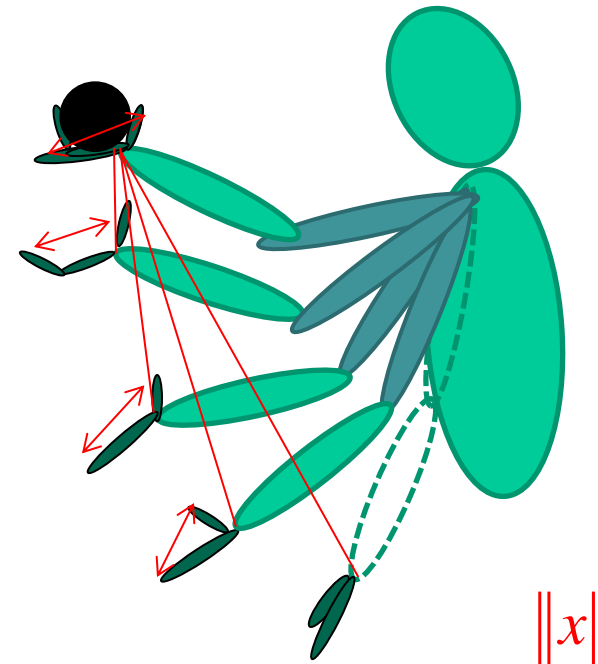
$$\dot{y} = f_y(y, \|x\|)$$

Dependency on distance to target

Learn $p(y, \|x\|)$ dependency on distance to target from human demonstrations

At run time, compute expected finger aperture:

$$\hat{y} = E\{p(y | \|x\|)\}$$



Learning the coupling

At run time, compute expected finger aperture:

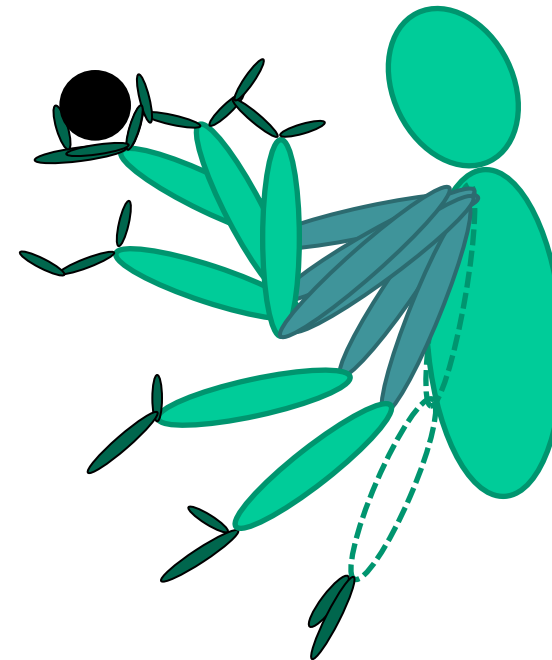
$$\hat{y} = E \left\{ p(y \mid \|x\|) \right\}$$

Drive finger motion:

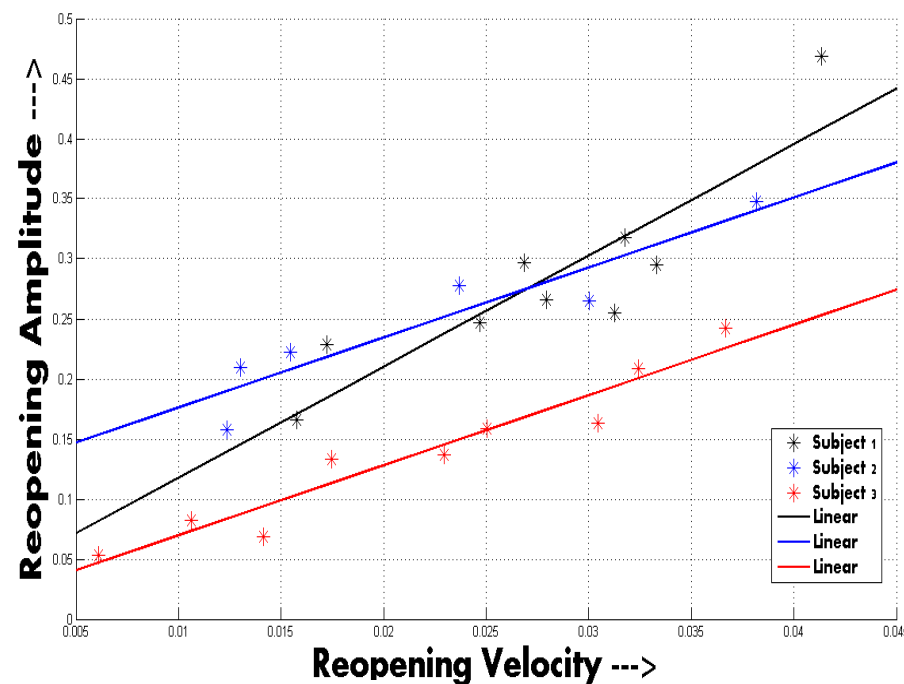
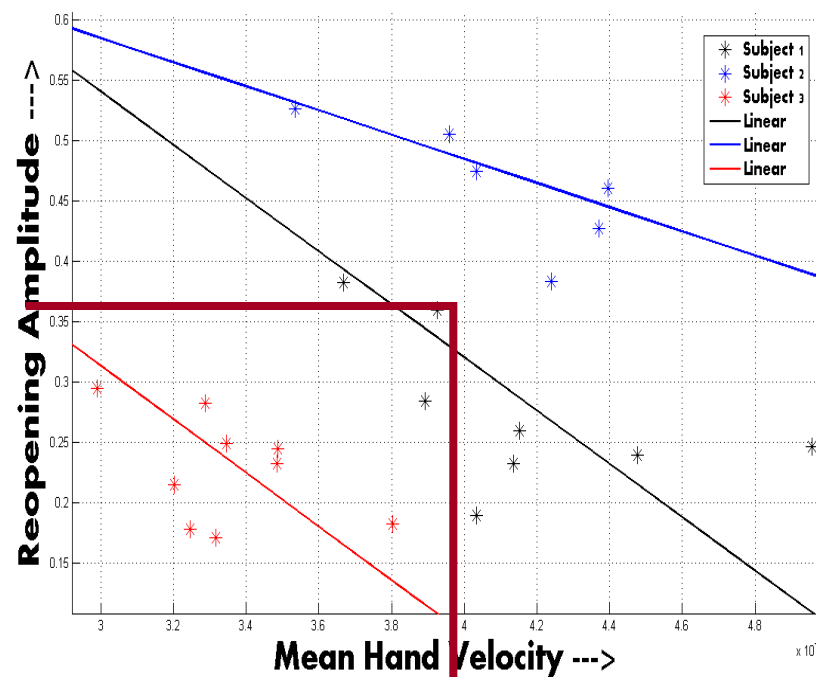
$$\dot{y} = f_y(y - \beta \hat{y}), \quad y_{t+1} = y_t + \alpha \dot{y}_t$$

Amplitude and speed of finger reopening

Learn α and β from human demonstrations

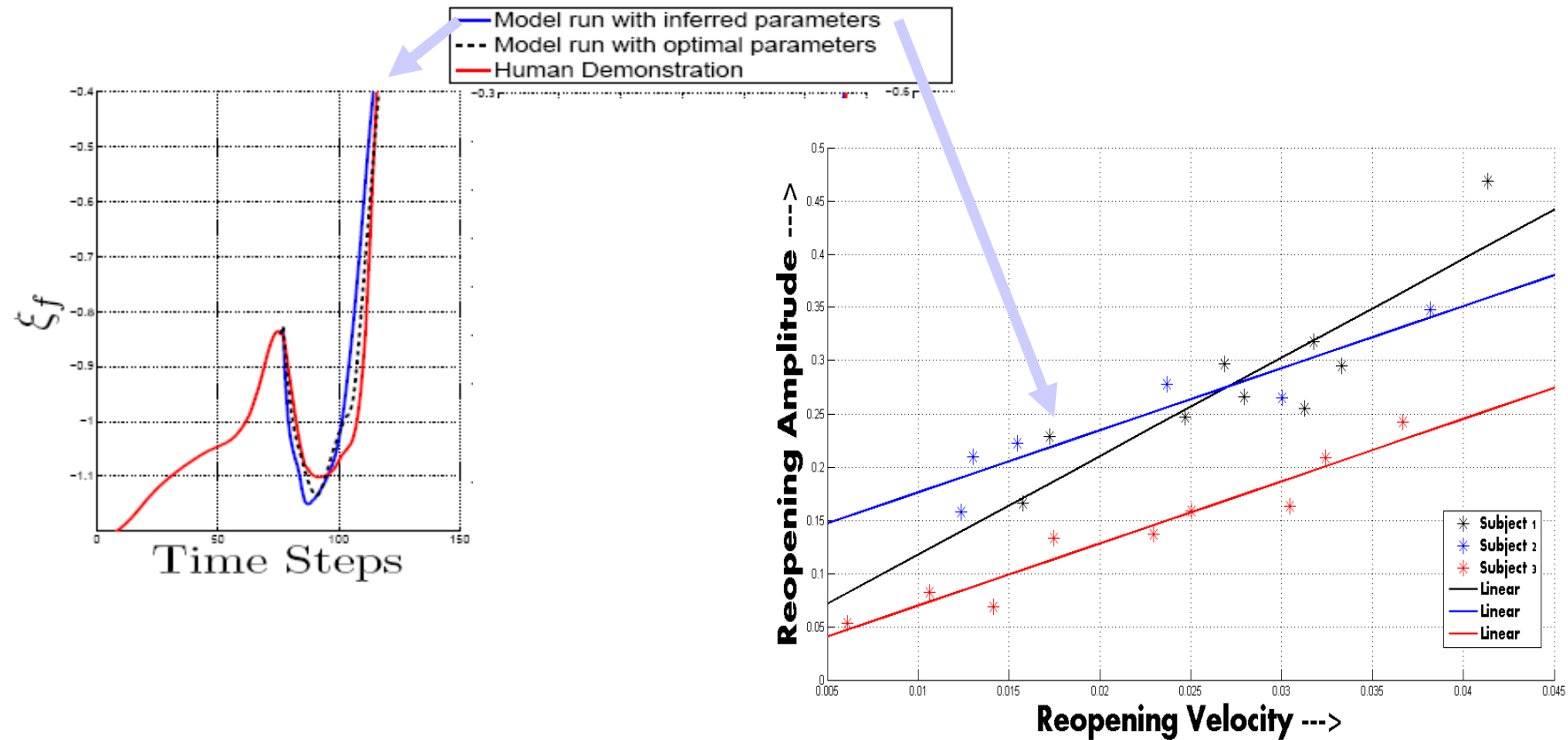


- α Re-opening velocity can be inferred from measuring mean hand velocity prior to perturbation
- β Re-opening amplitude can be inferred from reopening velocity

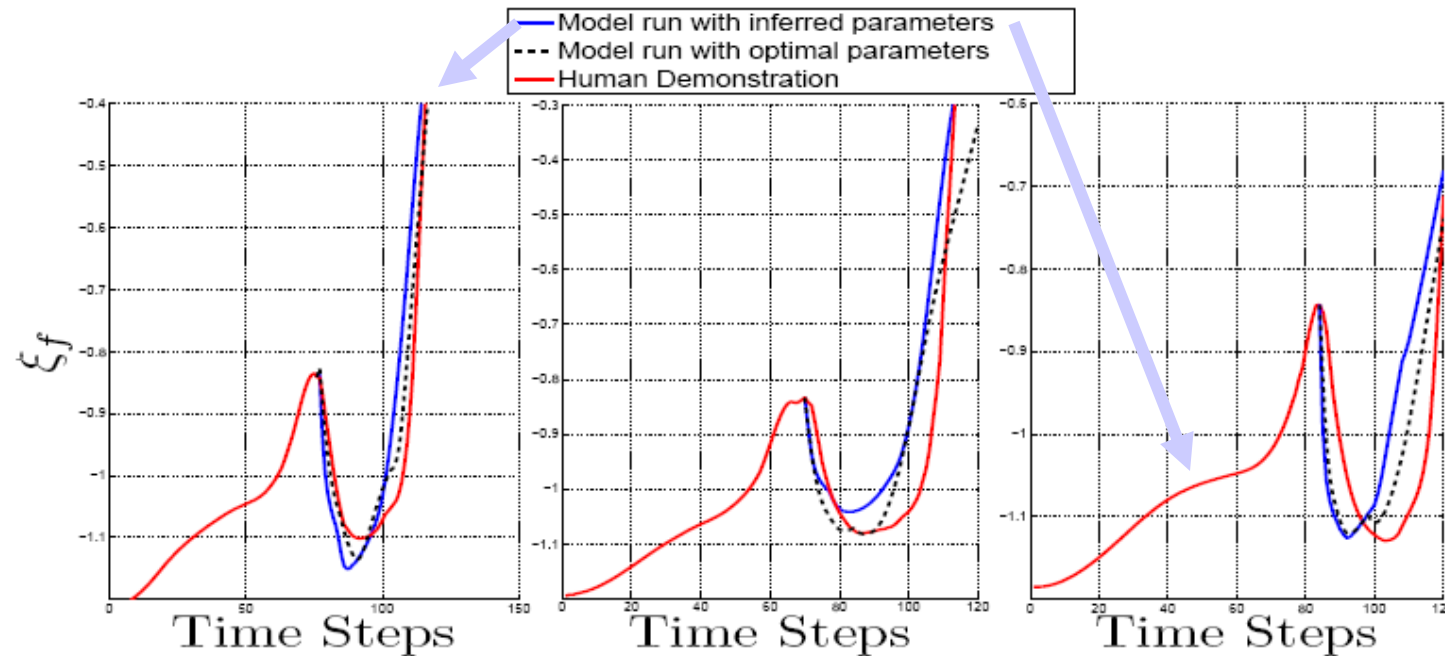


$$\dot{y} = f_y(y - \beta \hat{y}), \quad y_{t+1} = y_t + \alpha \dot{y}_t$$

α parameter inferred from human data

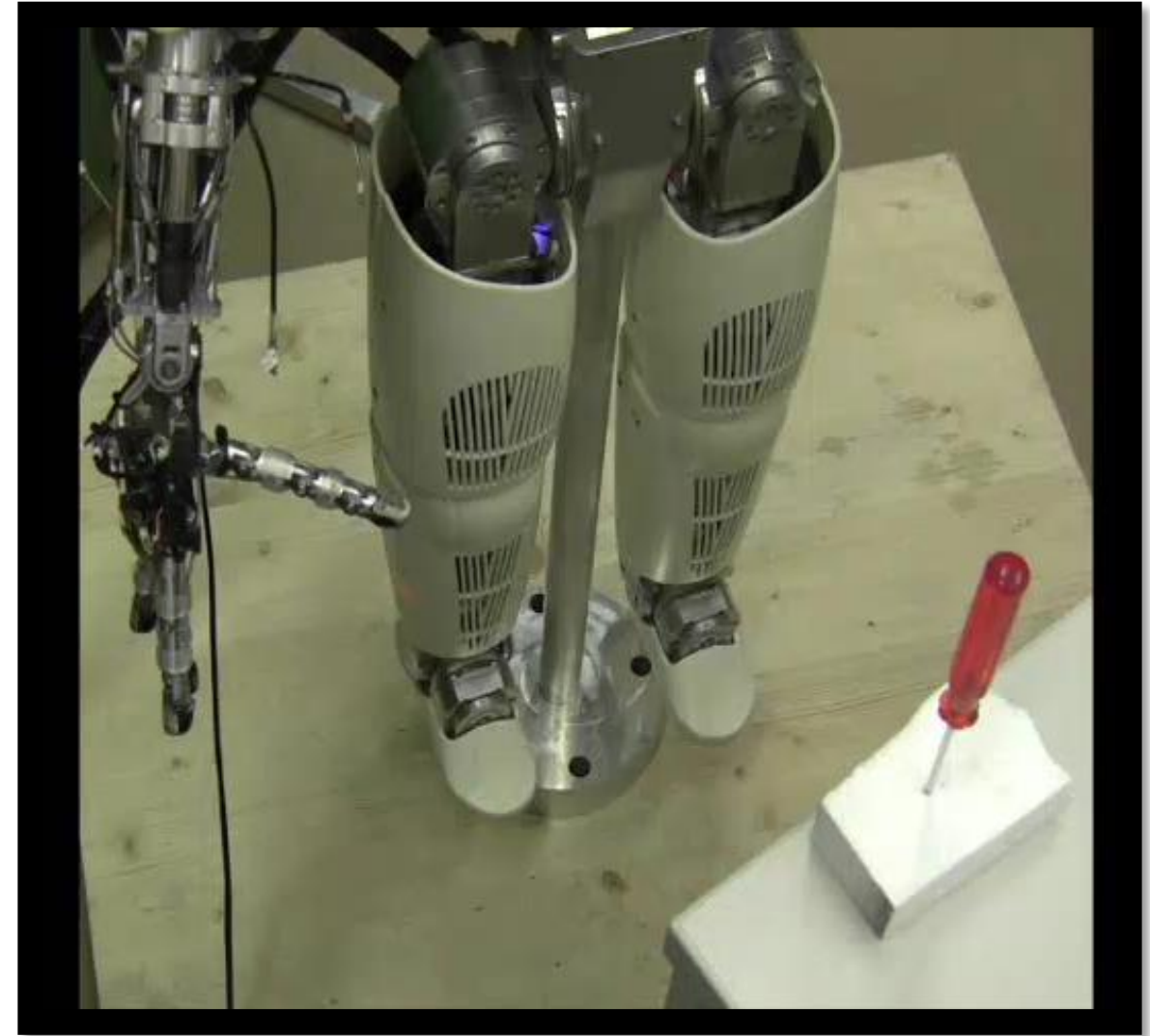
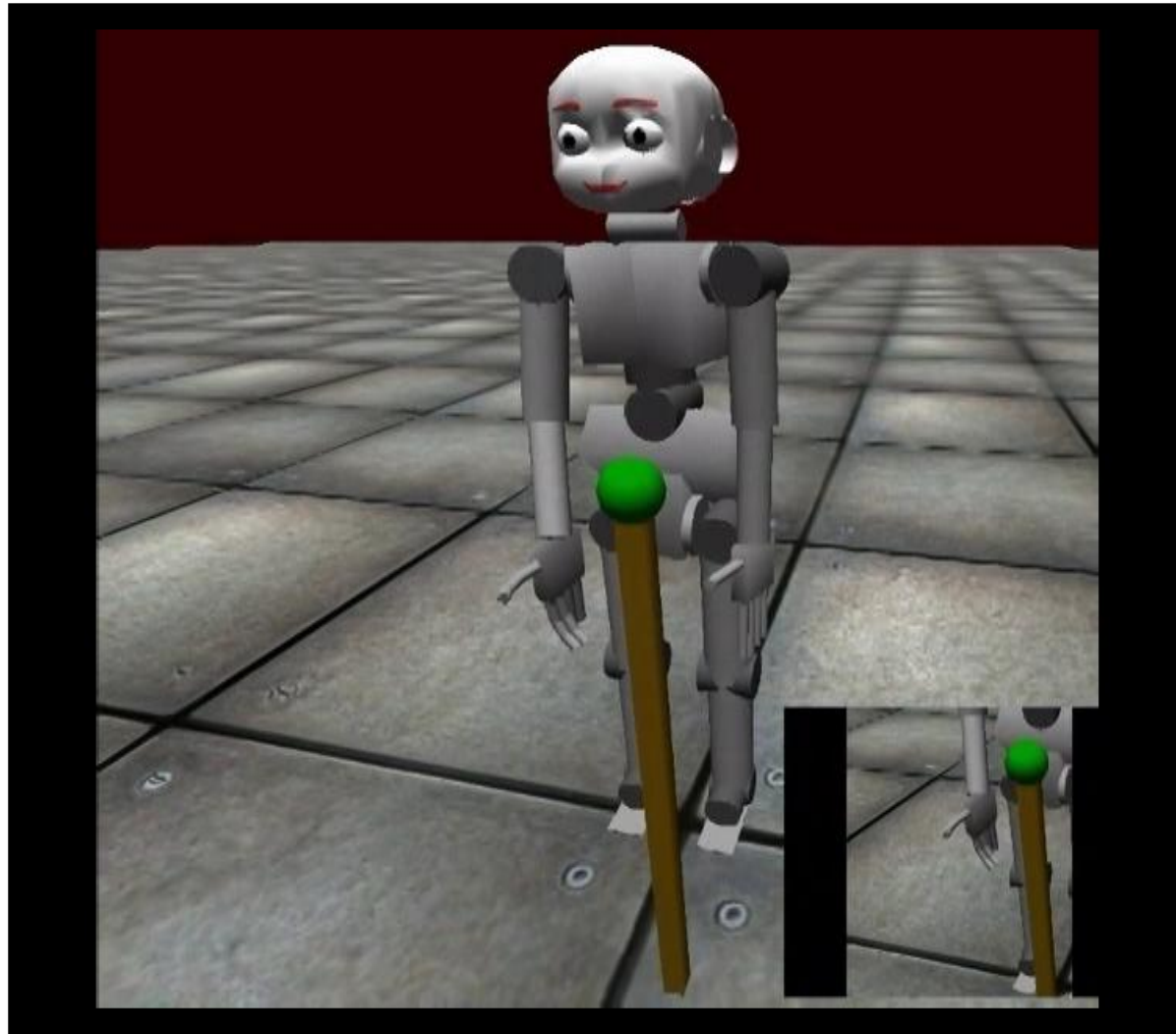


α parameter inferred from human data

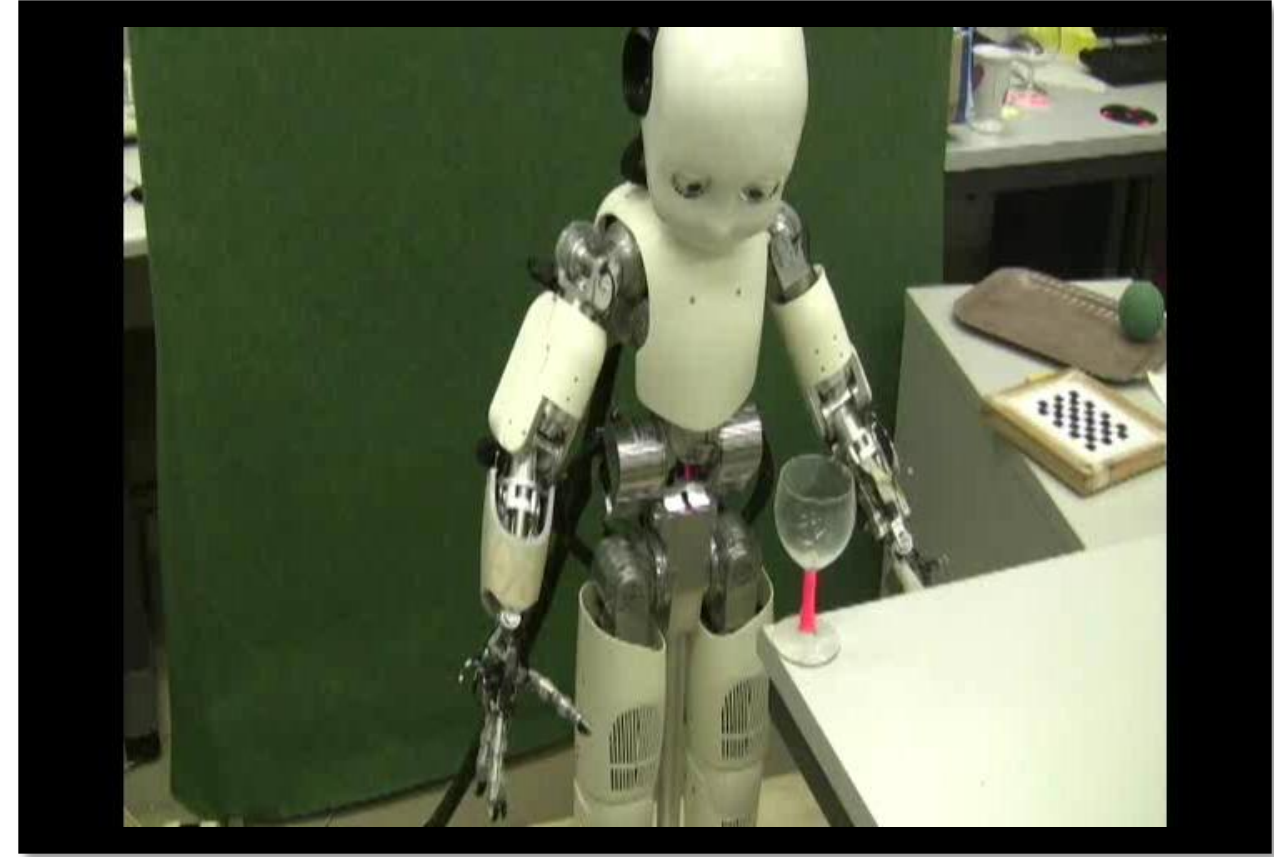
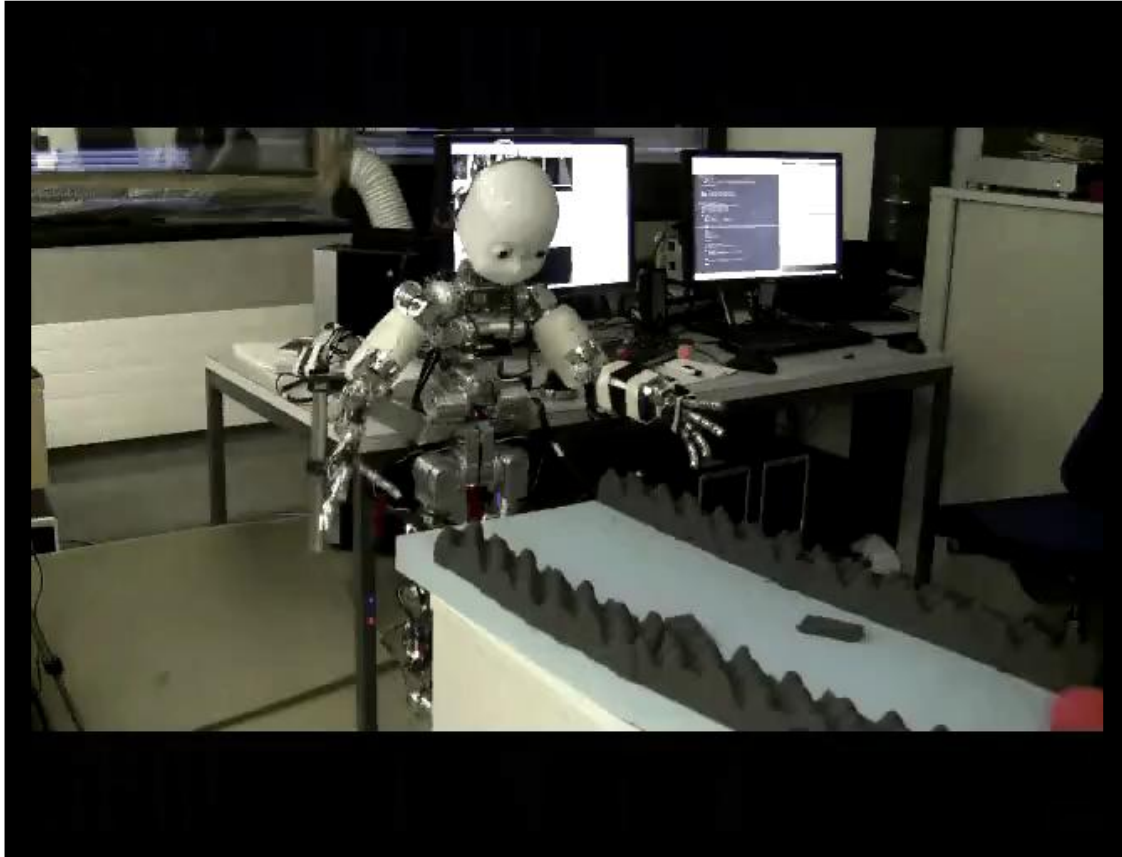


3 trials for the same subject

The Coupled DS model gives both a good qualitative and quantitative assessment of human motion.



Adaptation from pinch to power grasp
(train two separate Coupled DS for each grasp type)



Adaptation under visual or tactile disturbances

Example
Coupled DS for Bimanual Coordination

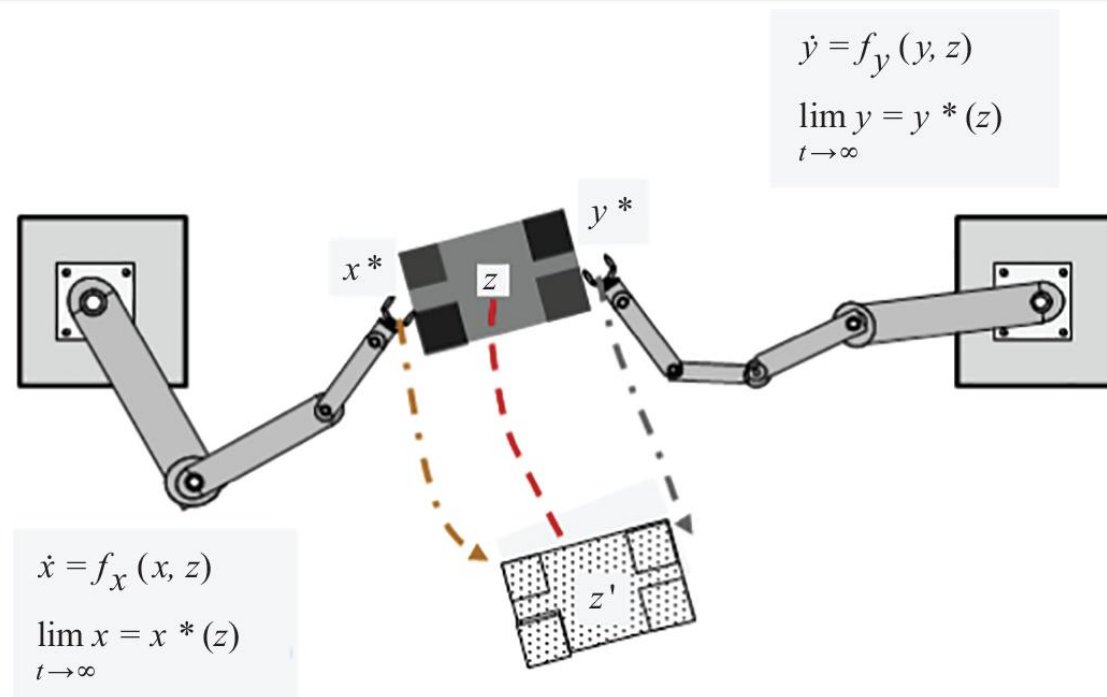
Coupling through External Variable

DS-s can be coupled through an external variable.

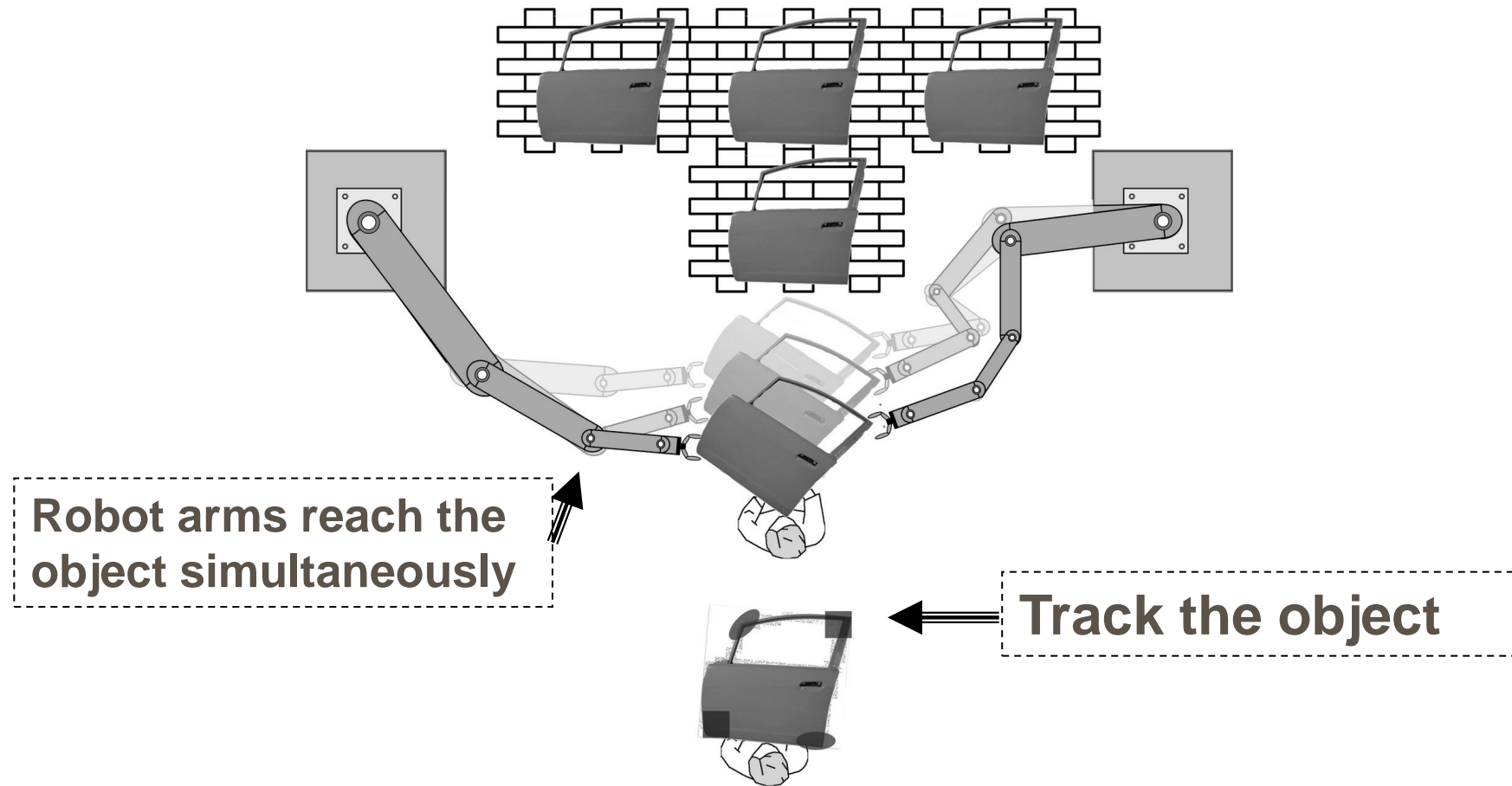
Let z be a virtual variable. x and y can be coupled through z :

$$\dot{x} = f_x(x, z)$$

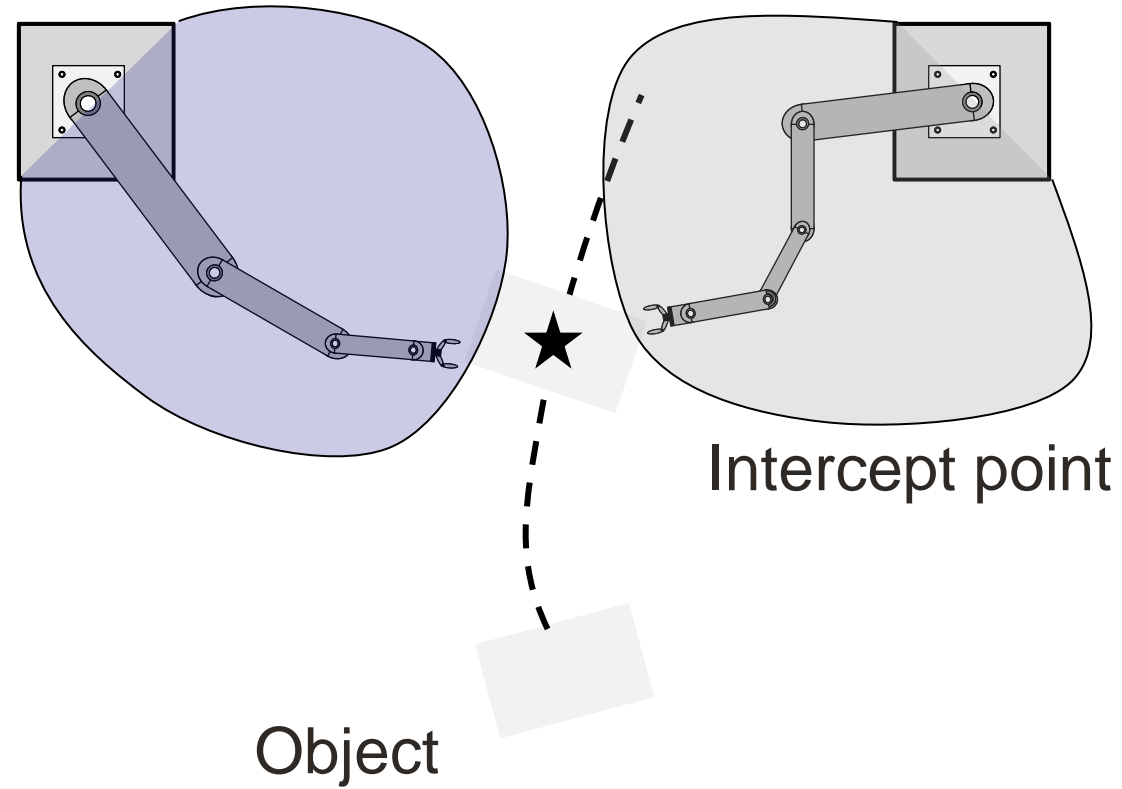
$$\dot{y} = f_y(y, z)$$

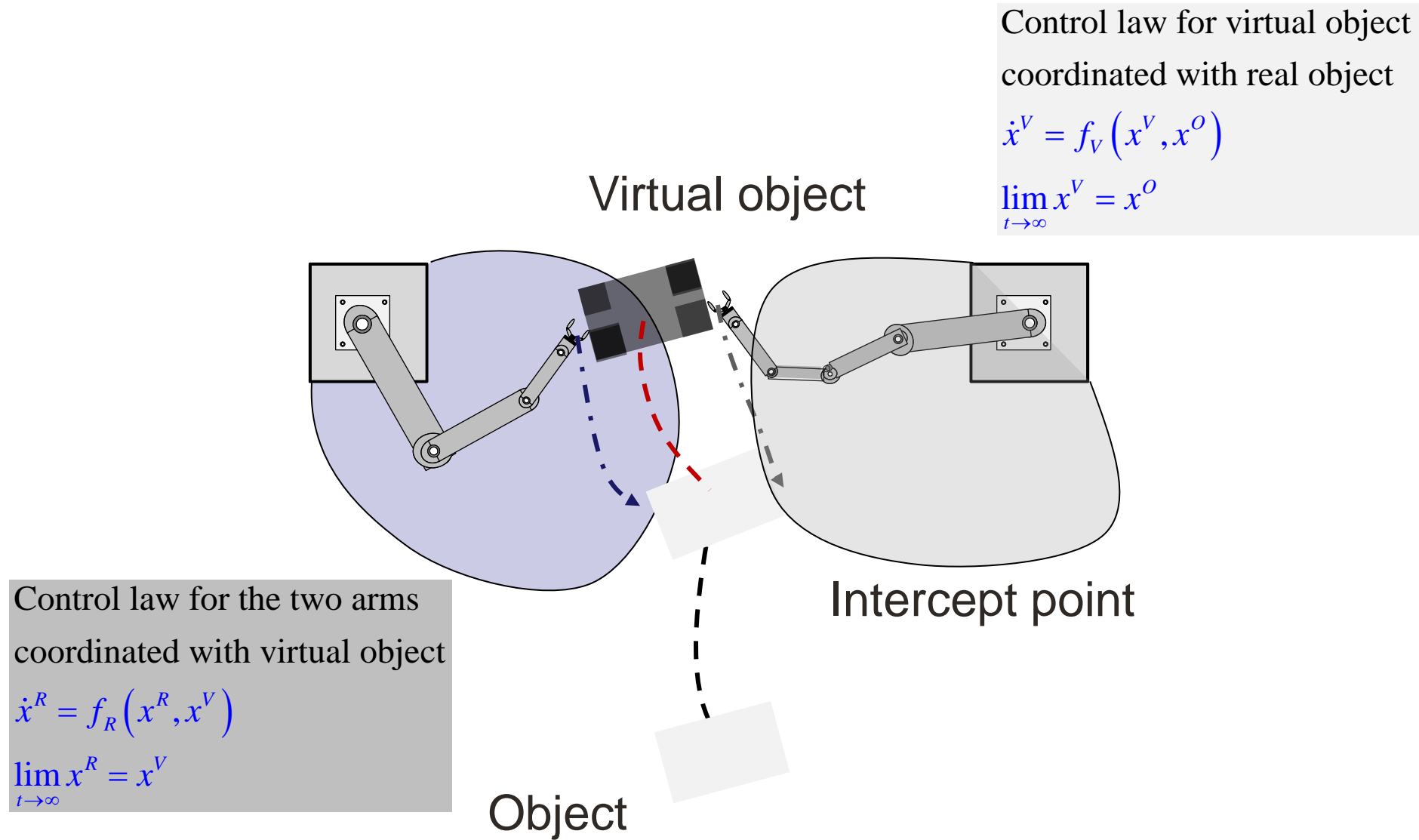






Robots' workspaces

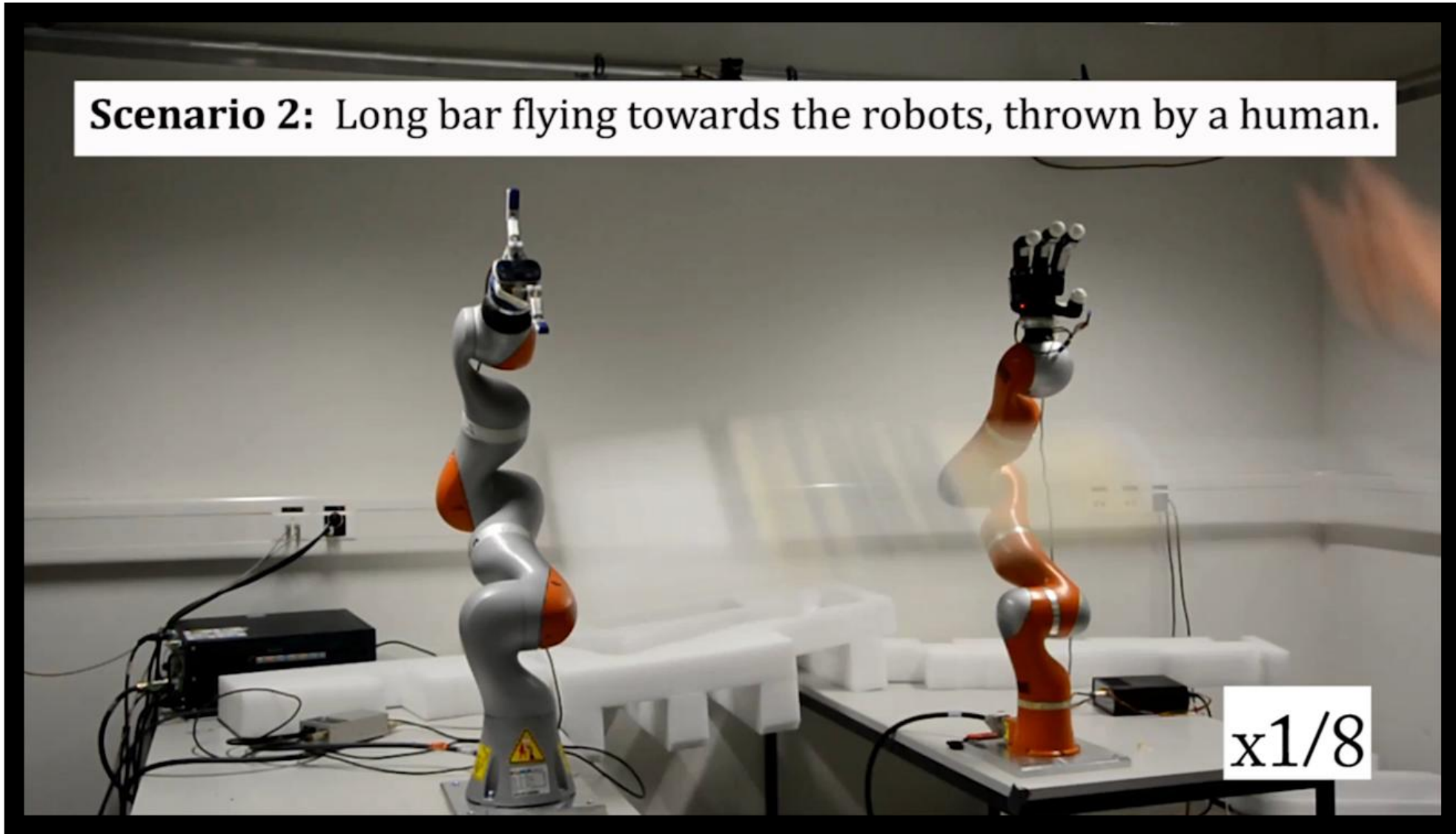






KUKA AWARD FINALIST 2017

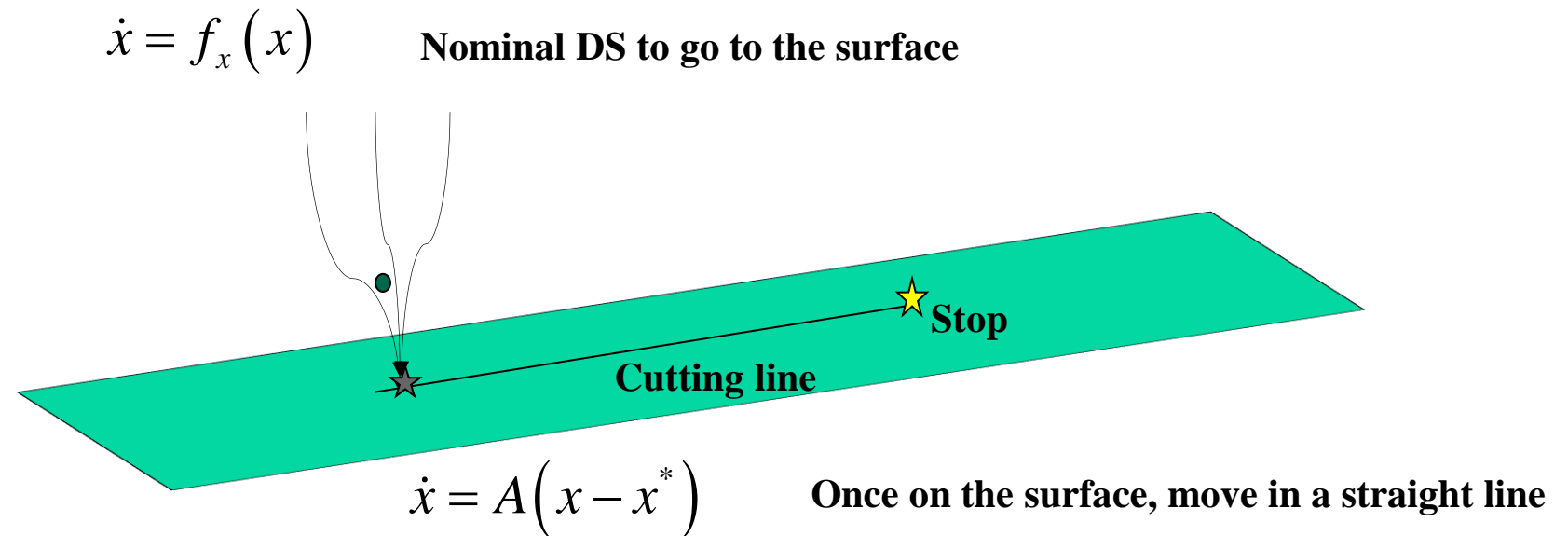
Scenario 2: Long bar flying towards the robots, thrown by a human.



S.S. Mirrazavi Saliehian, N. Figueroa and A. Billard. RSS 2016,. Best Student Paper Award

Example
Coupled DS for Cutting Tissue

Cutting a tissue



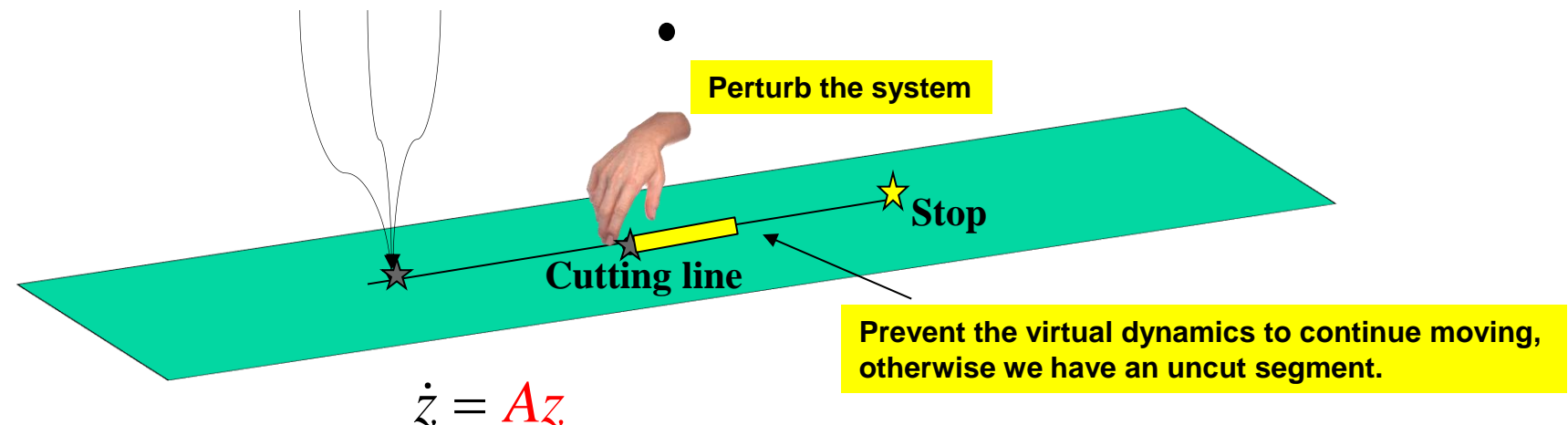
Coupling with Virtual Dynamics

Double Coupling across two DS-s.

$$\begin{aligned}\dot{x} &= f_x(x, z) \\ \dot{z} &= f_z(z)\end{aligned}$$

To make sure we move towards a single line, model it with a virtual system.

$\dot{x} = f_x(x, x^*(z)) = A(x - z)$: Attractor is on the location of the virtual system



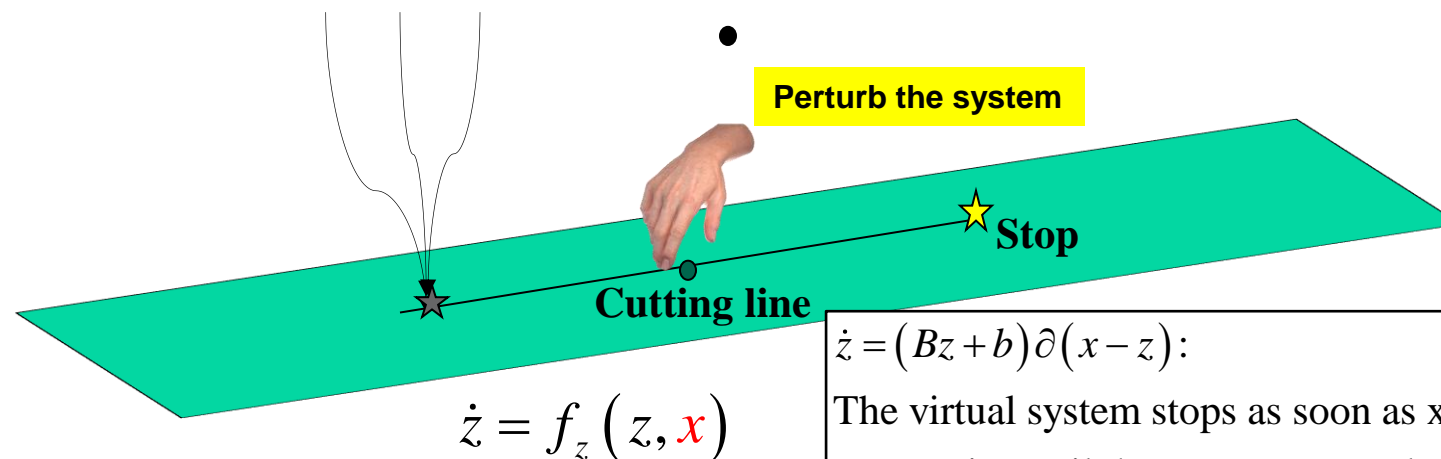
Coupling with Virtual Dynamics

Double Coupling across two DS-s.

$$\dot{x} = f_x(x, z)$$

$$\dot{z} = f_z(z, x)$$

$\dot{x} = f_x(x, x^*(z)) = A(x - z)$: Attractor is on the location of the virtual system

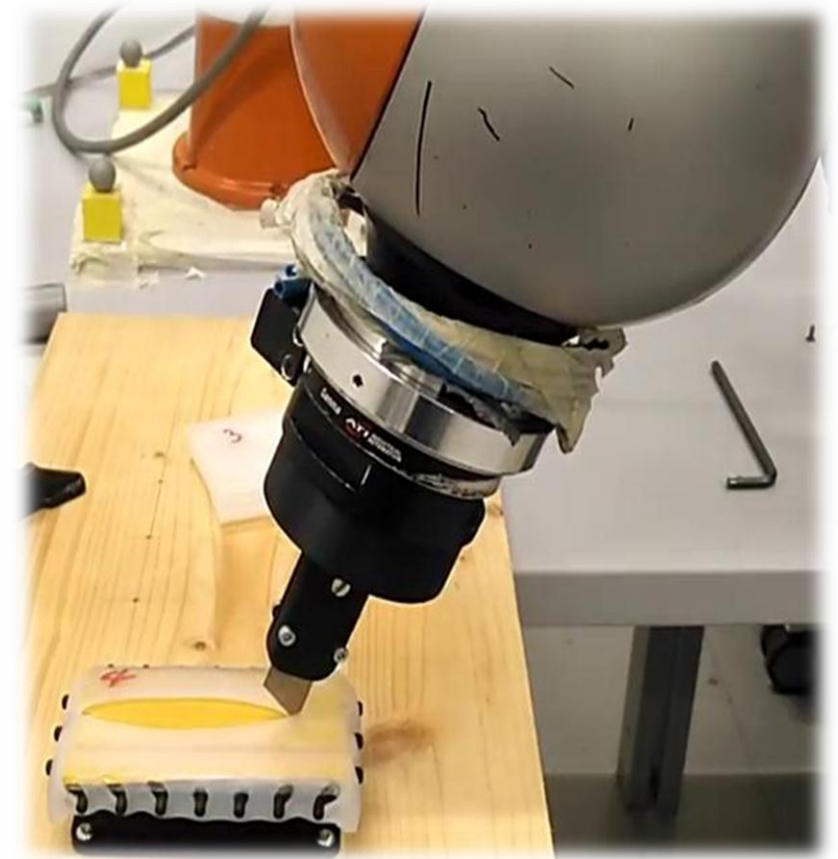
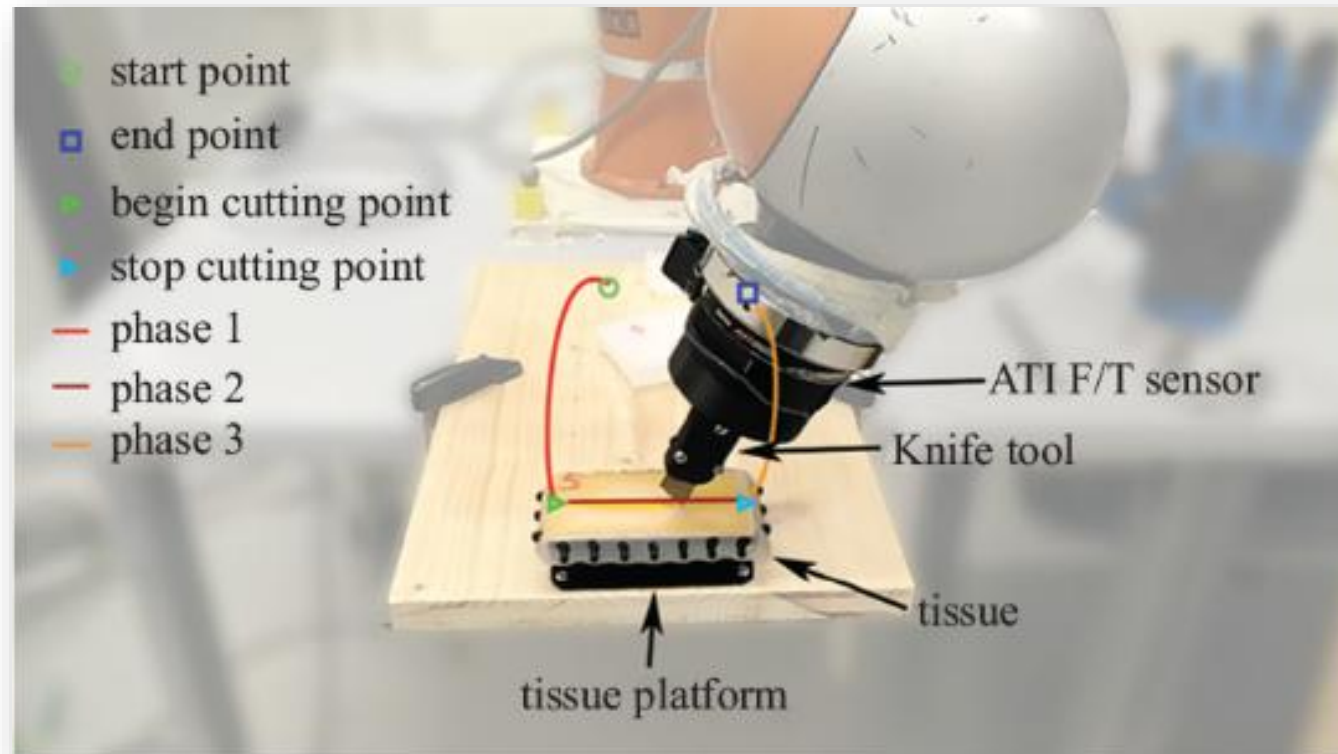


$$\dot{z} = (Bz + b)\partial(x - z):$$

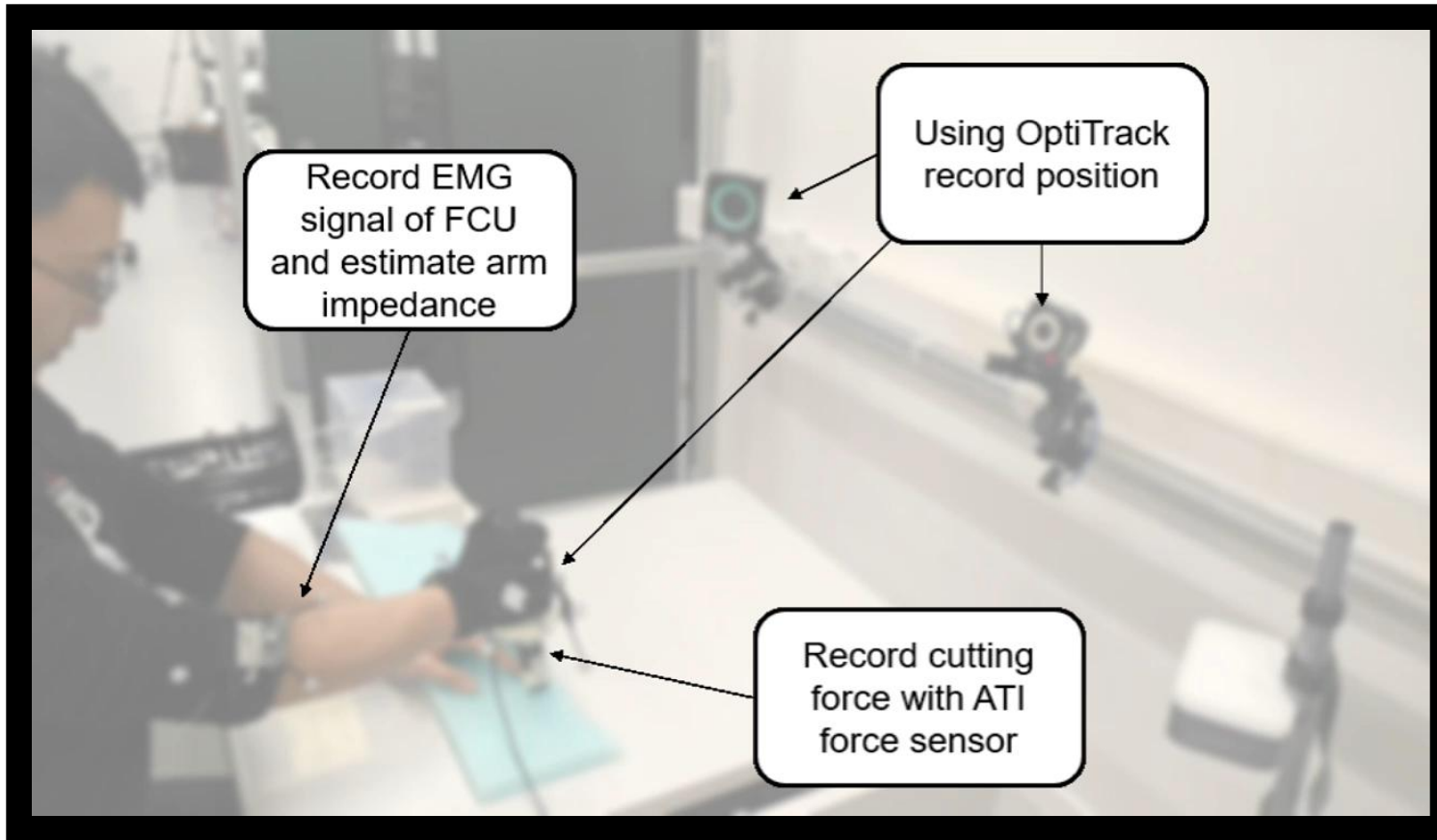
The virtual system stops as soon as x and z are not collocated
 → It waits until the system comes back to cutting point.

Application for Cutting Soft Tissue

Goal: cut a piece of silicon in a straight line



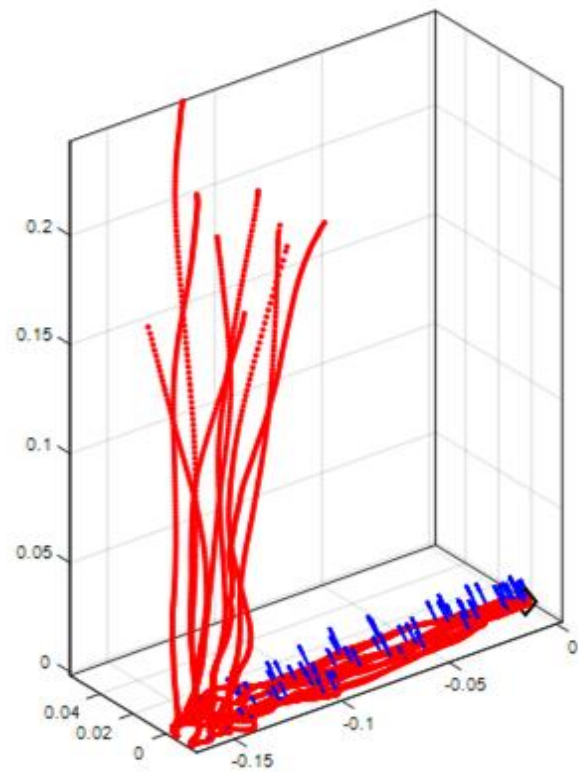
Human Demonstration



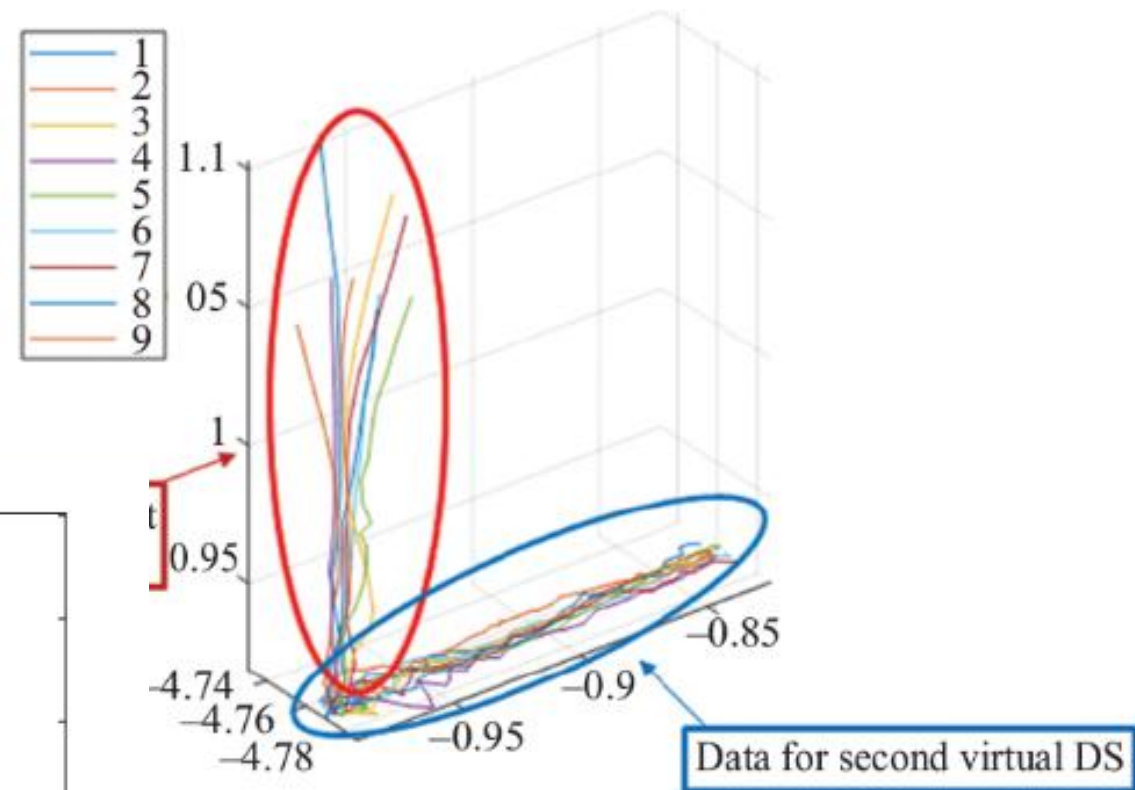
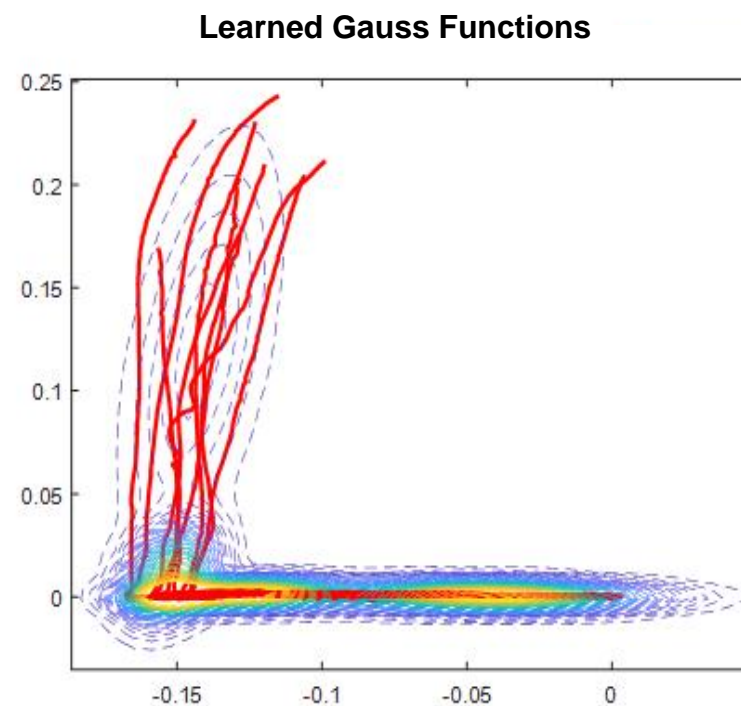
Recorded data:

- Knife position
- Interaction force
- Human arm impedance

From Raw Data to DS

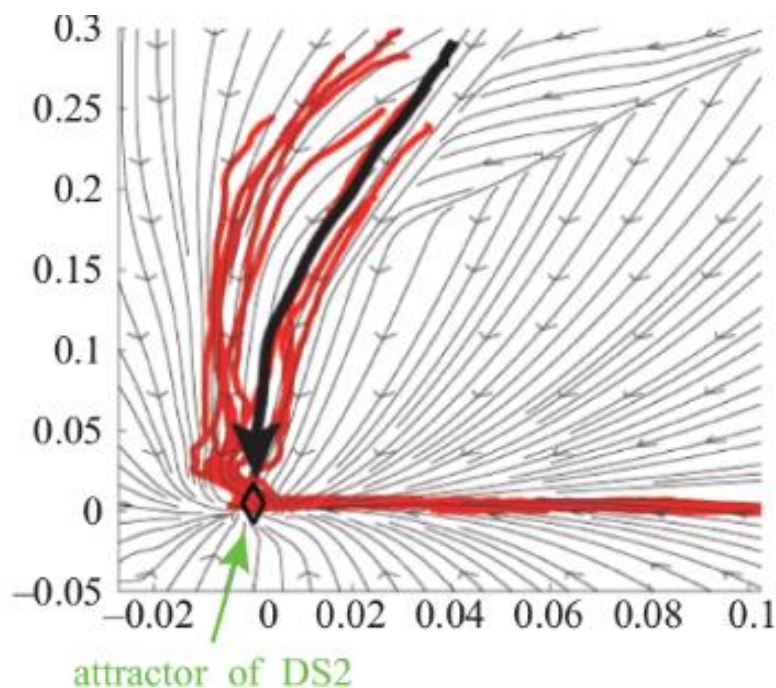


Raw Data

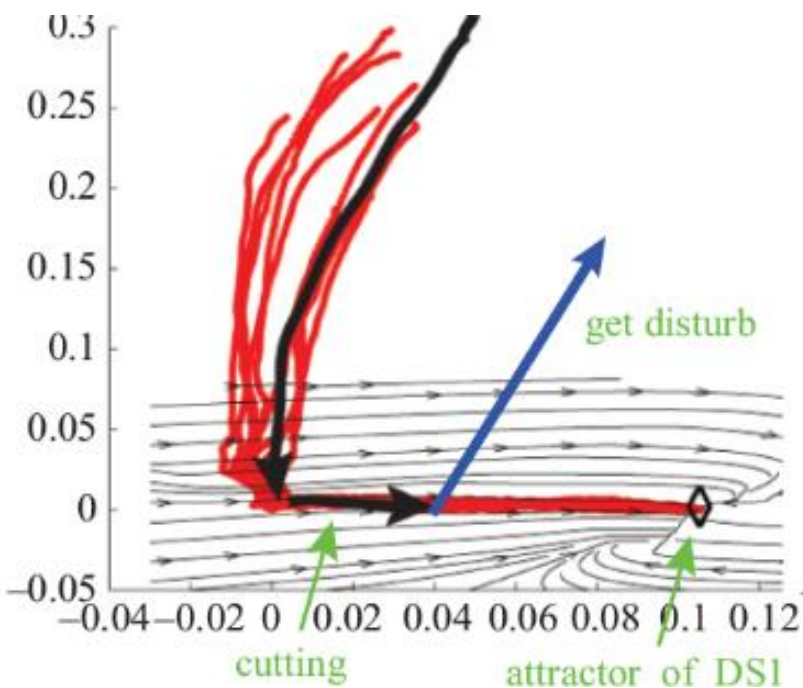


Modeling with Coupled DS

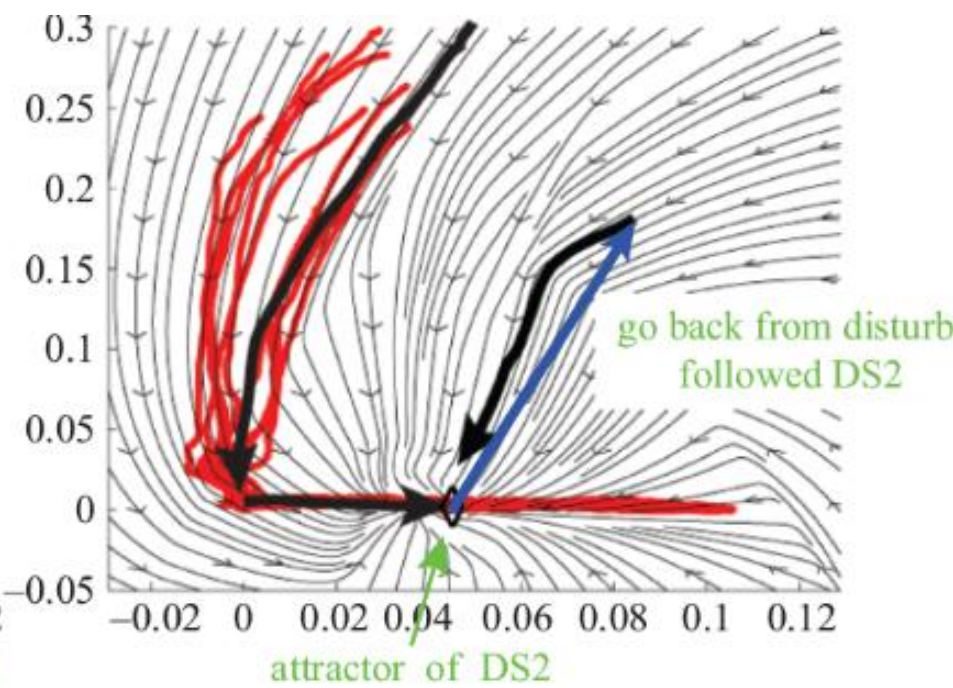
Approaching the Tissue



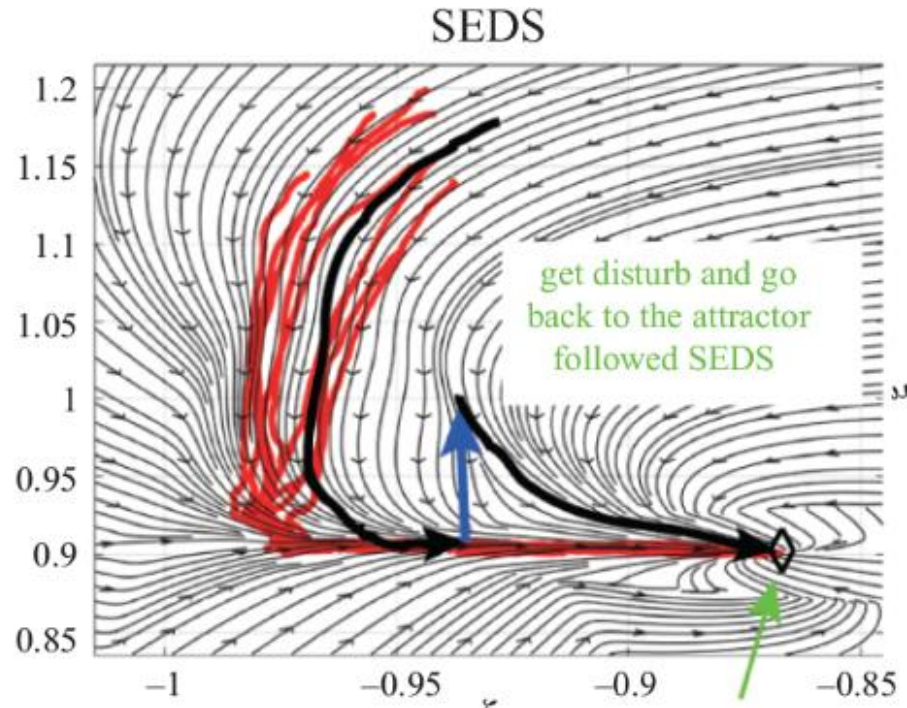
Cutting Phase



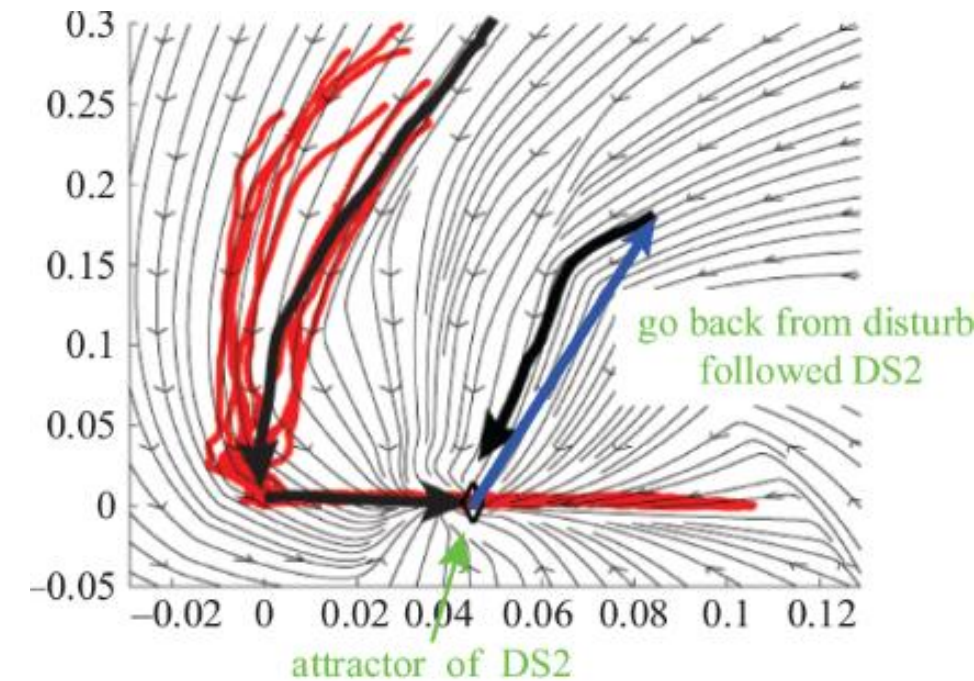
Recovery after disturbance



Comparison Coupled DS with SEDS

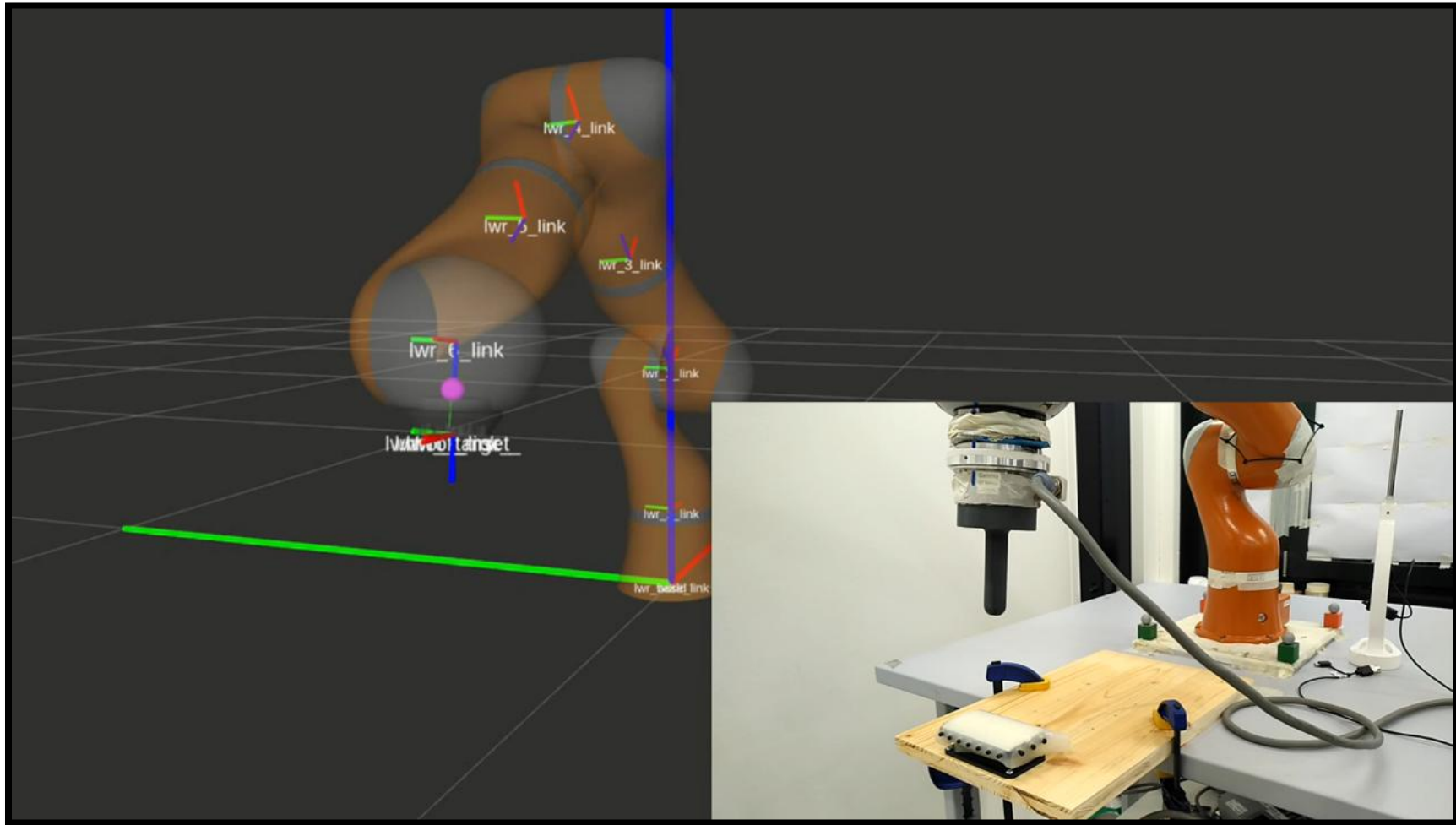


Recovery after disturbance

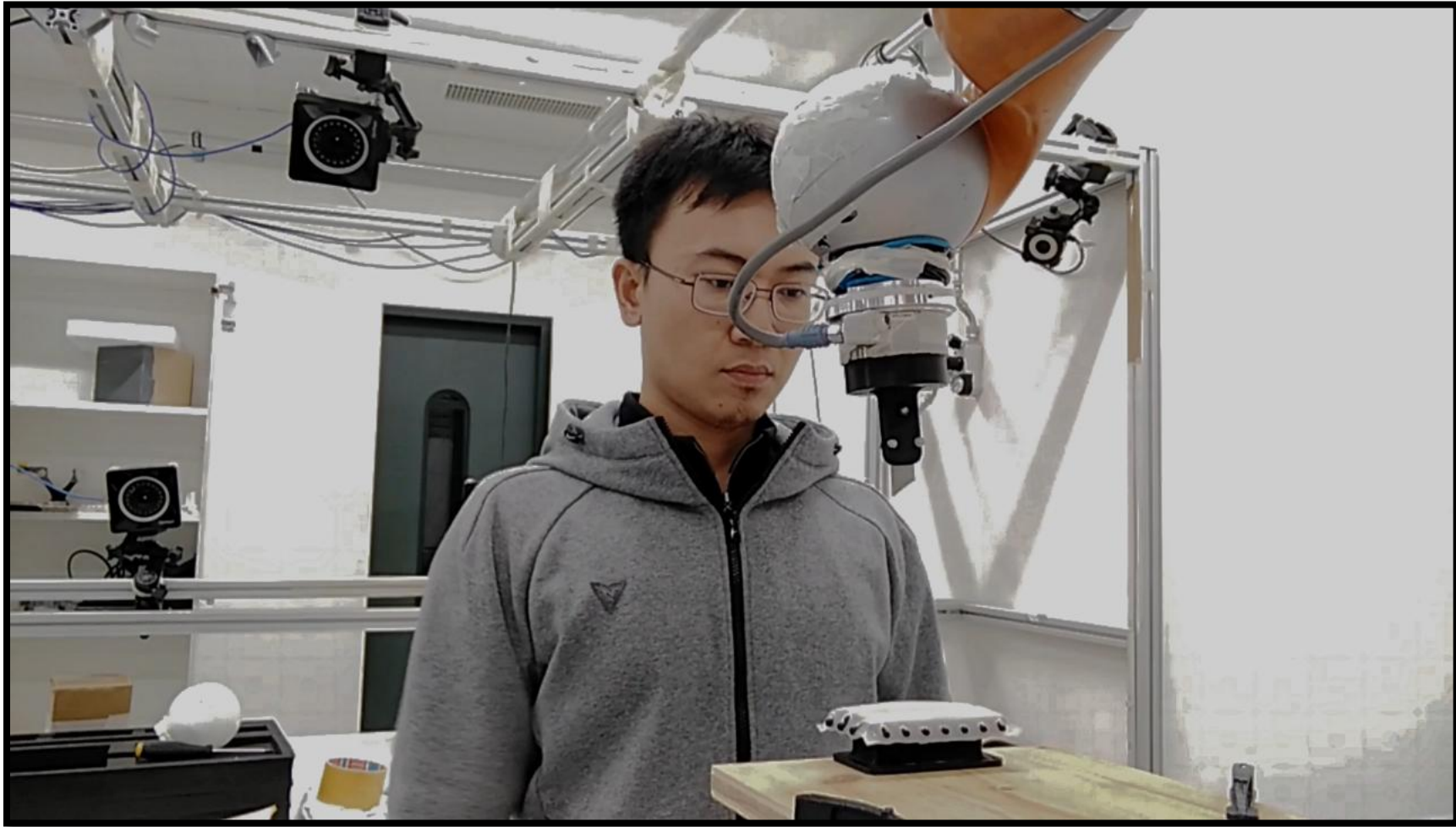


SEDS would simply send the knife to the end-effector but it would not come back to the cutting trajectory.

Robotic Implementation

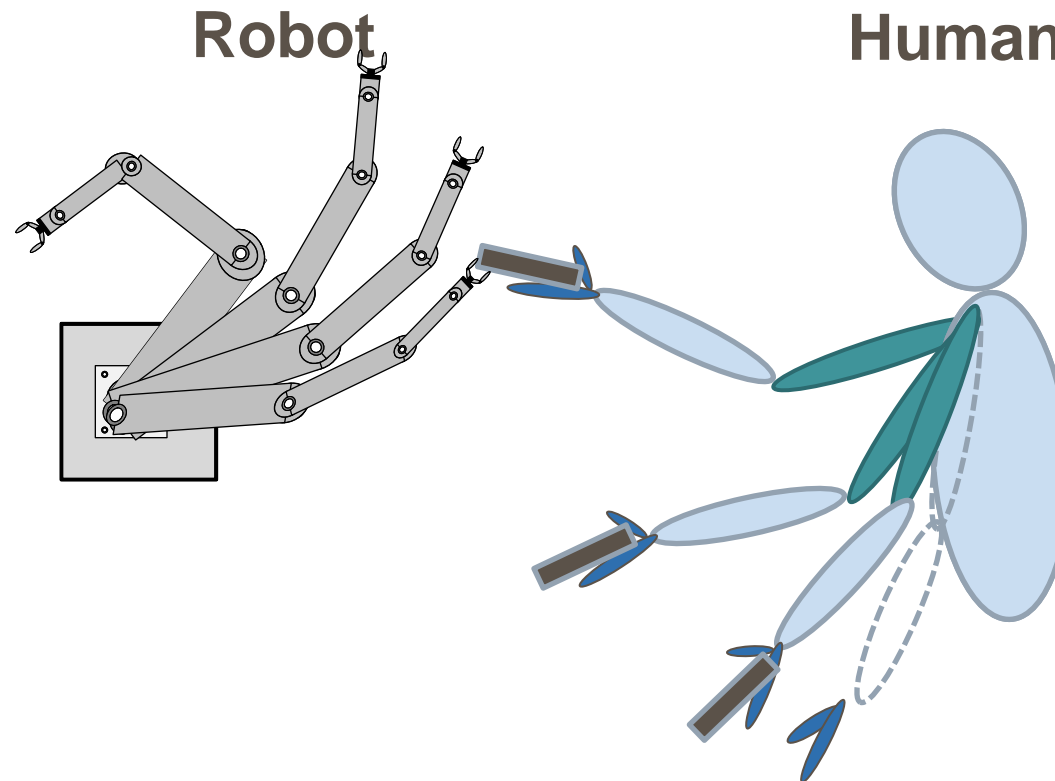


Robotic Implementation



DS attractor modulated by external input

Object Handover



Industrial Application





Control law for estimating human motion

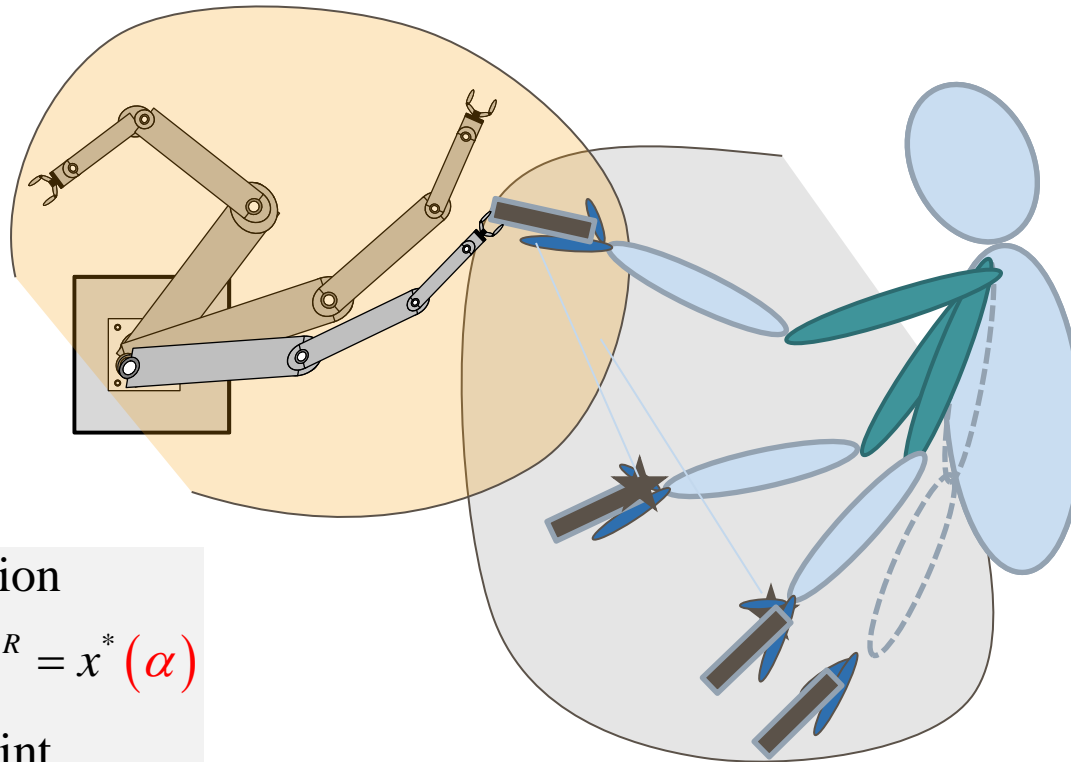
$$\dot{x}^H = A_H x^H$$

α : Load Share

Control law for robot motion

$$\dot{x}^R = f(x^R, g(\alpha)), \quad \lim_{t \rightarrow \infty} x^R = x^*(\alpha)$$

Origin on the intercept point



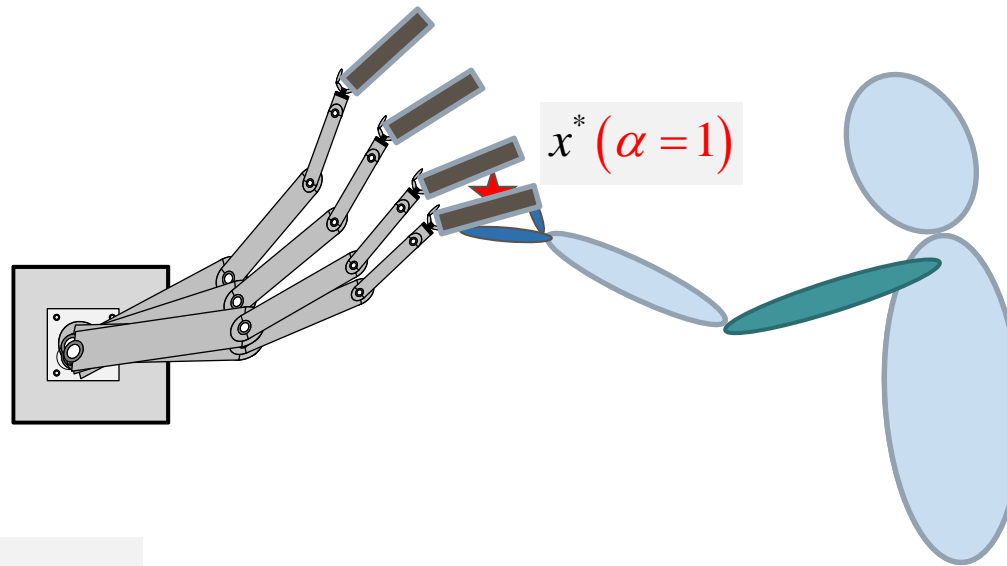
α : Load Share

Control law for robot motion

$$\dot{x}^R = f(x^R, g(\alpha)), \quad \lim_{t \rightarrow \infty} x^R = x^*(\alpha)$$

Origin on the intercept point

$\alpha = 1$: Full load on robot

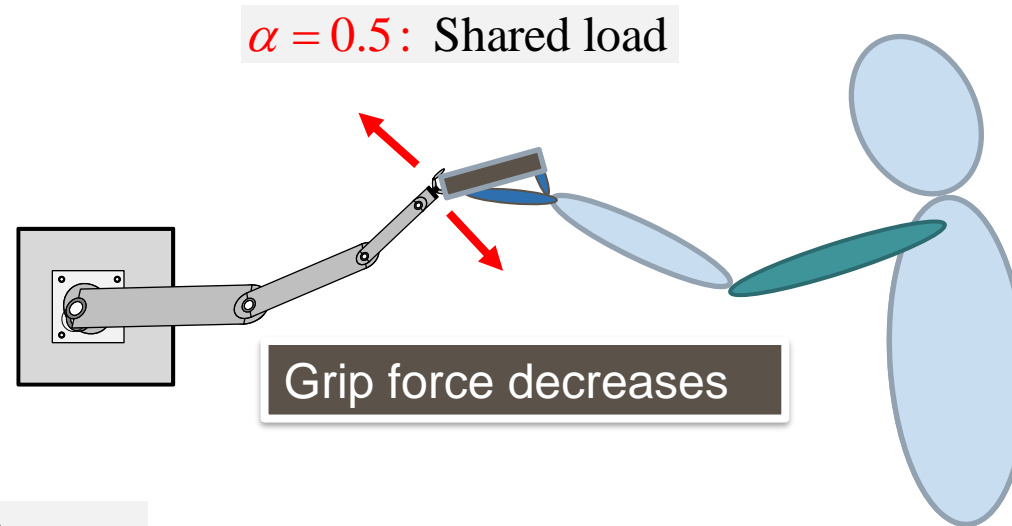


α : Load Share

Control law for robot motion

$$\dot{x}^R = f(x^R, g(\alpha)), \quad \lim_{t \rightarrow \infty} x^R = x^*(\alpha)$$

Origin on the intercept point



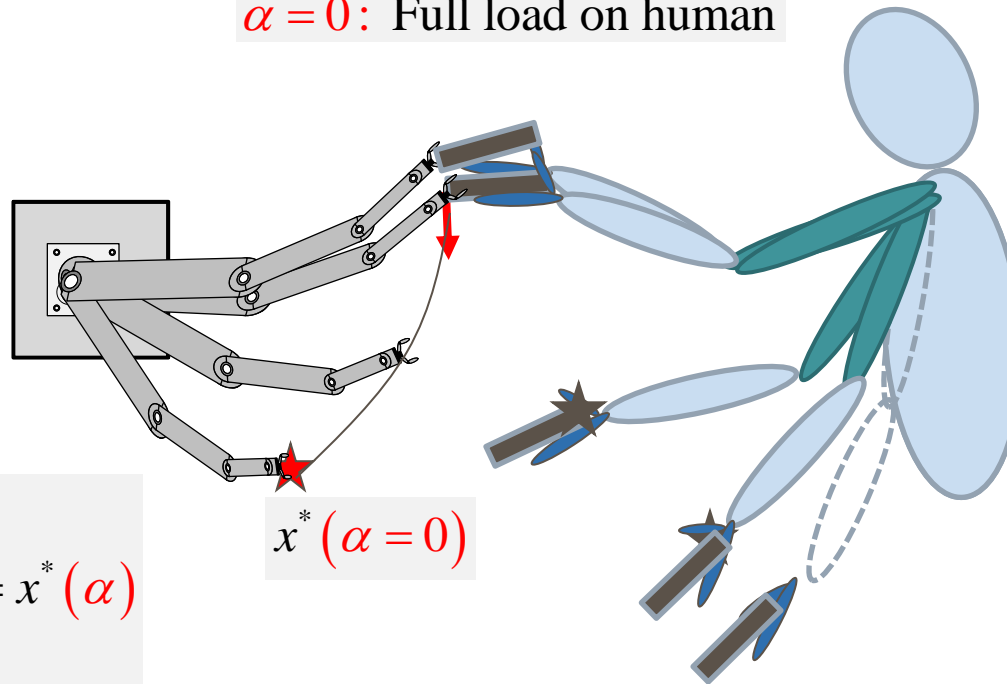
α : Load Share

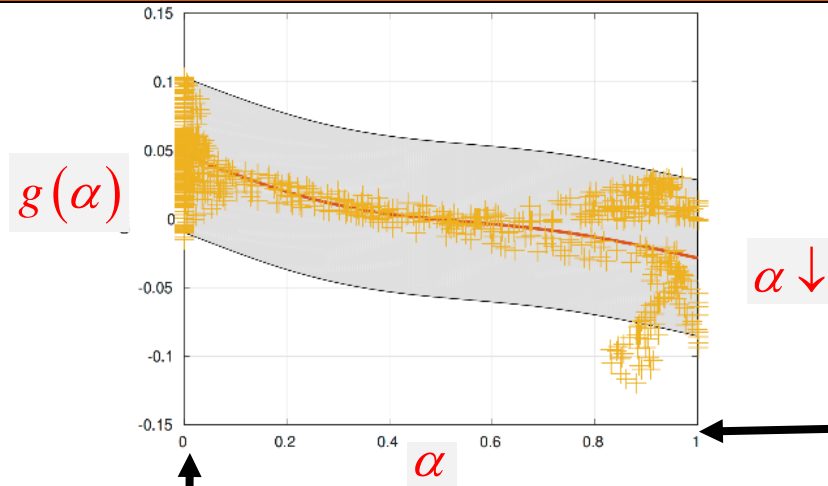
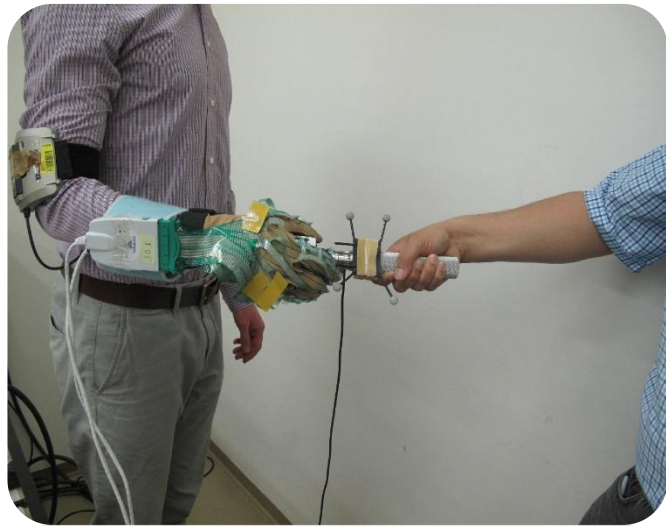
Control law for robot motion

$$\dot{x}^R = f(x^R, g(\alpha)), \quad \lim_{t \rightarrow \infty} x^R = x^*(\alpha)$$

Origin on the intercept point

$\alpha = 0$: Full load on human



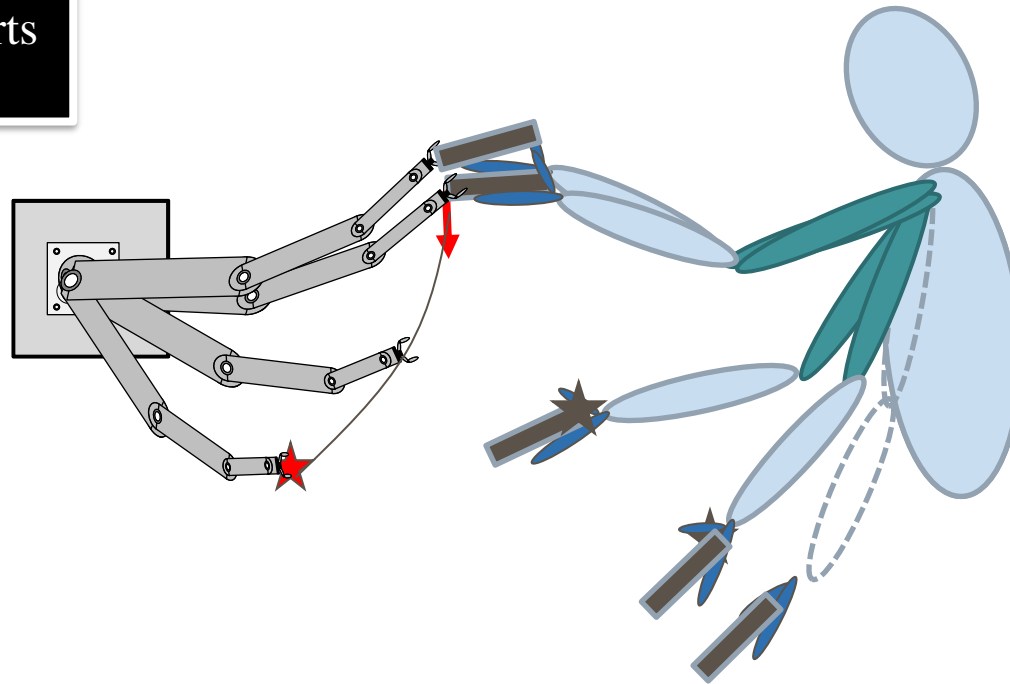


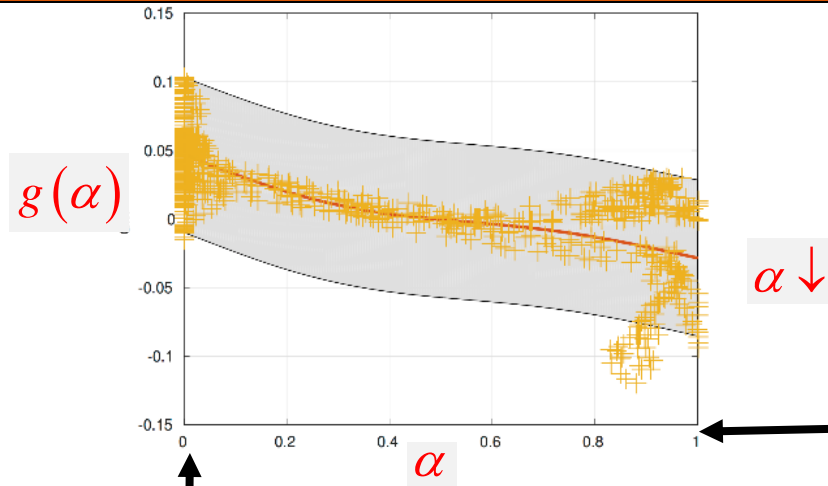
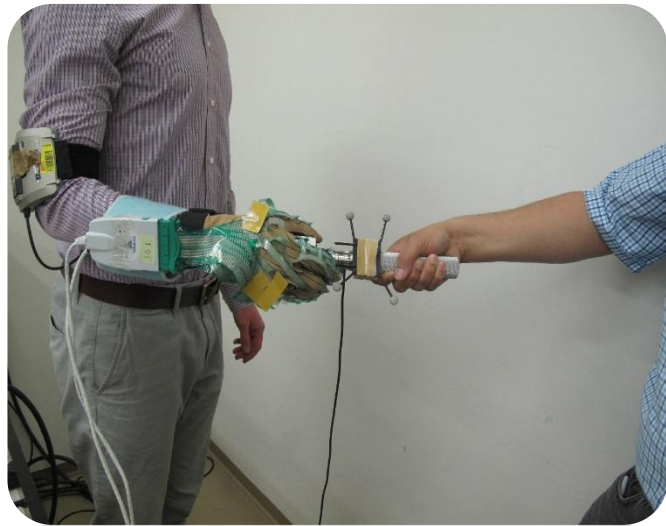
Giver supports object

Receiver supports object

Modeling $g(\alpha)$ from human data

α : Load Share



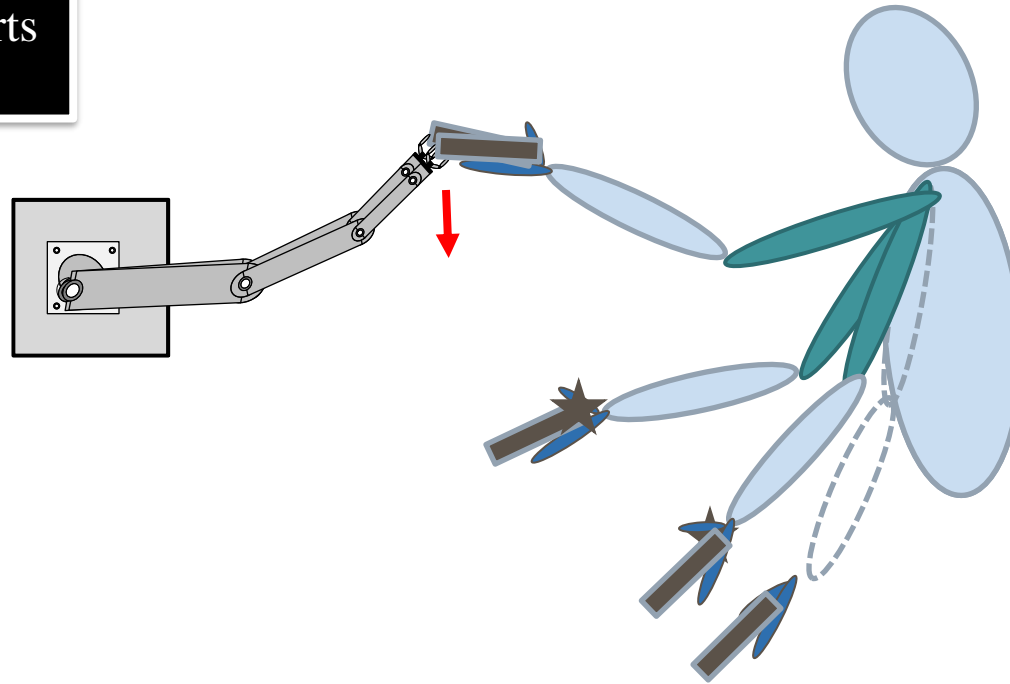


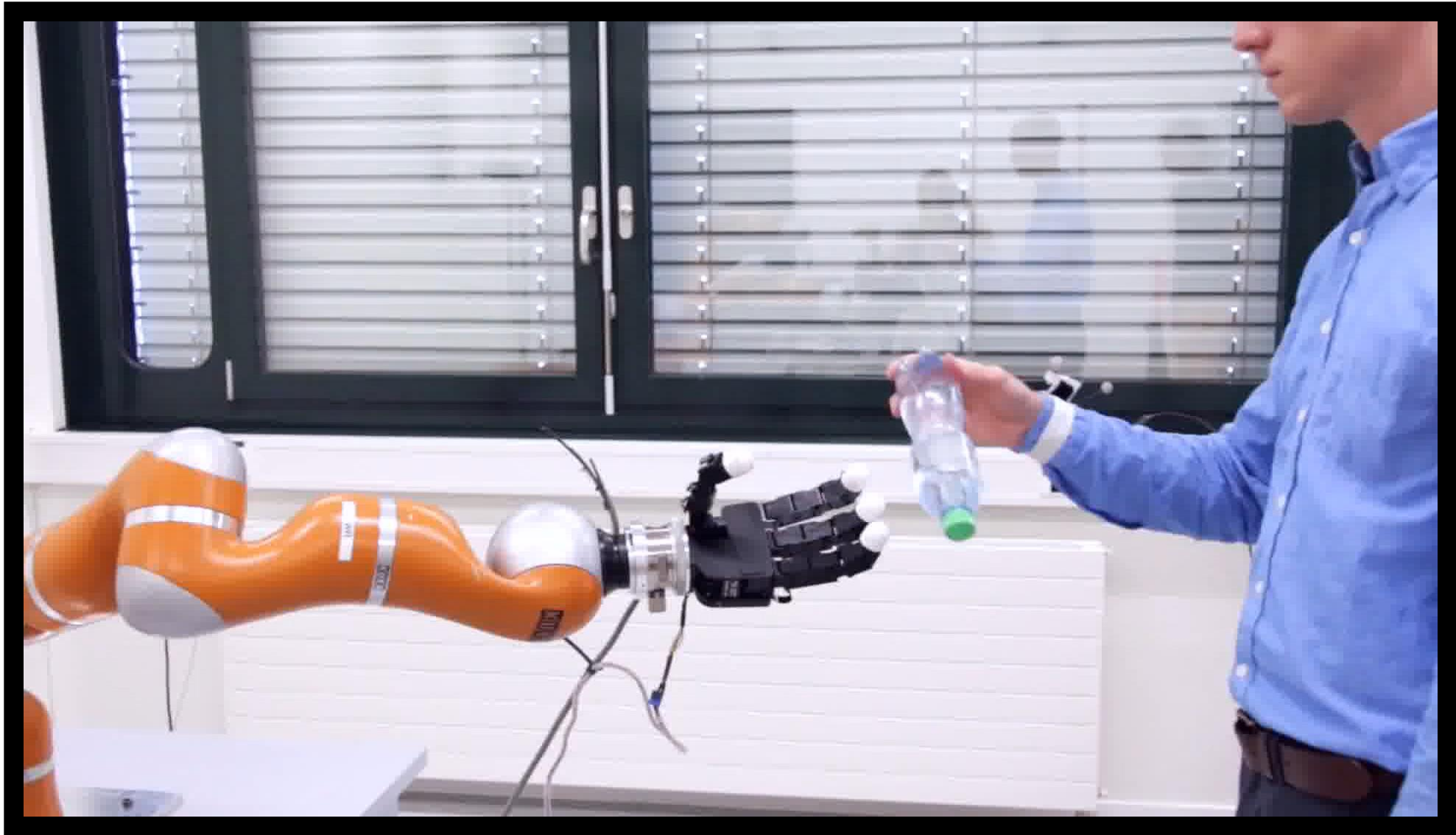
Giver supports
object

Receiver
supports object

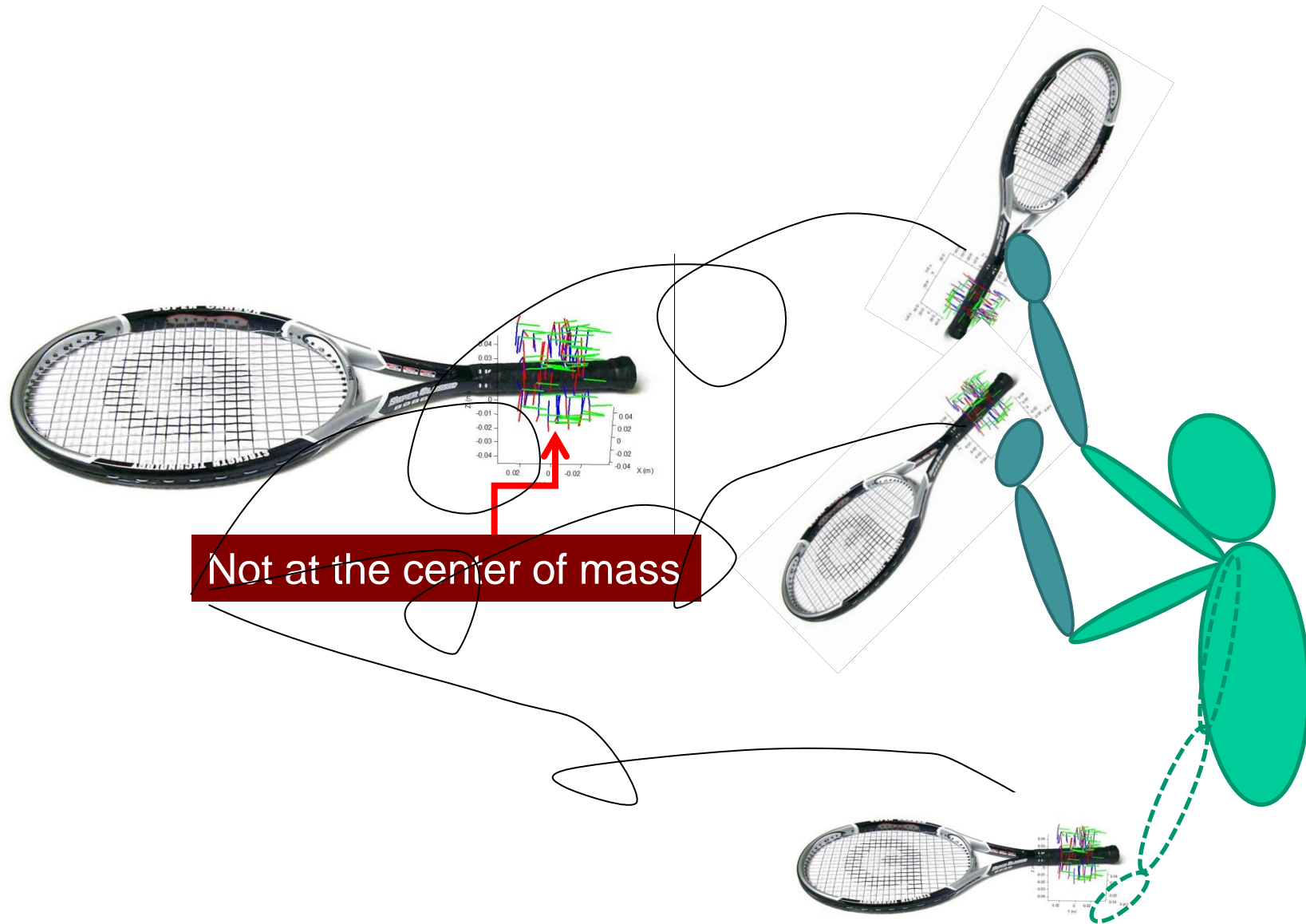
Modeling $g(\alpha)$ from human data

α : Load Share





DS for tracking moving objects



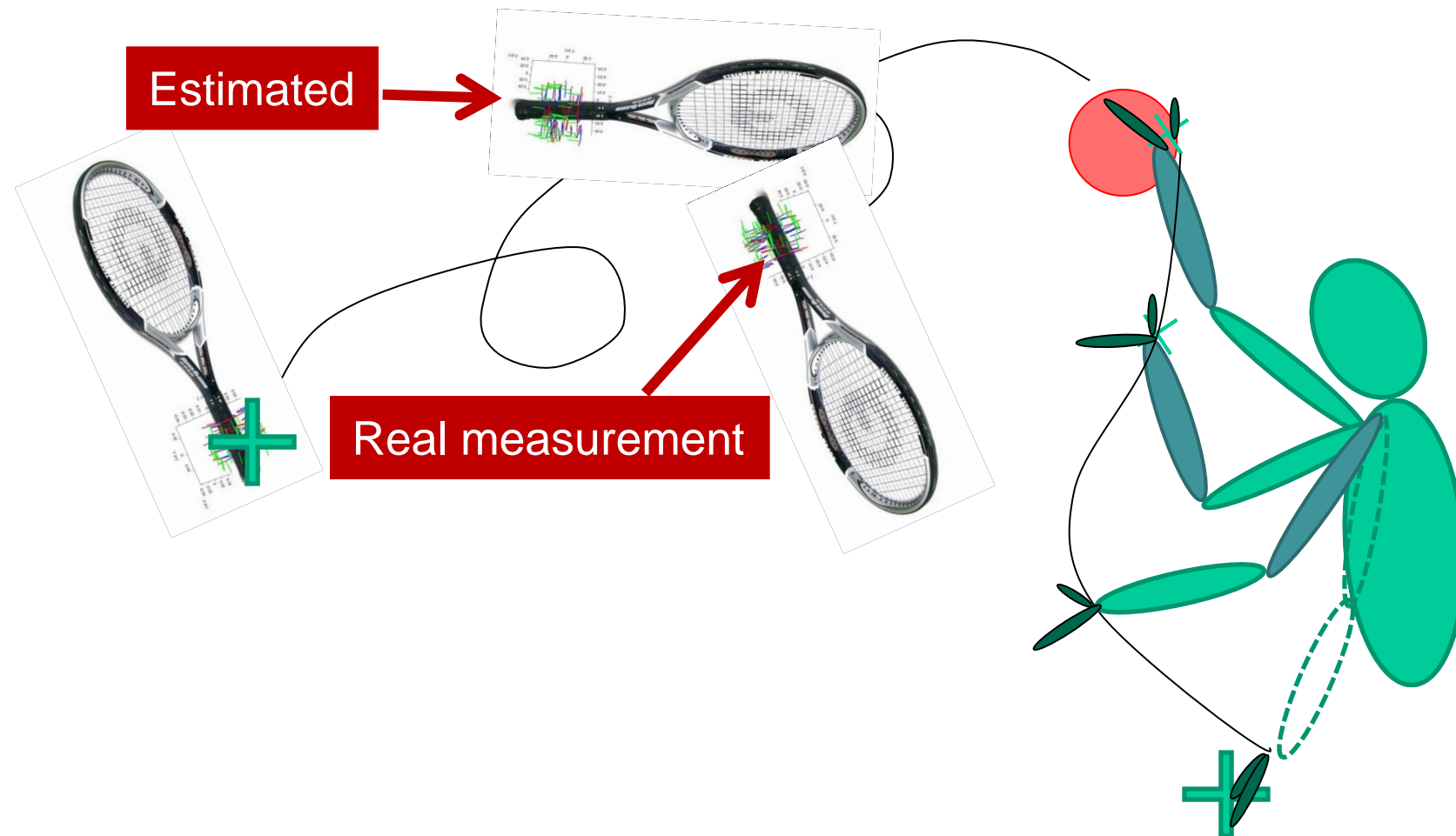
Modeling Object's Nonlinear Flying Dynamics

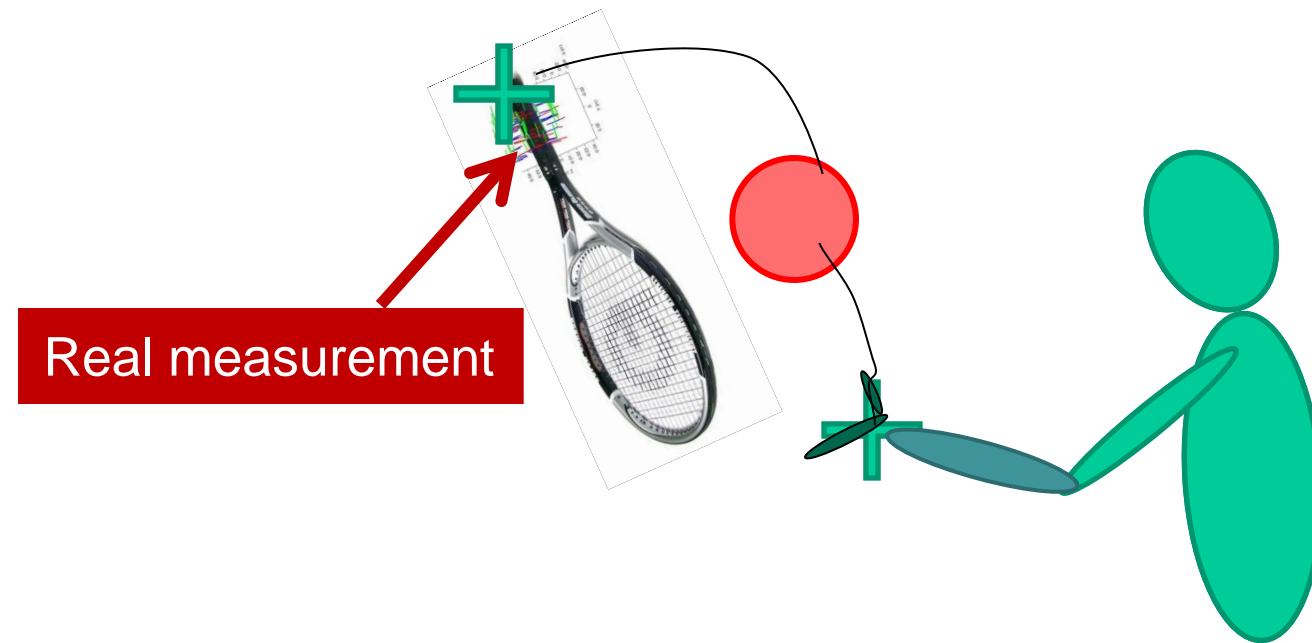


Build model of dynamics using
Support Vector Regression

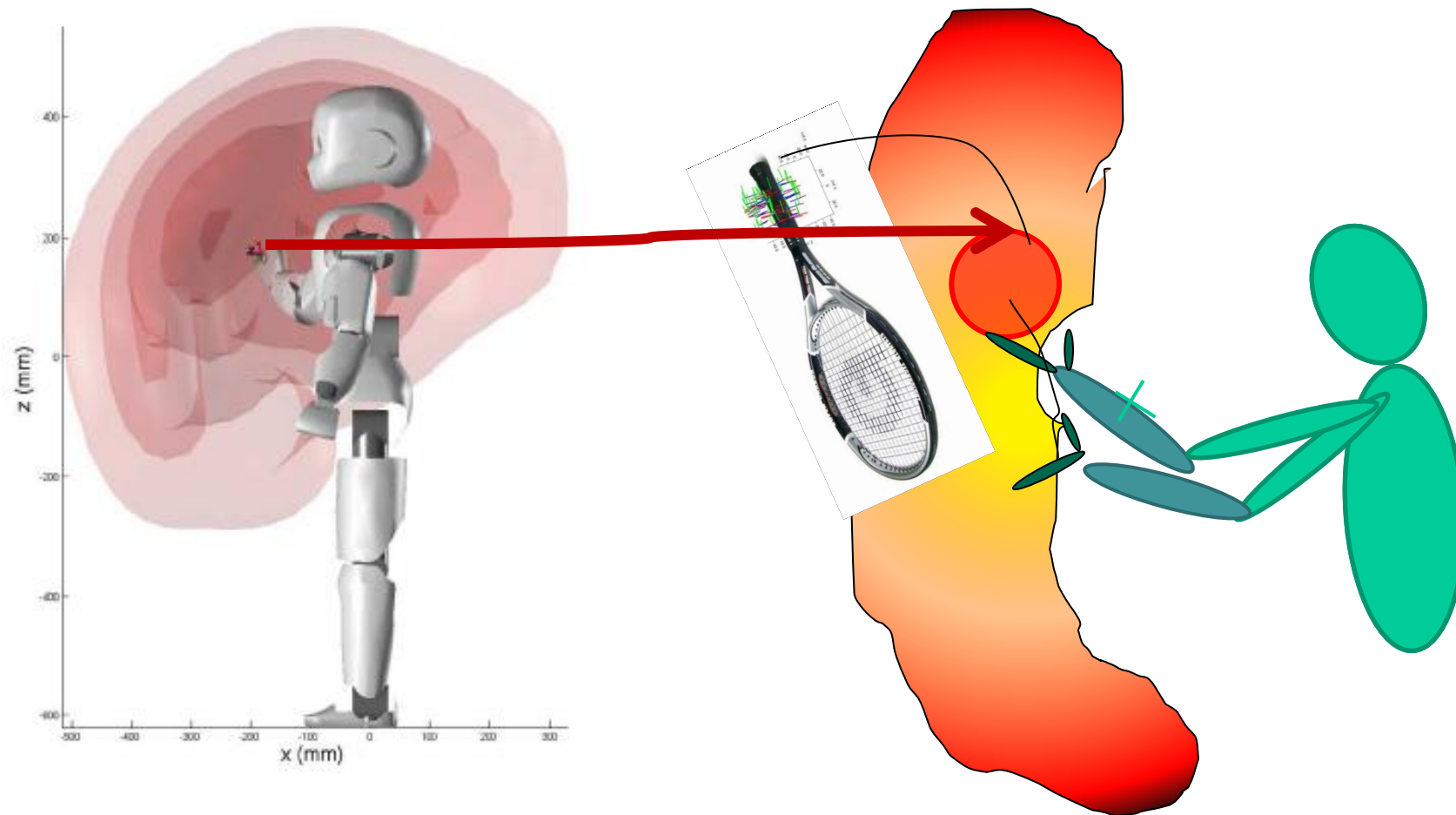
$$\ddot{x} = \sum_{i=1}^M \alpha_i k \left(\begin{bmatrix} x^i & \dot{x}^i \end{bmatrix}^T, \begin{bmatrix} x & \dot{x} \end{bmatrix}^T \right) + b$$

Compute derivative (closed-form) for Extended Kalman Filter

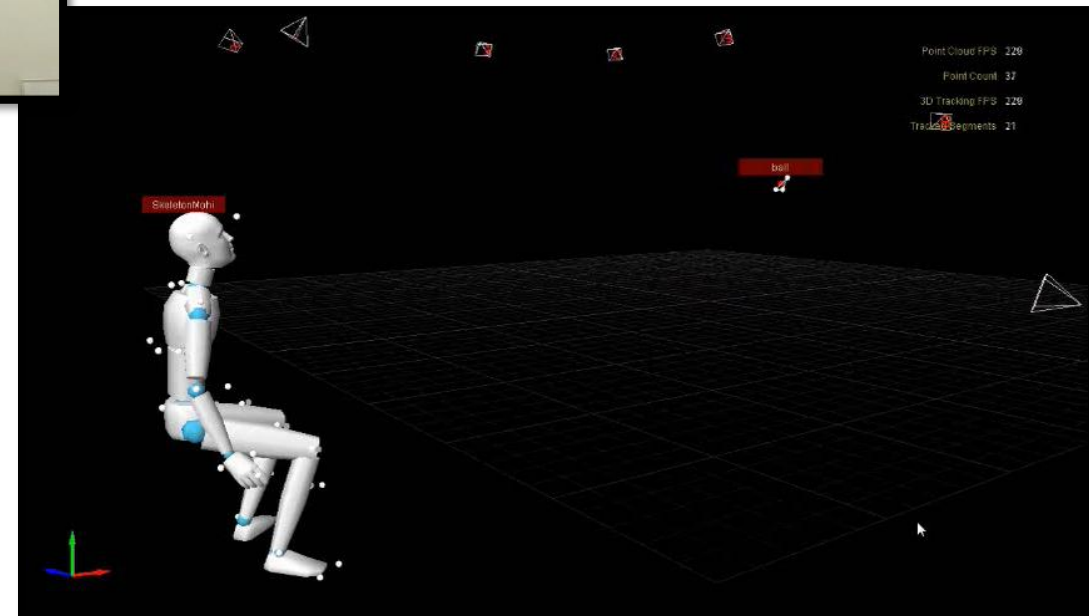
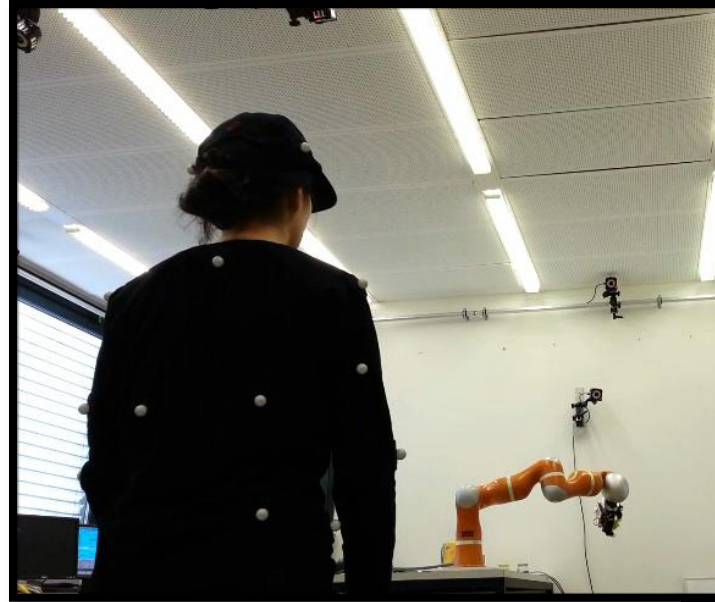


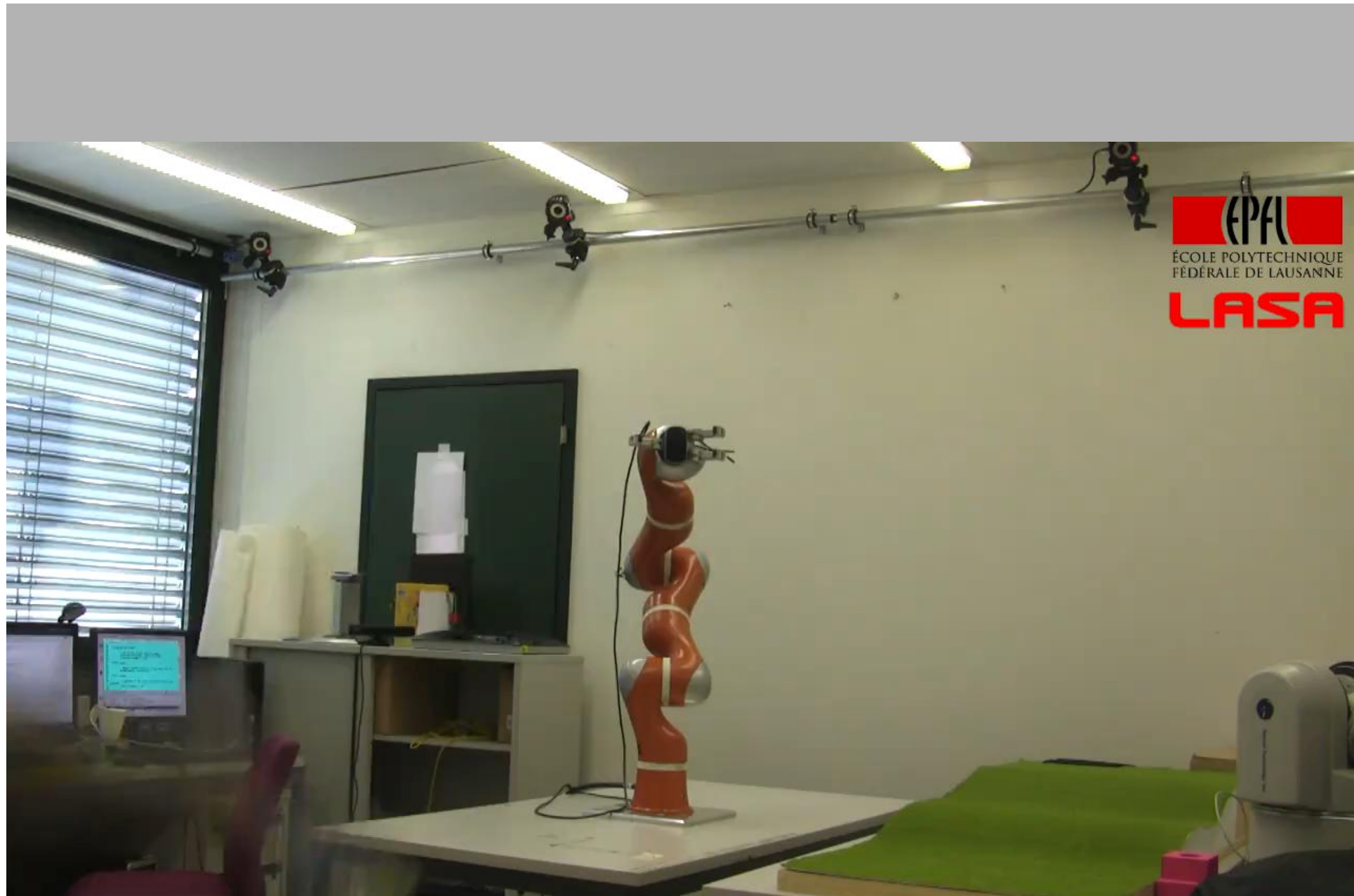


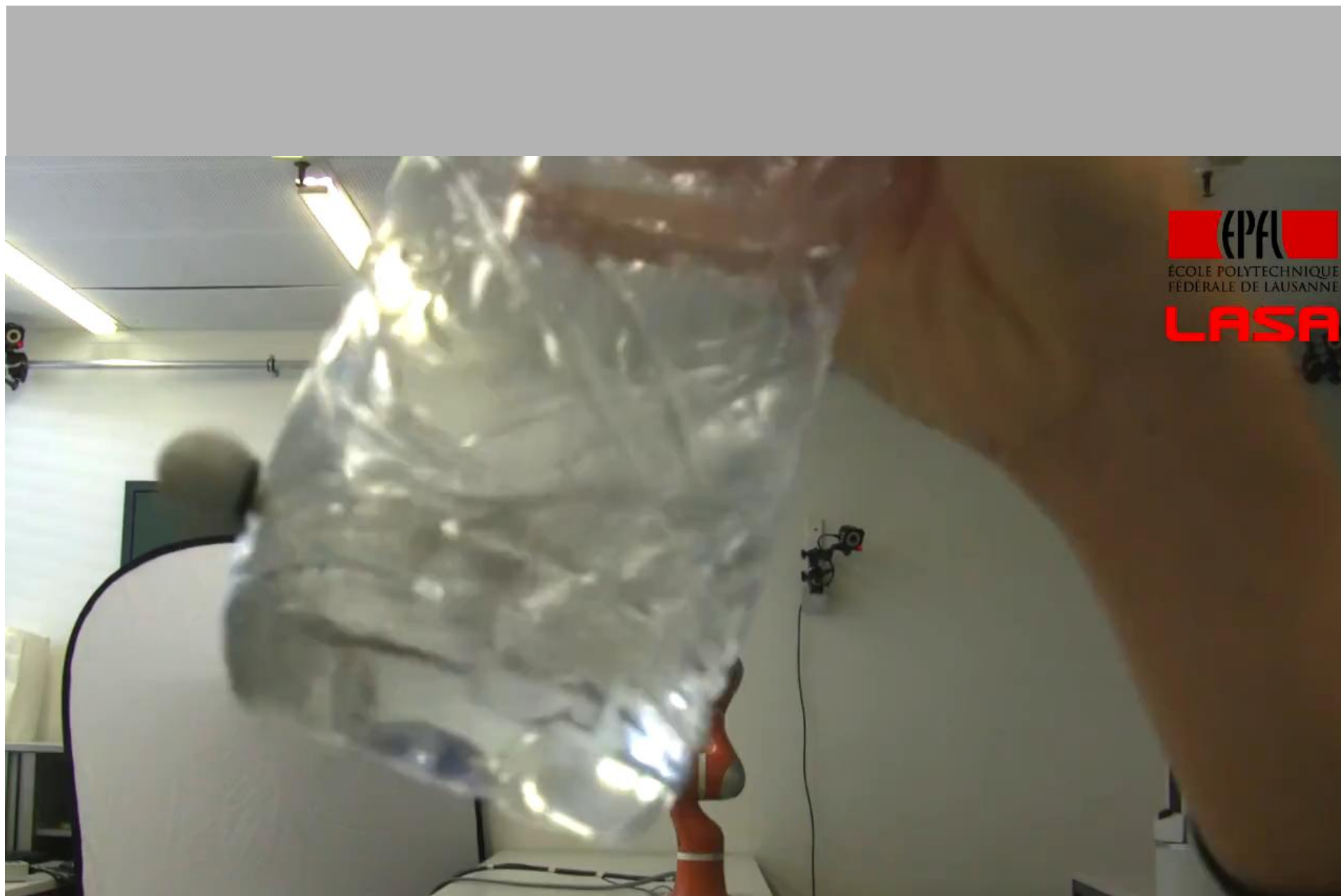
Learn most likely region to catch object

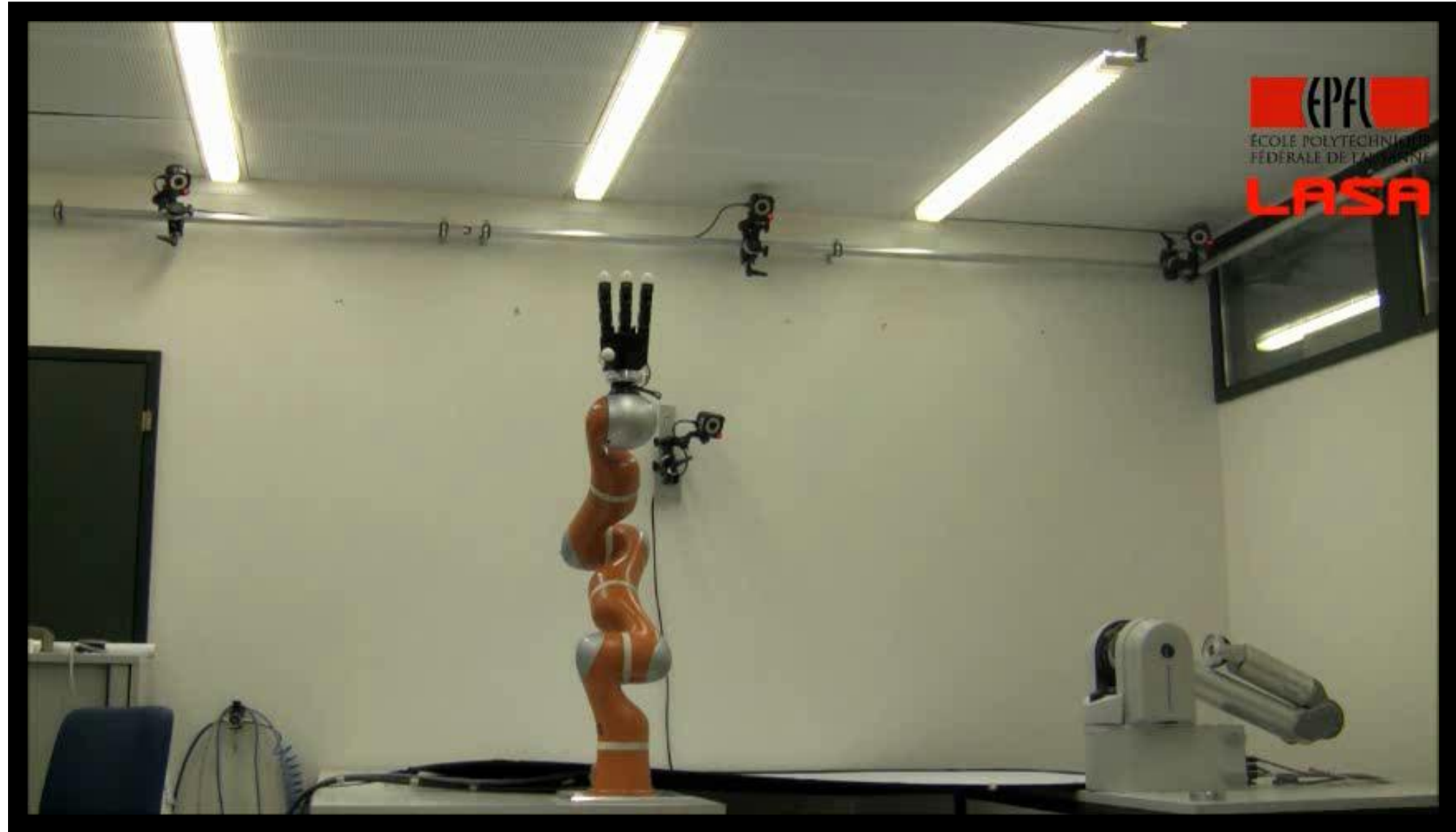


Learn arm-hand coupling



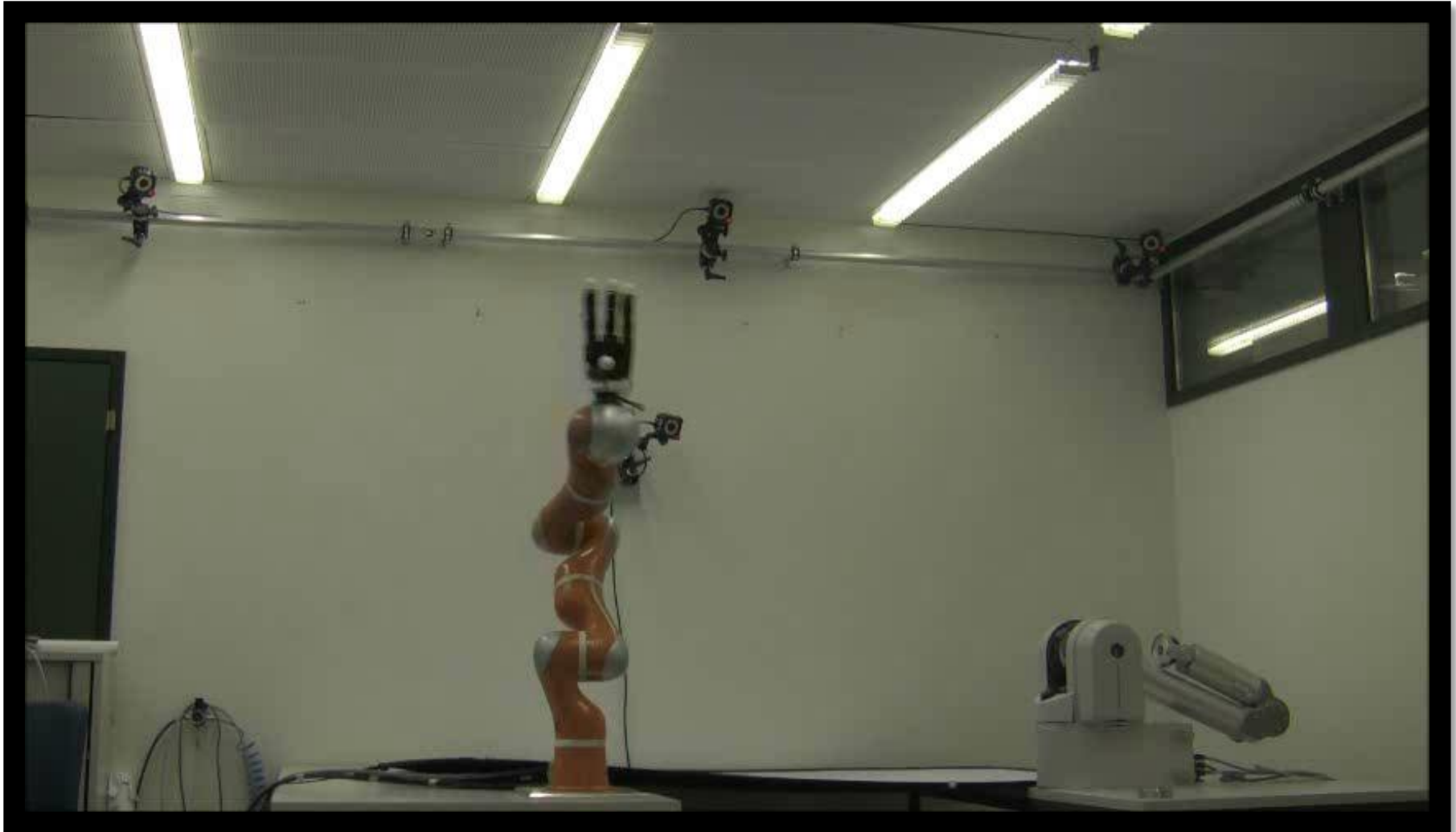




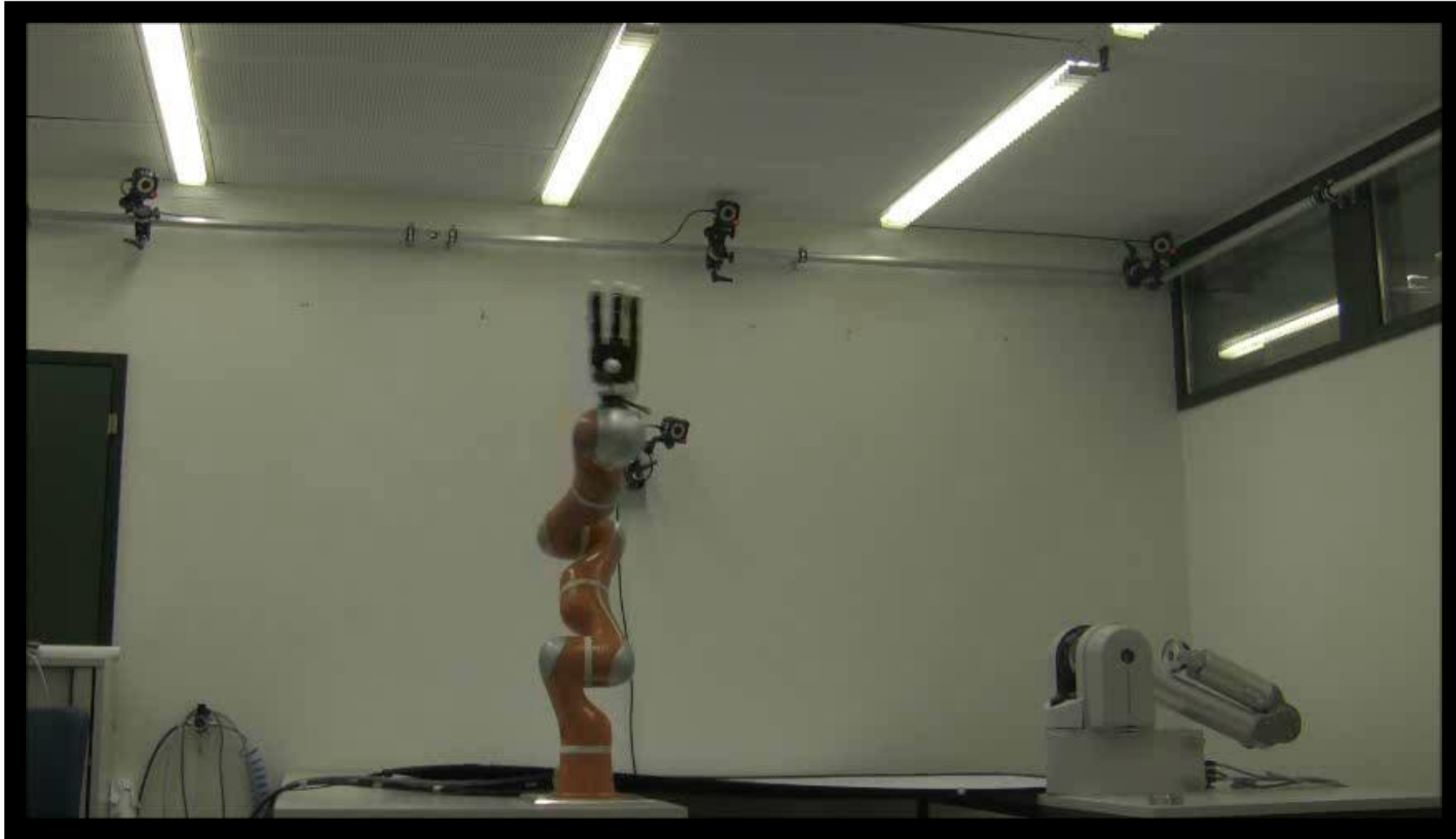


When it fails ...

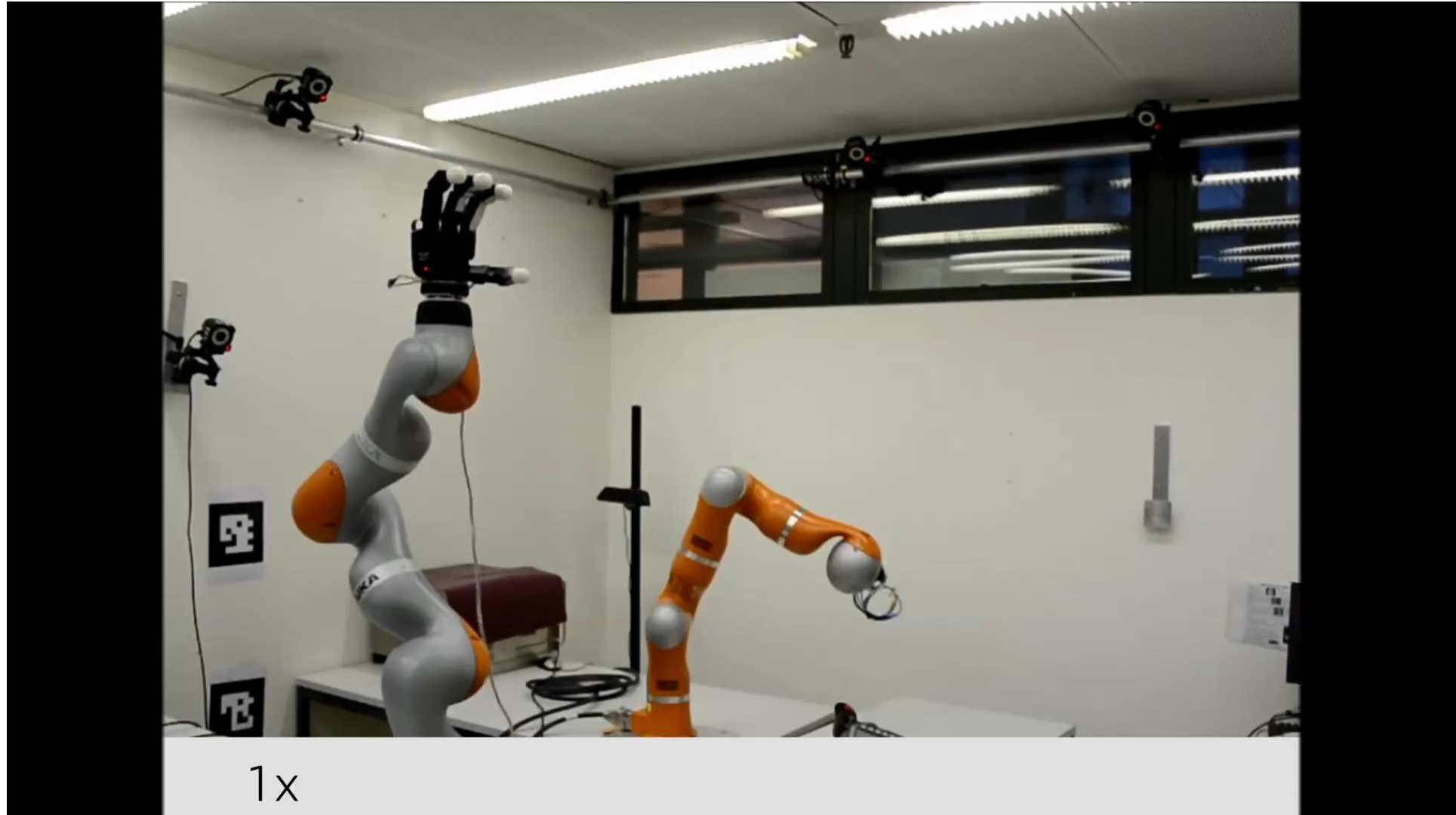
Just on time!



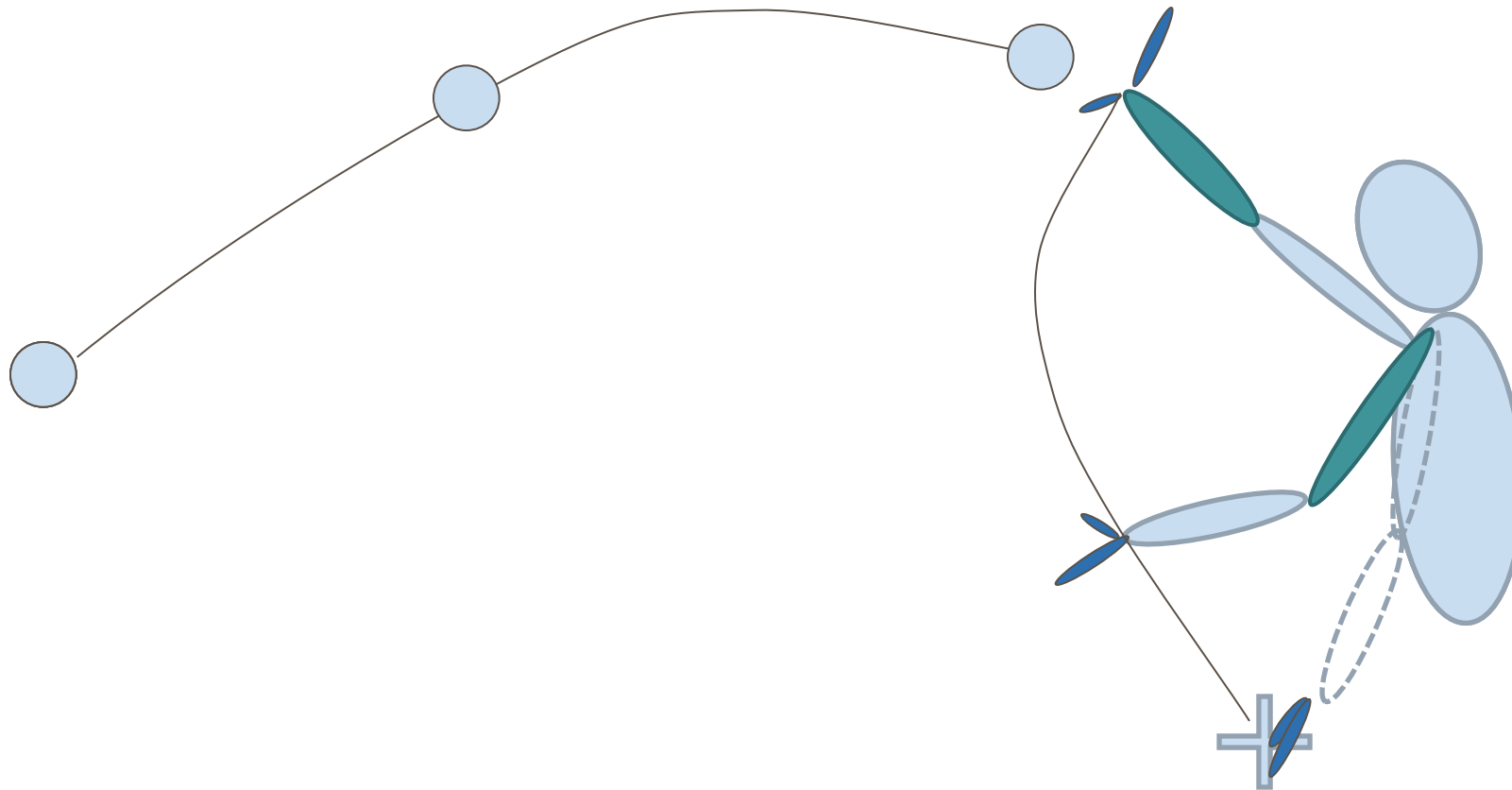
Failure due to imprecise closing of fingers



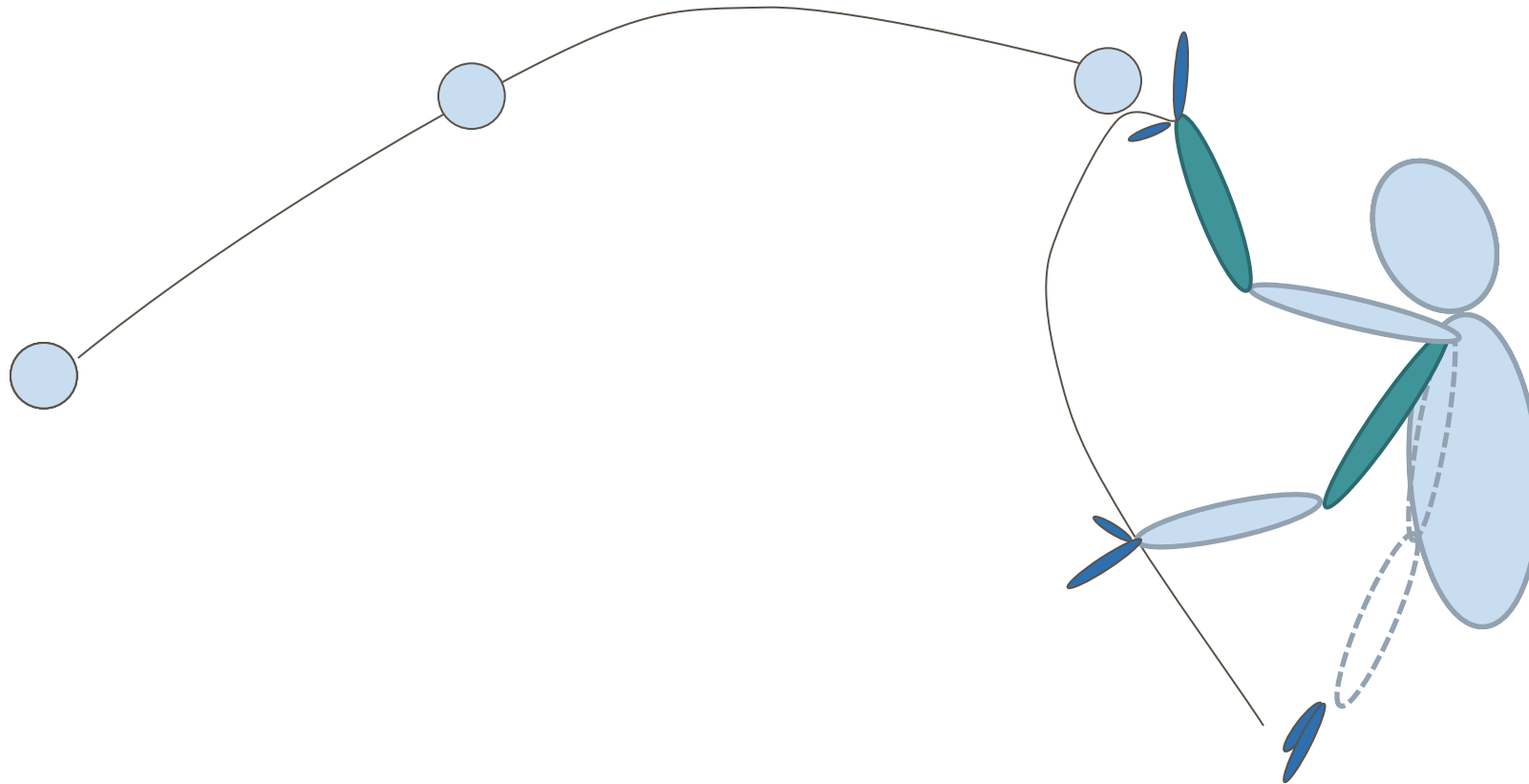
Failure for lack of time to close the fingers



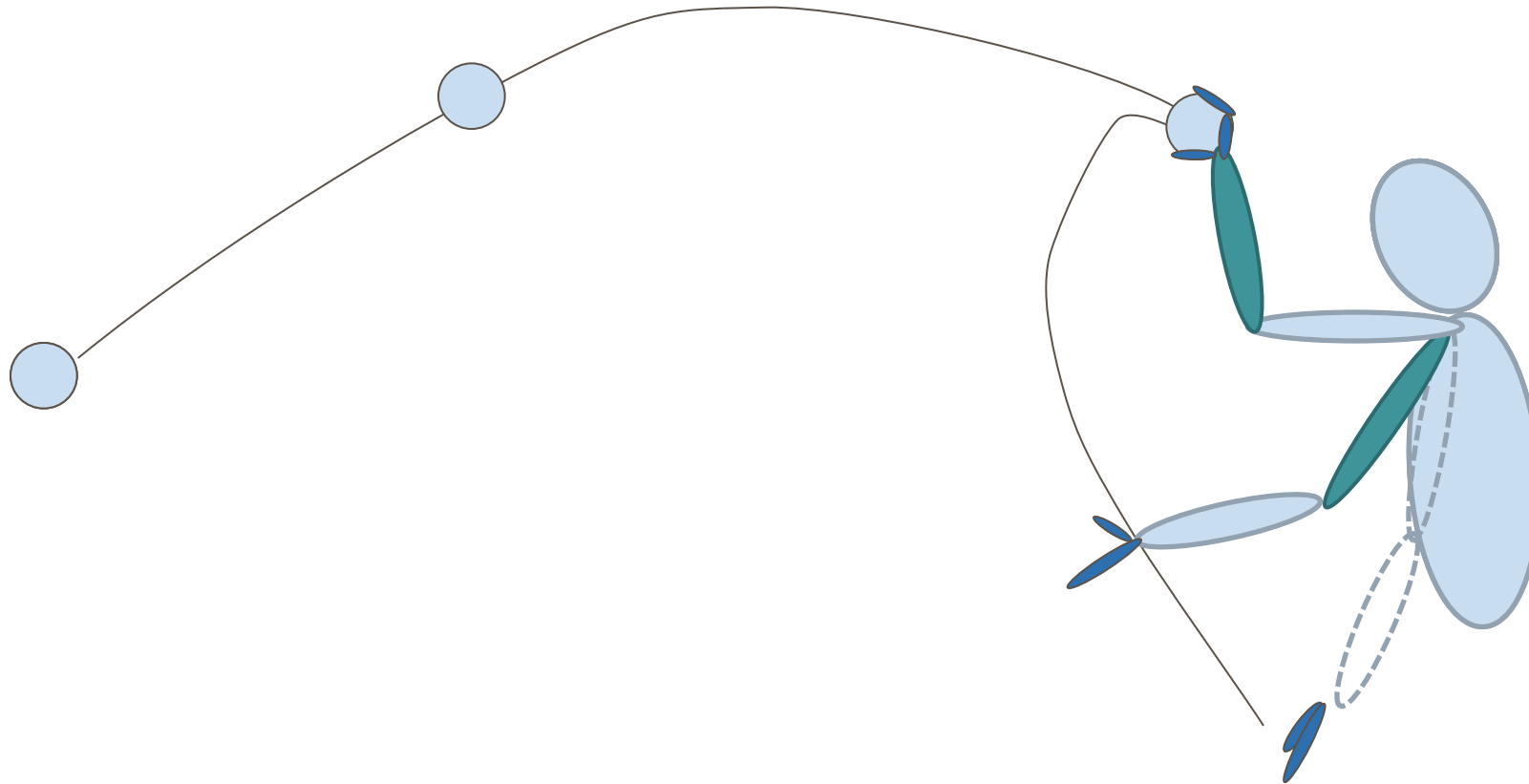
Soft Catching Strategy



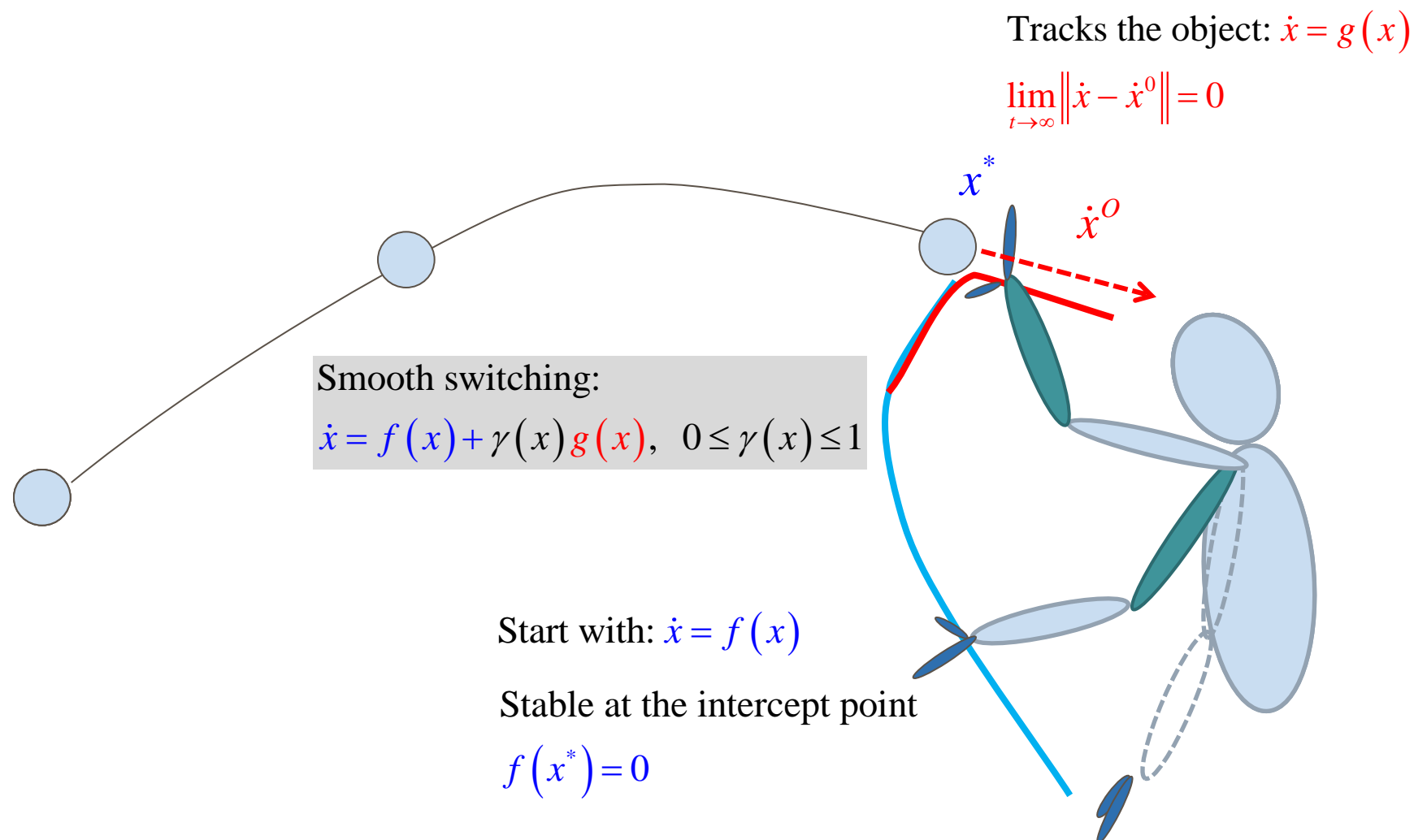
Soft Catching Strategy



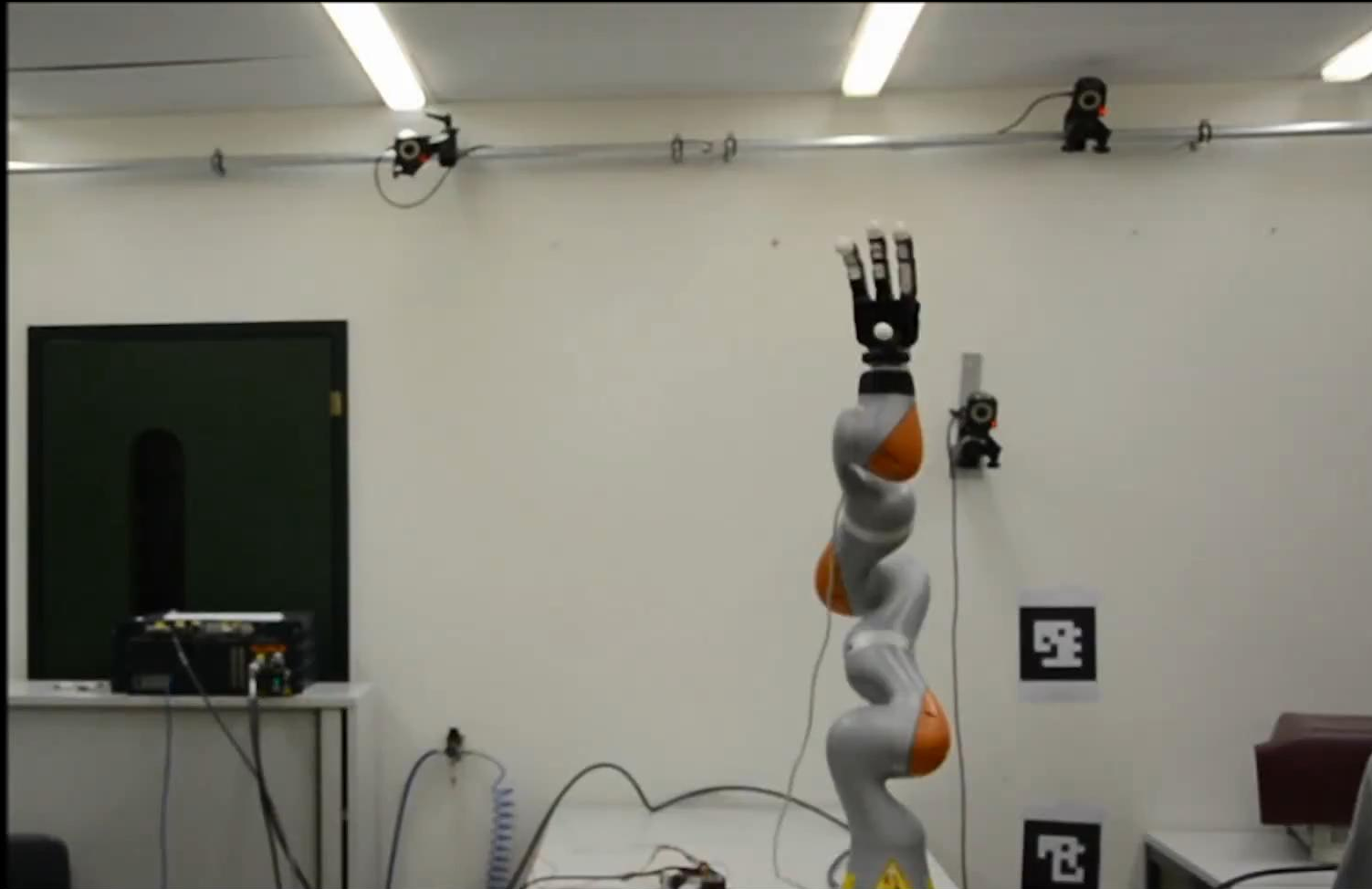
Soft Catching Strategy



Soft Catching Strategy



Increase performance to more than 80%

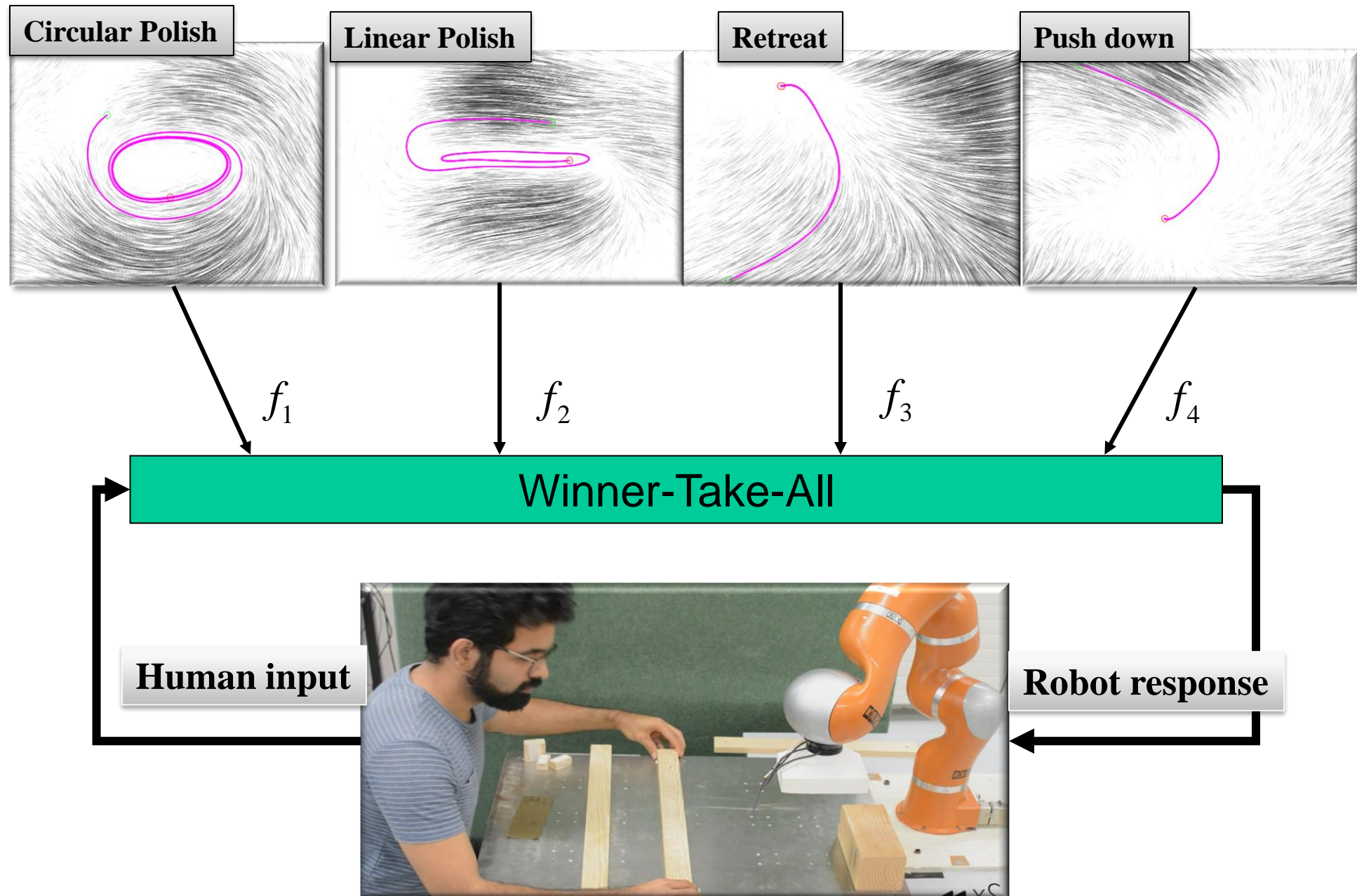


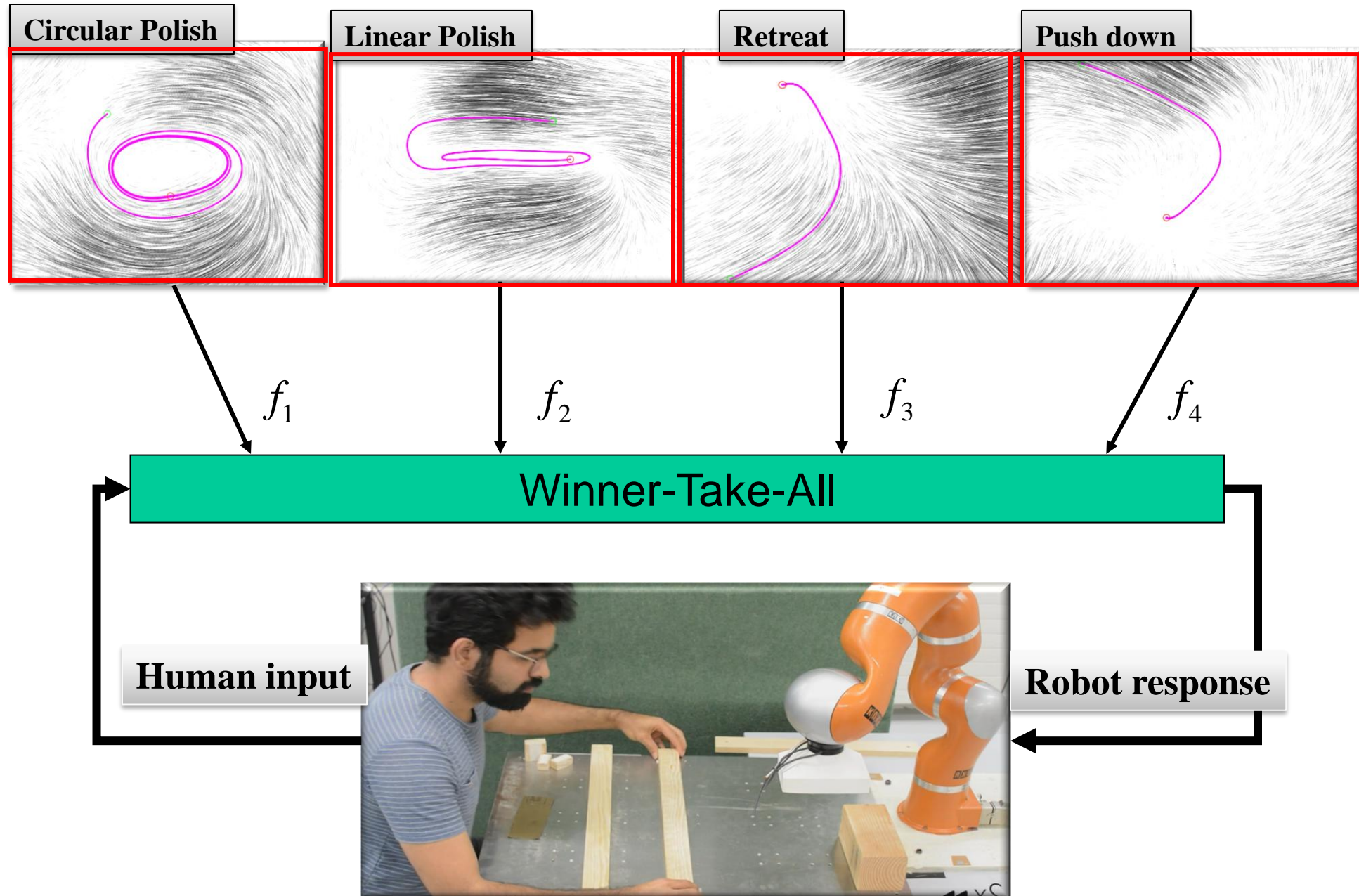
1x



Finalist KUKA Award 2017

Switching across DS







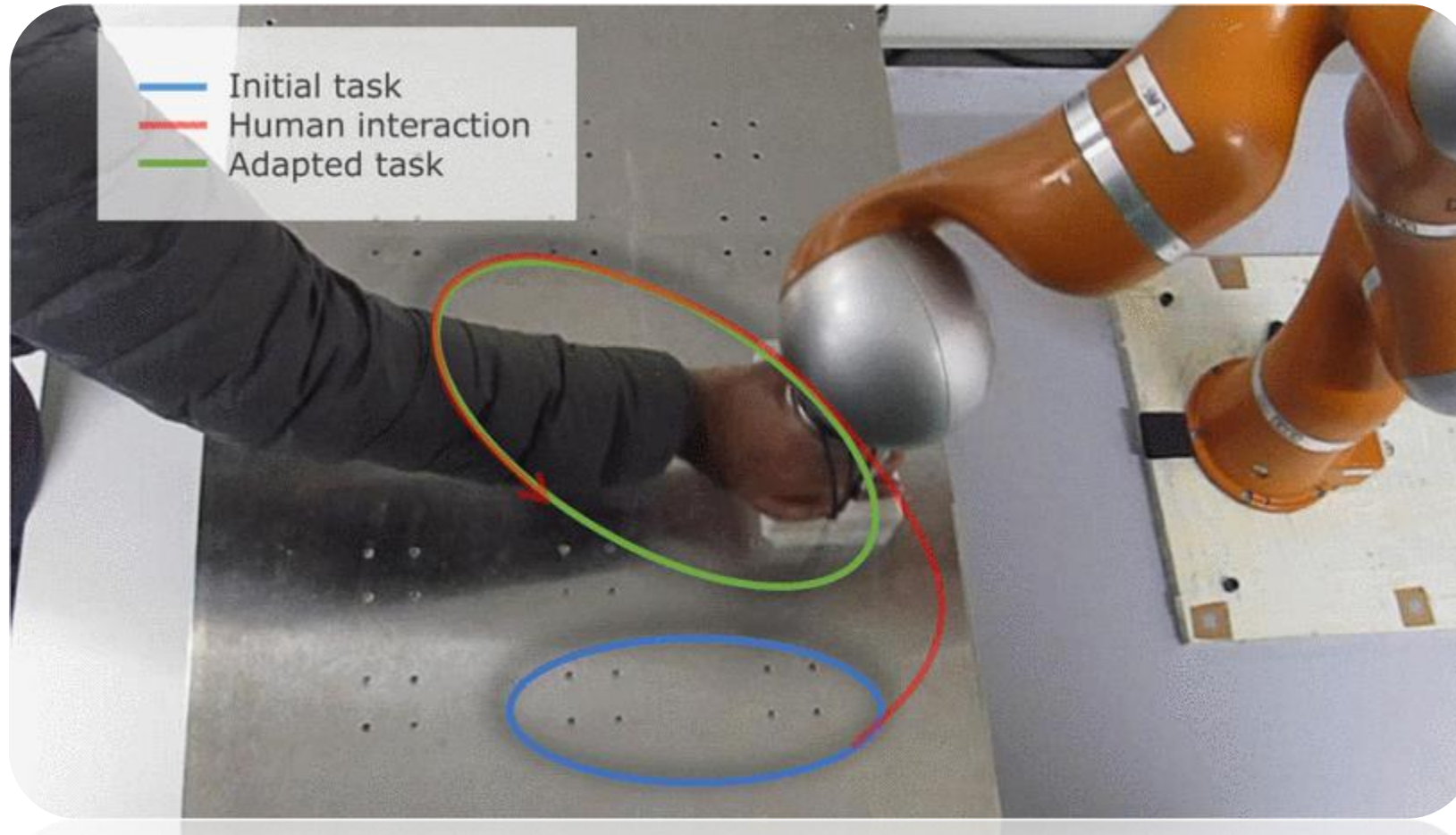
The human intends to place the box on either the right or left side.
The exact location for the placement is unknown to the robot.

Adapted
Task:

Robot's view
with four possible tasks:

- Move forward
- Move backward
- Place right
- Place left

Adapting DS Parameters on-line



Automatically adapt center and size of the limit cycle, based on human demonstration

Control law for robot motion

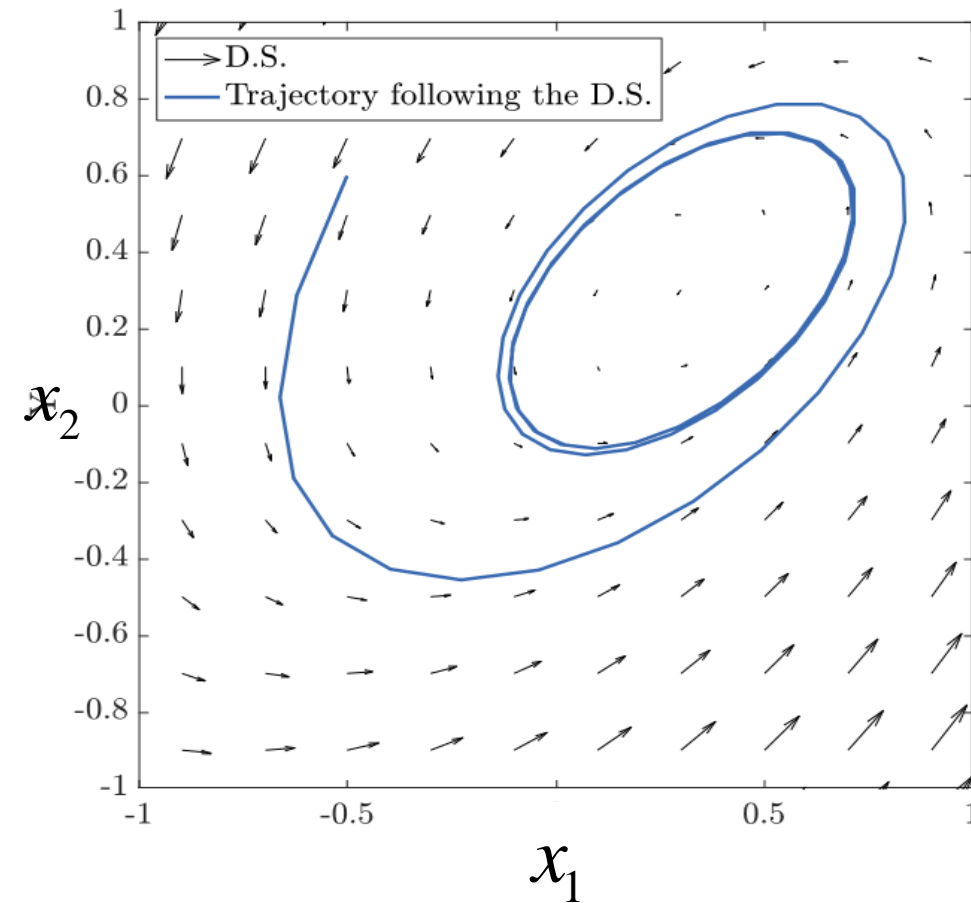
$$\dot{x} = f(x; \theta), \quad \theta: \text{parameters to be adapted}$$

Limit cycle in polar coordinates:

$$\dot{r} = -\alpha(r - r^*), \quad \alpha: \text{radial velocity}$$

$$\dot{\phi} = \omega, \quad \omega: \text{angular velocity}$$

α, ω : parameters to be adapted



Estimate parameters on-line as soon as a discrepancy is recorded

Error measure over K points measured every Δt :

$$e = \frac{1}{K} \sum_{k=0}^K \left(f(x(t-k\Delta t); \theta) - \dot{x}(t-k\Delta t) \right)$$

Departs from original vector field

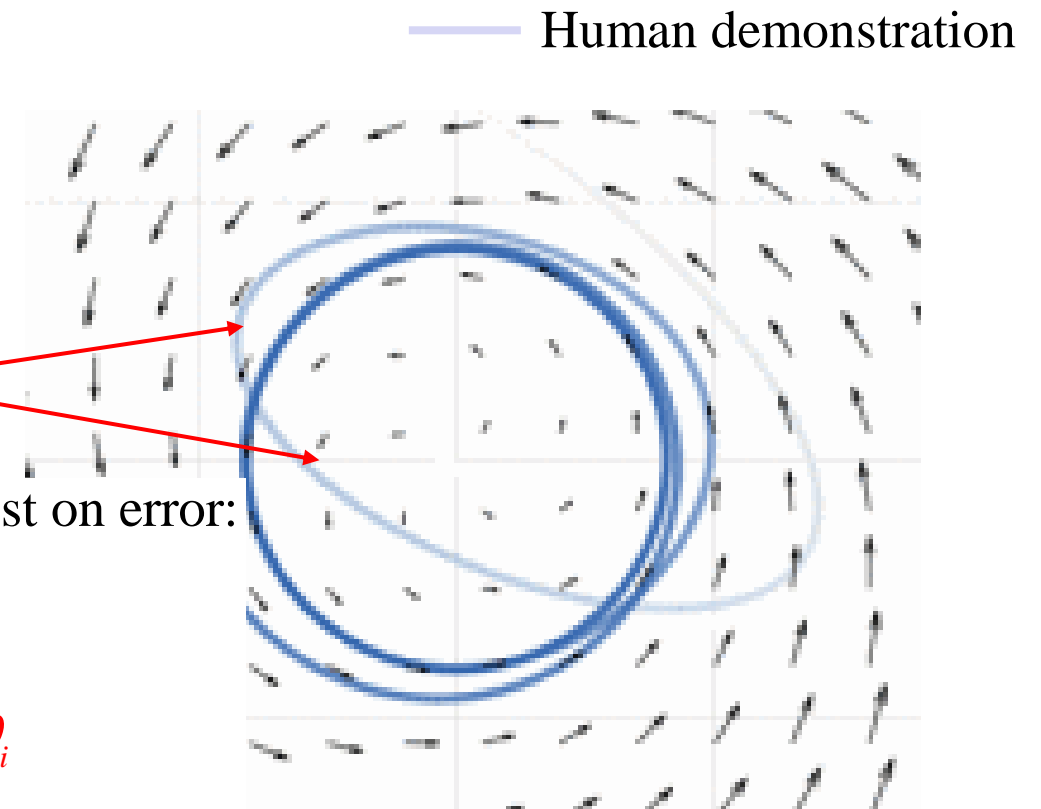
Update the parameters following gradient of the quadratic cost on error:

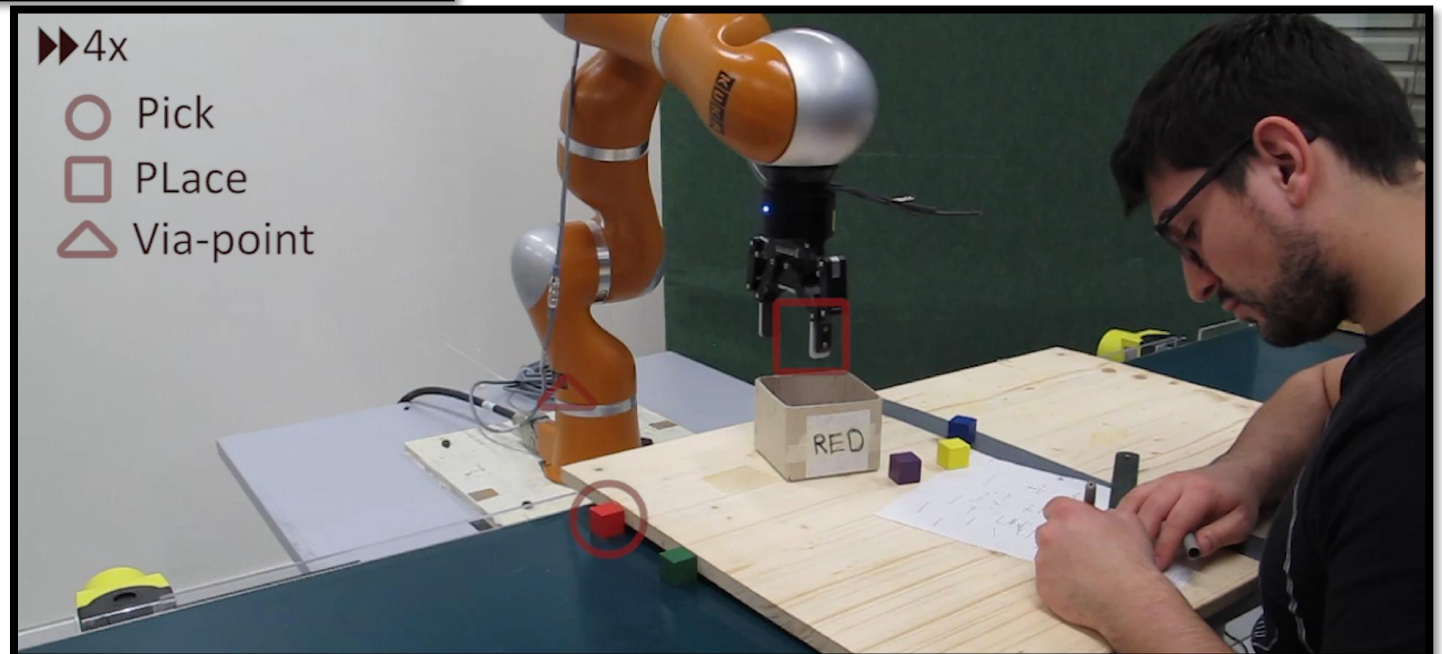
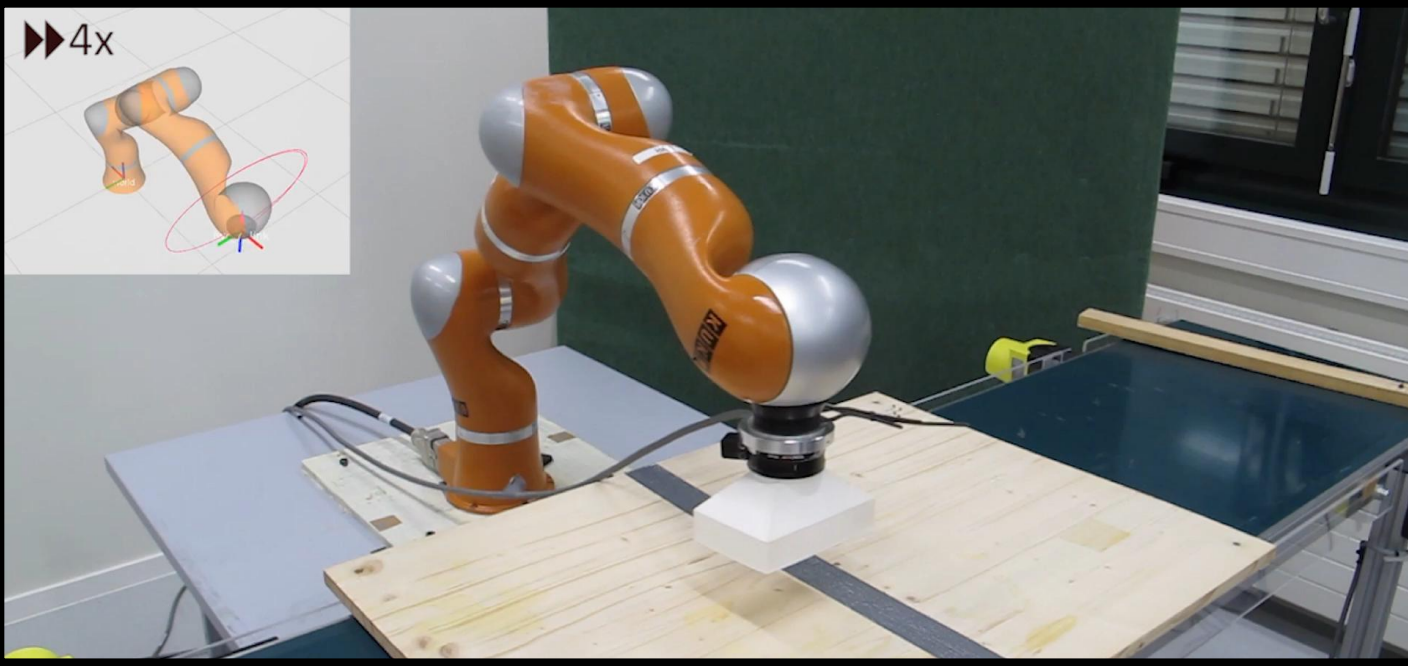
Quadratic cost: $J(\theta) = e(\theta)e^T(\theta)$

$$\frac{\partial J}{\partial \theta_i} = \frac{1}{K} e^T(\theta) \sum_{k=0}^{K-1} \frac{\partial f(x(t-k\Delta t); \theta)}{\partial \theta_i}; \text{ for each parameter } \theta_i$$

One can follow the same procedure to update the parameters of a linear DS: $\dot{x} = A(x - x^*)$.

Parameters to be learned are all elements of A and of x^* .





Khoramshahi, M., Laurens, A., Triquet, T. and Billard, A., 2018, October. From human physical interaction to online motion adaptation using parameterized dynamical systems. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1361-1366).

Summary

- ❑ Trajectory generation with DS is a powerful and **versatile** technique, allowing for the generation of a variety of behavior to control one or multiple robots.
 - ❑ They can be parameterized to introduce external dependencies such as dependency on force, or dependency to another dynamics for tracking moving targets.
- ❑ **Coupling DS** allows to generate temporal and spatial dependencies between different dynamics.
 - ❑ This can be applied to **control different robots** (or robot's parts) in **synchrony**.
 - ❑ When combined with the concept of **virtual dynamics**, it can force the system to track a desired path in space.
- ❑ We have seen methods to combine multiple DS
 - ❑ To switch at run time across different DS through comparison across observed and control dynamics
 - ❑ To re-estimate the parameters of a known DS (limit cycle, linear DS) online through simple gradient descent.