Learning from Demonstration in Robotic Surgery

Elite Robotics Summer School 2022

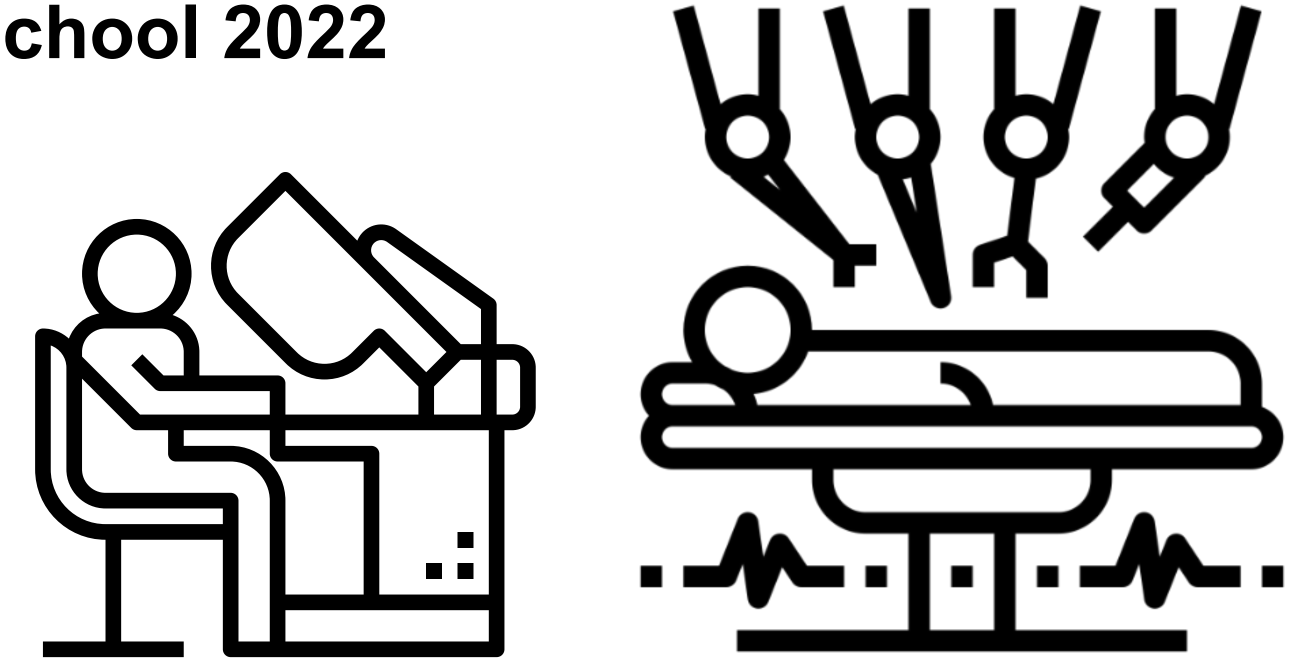
Iñigo Iturrate

Assistant Professor

Ø27-604-3

inju@mmmi.sdu.dk

SDU Robotics
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark

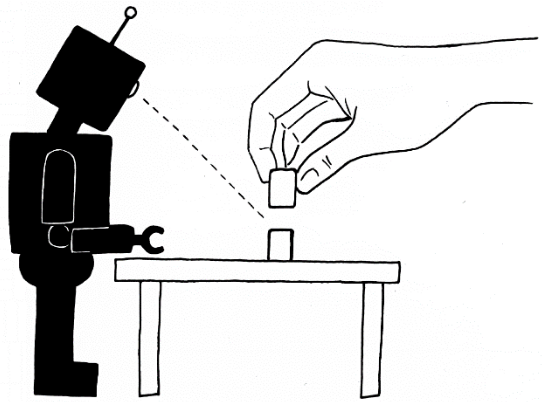


SDU

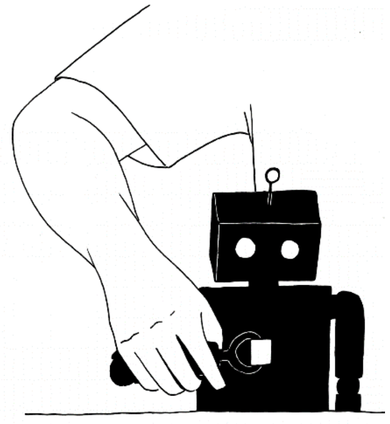
My Background

- MSc Robot Systems Engineering (SDU)
  - Student worker at Universal Robots for ~1 year
- Employed at Universal Robots (~4 years)
  - Software developer (½ year)
  - Industrial PhD with UR/SDU (3½ years)
- Postdoctoral Researcher at SDU (2019 – 2021)
- Assistant Professor at SDU (2021 – present)

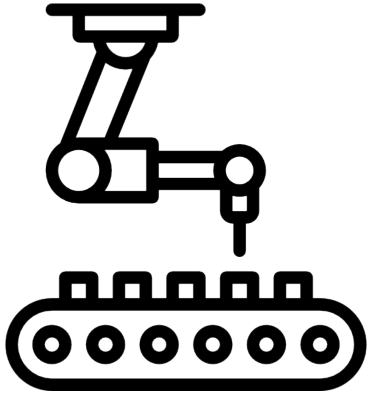
My Research



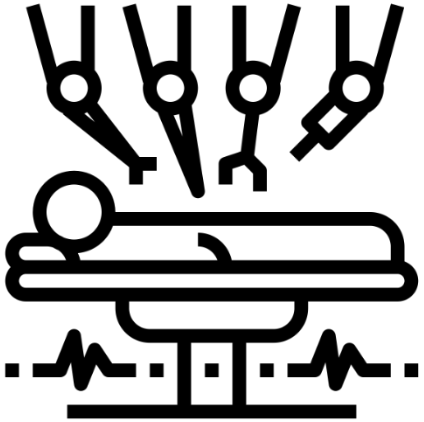
Robot Learning from Demonstration



Force Control for Human-Robot Interaction



Industrial Assembly



Surgical Robotics

Agenda

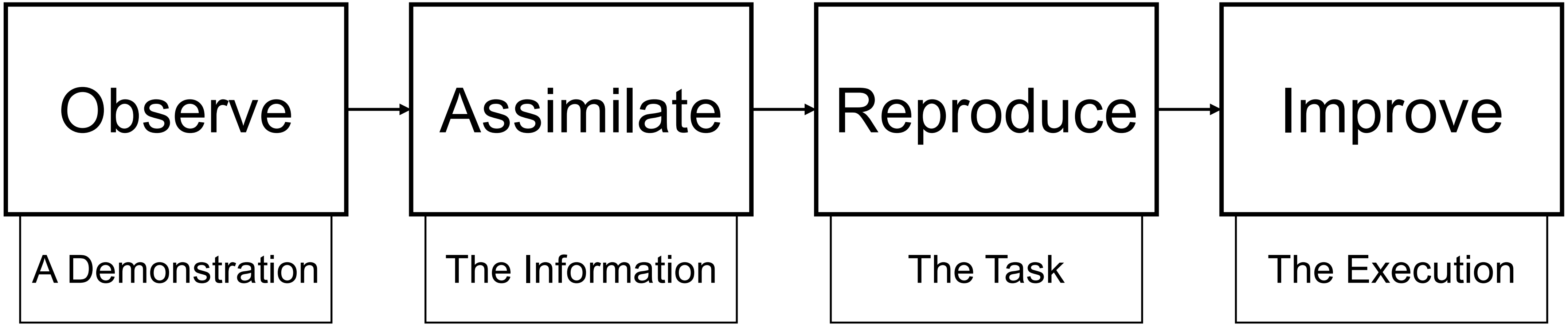
I. An Introduction to Learning from Demonstration

What is Learning from Demonstration?

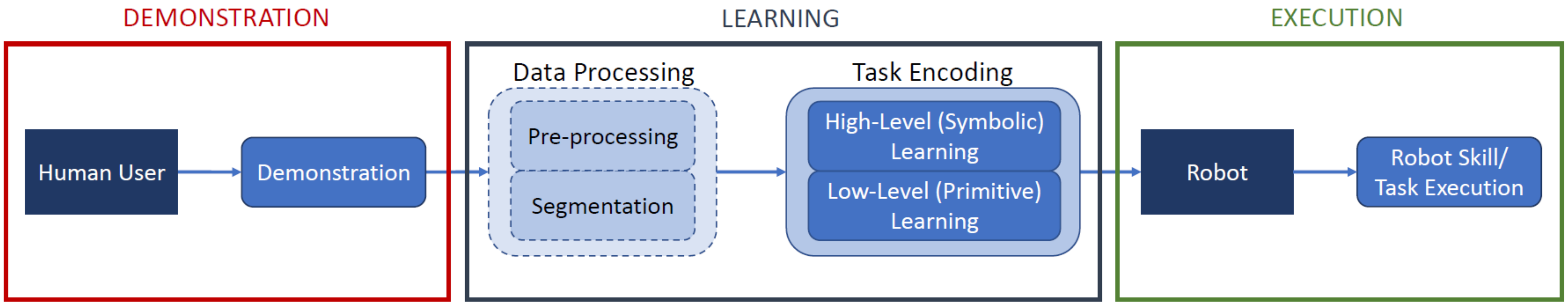
(or Learning by Imitation) [1]

What is Learning from Demonstration?

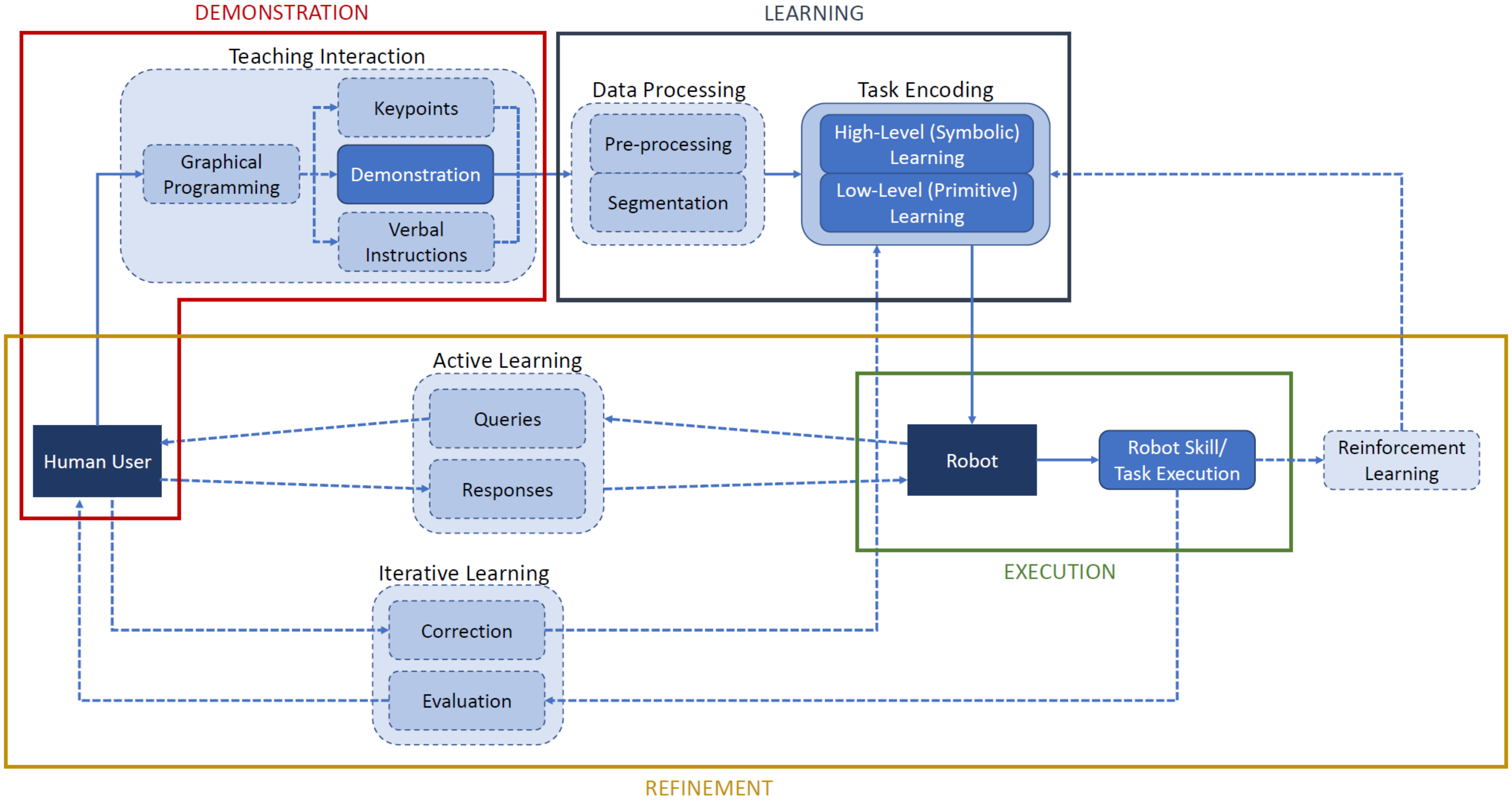
(or Learning by Imitation) [1]



Basic Process Flow

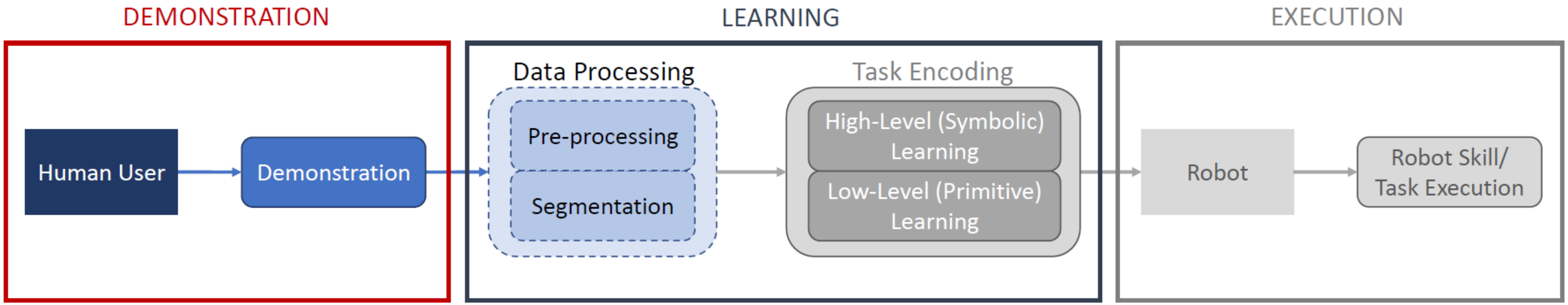


Expanded Process Flow



SDU

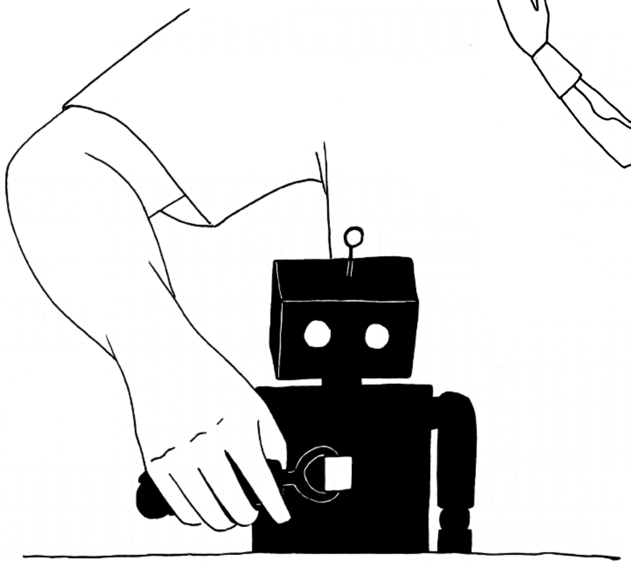
Part I: Demonstration Recording Phase



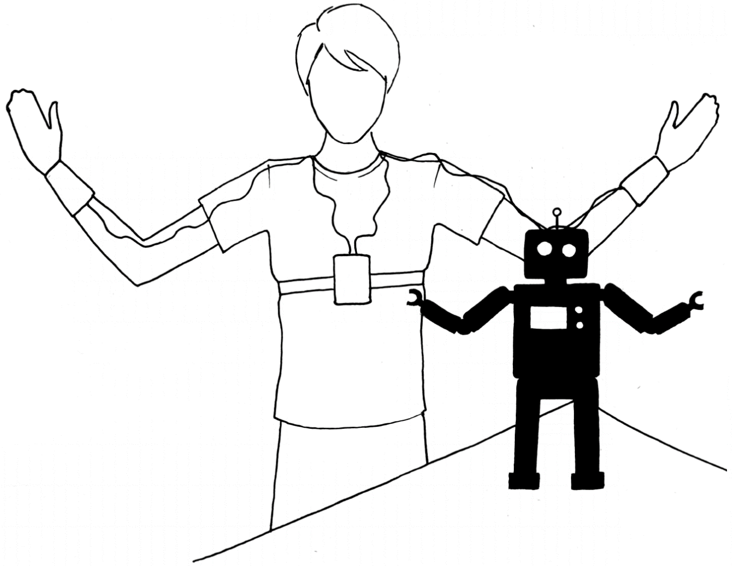
You have seen one way to record a demonstration.

Can you think of other ways?

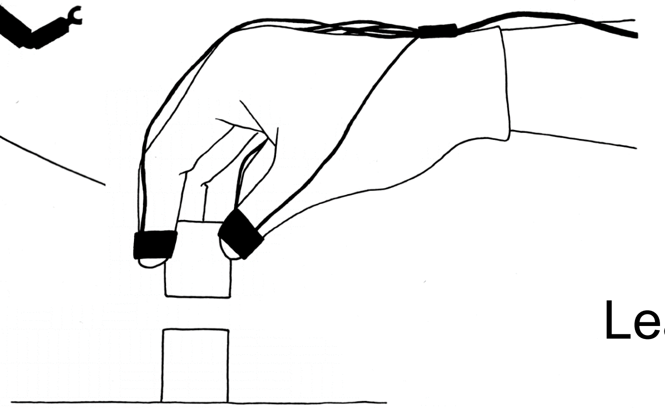
Demonstration Modalities



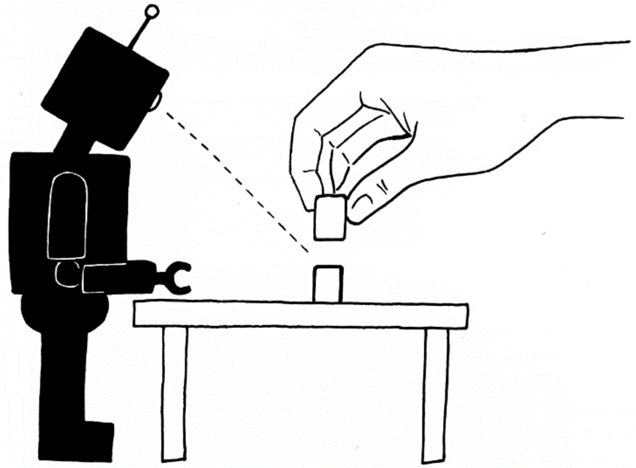
Kinesthetic Teaching



Shadowing



Sensors on Teacher



Learning from Observation

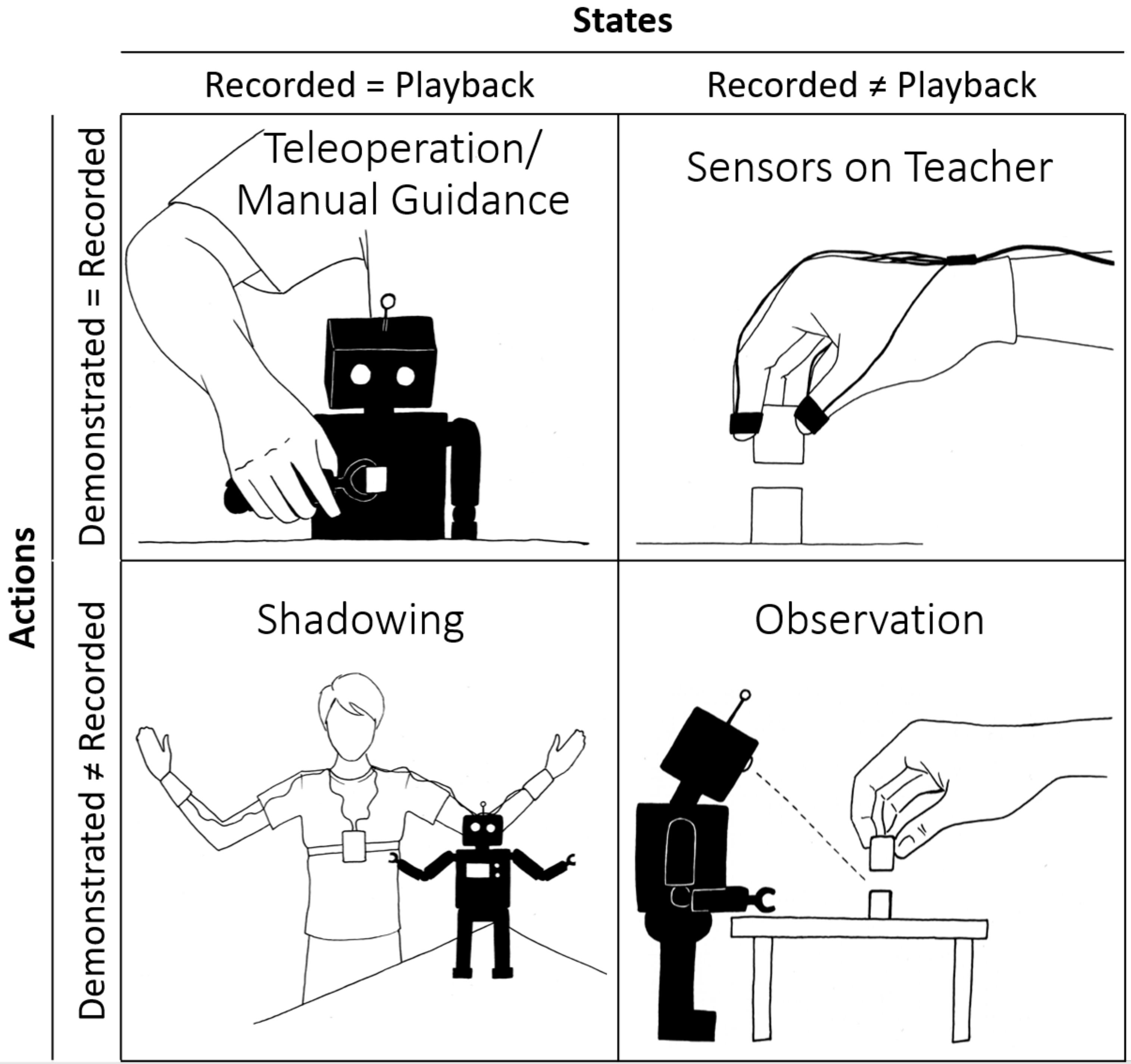
The Correspondence Problem

Sometimes, the **bodies** of the ***teacher*** and the ***learner*** will **not** be **the same**.

It is still possible to learn in this scenario by establishing a ***mapping*** or ***correspondence*** **between the two bodies**. [2]

This **introduces complexity** and the potential for **unreliability**.

SDU



Demonstration Modalities [3]

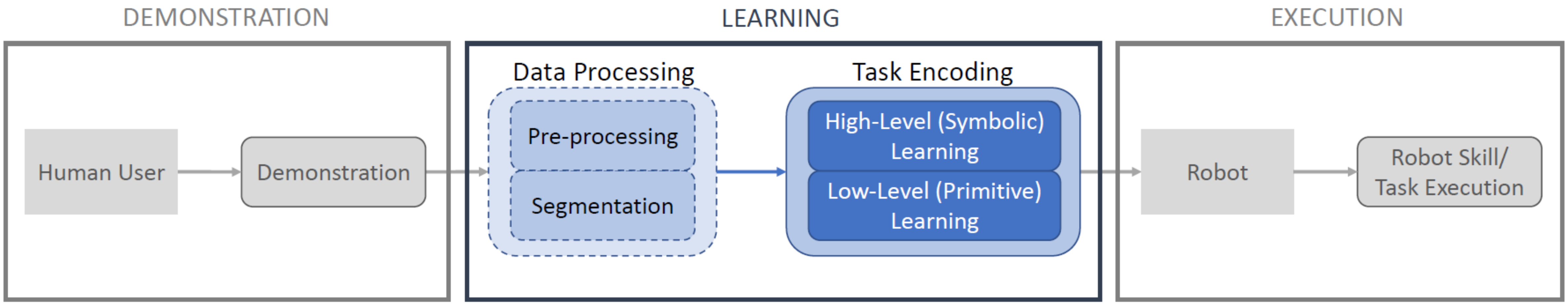
That’s all good… but what data should we record?

- **Joint positions** (velocities, accelerations).
- End-effector **Cartesian-space positions** (velocities, accelerations).
- End-effector **forces and torques**.
- **Object poses** (?)
- **Relative poses** (?)

SDU

II. Data Representations for Learning

Part II: Learning Phase



Learning at Different Levels of Abstraction

Learning can be performed at multiple levels:

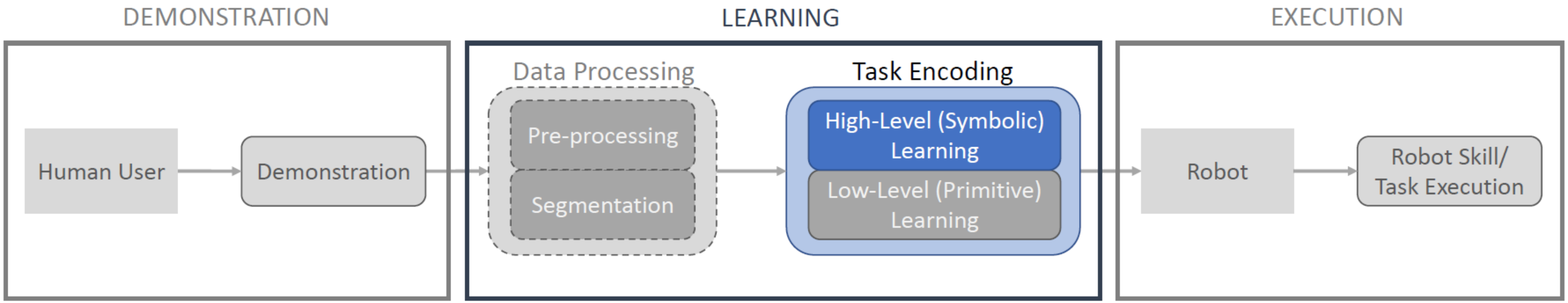
High-level:

- Learning task structure.
- Learning intentionality.
- Symbolic language representation.

Low-Level:

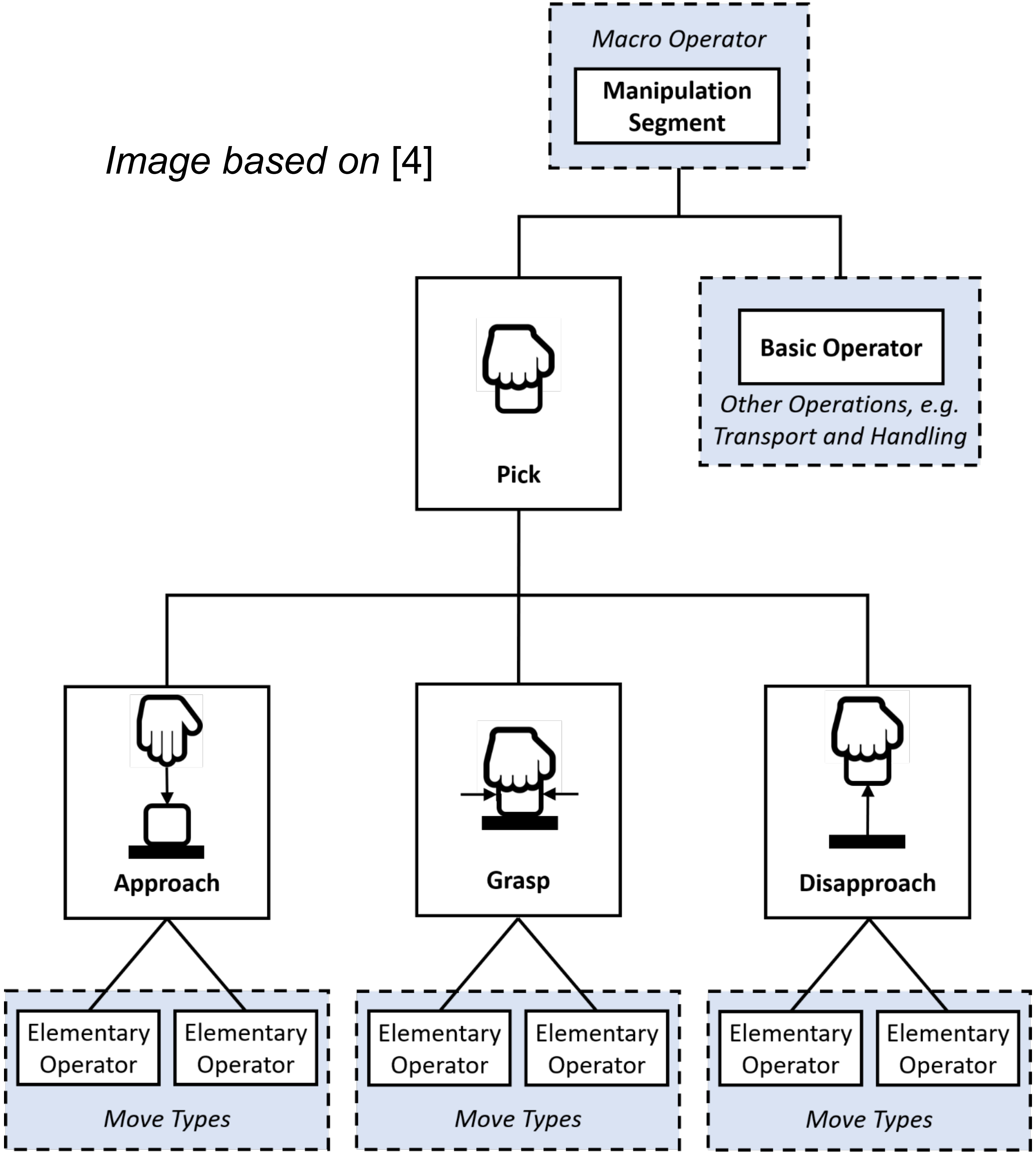
- Learning sensorimotor coupling.
- Learning reusable movement primitives and manipulation skills.
- Typically dynamical system or probabilistic representation.

High-Level Learning



SDU

Image based on [4]



High-level (Symbolic Abstraction) Learning

Focus on learning:

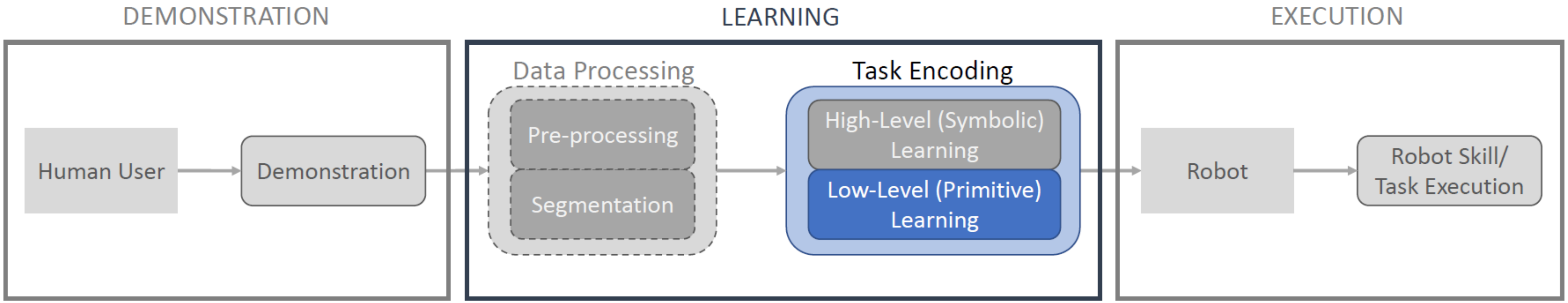
- Relationships between action and effects.
- Relationships between objects.
- Sequences of actions.

Highly task- and setup-dependent.

Typically requires many demonstrations.

SDU

Low-Level Learning



Low-level (Sensorimotor Primitive) Learning

Focus on learning:

- Relationships between sensor input and motor output (sensorimotor coupling).
- Reusable sets of primitive movements/skills.

Two general approaches:

- ~~**Probabilistic Models**~~
  ~~e.g. Gaussian Mixture Models and Regression, Probabilistic Movement Primitives, Kernelized Movement Primitives~~
- **Dynamical Systems**
  e.g. Dynamic Movement Primitives

We will not talk about this today

SDU

III. Dynamic Movement Primitives

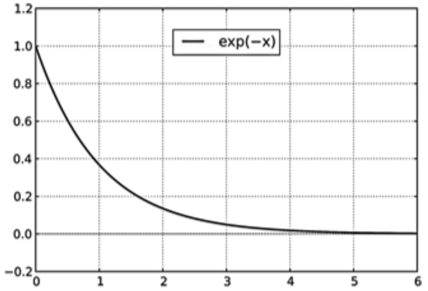
Dynamic Movement Primitives [5]

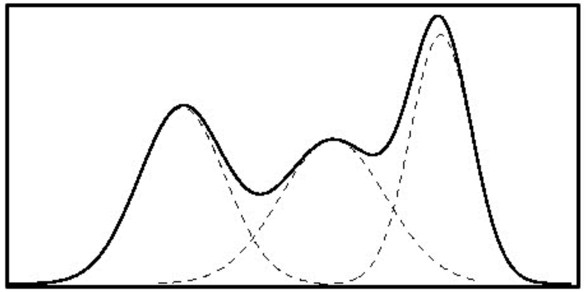
Trajectory Dynamics
Transformation System

$$\tau\dot{z} = K(g-y) - Dz + f(x)$$
$$\tau\dot{y} = z$$

Temporal Evolution
Phase System

$$\tau\dot{x} = -\alpha x$$

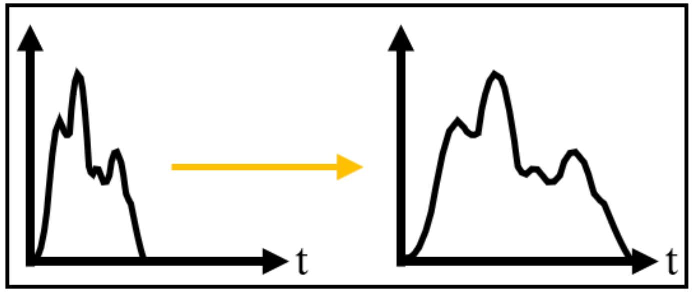




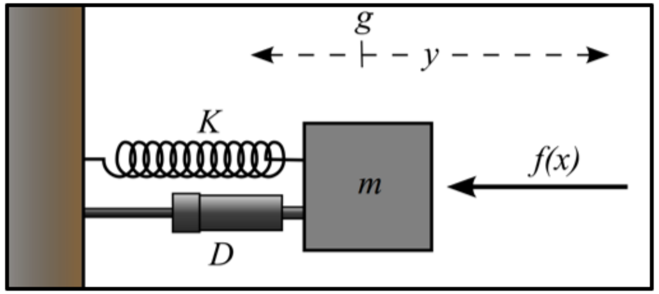
Shape Approximator
Forcing Term
(Learned from demonstration)

$$f(x) = \frac{\sum_{i=1}^{N} w_i \psi_i(x)}{\sum_{i=1}^{N} \psi_i(x)} x(g-y_0)$$

$$\psi_i(x) = e^{-\frac{(x-c_i)^2}{\sigma^2}}$$



Time Constant
Controls Duration of Movement



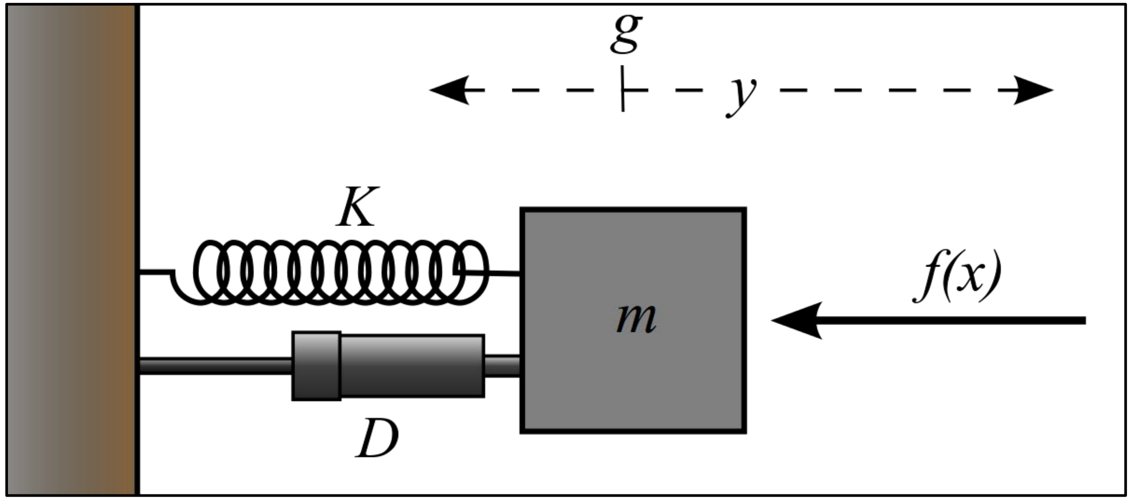
2nd Order Dynamical System
(Spring-damper system)

SDU

DMPs: Properties

- Convergence to goal.
- Time independence (autonomous system).
- Robustness to perturbation.
- Online trajectory (goal and time) modulation.
- Ability to incorporate coupling terms.
- Easy to learn (linear in its parameters).
  - Can learn from a single demonstration.
- Invariant in scale and time.

DMPs: Transformation System



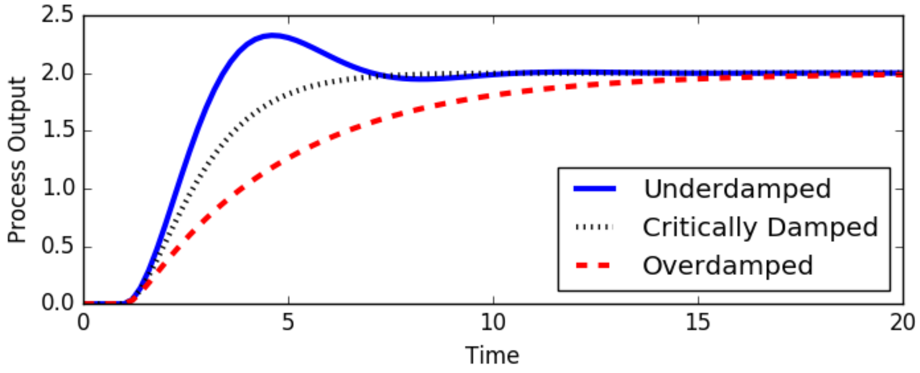
Second-order Spring-mass-damper system:

$$\tau^2 \ddot{y} = K(g - y) - D\tau\dot{y}$$

Typically written as:

$$\tau\dot{z} = \alpha_y\left(\beta_y(g - y) - z\right)$$
$$\tau\dot{y} = z$$

The parameters α_y and β_y are chosen in order to have a critically-damped system: $\beta_y = \frac{\alpha_y}{4}$



We can then use any integration scheme, e.g. simple forward Euler, to get velocity and position targets:

$$\dot{y}(t + \Delta t) = \dot{y}(t) + \ddot{y}(t + \Delta t)\Delta t$$
$$y(t + \Delta t) = y(t) + \dot{y}(t + \Delta t)\Delta t$$

DMPs: Forcing Term

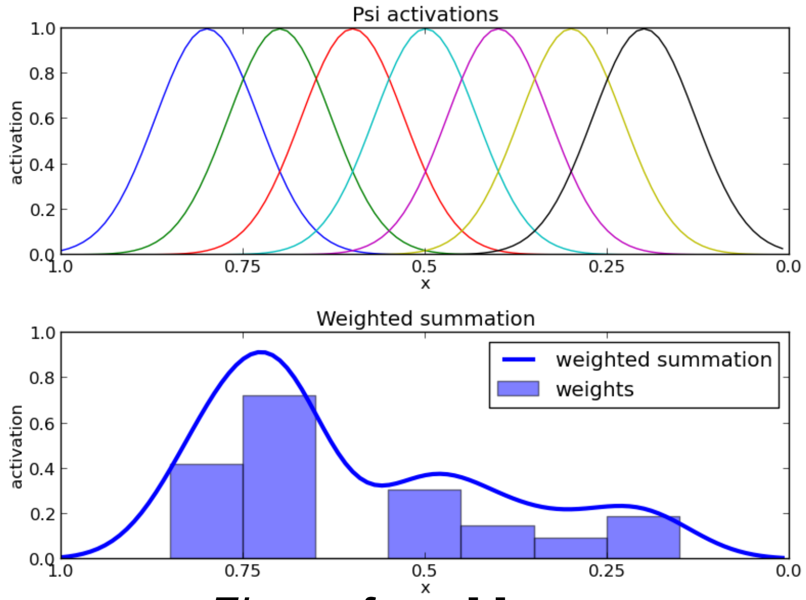


Figure from [-]

Weighted mixture of Gaussian Basis Functions:

$$f(x) = \frac{\sum_{i=1}^{N} w_i \psi_i(x)}{\sum_{i=1}^{N} \psi_i(x)} x(g - y_0)$$

Each function defined as:

Gaussian BF

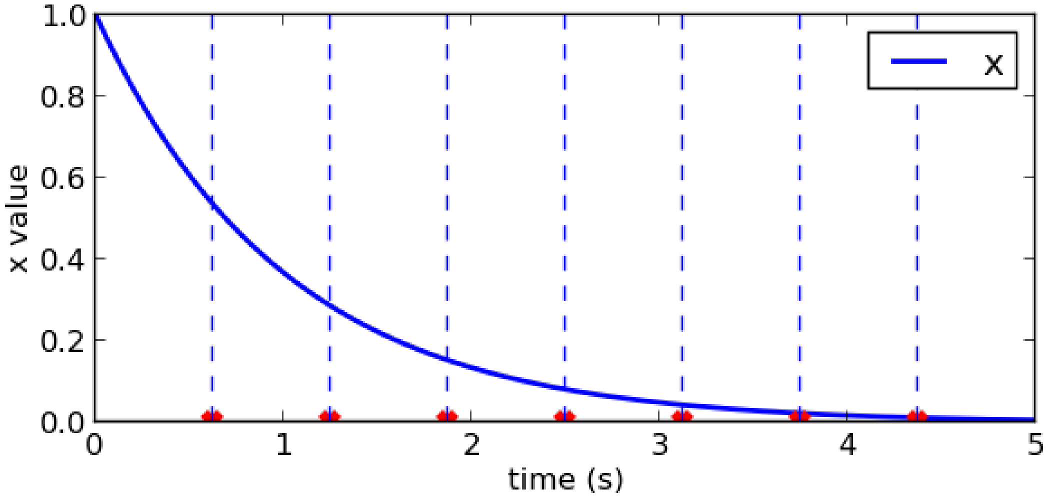
$$\psi_i(x) = e^{-\frac{(x-c_i)^2}{\sigma^2}}$$

Notice that:

- We can change the number of components, N
- Each Gaussian BF is located at a center, c_i.
- Each Gaussian BF has width-affecting variance, σ.
- The forcing term $f(x)$ is a function of the phase, x, and not time.

SDU

DMPs: Canonical (Phase) System



First order exponential decay:

$$\tau \dot{x} = -\alpha x$$

With: $x_0 = 1, \lim_{t \to \infty} x = 0$

Controls:

- The **temporal evolution** of the forcing term
- **Gating** of the forcing term as $t \to \infty$
- **Synchronization** across DOF.

Notice that:

- The position of the centers of the BF, c_i, are placed in x, not t.
- We want a **linear spacing in t.**
  - We can solve $\tau \dot{x} = -\alpha x$ to find a spacing that is linear in t.

SDU

DMPs: Learning (I)

Assuming we have a *demonstration* consisting of: $y_{\text{demo}}, \dot{y}_{\text{demo}}, \ddot{y}_{\text{demo}}$ for a duration τ_{demo}.

1. Ensure that CS converges ”close enough” to 0 at the end of the movement, when $\mathrm{t} = \tau$, i.e. calulate α typically for 99% convergence, based on:

$$\tau \dot{x} = -\alpha x \rightarrow x(t) = C_0 e^{-\dfrac{\alpha t}{\tau}}$$

$$x(t = \tau) = 0.01 = e^{-\alpha}$$

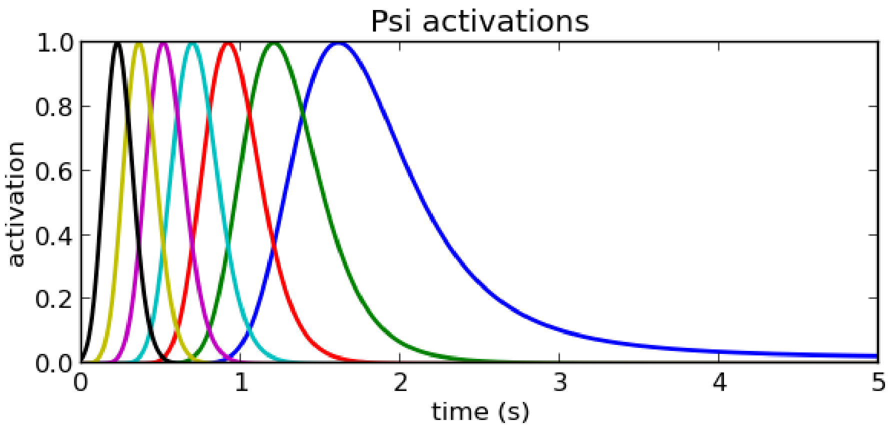
$$\alpha = -\ln 0.01$$

SDU

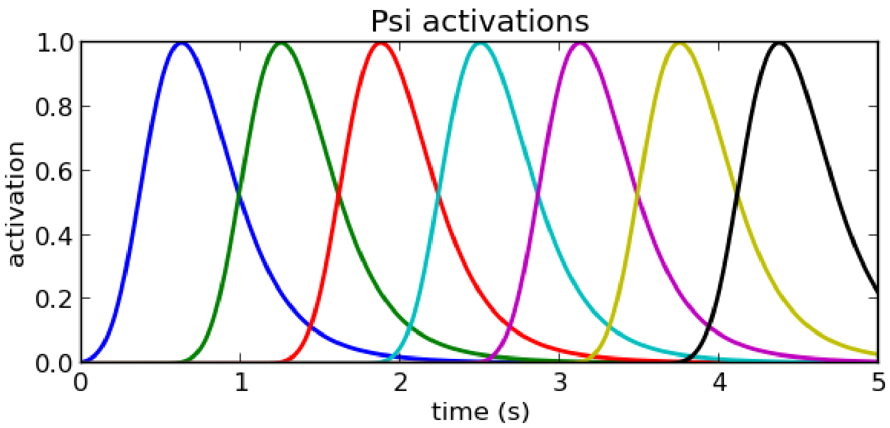
DMPs: Learning (II)

2. Choose N, σ, c_i for the FT. Usually:
   - The lower N the ”smoother” the movement, but it will also lose important features. $N = 50$ is a good starting point.
   - σ is usually found through trial and error, with a typical value being $\sigma = \frac{1}{0.025}$
   - The centers of the BFs, c_i, should be **spaced linearly in t not in x**.

Spacing Linear in x



Spacing Linear in t



SDU

DMPs: Learning (III)

3. Our desired forcing term is:

$$f_{\text{target}} = \tau_{\text{demo}}^2 \ddot{y}_{\text{demo}} - \alpha_y \left(\beta_y(g - y_{\text{demo}}) - \tau_{\text{demo}}\dot{y}_{\text{demo}}\right)$$

   We can then minimize:

$$J_i = \sum_{t=1}^{P} \psi_i(t) \left(f_{\text{target}}(t) - w_i \xi()t)\right)^2 \quad \text{with} \quad \xi(t) = x(t)(g - y_0)$$

   Which is a least-squares locally-weighted linear regression (LWLR) problem with solution:

$$w_i = \frac{\mathbf{s}^T \mathbf{\Gamma_i} \mathbf{f}_{\text{target}}}{\mathbf{s}^T \mathbf{\Gamma_i} \mathbf{s}}$$

   where

$$\mathbf{s} = \begin{pmatrix} \xi(1) \\ \xi(2) \\ \dots \\ \xi(3) \end{pmatrix} \qquad \mathbf{\Gamma}_i = \begin{pmatrix} \psi_i(1) & \dots & & 0 \\ \vdots & \psi_i(2) & & \vdots \\ & & \ddots & \\ 0 & \dots & & \psi_i(P) \end{pmatrix} \qquad \mathbf{f}_{\text{target}} = \begin{pmatrix} f_{\text{target}}(1) \\ f_{\text{target}}(2) \\ \dots \\ f_{\text{target}}(P) \end{pmatrix}$$

SDU

DMPs: Cartesian-space Orientation [6]

Take the same overall ideas as positional DMPs, but:
Represent orientation as unit quaternions:

$$\tau\dot{\boldsymbol{\eta}} = \alpha_z\left(\beta_z 2\log(\mathbf{g}_o * \bar{\mathbf{q}})) - \boldsymbol{\eta}\right) + \mathbf{f}_o(x)$$

$$\tau\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\eta} * \mathbf{q}$$

$$\mathbf{f}_o(x) = \frac{\sum_{k=1}^{N}\mathbf{w}_k^o\boldsymbol{\psi}_k(x)}{\boldsymbol{\psi}_k(x)}x$$

Quaternion Logarithm & Exponential Maps

$$\log : \mathrm{S}^3 \mapsto \mathbb{R}^3$$

$$\log(\mathbf{q}) = \log(v + \mathbf{u}) = \begin{cases} \arccos(v)\frac{\mathbf{u}}{\|\mathbf{u}\|}, & \text{if } \mathbf{u} \neq 0 \\ [0,0,0]^\mathsf{T}, & \text{otherwise} \end{cases}$$

$$\exp : \mathbb{R}^3 \mapsto \mathrm{S}^3$$

$$\exp(\mathbf{r}) = \begin{cases} \cos(\|\mathbf{r}\|) + \frac{\mathbf{r}}{\|\mathbf{r}\|}\sin(\|\mathbf{r}\|), & \text{if } \mathbf{r} \neq 0 \\ 1 + [0,0,0]^\mathsf{T}, & \text{otherwise} \end{cases}$$

Where $\boldsymbol{\eta} = [0, \boldsymbol{\omega}]$ is a quaternion with real part 0 and vector part equal to an angular velocity, $\boldsymbol{\omega}$.

We can then integrate: $\boldsymbol{\eta}(t + \Delta t) = \boldsymbol{\eta}(t) + \dot{\boldsymbol{\eta}}(t + \Delta t)\Delta t$

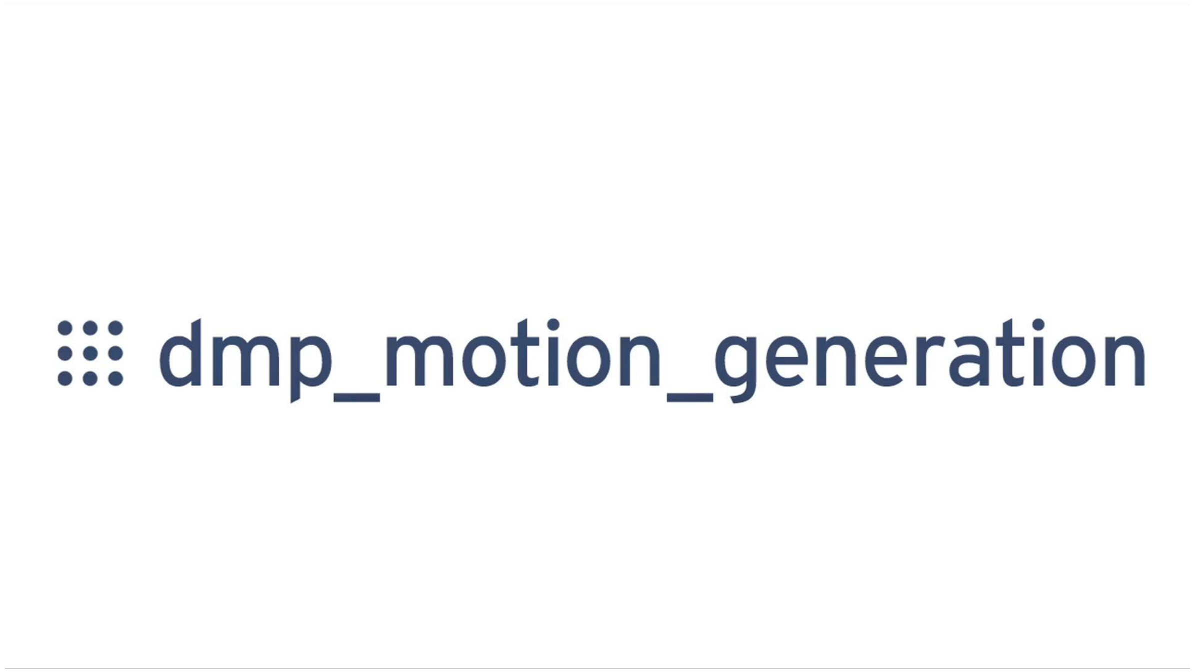
From $\boldsymbol{\eta}$, a new quaternion can be integrated using the exponential map:

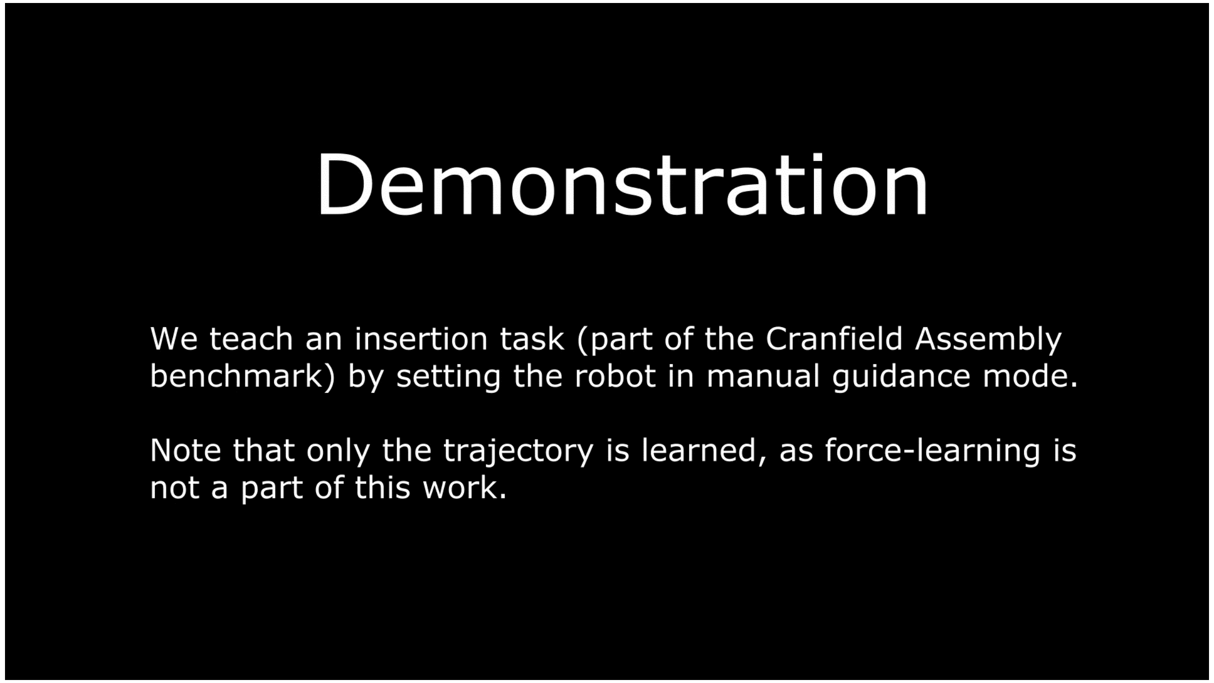
$$\mathbf{q}(t + \Delta t) = \exp\left(\frac{\Delta t}{2}\frac{\boldsymbol{\eta}(t)}{\tau}\right) * \mathbf{q}(t)$$

The phase and basis functions are the same as for the positional DMP.

SDU

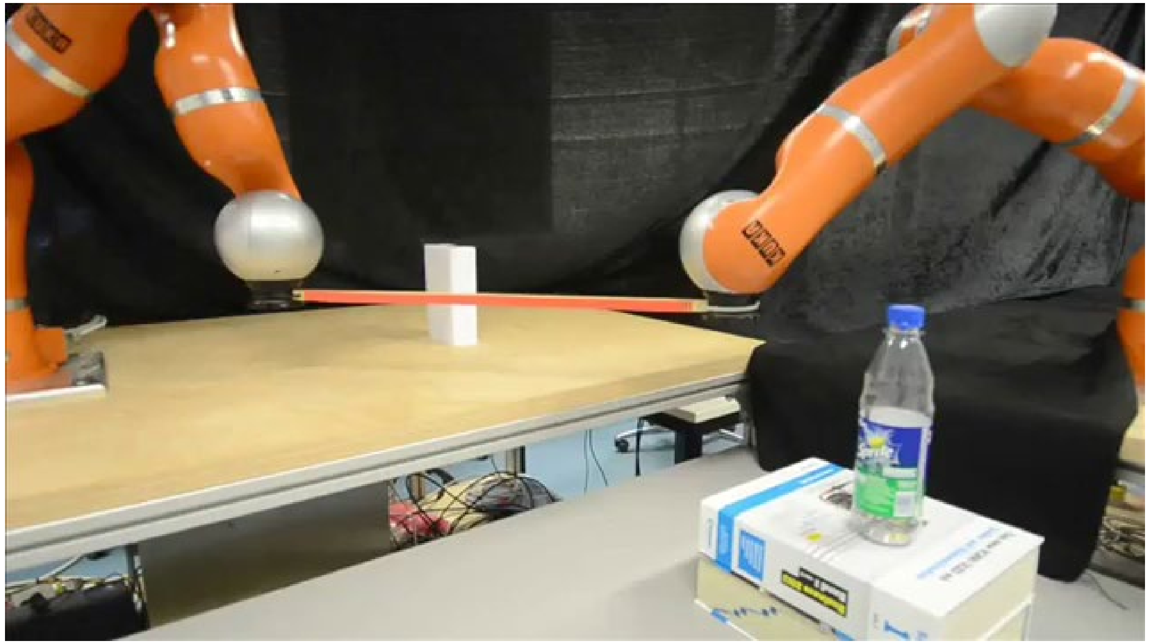
Examples (I)





Video from [7]

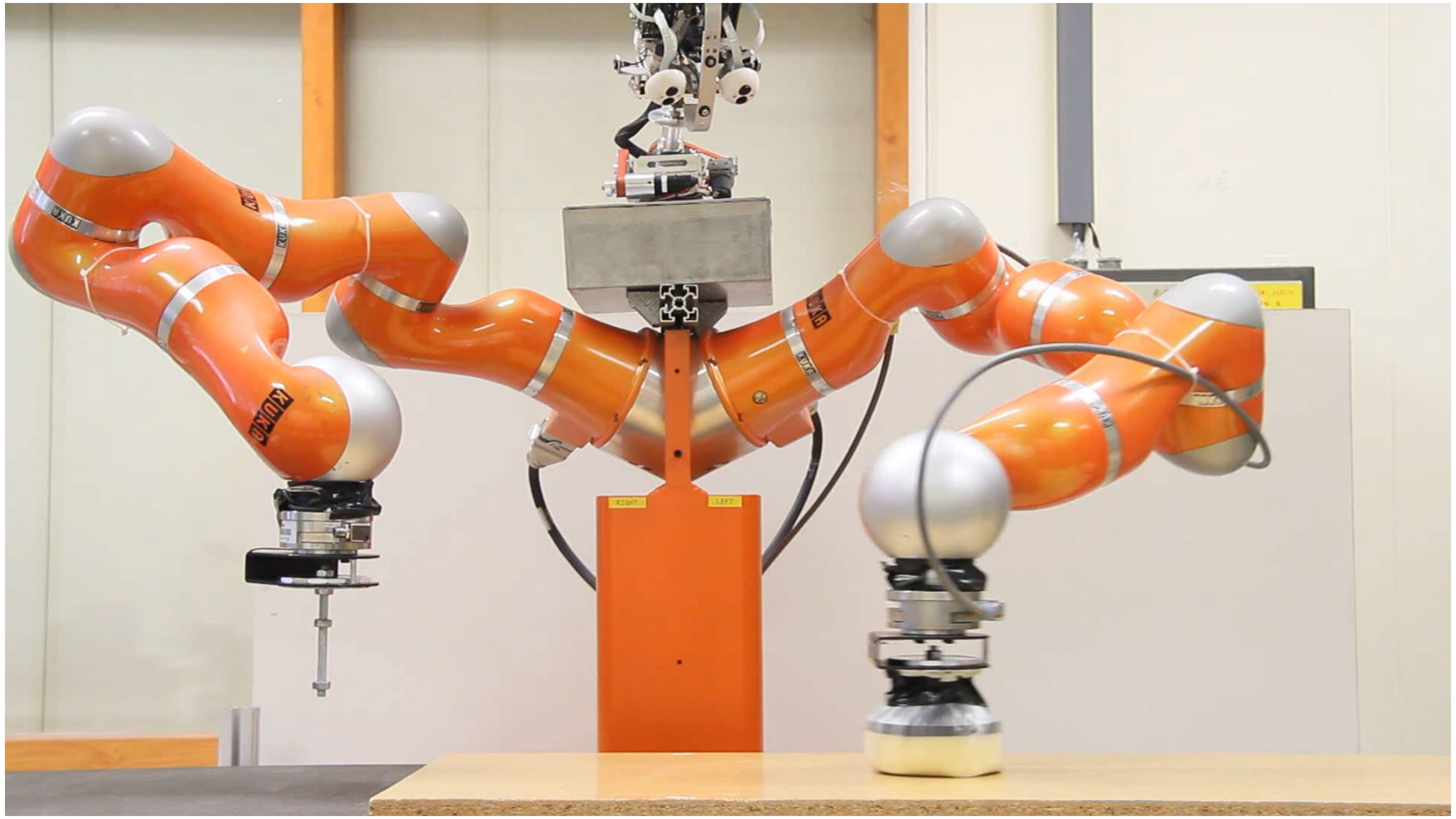
Examples (II)



https://www.youtube.com/watch?v=CDJwEAtEOR0

https://www.youtube.com/watch?v=W_gxLKSsSIE&t=37s

Examples (III) [8]



SDU

Examples (IV) [9]

IntellAct

Online Monitoring and Execution

Thiusius R. Savarimuthu, Anders G. Buch, Wail Mustafa, Yang Yang, Eren Aksoy, Jeremie Papon, Simon Haller, David Martinez, Aljaz Kramberger, Bojan Nemec

IntellAct (2011-2014): Intelligent Observation and Execution of Actions and Manipulations

SDU

IV. Learning Robotic Suturing from Demonstration

Why Learn from Demonstration in Robotic Surgery?

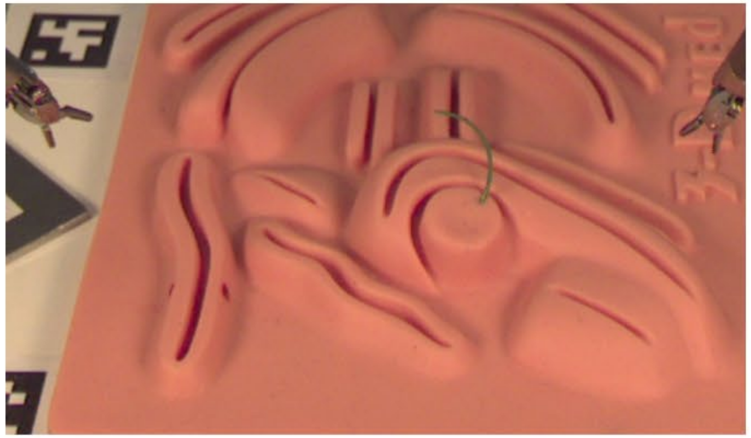
- The robots are naturally **teleoperated**.
- Data of surgical procedures is **recorded**.
- **Multimodal data** is available:
  - Kinematics of all arms
  - States of graspers/instruments
  - Endoscopic camera images

Image from Intuitive Surgical, Inc.

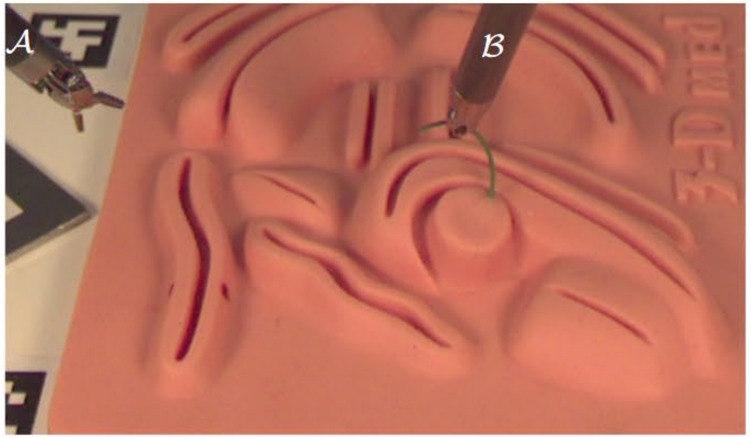
Autonomous Bi-Manual Surgical Suturing Based on Skills Learned from Demonstration [10]

Kim L. Schwaner, Iñigo Iturrate, Jakob K. H. Andersen, Pernille T. Jensen and Thiusius R. Savarimuthu

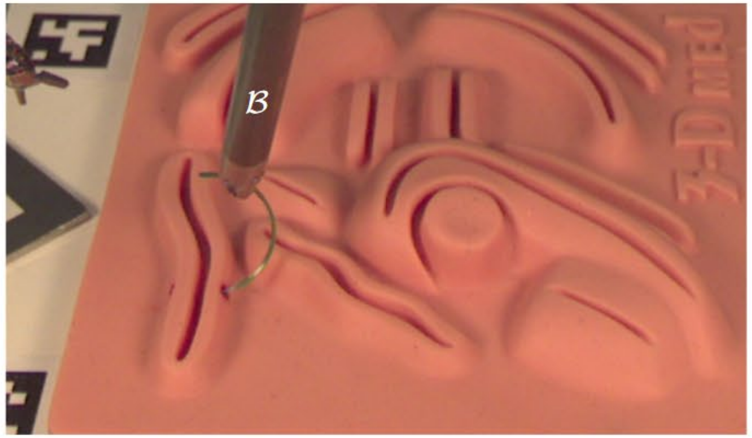
The Surgical Suturing Task [10]



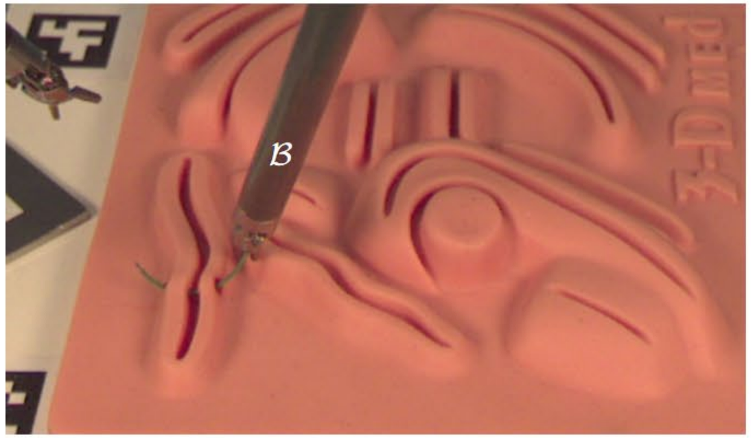
(a) Initial condition.



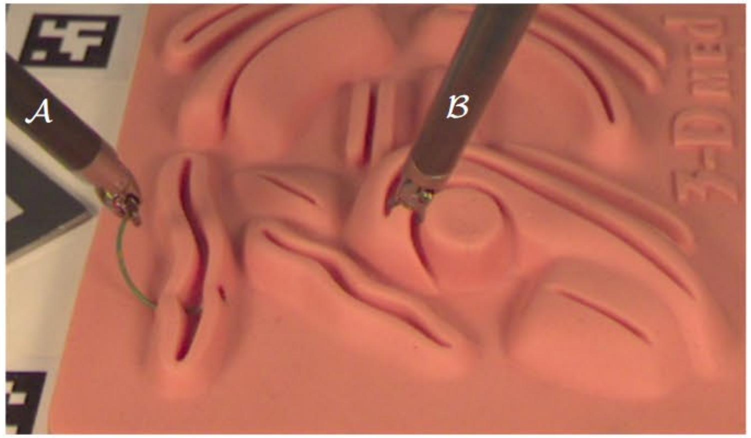
(b) Pick up needle (subtask I).



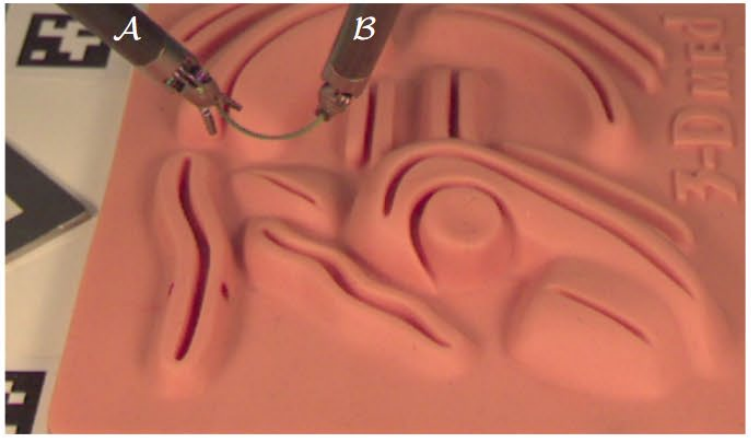
(c) Position needle for insertion (subtask II).



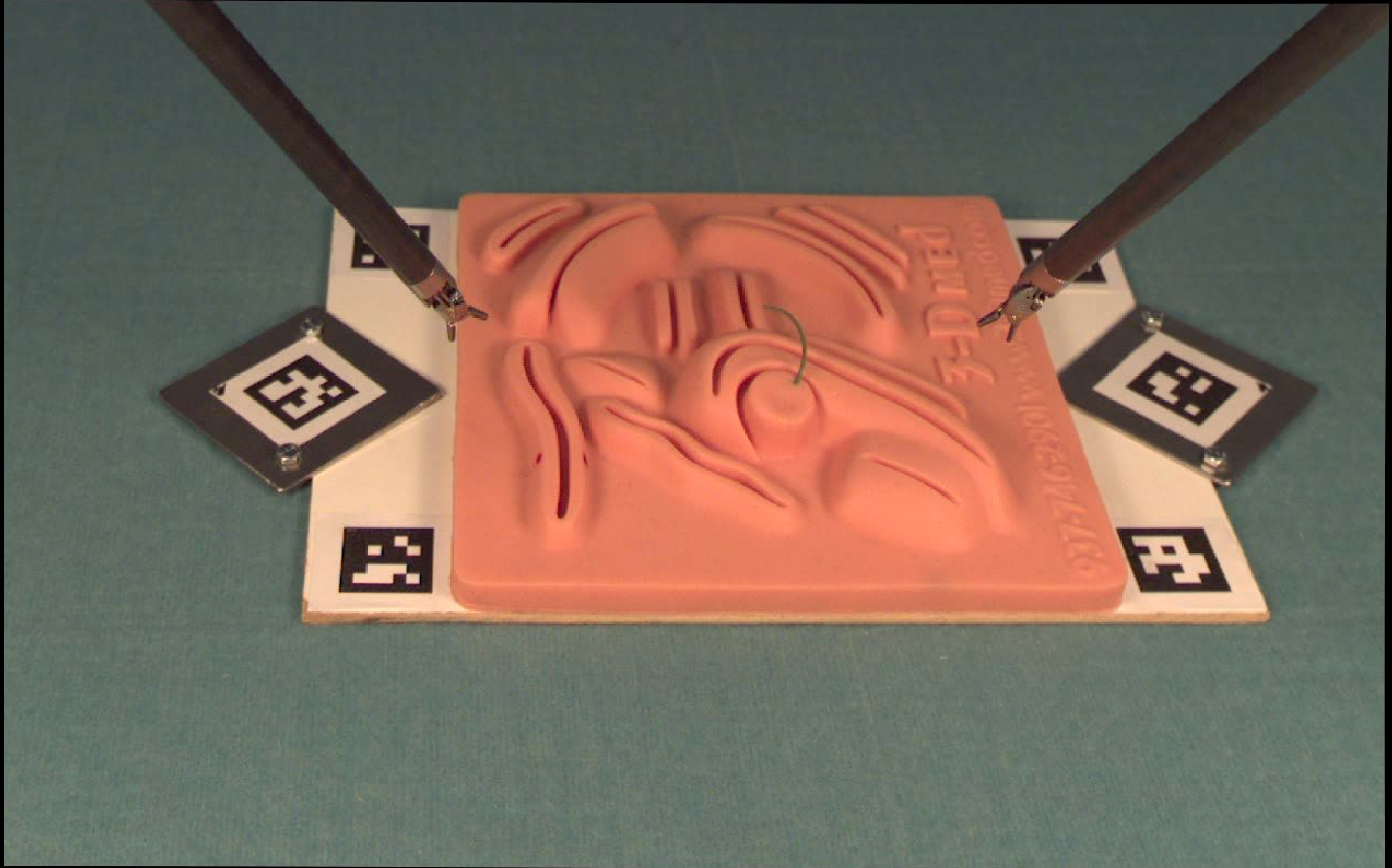
(d) Insert needle, initiating the throw (subtask III).



(e) Extract needle, completing the throw (subtask IV).



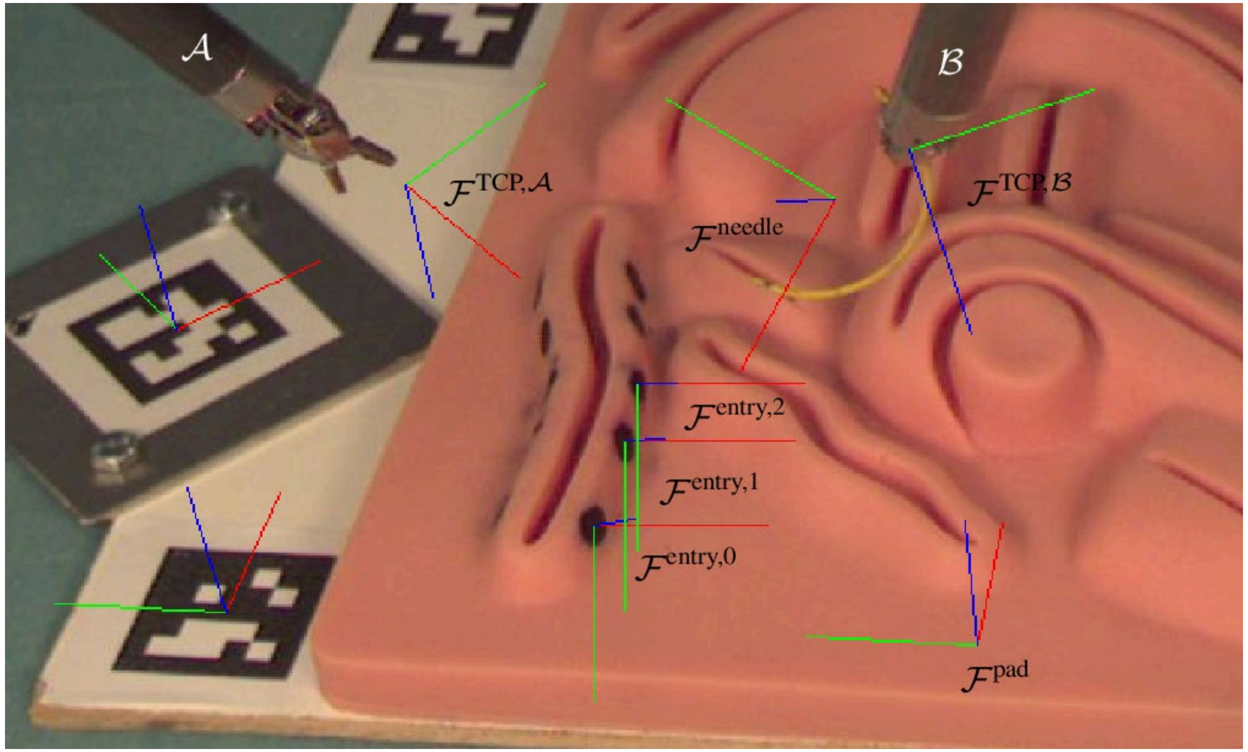
(f) Hand over needle from $\mathcal{A}$ to $\mathcal{B}$ (subtask V).



Computer Vision-based Tracking [10]

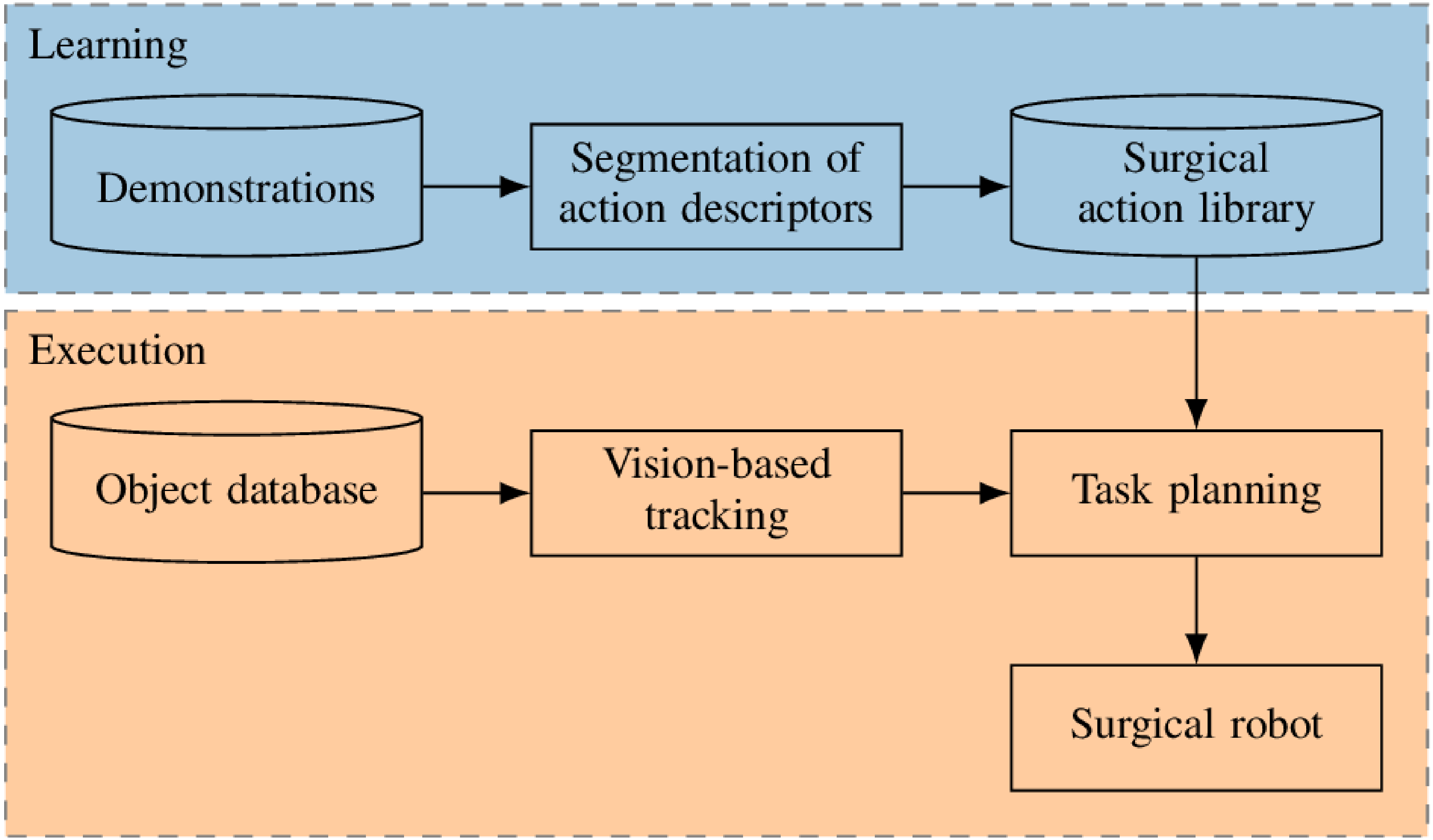
- **Hardware:** Basler acA2500-20gc GigE cameras with TamronM111FM08 8 mm lenses

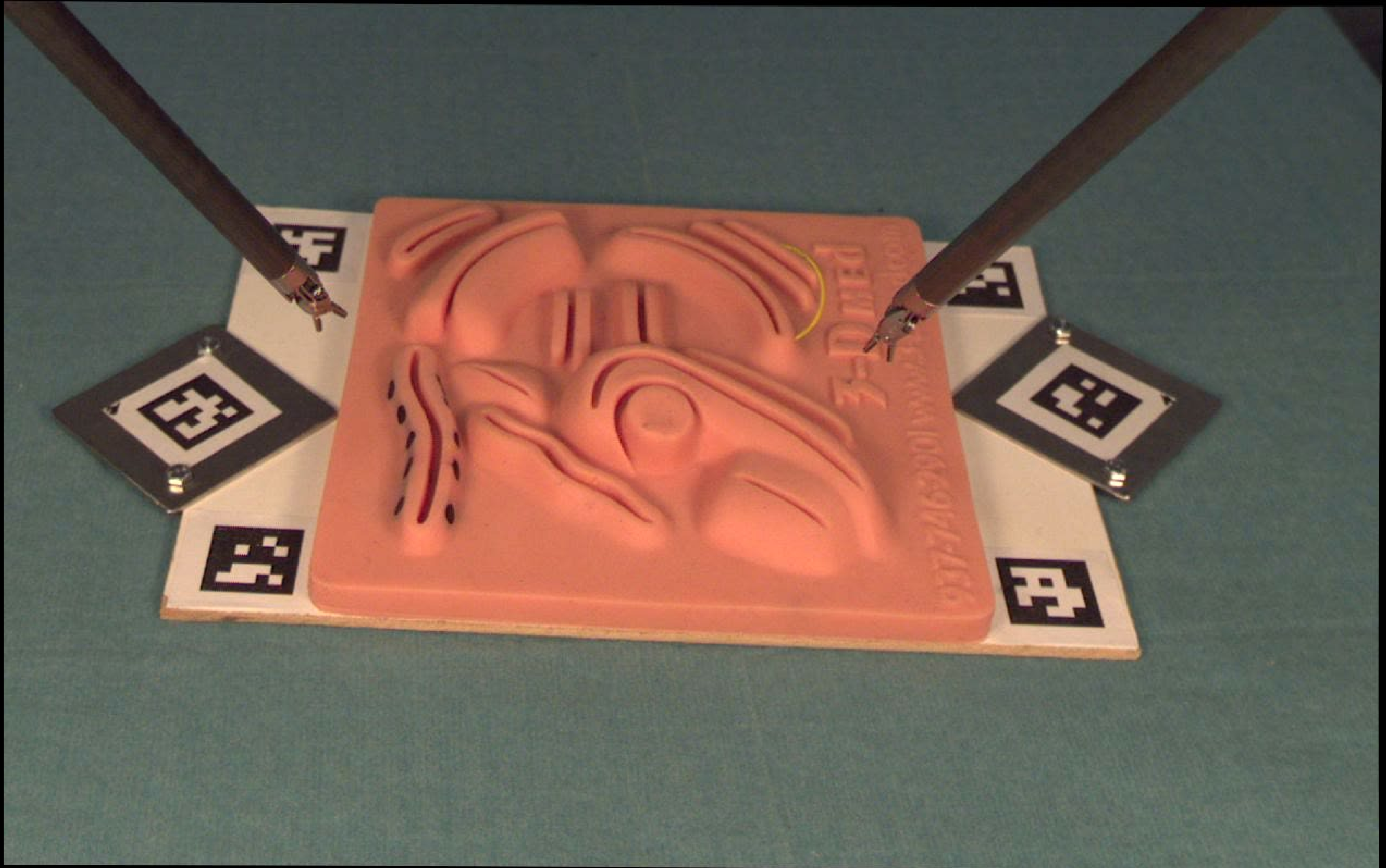
- **Algorithms**:
  - **Suture pad**: Pose estimation based on ArUco markers fixed w.r.t. suture pad known frames.
  - **Needle entry points**: Centroids of contours sorted w.r.t. suture pad frame.
  - **Needle segmentation**: HSV color segmentation, fit ellipses, 3D reconstruction by triangulation and least-squares 3D plane fitting.



SDU

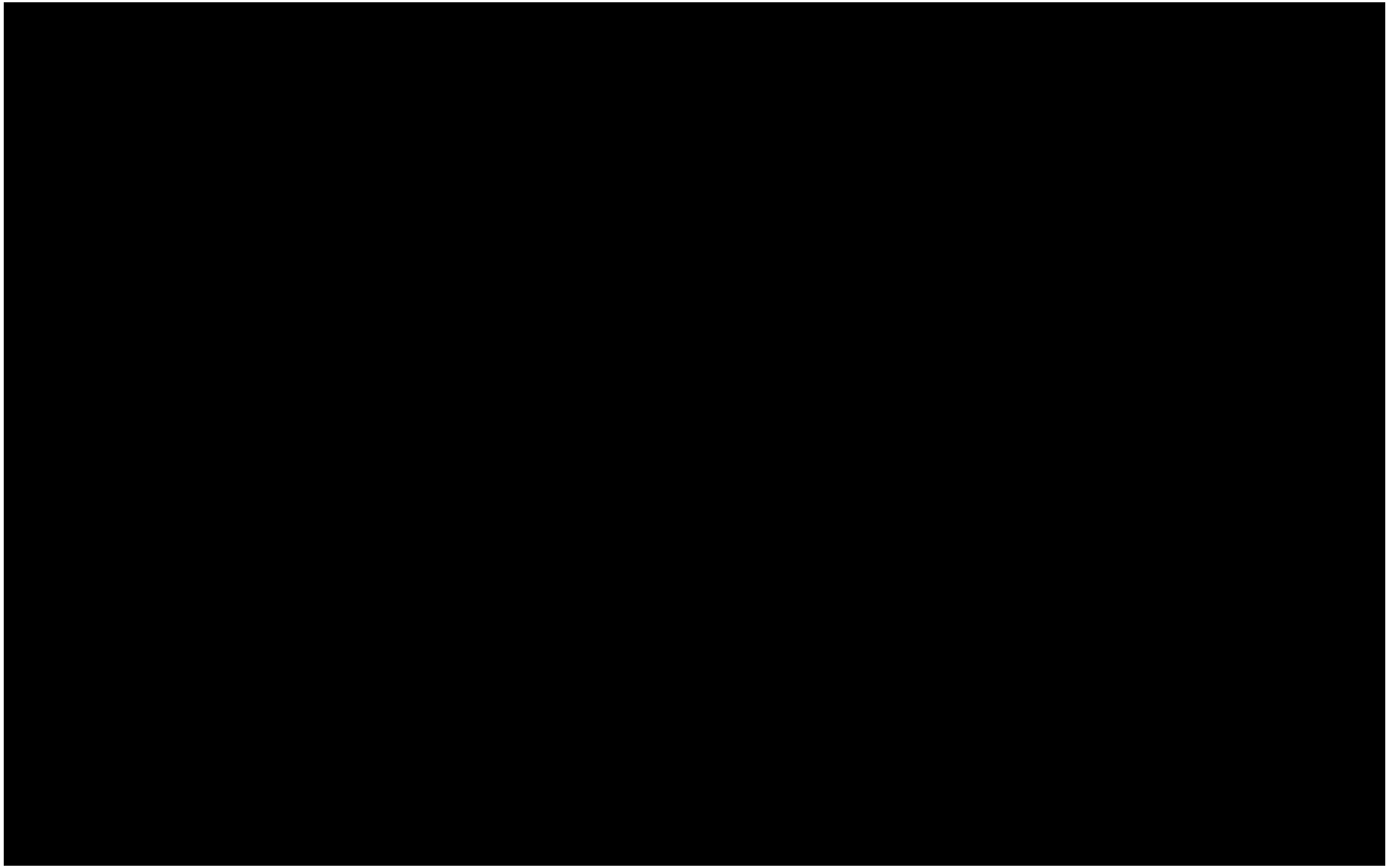
Surgical Action Framework: Workflow [10]



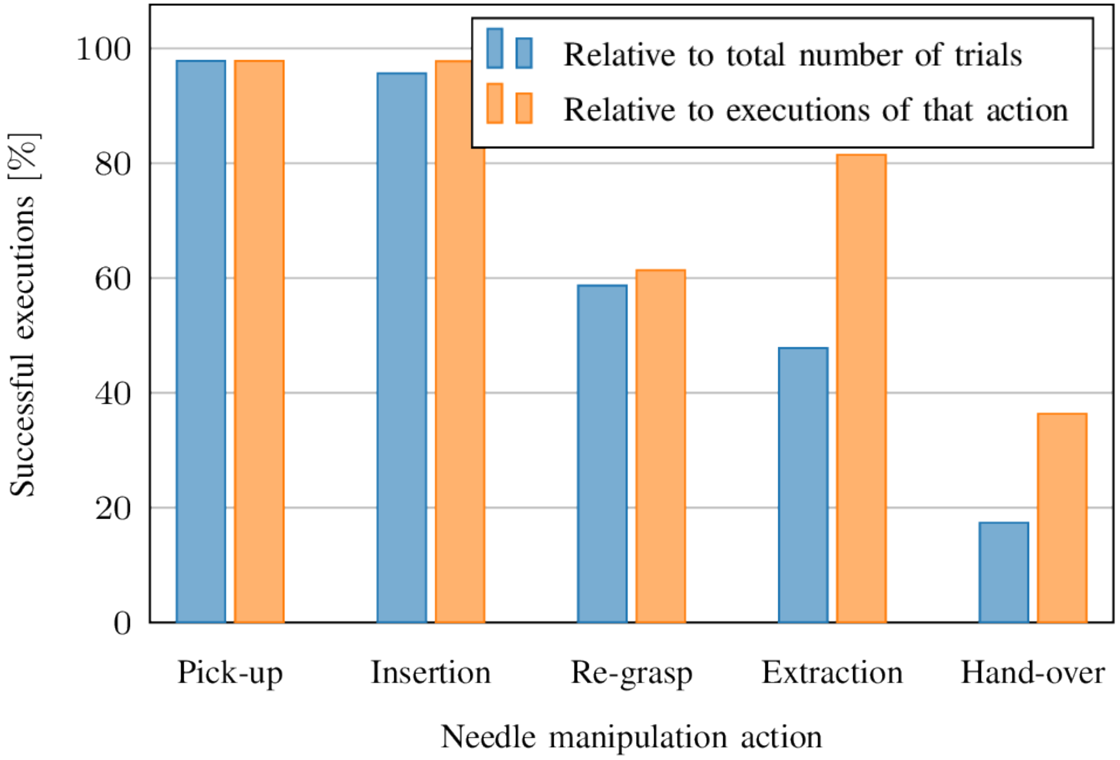


Execution: Generalization Examples [10]

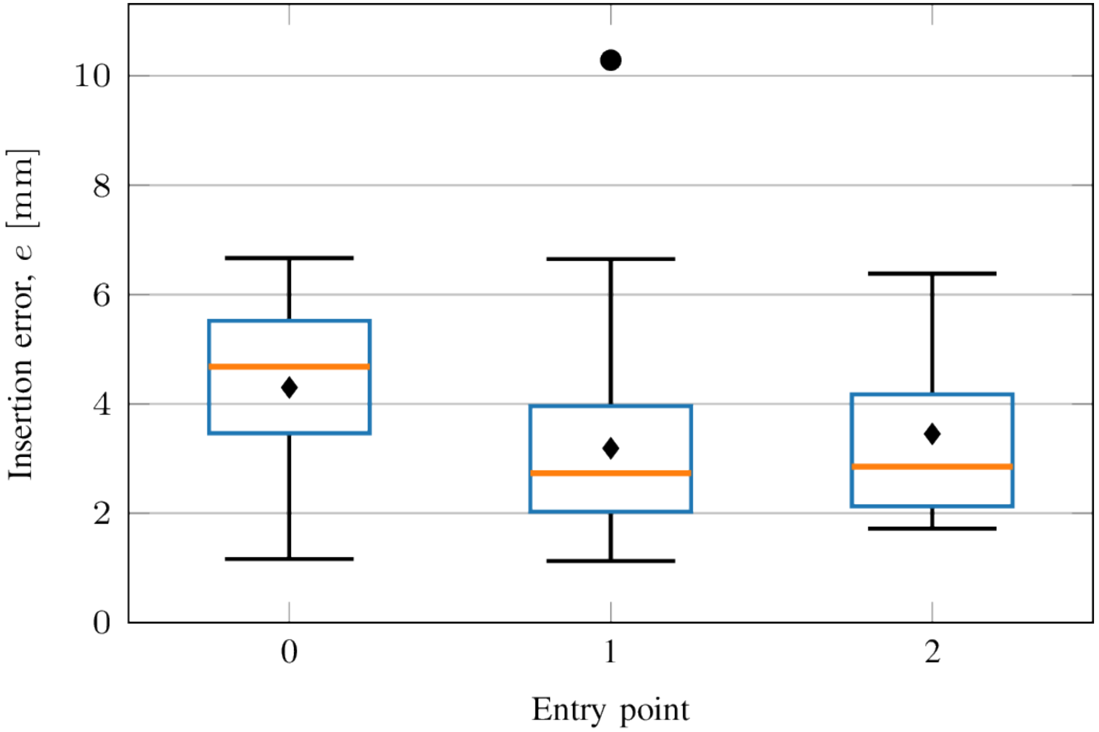
Not everything is a walk in the park… [10]



Results [10]



Relative to total number of trials
Relative to executions of that action
Successful executions [%]
100
80
60
40
20
0
Pick-up
Insertion
Re-grasp
Extraction
Hand-over
Needle manipulation action



Insertion error, e [mm]
10
8
6
4
2
0
0
1
2
Entry point

Concluding Remarks

- Learning from Demonstration (LfD) allows robots to quickly learn tasks from human demonstrations.
- Robotic Minimally Invasive Surgery (RMIS) lends itself well to LfD, as it is naturally teleoperated and operation data is recorded.
- Dynamic Movement Primitives (DMPs) allow for one-shot learning of trajectories and exhibit nice properties when it comes to robustness, coupling and generalizability.
- When coupled with a high-level task planner semantic segmentation and computer vision feedback, DMPs can be used to fully automate suturing in research RMIS setting.

Exercise

1. Download the file *pydmp_exercise.zip* and extract its contents.
2. The *README* file contains the exercise instructions.
3. Once you are done with the exercise, you can try executing the DMP on a Universal Robot:
   a. Download the Universal Robots Simulator
   b. To read data from the robot and send it movement commands, we will use the Universal Robots Real-Time Data Exchange interface. SDU Robotics has compiled a SW library that wraps this functionality. Check the documentation here: https://sdurobotics.gitlab.io/ur_rtde/
   c. Try a few of the UR RTDE examples: https://sdurobotics.gitlab.io/ur_rtde/examples/examples.html
   d. Write a script that steps through the DMP output positions and sends them to the robot using UR RTDE. You will want to make use of the *servoL* or *speedL* commands.

References (I)

[1] A.G. Billard, S. Calinon, R. Dillmann, "Learning from Humans," in Springer Handbook of Robotics, B. Siciliano, O. Khatib, Ed. Springer, Cham, 2016, pp.1995-2014. DOI: 10.1007/978-3-319-32552-1_74

[2] Chrystopher L. Nehaniv, Kerstin Dautenhahn, and Kerstin Dautenhahn. Imitation in animals and artifacts. MIT Press, 2002.

[3] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. “A survey of robot learning from demonstration”. In: Robotics and Autonomous Systems vol. 57, no. 5, 2009, pp. 469–483.

[4] Holger Friedrich, S. Münch, Rüdiger Dillmann, Siegfried Bocionek, and Michael Sassin. “Robot programming by demonstration (RPD): Supporting the induction by human interaction”. In: Machine Learning vol. 23, no. 2-3, 1996, pp. 163–189.

[5] A.J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, S. Schaal. "Dynamical movement primitives: learning attractor models for motor behaviors," Neural computation, vol. 25, no. 2, pp. 328-373, 2013. DOI: 10.1162/NECO_a_00393

[6] Aleš Ude, Bojan Nemec, Tadej Petrić, and Jun Morimoto. “Orientation in cartesian space dynamic movement primitives”. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). 2014, pp. 2997–3004.

[7] Iñigo Iturrate, Christoffer Sloth, Aljaž Kramberger, Henrik Gordon Petersen, Esben Østergaard, and Thiusius Rajeeth Savarimuthu. “Towards Reversible Dynamic Movement Primitives”. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2019, pp. 5063-5070.

References (II)

[8] Aljaž Kramberger, Erfan Shahriari, Andrej Gams, Bojan Nemec, Aleš Ude, and Sami Haddadin. “Passivity Based Iterative Learning of Admittance-Coupled Dynamic Movement Primitives for Interaction with Changing Environments”. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2018, pp. 6023–6028.

[9] Thiusius Rajeeth Savarimuthu, Anders Glent Buch, Christian Schlette, Nils Wantia, Jürgen Roßmann, David Martínez, Guillem Alenyà, Carme Torras, Ales Ude, Bojan Nemec, et al. “Teaching a robot the semantics of assembly tasks”. In: IEEE Transactions on Systems, Man, and Cybernetics: Systems vol. 48, no. 5, 2018, pp. 670–692.

[10] Kim L. Schwaner, Iñigo Iturrate, Jakob K. H. Andersen, Pernille T. Jensen, and Thiusius R. Savarimuthu. “Autonomous Bi-Manual Surgical Suturing Based on Skills Learned from Demonstration”. To appear in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2021.

Thank you for today.

Iñigo Iturrate

Ø27-604-3

inju@mmmi.sdu.dk